# Table of Contents

## Full Papers

2. Model based Online Fault Diagnosis of Automotive Engines using Joint State and Parameter Estimation  
*Nadeer E P, Amit Patra, and Siddhartha Mukhopadhyay*

11. D-matrix Based Fault Modeling for Cryogenic Loading Systems  
*Anuradha Kodali, Ekaterina Ponizovskaya-Devine, Peter Robinson, Dmitry G. Luchinsky, Anupa Bajwa, Michael Khasin, Jose Perotti, and Barbara Brown*

21. Fault diagnostics and evaluation in cryogenic loading system using optimization algorithm  
*Ekaterina Ponizovskaya-Devine, Dmitry G. Luchinsky, Anuradha Kodali, Michael Khasin, Dogan Timucin, Jarred Sass, Jose Perotti, and Barbara Brown*

30. Interactive Multiple-Model Application for Hydraulic Servovalve Health Monitoring  
*Wlamir Olives Loesch Viana and Takashi Yoneyama*

38. Lessons Learned in Implementing a Practical Aircraft System Health Management (ASHM) System  
*Joshua Koelle, Matthew Smith, Peter Sulcs, Greg Kacprzynski, and Rhonda Walthall*

46. Real-time Prognostics of a Rotary Valve Actuator  
*Matthew Daigle*

57. Model-based Prognostics of Hybrid Systems  
*Matthew Daigle, Indranil Roychoudhury, and Anibal Bregon*

67. A Systems Approach To PHM  
*Thomas D. Mooney*

75. Online Monitoring and Fault Diagnosis of Hybrid Systems Using Switched Dynamic Bayesian Networks  
*Gan Zhou, Gautam Biswas, Wenquan Feng, and Xiumei Guan*

86. Characterization of Phase Space Topology Using Density: Application to Fault Diagnostics  
*Mohsen Samadani, Cedrick A. Kitio Kwuimy, and C. Nataraj*

94. From Theory to Practice: Model-Based Diagnosis in Industrial Applications  
*Roxane Koitz and Franz Wotawa*

103. Prognostics and Health Management of an Electro-Hydraulic Servo Actuator  
*Andrea Mornacchi, George J. Vachtsevanos, and Giovanni Jacazio*

115. Application of Health Management and Diagnostics for Synthetic Aperture Radar (SAR) Payloads  
*Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford*

127. Variable Selection and Indices Proposal for the Determination of an Aeronautic Valve Degradation  
*Ananda S. Ribeiro, Takashi Yoneyama, Rodrigo F. Souto, and Wallace Turcio*

138. Estimating Remaining Useful Life Using Actuarial Methods  
*Eric Bechhoefer, Rude Schlunbusch, and Tor Inge Waag*

146. Remaining Useful Life Prediction through Failure Probability Computation for Condition-based Prognostics  
*Shankar Sankararaman*

*Jacob Loverich, Jeremy Frank, David Kraige, Gary Koopmann, Ashok Belegundu, and Michael Grissom*

165. Battery Capacity Anomaly Detection and Data Fusion  
*John Weddington, Wazhao Yan, Wanchun Dou, and Bin Zhang*

173. Parameters Adaption of Lebesgue Sampling-based Diagnosis and Prognosis for Li-ion Batteries  
*Wazhao Yan, Wanchun Dou, Datong Liu, Yu Peng, and Bin Zhang*

182. Lithium-Ion Battery End-of-Discharge Time Estimation and Prognosis based on Bayesian Algorithms and Outer Feedback Correction Loops: A Comparative Analysis  
*Carlos Tampier, Aramis Pérez, Francisco Jaramillo, Vanessa Quintero, Marcos E. Orchard, and Jorge F. Silva*
Towards a Methodology for Design of Prognostic Systems
Jose Ignacio Aizpurua and Victoria M. Catterson

System Interdependency Modeling in the Design of Prognostic and Health Management Systems in Smart Manufacturing
M.L. Malinowski, Peter A. Beling, Yacov Haimes, Amy E. LaViers, Jeremy A. Marvel, and Brian A. Weiss

Device Health Estimation by Combining Contextual Control Information with Sensor Data and Device Health Prognostics Utilizing Restricted Boltzmann Machine
Linxia Liao, Tomonori Honda, and Radu Pavel

Distributed Adaptive Fault-Tolerant Consensus Control of Multi-Agent Systems with Actuator Faults
Mohsen Khalili, Xiaodong Zhang, Yongcan Cao, and Jonathan A. Muse

A Review of Prognostics and Health Management for Power Semiconductor Modules
Nicolas Degrenne, Jeffrey Ewanchuk, Erwan David, Roman Boldyrjew, and Stefan Mollov

A Self-Organizing Map-Based Monitoring System for Insulated Gate Bipolar Transistors Operating in Fully Electric Vehicle
Marco Rigamonti, Piero Baraldi, Enrico Zio, Allegra Alessi, Daniel Astigarraga, and Ainhoa Galarza

Analysis of Electrolytic Capacitor Degradation under Electrical Overstress for Prognostic Studies
Jason Renwick, Chetan S. Kulkarni, and José R. Celaya

Review of Markov Models for Maintenance Optimization in the Context of Offshore Wind
Rafael Dawid, David McMillan, and Matthew Revie

Detection of pitch failures in wind turbines using environmental noise recognition techniques
Georgios Alexandros Skrimpas, Kun S. Marhadi, Robert Gomez, Christian Walsted Sweeney, Bogi Bech Jensen, Nenad Mijatovic, and Joachim Holboell

PHM Based Predictive Maintenance Option Model for Offshore Wind Farm O&M Optimization
Xin Lei, Peter A. Sandborn, Navid Goudarzi, and Maira A. Bruck

An Influence Gauge to Detect and Explain Relations between Measurements and a Performance Indicator
Jérôme Lacaille

Omni-Directional Regeneration (ODR) of Gap Sensor Signal for Journal Bearing System Diagnosis
Joon Ha Jung, Byung Chul Jeon, Byeng D. Youn, Donghwan Kim, and Yeonwhan Kim

Quadrotor Accelerometer and Gyroscope Sensor Fault Diagnosis with Experimental Results
Remus C Avram, Xiaodong Zhang, and Jonathan A. Muse

Measurement Science for Prognostics and Health Management for Smart Manufacturing Systems: Key Findings from a Roadmapping Workshop
Brian A. Weiss, Gregory W. Vogl, Moneer Helu, Guixiu Qiao, Joan Pellegrino, Mauricio Justiniano, and Anand Raghu Nathan

A New Scheme for Monitoring and Diagnosis of Multistage Manufacturing Processes Using Product Quality Measurements
Hossein Davari Ardakani, Brian Phillips, Behrad Bagheri, and Jay Lee

A data pipeline for PHM data-driven analytics in large-scale smart manufacturing facilities
P. O’Donovan, K. Leahy, D. Og Cusack, K. Bruton, and D. T. J. O’Sullivan

Early Detection of Combustion Instability from Hi-speed Flame Images via Deep Learning and Symbolic Time Series Analysis
Soumalya Sarkar, Kin G Lore, Soumik Sarkar, Vikram Ramanan, Satyanarayanan R Chakravarthy, Shashi Phoha, and Asok Ray

Railcar Bogie Performance Monitoring using Mutual Information and Support Vector Machines
Parham Shahidi, Daniel Maraini, Brad Hopkins, and Andrew Seidel

Neural Network-Based Gear Failure Prediction in a Brushless DC Actuation System
Yan Chen, Christina Lucas, Jay Lee, and Michael Buehner
Verification of a Remaining Flying Time Prediction System for Small Electric Aircraft
Edward F. Hogge, Brian M. Bole, Sixto L. Vazquez, José R. Celaya, Thomas H. Strom, Boyd L. Hill, Kyle M. Smalling, and Cuong C. Quach

A V-Diagram for the Design of Integrated Health Management for Unmanned Aerial Systems
Andrew E Heaton, Ip-Shing Fan, Ian K. Jennions, Craig Lawson, and Jim McFeat

Diagnostic Reasoning using Prognostic Information for Unmanned Aerial Systems
Johann Schumann, Indranil Roychoudhury, and Chetan S. Kulkarni

A Sensor-Based Method for Diagnostics of Machine Tool Linear Axes
Gregory W. Vogl, Brian A. Weiss, and M. Alkan Donmez

Quality Control Based Tool Condition Monitoring
Amit Kumar Jain and Bhupesh Kumar Lad

Adaptive Multi-Scale PHM for Robotic Assembly Processes
Benjamin Y. Choo, Peter A. Beling, Amy E. LaViers, Jeremy A. Marvel, and Brian A. Weiss

On Accurate and Reliable Anomaly Detection for Gas Turbine Combustors: A Deep Learning Approach
Weizhong Yan and Lijie Yu

Adapting nearest neighbors-based monitoring methods to irregularly sampled measurements
Inês M. Cecílio, James R. Ottewill, and Nina F. Thornhill

Comparison and ensemble of temperature-based and vibration-based methods for machinery prognostics
James Kuria Kimotho and Walter Sextro

Separated two-phase flow model of cryogenic loading operation
Dmitry G. Luchinsky, Ekaterina Ponizovskaya-Devine, Michael Khasin, Dogan Timucin, Jarred Sass, Jose Perotti, and Barbara Brown

Modeling and Prediction of Criminal Activity Based on Spatio-Temporal Probabilistic Risk Functions
Paulina Flores, Mario Vergara, Pablo Fuentes, Francisco Jaramillo, David Acuña, Aramis Perez, and Marcos E. Orchard

Active Mission Success Estimation through PHM-Informed Probabilistic Modelling
Adam R. Short and Douglas L. Van Bossuyt

Aircraft Line Maintenance Planning Based on PHM Data and Resources Availability Using Large Neighborhood Search
Wlamir Olivares Loesch Vianna, Leonardo Ramos Rodrigues, and Takashi Yoneyama

Cost-benefit Analysis of Prognostics and Condition-based Maintenance Concepts for Commercial Aircraft Considering Prognostic Errors
Nico B. Hölzel and Volker Gollnick

Cost-Wise Readiness Enabled Through Condition Based Maintenance Plus (CBM+)
Josh Kennedy, Jim Carter, and Matt Carter

Cyber Defense of Rotating Machinery Using an Integrated ‘Fuse’ Bearing
Evyatar Cohen, Jacob Bortman, and Renata Klein

Asynchronous Motor Test Bench for the Generation and Current Signal Diagnostics of Accelerated Bearing Damage
Christoph Anger, Christian Preusche, and Uwe Klingauf

Experimental Approach for Estimating Mesh Stiffness in Faulty States of Rotating Gear
Jungho Park, Jong M. Ha, Hyunseok Oh, Byeng D. Youn, Sungho Park, and Joo-Ho Choi

Processing and Interpretation of Crack-Propagation Sensors
Nenad G. Nenadic, Paul A. Ardis, Adrian Hood, Michael G. Thurston, and David G. Lewicki

Advances in Ultrasonic Imaging for Internal Flaws in Structures
Thompson Vu Nguyen, Simone Sternini, and Francesco Lanza di Scalea
Analyzing high-dimensional thresholds for fault detection and diagnosis using active learning and Bayesian statistical modeling
Yuning He

Improved Fault Detection by Appropriate Control of Signal Bandwidth of the TSA
Eric Bechhoefer and Xinghui Zhang

Abort Trigger False Positive and False Negative Analysis Methodology for Threshold-based Abort Detection
Kevin J. Melcher, José A. Cruz, Stephen B. Johnson, and Yunnhon Lo

Model-based Reasoning Approach for Automated Failure Analysis: An Industrial Gas Turbine Application
Gulnar Mehdi, Davood Naderi, Giuseppe Ceschini, and Mikhail Roshchin

Vibration-based Fault Diagnostics of Planetary Gearbox Using Time Synchronous Averaging with Multiple Window Functions
Jong M. Ha, Jungho Park, Hyunseok Oh, and Byeng D. Youn

A New Generic Approach to Convert FMEA in Causal Trees for the Purpose of Hydro-Generator Rotor Failure Mechanisms Identification
Olivier Blancke, Normand Amyot, Claude Hudon, Mélanie Lévesque, and Antoine Tahan

Model-based Approach to Automated Calculation of Key Performance Indicators for Industrial Turbines
Gulnar Mehdi, Davood Naderi, Giuseppe Ceschini, Alexey Fishkin, Sebastian Brandt, Stuart Watson, and Mikhail Roshchin

Linear Temporal Logic (LTL) Based Monitoring of Smart Manufacturing Systems
Gerald Heddy, Umer Hzafa, Peter A. Beling, Yacov Haimes, Jeremy A. Marvel, Brian A. Weiss, and Amy E. LaViers

Early Concepts for the Coupling of a Nuclear Plant Computer to a Computerized Maintenance Management System for Autonomous Prognostic Model Development
Zachary Welz, J. Wesley Hines, and Belle R. Upadhyaya

Sparse Coding-Based Failure Prediction for Prudent Operation of LED Manufacturing Equipment
Jia-Min Ren, Chuang-Hua Chueh, and H. T. Kung

Testing Diagnostics Components Supervising Functional Safety Requirements
Mihai Nica, Ingo Pill, and Franz Wotawa

Direct analysis of non-quadratic phase coupling for detection of linearly modulated signals
Fidel Hernández and Aitzol Iturrospe

Speed Estimation in Geared Wind Turbines Using the Maximum Correlation Coefficient
Georgios Alexandros Skrimpas, Kun S. Marhadi, Bogi Bech Jensen, Christian Walsted Sweeney, Nenad Mijatovic, and Joachim Holboell

Sliding Wear Particle Mass Distribution Assessment for Wear Mode Diagnosis
Surapol Raadnai

Creep Mechanisms vis-à-vis Power Law vs. Grain Boundary Sliding in α-β Titanium Alloys for Physics Based Prognostics
Amar Kumar, Alka Srivastava, Nita Goel, Avi Banerjee, and Ashok K Koul

Hardware Development for the Controlled Fault Injection into a Turbofan Engine Air-Bleed Valve
George E. Goersepe, Donald L. Simon, and Kai F. Goebel

Distributed Fault Detection and Estimation for Cooperative Adaptive Cruise Control System in a Platoon
Zoleikha Abdollahi Biron and Pierluigi Pisu

A Probabilistic Machine Learning Approach to Detect Industrial Plant Faults: PHM15 Data Challenge
Wei Xiao
727  Feature Extraction and Ensemble Decision Tree Classifier in Plant Failure Detection
    Cong Xie, Donglin Yang, Yixiang Huang, and Donglai Sun

736  Fault Log Recovery Using an Incomplete-data-trained FDA Classifier for Failure Diagnosis of Engineered Systems
    Hyunjae Kim, Jong Moon Ha, Jungho Park, Sunuwe Kim, Keunsu Kim, Beom Chan Jang, Hyunseok Oh, and Byeng D. Youn

747  Author Index
Full Papers
Model based Online Fault Diagnosis of Automotive Engines using Joint State and Parameter Estimation

Nadeer E P¹, Amit Patra², and Siddhartha Mukhopadhyay³

¹²³ Indian Institute of Technology, Kharagpur, West Bengal, 721302, India
epnadeer@iitkgp.ac.in
amit@ee.iitkgp.ernet.in
smukh@ee.iitkgp.ernet.in

ABSTRACT

In this work, an Extended Kalman Filter (EKF) based tunable diagnoser, which uses a minimal hybrid nonlinear state space model of a spark ignition (SI) four stroke engine, is used for the detection and isolation of a variety of engine system faults including intake manifold leak, injector fault and exhaust manifold leak. The state estimates and innovation sequences from the EKF based estimator are shown to be adequate for the detection and isolation of the faults under consideration. Once a fault is detected and isolated, the diagnoser could be tuned online to perform fault identification by redefining a model/fault parameter as an additional state to be estimated, and then performing a joint state and parameter estimation. The engine model and diagnoser are implemented in Simulink™ and are validated against an AMESim™ model of the engine. For the nominal engine model, the performance of the EKF estimator is compared with two other computationally more expensive nonlinear estimators, namely the Unscented Kalman Filter (UKF) and Rao-Blackwell Particle Filter (RBPF).

1. INTRODUCTION

Traditionally, automotive engine modelling has been carried out using Mean value engine models (MVEMs). MVEMs are formulated based on the assumption that all processes and effects are spread out over the entire engine cycle, and do not take into account the reciprocating nature of the engine (Guzzella & Onder, 2010). An alternative to MVEM, which takes into account the within-cycle reciprocating nature of the engine cylinders, is the Within Cycle Crank-angle-based Model (WCCM) (Sengupta S., Mukhopadhyay, Deb, Pattada, & De, 2012). WCCMs are instantaneous physics-based models and have the advantage of being able to detect, isolate and identify small faults within shorter time duration as compared with MVEMs. This, however, comes at a greater computational effort.

WCCM based fault diagnosis of engines has been carried out using a bank of Extended Kalman Filters (EKF) in (Sengupta, Mukhopadhyay, & Deb, 2011). However, presence of multiple estimators and calculation of Jacobian matrix by simulation makes the use of such bank of estimators prohibitive for online implementation in a real time fault diagnosis system for engines. In this work, these shortcomings are overcome by three modifications. Firstly, a minimal hybrid state space model is derived by reformulating the mass and energy balance equations. Secondly, the EKF uses analytically derived expressions for the Jacobian matrix instead of employing simulation to find it. Finally, instead of a bank of estimators, a single tunable estimator is employed for fault diagnosis. The estimator identifies a fault parameter only when a fault is detected and/or isolated using the residuals and state estimates from a nominal EKF.

Figure 1 shows the proposed fault diagnosis framework. The control inputs from the Engine Control Unit (ECU) and measurements from the engine are inputs to the estimator. Fault-detection and isolation (FDI) logic monitors the state estimates and residuals (innovation sequence) from the estimator. Once a fault is detected and isolated, the FDI sends commands to the estimator for identification of the concerned parameter(s). A joint state and parameter estimation is then performed to identify the fault magnitude. The estimation accuracy depends on the input conditions also.

2. ENGINE MODELING, ESTIMATION AND FAULT DIAGNOSIS

In this section, the minimal hybrid state space modeling of the engine, nonlinear estimation of engine states, the fault detection and isolation methodology and the fault identification method using joint state and parameter estimation are described.

Nadeer E P et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
2.1. Minimal hybrid state space model of the engine

A naturally aspirated spark ignition (SI) gasoline engine has the basic subsystems and components shown in Figure 2, which include the intake manifold, combustion chamber (cylinder and piston), exhaust manifold, crank assembly, exhaust gas recirculation (EGR), ECU, fuel injection and sensor and actuator assembly. A physical system such as an engine consists of two kinds of objects: reservoirs and flows. Reservoirs store thermal or kinetic energy, mass, information, etc., whereas flows have these storage entities flowing between reservoirs, typically driven by differences in reservoir levels (Guzzella & Onder, 2010). In the case of the engine, we consider the intake manifold (IM) or Air Intake System (AIS), cylinder and exhaust manifold (EM) as reservoirs. The throttle, exhaust gas recirculation (EGR) and muffler are considered as flow elements. The equations we use here are from standard models available in duly mentioned references, the novelty is only in the state space formulation. We start the development by considering the regular direction of mass and energy – from throttle to exhaust.

A full 4-stroke engine cycle corresponds to a crank shaft rotation of 0-720°, approximately 180° each for intake, compression, expansion and exhaust strokes. The WCCM equations essentially describe two things: the continuous time dynamics (or flows) of the variables or system states of interest – such as pressure, temperature and mass flow rates – and the jump conditions, or events, which cause transition between different modes of the hybrid system. These transitions between modes could be triggered either by control actions or by the instantaneous state variable values themselves (e.g., sub-sonic or sonic, positive flow or negative flow).

We assume that all the gases in the engine obey the ideal gas law, i.e.,

\[ PV = mRT \] or \[ P = mRT/V \]  

(Please refer to nomenclature section for symbols and notations).

The differential equations describing the reservoir can be written in terms of the mass \( m \), the internal energy \( U \), and the heat energy \( Q \), as:

\[
\dot{m} = \dot{m}_{in} - \dot{m}_{out}
\]

\[
\dot{U} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q}_{in} - \dot{Q}_{out}
\]

The subscripts \( in \) and \( out \) respectively denote variables entering and leaving the reservoir. The flow equation for valves can be written as (Guzzella & Onder, 2010):

\[
\dot{m} = C_d A \frac{P_{in}}{\sqrt{R_T T_{in}}} \psi \left( \frac{P_{in}}{P_{out}} \right)
\]

where,

\[
\psi \left( \frac{P_{in}}{P_{out}} \right) = \begin{cases} 
\frac{1}{1 + \gamma} & \text{if } P_{out} < P_{in} \\
\frac{\gamma}{1 + \gamma} P_{in} & \text{if } P_{out} \geq P_{in}
\end{cases}
\]

\[
P_{cr} = 2 / (\gamma_{in} + 1) \gamma_{in}^{-1} P_{in}
\]

\( P_{in} \) and \( P_{out} \) will be interchanged and \( \gamma_{cr} \) values will be replaced by \( \gamma_{out} \) for reverse flow (negative).

The model inputs are: Throttle position, Fuel control signal, EGR control signal, speed and crank angle.

At each storage element, we approximate the specific gas constant \( R \) (for air, burnt gases and fuel) and specific heats \( C_p \) and \( C_v \) in terms of mass fractions as:

\[
R = \sum \frac{m_i R_i}{m}, \quad C_p = \sum \frac{m_i C_{p,i}}{m}, \quad C_v = \sum \frac{m_i C_{v,i}}{m}, \quad \gamma = \frac{C_p}{C_v}
\]

Where, \( i=a,b,f \) (for air, burnt gases and fuel), and \( m = m_a + m_b + m_f \).

The time derivatives of internal energy and enthalpy can be expressed in terms of masses, specific heats and temperature as:
Throughout this paper, subscript $i$ denotes terms in the state of the system and the ratio of specific heats $\gamma$ and EGR respectively, and $m^i$ is the flow rate of gases entering the cylinder from IM. These can be obtained by replacing the flows in Eq. (3) by Eq. (1).

The mass balance at IM is:

$$\dot{m}_i = \dot{m}_{in} - \dot{m}_{out}$$

(11)

Where $\dot{m}_i$ is the mass flow rate at the intake manifold and $\dot{m}_{in}$ and $\dot{m}_{out}$ are the intake and exhaust manifold respectively. $m^i$ and $\gamma^i$ are vectors and elements giving the mass and temperature of the state. It is possible to express the nonlinear engine dynamical equations in the form $\dot{x} = f(x,u,d)$, where $u$ is the input vector.

2.1.2. Cylinder states

The mass balance equation for individual cylinders could be written as:

$$\dot{m}_i = \dot{m}_{in} - \dot{m}_{out}$$

(9)

where $m^i_2$ is the mass of fuel accumulated in cylinder $i$ before combustion, $\zeta$ is the engine angular speed, $\Delta P$ is the combustion pressure, $\zeta$ is the combustion airfuel ratio, and $\Delta P$ is the combustion pressure drop.

The enthalpy balance for the cylinder should include the heat energy added during combustion, 1989, and due to radiation. The enthalpy balance for the cylinder is approximated by the stoichiometric airfuel ratio, $\zeta$ is the engine angular speed, $\Delta P$ is the combustion pressure, $\zeta$ is the combustion airfuel ratio, and $\Delta P$ is the combustion pressure drop.

$$\dot{Q}_i = k(\dot{T}_i - \dot{T}_c)$$

(13)

where $\dot{Q}_i$ is the heat flow rate at the intake manifold gives:

$$\dot{m}_i = \dot{m}_{in} - \dot{m}_{out}$$

(7)
Further, from the first law of thermodynamics, the mechanical work done should be subtracted from the heat energy added to get the change in internal energy. Thus for the $i^{th}$ cylinder, the expression for cylinder temperature derivative is:

$$T_{ci}^{(i)} = \frac{1}{m_{cyl}^{(i)} C_{v,ci}^{(i)}} \left[ \dot{Q}_{in,ci}^{(i)} - \dot{Q}_{out,ci}^{(i)} + H_{i,c}^{(i)} - H_{i,ci}^{(i)} \right]$$

(14)

The state space formulation requires that all the variables be expressed in terms of states and their first order derivatives. Eqs. (1) and (4) can be used to express $P_c$ and $C_{cyl}$ as algebraic states. However, the derivative of $C_{cyl}$ should be expressed in terms of the states and their derivatives only. To this end, from the assumptions made in Eqs. (4) and (5), we write:

$$m_{cyl}^{(i)} C_{v,ci}^{(i)} + m_{cyl}^{(i)} \dot{C}_{v,ci}^{(i)} = \sum m_{cyl,i}^{(i)} C_{v,j}$$

Where, $i = a,b,f$ (air, burnt gases, fuel)

A similar technique could be used for the derivatives of enthalpy terms $H$ which in turn depend on the derivatives of $C_{cyl}$. Further, if the cylinder dimensions are known, the derivative of the cylinder volume, $V_c$, in the above equation can be expressed as a function of angular speed. Hence, all of the terms in the right hand side of Eq. (14) can be expressed as a function of states only, and without their derivatives. Eqs. (7)-(14) constitute the first order non-linear differential equations necessary for forming the state space model.

2.2. Estimation of Nominal Engine States

The engine model equations described in the previous section are nonlinear and hybrid in nature. However, the model exhibits no state resets when transitioning from one discrete mode to other. The optimal state estimation problem of nonlinear systems is intractable for many practical problems. Hence extensions of linear system techniques for state estimation, like the Kalman filter (KF), are often used in practice. The most common techniques in use are the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF) and their variants. For hybrid systems, the optimal estimation of discrete modes and continuous states would mean that the number of modes to be estimated grows exponentially with each time step. To reduce the complexity, in some techniques, $N$ modes with largest probability are kept and the rest are discarded, and probabilities are renormalized to sum up to unity (Bar-Shalom, Li, & Kirubarajan, 2004). The generalized pseudo-Bayesian (GPB) approaches and interacting multiple model (IMM) estimation (Bar-Shalom, Li, & Kirubarajan, 2004) fall in this category. Particle filters (PF) (Doucet, De Freitas, Gordon, & others, 2001), (Arulampalam, Maskell, Gordon, & Clapp, 2002) are a class of numerical methods for the solution of the optimal estimation problem in non-linear non-Gaussian scenarios and come under the generic name Sequential Monte Carlo algorithms. In UKF, the weights for the sigma points are fixed, whereas in PF, the weights of particles are dependent on the posterior probabilities, and hence can be expected to perform better. The Rao-Blackwell particle filter (RBPF) is a special kind of particle filter that reduces the variance in estimates. Specifically, the estimation state space is partitioned so that one set is updated with sampling whereas the other set is updated analytically using KF, EKF or UKF. In the hybrid estimation problem, the RBPF has been applied with sampling for discrete states and exact computations of mean and variances using KF for continuous states (De Freitas, 2002). In our case, the particles do not have a one-to-one correspondence with discrete modes.

In this work, sensor measurements that are assumed to be available to the estimator are: throttle position (from which throttle area could be found out), IM pressure, IM temperature, EM pressure and EM temperature. The engine speed and crank angle are considered to be inputs. In the case of an actual engine, these signals have to be derived from the crank position sensor signal. The estimator is a continuous-discrete one, i.e., the engine dynamical equations are continuous, but measurements are assumed to be discrete. The integration of continuous dynamics has been carried out using RK4 method. Estimation of nominal engine states has been carried out using EKF, UKF and RBPF and results were compared. The EKF uses analytical expressions for the Jacobian matrix entries.

2.2.1. The EKF Algorithm (Sarkka, 2006)

**Prediction:** Integrate the following differential equations to get $m_{cyl}$ and $P_c$:

$$\dot{x} = f(x,u)$$

$$\dot{P}_k = FP_k + P_kF_{+} + Q_k$$

(15)

Where $F$ is the Jacobian of $f$ w.r.t the state vector $x$, $Q_k$ is the process noise covariance and $P_k$ is the estimation error covariance. Variables $x$ and $P_k$ are assigned the values $m_{cyl-1}$ and $P_{k-1}$ from previous time step before solving the above equations.

**Update:**

$$v_k = y_k - h(m_{cyl})$$

$$S_k = H_k P_k H_k^T + R_k$$

$$K_k = P_k H_k^T S_k^{-1}$$

$$m_k = m_{cyl} + K_k v_k$$

$$P_k = P_k - K_k S_k K_k^T$$

(16)

Where $h(.)$ denotes the measurement function and $H$, its Jacobian. $R_k$ is the measurement noise covariance.
2.2.2. The RBPF Algorithm (Hutter, Dearden, & others, 2003)

1. For N particles \( p^{(i)} \), \( i = 1: N \), sample discrete modes \( z_{0}^{(i)} \) from the prior \( p(Z_{0}) \).
2. For each particle \( p^{(i)} \), set \( \hat{\mu}_{i}, \Sigma_{i} \) to the prior mean and variance in state \( z_{0}^{(i)} \).
3. For each time-step \( t \) do
   a) For each particle \( p^{(i)} = (z_{t-1}^{(i)}, \hat{S}_{t}^{(i)}) \) do
      i) Sample a new mode:
         \[ \hat{z}_{t}^{(i)} \sim P\left(Z_{t} \mid z_{t-1}^{(i)}\right) \]
      ii) Perform EKF update using parameters from mode \( \hat{z}_{t}^{(i)} \):
         \[ (y_{vt+1}^{(i)}, \hat{S}_{t}^{(i)}, \hat{\mu}_{t}^{(i)}, \Sigma_{t}^{(i)}) \leftarrow \text{EKF}(\hat{z}_{t}^{(i)}, \Sigma_{t}^{(i)}, y_{t}) \]
      iii) Compute the weight of particle \( p^{(i)} : P(p^{(i)} = p^{(i)}) \propto w_{i}^{(i)} \)
   b) Resample \( N \) new samples \( p^{(i)} \) where:
      \[ P(p^{(i)} = p^{(i)}) \propto w_{i}^{(i)} \]

2.3. Fault Modelling

In this work, three faults are considered: intake manifold leak, fuel injector choking, and exhaust manifold leak. The IM leak could be modelled by an additional leak area in the throttle flow orifice equation. The injector fault could be modelled by changing the fuel injector signal to one of the cylinders. In our case the fuel injection duration was halved. The exhaust manifold leak can be modelled by an additional area in the muffler flow equation.

2.4. Online Fault Detection and Isolation

The fault detection algorithm assumes that only one fault occurs at a time. This is not unreasonable, because the algorithm tries to detect faults early using an instantaneous model, and hence it is very unlikely for multiple faults to occur at exactly the same sampling instant. To enhance the prospects of online implementation, only one estimator, namely the one for the nominal case, runs at a time. The fault detection and isolation algorithm monitors the measurement residuals and states from the estimator. An estimator which considers all the fault parameters as states to be estimated will have very poor observability because of the fewer measurements compared to the number of states.

Part of the fault detection logic runs with the nominal estimator and tries to predict the measurement residuals in presence of single faults, captured by corresponding parameters. For example, the intake manifold leak can be captured by an additional area in the throttle. By defining this area to be a parameter, it is possible to predict the fault residuals in presence of IM leak. A similar method could be adopted for EM leak with the muffler area. If \( w_{i} \) is the parameter associated with fault \( i \), and \( \Delta w_{i} \) a small change in \( w_{i} \), indicating a fault, then from Eq.(16) we can predict the residual for fault \( i \) as:

\[
\Delta v_{i} = - \left( \frac{\partial h}{\partial m_{k}} + \frac{\partial h}{\partial w_{i}} \right) \Delta w_{i} \tag{17}
\]

Note that \( \frac{\partial h}{\partial m_{k}} = H_{k} \), which is already available from the EKF routine. The term \( \frac{\partial m_{k}}{\partial w_{i}} \) could be calculated recursively by a technique similar to dual estimation in (Haykin, 2001):

\[
\frac{\partial m_{k+1}}{\partial w_{i}} = \left( I + \frac{\partial f}{\partial m_{k}} \Delta T \right) \frac{\partial m_{k}}{\partial w_{i}} + \frac{\partial f}{\partial w_{i}} \Delta T \tag{18}
\]

\[
\frac{\partial m_{k}}{\partial w_{i}} = \frac{\partial m_{k}}{\partial w_{i}} - K \left( \frac{\partial h}{\partial m_{k}} \right) \frac{\partial m_{k}}{\partial w_{i}} \frac{\partial h}{\partial w_{i}} + \frac{\partial h}{\partial w_{i}} \Delta T
\]

Where we have discretized the continuous dynamics by Euler approximation. \( K \) is the Kalman gain and \( \Delta T \) is the sampling interval. Note that \( \frac{\partial f}{\partial m_{k}} = F \), already available from EKF routine in Eq. (15).

If the sign-reversed residual from the estimator is \( \Delta v_{k} \), and a fault of magnitude \( \Delta w_{i} \) is present, we expect \( \Delta v_{k} \) to be approximately equal to \( \Delta v_{i} \). However, in actual case, neither the fault type nor its magnitude is known. An intuitive way to detect the fault is to normalize both \( \Delta v_{k} \) and \( \Delta v_{i} \), calculate a moving average (with window length \( M \) of their inner product, and choose the fault \( i \) for which the inner product is maximum:

\[
g_{k} = \sum_{k-M+1}^{k} \left( \Delta v_{k} \right)^{T} \Delta v_{i} \tag{19}
\]

\[
\text{fault index} = \arg \max_{i} g_{k}
\]

The above condition is applied only when the products are more than a set threshold value between 0 and 1. Otherwise, a fault free condition is inferred. To save memory, the moving average filter could be replaced by an Infinite Impulse Response (IIR) filter which requires only one delay
storage per measurement. With IIR filter, the overall process is similar to a likelihood ratio test with geometric moving average (Basseville & Nikiforov, 1993).

When the above procedure is not sufficient to detect and/or isolate the faults, the state residual and state estimate signals from the Kalman filter are checked. This is needed to detect and isolate the injector fault. The injector fault too could be captured by defining a multiplicative parameter in the model; however, an ad-hoc method using state residuals from the Kalman filter was employed here. Since the injector fault also affects exhaust manifold states and intake manifold states (through EGR), isolation is difficult. The fault detection function \( g_t \) for EM leak gives a fairly high value for injector fault too. However, the state correction term \( K_t \) from Kalman filter can be used to isolate injector fault in one of the cylinders, assuming single fault. The state error correction signal for fuel mass in cylinder with faulty injector will be higher in absolute value than non-faulty ones. The filtered state error correction signal \( \Delta m_{ji} \) for each cylinder fuel mass could be compared with their mean, and if the deviation from the mean is more than a threshold for a particular cylinder, the corresponding injector is faulty.

2.5. Joint State and Parameter Estimation for Fault Identification

Once the fault is detected and isolated, the nominal estimator could be modified to identify only the parameter associated with the detected fault. The estimator error is minimized in the least square sense by updating the parameter at each time step as (Haykin, 2001):

\[
w_{k+1} = w_k + r_k
\]

Where, \( w_k \) is the parameter vector and \( r_k \) is a small noise term added for numerical stability of the estimator. The new state vector is formed by appending \( w_k \) to the original state vector \( x_k \). The state and measurement equations will now be functions of this combined state and parameter vector. In this paper, only the intake manifold leak area estimation has been performed.

3. SIMULATION RESULTS AND DISCUSSION

The estimation and fault diagnosis techniques described were tested against data generated from an AMESim\textsuperscript{TM} model of the engine with 4 cylinders.

3.1. Estimation of Nominal Engine States

The estimates of nominal engine states using EKF, UKF and RBPF are shown in Figure 3. The UKF uses 50 weights and RBPF uses 12 particles. The normalized root mean square error (NRMSE) was used as a performance indicator and is shown in Table 1. The computation times taken for a 4 second simulation of the engine model and estimator are also given in the table. It is seen that the RBPF and UKF are marginally better in performance than EKF; however, this comes at a much greater computational effort. The UKF, even with more number of weights than number of particles for RBPF, performs worse than RBPF in terms of computation time and estimation accuracy. The EKF could be employed for online estimation on account of its fast execution.

![Figure 3. Comparison of state estimates using EKF, UKF and RBPF: a) Intake manifold temperature, b) Cylinder 1 temperature and c) Exhaust manifold pressure](image)

Table 1. Comparison of nominal state estimators

<table>
<thead>
<tr>
<th>Estimator</th>
<th>NRMSE</th>
<th>Computation Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>0.0417</td>
<td>55</td>
</tr>
<tr>
<td>UKF</td>
<td>0.0412</td>
<td>1360</td>
</tr>
<tr>
<td>RBPF</td>
<td>0.0404</td>
<td>620</td>
</tr>
</tbody>
</table>

3.2. Online Fault Detection and Isolation

The EKF estimation based on nominal model was carried out for three kinds of faults considered.

3.2.1. Detection and Isolation of Intake Manifold Leak Fault

An intake manifold leak fault of 50mm\textsuperscript{2} area was introduced at 3sec. in the AMESim\textsuperscript{TM} model. The nominal estimator was run with data from AMESim\textsuperscript{TM} simulation. The throttle
signal, residuals from the estimator and fault detection function $g_k$ in Eq.(19), for both IM leak and EM leak are all shown in Figure 4. The threshold value used for $g_k$ was 0.4. It could be seen that the $g_k$ for EM leak is always less than this value and hence, for the fault magnitude of 50mm$^2$, IM leak is detectable and isolable for low throttle values.

![Figure 4. Measurement residuals and fault detection functions in presence of intake manifold leak](image)

### 3.2.2. Detection and Isolation of Injector Fault

In this case, the data was collected by halving the pulse width of cylinder 1 fuel injector signal. None of the measurements available provides direct data from the cylinders, but EKF is able to estimate the cylinder states. The EM temperature shows a dip at every fourth cycle, which indicates fault with one of the cylinders. The state error correction signals for cylinder masses obtained from the estimator were used to detect the injector fault. The filtered state error correction signals and results of fault detection are shown in Figure 5. It is seen that the fault detection function $g_k$ for EM leak gives a false alarm in this case. To resolve this fault from EM leak, checking the state error correction signal is necessary.

![Figure 5. State error correction signals, exhaust manifold temperature and fault detection functions for injector fault](image)

### 3.2.3. Detection and Isolation of Exhaust Manifold Leak Fault

An exhaust manifold leak fault of 50mm$^2$ area was introduced at 3sec. in the AMESim™ model. The throttle signal, residuals from the estimator and fault detection function $g_k$ in Eq.(19), for both IM leak and EM leak are all shown in Figure 6. The threshold value used for $g_k$ was 0.4. As pointed out earlier, the $g_k$ for EM leak gives a false alarm for injector fault, and should be resolved by checking mass correction signals from estimator. It could be seen that the $g_k$ for IM leak is always less than that for EM leak and hence, for the fault magnitude of 50mm$^2$, EM leak is detectable and isolable for low throttle values.

![Figure 6. Measurement residuals and fault detection functions in presence of intake manifold leak](image)

### 3.3. Fault Identification by Joint State and Parameter Estimation

The joint state and parameter estimation was carried out for IM leak fault. To this end, once the leak is detected, a fault flag is sent to the EKF estimator and it is tuned to detect the leak area, now defined as a new state. Once the leak area estimation is started, the fault detection signal for IM leak is no longer valid, since the residuals will now tend to zero.
The fault detection function $g_k$ for IM leak and leak area estimates are shown in Figure 7. For a 50 mm$^2$ leak fault, the final value of estimated leak is 48.3 mm$^2$ the error in estimation being only 1.7 mm$^2$.

![Figure 7. IM leak detection function and leak area estimate](image)

### 4. CONCLUSION

An Extended Kalman Filter (EKF) based estimator that uses an instantaneous physics based model of the spark ignition automotive engine was shown to be sufficient for online estimation of engine states with estimation accuracy comparable to other nonlinear estimators like Unscented Kalman Filter (UKF) and Rao-Blackwell Particle Filter (RBPF). The use of such an estimator for fault detection and isolation of intake manifold leak, injector fault and exhaust manifold leak was demonstrated. Online identification of faults by tuning the estimator for joint and state and parameter estimation, once a fault is detected, was also demonstrated for the intake manifold leak fault.

### ACKNOWLEDGEMENT

Research reported in this publication was supported by National Program for Micro and Smart Systems (NPMASS), Govt. of India.

### NOMENCLATURE

- $A$: Area of cross section for flow (m$^2$)
- $C_d$: Coefficient of discharge
- $C_P$: Constant pressure specific heat of gas (J/kg K)
- $C_v$: Constant volume specific heat of gases (J/kg K)
- $C_{pa}$: $C_P$ for air (J/kg K)
- $C_{pb}$: $C_P$ for burnt gases (J/kg K)
- $C_{pf}$: $C_P$ of fuel (J/kg K)
- $C_{va}$: $C_v$ for air (J/kg K)
- $C_{vb}$: $C_v$ for burnt gases (J/kg K)
- $C_{vf}$: $C_v$ of fuel (J/kg K)
- $m$: Mass of the gas in a reservoir element (kg)
- $\dot{m}$: Mass flow rate (kg/s)
- $P$: Pressure (N/m$^2$)
- $Q_{LHV}$: Lower Heating Value of fuel, accommodating latent heat of vaporization for fuel (J/kg)
- $\dot{Q}_{\text{heatloss}}^{(i)}$: Convective and radiative heat loss in $i^{th}$ cylinder (J/s)
- $R_{i}$: Specific gas constant for a material $i$ (J/kg K)
- $T$: Temperature at the reservoir (K)
- $T_a$: Ambient Temperature (K)
- $T_{\text{cool}}$: Environment temperature for convection (K)
- $h_c$: Heat transfer coefficient (W/m$^2$ K)
- $V$: Volume of the element (m$^3$)
- $\theta$: Crank angle (rad)
- $\theta_{\text{soc}}$: Crank angle for start of combustion (rad)
- $\alpha$: Throttle Angle (rad)
- $\omega$: Angular engine speed (rad/s)
- $\gamma$: Ratio of specific heats for gas
- $\eta_c$: Combustion efficiency
- $\rho$: Density of gas
- $\lambda$: Lambda value from lambda sensor
- $\lambda_{af}$: Stoichiometric air fuel ratio

### REFERENCES


**Biographies**

**Nadeer E P** received the B.Tech. degree in Applied Electronics and Instrumentation Engineering from Kerala University, Trivandrum, India in 2006 and M.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kharagpur, in 2008. During 2008-2010, he worked at SanDisk Semiconductors, Shanghai, China. Since 2010, he has been pursuing his Ph.D degree in the Advanced Technology Development Centre (ATDC), Indian Institute of Technology, Kharagpur.

**Amit Patra** received the B.Tech., M.Tech. and Ph.D. degrees from the Indian Institute of Technology, Kharagpur in 1984, 1986 and 1990 respectively. During 1992-93 and in 2000 he visited the Ruhr University, Bochum, Germany as a Post-Doctoral Fellow of the Alexander von Humboldt Foundation. He joined the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur in 1987 as a faculty member, and is currently a Professor. He was the Professor In-Charge, Advanced VLSI Design Lab, at IIT Kharagpur during 2004-07. Between 2007 and 2013 he served as the Dean (Alumni Affairs and International Relations) at IIT Kharagpur. His current research interests include power management circuits, mixed-signal VLSI design and embedded control systems. He has guided 17 doctoral students and published more than 200 research papers in various Journals and Conferences. He is the co-author of two research monographs entitled General Hybrid Orthogonal Functions and Their Applications in Systems and Control, published by the Springer Verlag in 1996, and Nano-Scale CMOS Analog Circuits Models and CAD Tools for High-Level Design, published by CRC Press in 2014. Dr. Patra received the Young Engineer Award of the Indian National Academy of Engineering in 1996 and the Young Teachers Career Award from the All India Council for Technical Education in 1995. He has been a Young Associate of the Indian Academy of Sciences during 1992-97. He is a member of IEEE (USA), Institution of Engineers (India) and a life member of the Systems Society of India.

**Siddhartha Mukhopadhyay** received his B.Tech. (Hons.), M.Tech. and Ph.D., all from IIT Kharagpur in 1985, 1987, and 1991, respectively. He is a professor in the Department of Electrical Engineering, IIT, Kharagpur, at present. His current research interests are in aerospace tracking, guidance and control, estimation and control theory, process monitoring, automotive failure diagnosis and prognosis, industrial automation and behavioral modeling of AMS VLSI circuits for CAD and test development. He is involved with several consulting and research projects in the above areas sponsored by the government research funding agencies of India and the USA, such as DRDO, ISRO, DIT, Ministry of Power, and industrial organizations including SAIL, CESC, General Motors USA, National Semiconductor Corporation, USA, Freescale Semiconductor, USA. He was the Thrust Area Co-lead for IVHM of the General Motors Collaborative Research Lab on Electronic Controls and Software at IIT Kharagpur. Dr. Mukhopadhyay has received the Young Scientist Award from INSA and the Young Engineer Award from INAE. He is a member of the Editorial Board of the Journal of Systems Science and Engineering and the Journal of IETE. He was the National Coordinator of NPTEL-I in the discipline of electrical engineering.
D-matrix Based Fault Modeling for Cryogenic Loading Systems

Anuradha Kodali¹, Ekaterina Ponizovskaya-Devine², Peter Robinson³, Dmitry Luchinsky⁴, Anupa Bajwa⁵, Michael Khasin⁶, Jose Perotti⁷, Barbara Brown⁷

¹University of California Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035
anuradha.kodali@nasa.gov

³,⁵NASA Ames Research Center, Moffett Field, CA, 94035
peter.i.robinson@nasa.gov
anupa.r.bajwa@nasa.gov

²,⁶SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035
ekaterina.v.ponizovskayadevine@nasa.gov
michael.khasin@nasa.gov

⁴Mission Critical Technologies, Inc., El Segundo, CA, 94035, USA
dmitry.g.luchinsky@nasa.gov

⁷Kennedy Space Center, Florida, 32899, USA
jose.m.perotti@nasa.gov
barbara.l.brown@nasa.gov

ABSTRACT

The study is motivated by NASA plans to develop technology for an autonomous cryogenic loading operation including online fault diagnostics as a part of Integrated Health Management system. For years, the diagnostic modeling effort is performed in many paradigms. None of these paradigms independently can provide a complete set of efficiency metrics: better diagnostics, lower run-time, etc. D-matrix, a causal 0-1 relationship between faults and tests, is proposed as a single representation between different model-based diagnostic methods for comparison and communication. This framework is suitable to create a common platform for communication via D-matrix for systems engineering process. The knowledge transfer between modeling techniques is done via D-matrix. In addition, D-matrix provides a common paradigm to compare the embedded knowledge and performance of heterogeneous diagnostic systems. D-matrix is generated from physics models to be used with faster run-time performance D-matrix based diagnostic algorithms. Additionally, we will also investigate if the derived D-matrix and thereby the physics model is sufficient and accurate for efficient diagnostics via iDME tool.

1. INTRODUCTION

Systems engineering is an important field to design and manage complex engineering systems during their life cycle. According to the NASA Systems Engineering Handbook, System Engineering is a robust approach to the design, development, test, evaluation and operation (DDT&E) of cyber-physical systems. In simple terms, the approach consists of identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals (NASA Systems Engineering Handbook, 2007). In this paper, we will present integrated system health management (ISHM) techniques as a systems engineering process via a D-matrix framework. This common model can be migrated throughout the DDTEO process; thus enabling cost-effective system design to operations for NASA missions.

For systems engineering process, system health management is very critical during design, development, operation, and life cycle management of system components (Johnson, Gormley, Kessler, Mott, Patterson-Hine, Reichard, Karl & Scandura, 2011). This process is designed to improve system dependability while in operation. ISHM
is a parallel capability across the entire system whose objective is to avoid failures where possible, but primarily reverts the system back to nominal functional behavior. Even though ISHM is very critical for system’s operation, it is not fully accepted as an integral part in systems engineering process. This is partly due to the lack of a comprehensive framework that can well-define the requirements and knowledge in a simplistic way and can be easily interpreted by system engineers and health management community.

In ISHM process, system anomalous behavior is defined by low-level component failures. Fault diagnosis, specifically deals with detecting, isolating, and identifying the cause of failure. There are many fault diagnosis methods mainly categorized into model-based, data-driven, and knowledge-based. In this paper, we are defining a common representation for model-based methods via diagnostic matrix (D-matrix) (Luo, Tu, Pattipati, Qiao, & Chigusa, 2006). This is to give a global perspective for ISHM process in terms of the overall system. Traditional Hazard control lacks this global view to deal with cross-subsystem failure propagations.

D-matrix is a causal representation between faults and tests with 1 representing the relationship that the test can detect corresponding failure in the component and 0, otherwise. Our idea is to present D-matrix suitable to systems engineering process. This is performed in 2 ways. Firstly, D-matrix is defined as a communication platform between diagnostic modelers and system engineers. It is an ideal representation that can be easily understood by system engineers to approve or make changes with its closer to human reasoning. Secondly, it acts as a common conceptual diagnostic framework for knowledge transfer and compare among different diagnostic models. Importantly, it will help to analyze for the best diagnostic model representation. Additionally, any diagnostic model can be analyzed for errors using a tool called iDME via its representation as D-matrix. This is a simple effort compared to trying to analyze the original model itself for efficiency.

Diverse modeling techniques have different ways to interpret diagnosis. For years, the diagnostic modeling effort is performed in many paradigms. Fault trees (Vesely, 2002), failure modes and effects analysis (FMECA), graph-based dependency models (Deb, Pattipati, Raghavan, Shakeri & Shrestha, 1995) are some examples. None of these paradigms independently can provide a complete set of efficiency metrics: better diagnostics, lower run-time, etc. But, one thing that is common among all these techniques is the implicit knowledge of D-matrix. Not all techniques generate D-matrix for their diagnosis purpose. But, the information about fault-test dependencies can be easily established for any model via simulations or reachability analysis (Skiena, 2011) or interpretation of the model by an expert. This is why, as discussed earlier, D-matrix can serve as the common representation across models. To support this idea, in this paper, we will present the preliminary study to build D-matrix from closer to real system physics models operating at different system modes. Additionally, we will analyze the generated D-matrix via iDME tool for sensor optimization and diagnostic performance (Kodali, Robinson, & Patterson-Hine, 2013).

We will demonstrate deriving D-matrix for the cryogenic transmission line that includes pipes of different diameter, control and dump valves. The cryogenic transmission is a high-fidelity first-principles physics model. Thus, deriving D-matrix from this model would help to achieve run-time diagnostic performance and sensor optimization. The causal 0-1 relations between faults and responses of pressure and temperature sensors will be obtained empirically by simulations of a moving front homogeneous two-phase physics model of cryogenic chill down in transmission lines. There will be associated test logic to determine if the sensor measurements represent nominal (0) or any faulty condition (1). Specifically, the generated D-matrix contains more than one system mode; thus, the modeled faults have different signatures in different system mode. D-matrix always represents the system in a single system mode. Thus, multiple D-matrices are required, one for each system mode. But, in this paper, we built one aggregated D-matrix with each row corresponding to a failure mode and system mode. We will also investigate if the derived D-matrix is sufficient to obtain efficient diagnostics performance with the existing sensors. In this paper, we used the same simulated data for building and then validating the D-matrix. By doing so, we are training the model to correct answer. But, in our case, as we are analyzing only for observability, using the same data should be fine. Generally, it is advisable to have different datasets for both.

Thus, this paper presents the methodology to generate and analyze D-matrix from high-fidelity physics model of the cryogenic transmission line. This is our first step to define a unified systems engineering process across different modeling techniques. In Section 2, we will discuss D-matrix and iDME tool. In Section 3, we will describe the model for cryogenic transmission line. We will elaborate the contributions of this paper in Sections 4 and 5. The first contribution is to generate D-matrix from physics model of cryogenic system. This is done by simulating data from the physics model. Secondly, the generated D-matrix will be evaluated for diagnostic performance and sensitivity towards the defined test logic of each test. This is presented in Section 5. In this paper, we demonstrate that the current model is only partially observable and thus to improve efficiency, more tests need to be added. Tests need to be designed to disambiguate an ambiguous set of faults accordingly. This acts as guidance for the system designer. In Section 6, we will briefly discuss the innovation of this research. We will summarize the findings and present the future work in Section 7.
2. Diagnostic Matrix (D-Matrix) and Testability Analysis via iDME Tool

Most dependency modeling techniques represent the system in the failure space. It is sufficient to model only the fault propagation to various monitoring points (tests). Thus, this type of dependency modeling captures only the minimum necessary information. This is contrary to the regular qualitative and quantitative techniques (Kuijpers, 1993). They require complete specification of system components, the state and observed variables associated with each component, and the functional relationships among the state variables. Acquiring this precise information is not always possible with increasing complexity in systems. Even after modeling, it will be difficult to analyze these models for testability and diagnostic performance.

D-matrix provides the required simplistic view for our purpose that results in lesser footprint during real-time implementation and can be applied to large-scale systems with faster processing time. This matrix is also popularly known as dependency matrix, fault dictionary, or fault signature matrix. This matrix is obtained from directed graphs based on first principles via reachability analysis. Each test is analyzed to find the corresponding observed failure source (Deb, Pattipati, Raghavan, Shakeri & Shrestha, 1995). The dependency between a failure source and test is defined if the test can detect the fault when it occurs. This is identified as “1” in the D-matrix, otherwise it is “0”. These Boolean expressions can be conceived as test fail (1) or pass (0) in real sense. More than one test can detect a single failure source. Each test is identified by corresponding logic that determines if the test has failed or passed. The test logic can range from simple threshold checks to complicated signal processing techniques like Fourier transforms or statistical or trending tests. Dependency models include both D-matrix and test logic. The concept of D-matrix is popularized commercially by TEAMS software which employs multi signal modeling framework (Qualtech Systems Inc.).

The concept of D-matrix is quite popular in aerospace diagnostic community. Due to its widespread usage, it is standardized as “diagnostic inference model” (IEEE Std 1232-2002). Most diagnostic algorithms, for example Bayesian inference, case based reasoning, rule based inference, set partitioning can be applied easily for models based on D-matrix concept (Sheppard & Butcher, 2006).

2.1 iDME Tool: Analyze D-Matrix

The diagnostic information of the system is summarized by D-matrix. Hereafter, all diagnostic analysis is performed using this matrix. In other words, the original model is not required anymore unless the modeler wants to understand the trace back and modify the schematics. But in an another perspective, the major contribution of this paper is to utilize D-matrix to analyze efficacy of the corresponding physics model. In such case, the physics model is modified accordingly based on the findings from the D-matrix via iDME tool.

iDME tool, with the aid of supervised data (data is labeled with corresponding nominal or faulty state), debugs and proposes repair strategies to D-matrix by coordinating with the decision maker (user) (Kodali, Robinson, & Patterson-Hine, 2013).

iDME is defined as a combined process of computer and user decisive mechanisms where computer provides
platform of the diagnostic analysis of the system model with the aid of supervised data and the decision maker performs the role of accepting/declining repair strategies based on the analysis of performance metrics and technical expertise (see Figure 1). Five D-matrix repair strategies are identified arranged in ascending order of cost effectiveness. These strategies range from addressing duplicity in faults and tests, repairing the fault universe to accommodate lower/higher level fault modeling (re-define the level of fault modeling by adding or removing rows), repairing/changing the wrapper/test logic, repairing 0’s and 1’s in the D-matrix entries, and adding/removing tests. They are included in an iterative loop to experiment for better performance along with the decision maker. The performance criteria are based on fault detection and isolation metrics derived from the mission objectives by the user. Then, the decision maker accepts/declines the repair strategies based on before and after performance. More details of this framework can be found in (Kodali, Robinson, & Patterson-Hine, 2013).

The efficiency of the inference process is directly proportional to better coverage of the defined failures by tests and the separability between the rows of D-matrix (fault signatures) (Sheppard & Simpson, 1992). In this analysis, we will test for ambiguous, hidden, and masking faults (Kodali, Singh & Pattipati, 2013). Two failures are ambiguous if their fault signatures are similar i.e. the two corresponding rows are identical. The failures which are masked by a fault are its hidden failures, i.e., the fault signature of a hidden failure is the subset of the signature of the fault. A masking false failure occurs when the symptoms of two or more failures add up to mimic the failure of an unrelated element, i.e., the combination of their signatures produces the signature of another fault. The existence of these types of failures indicates partial coverage of the model. This reduced observability is due to increased failure definitions while the monitoring points are reduced. In such a case, the solution would be to add more tests to improve detectability of the failures. We will provide guidance about what tests need to be designed in terms of which fault they need to detect or isolate from other faults.

For the framework proposed in this paper, data is collected from high fidelity physics model (cryogenic) which is closer to reality. This is done by injecting faults and then collecting corresponding sensor data from the computer-aided simulation model. The simulated model is very close to reality as will be discussed in the next section. The same data is also used to build D-matrix.

### 3. DESCRIPTION OF CRYOGENIC SYSTEM

The fault diagnostics was applied to the chill down stage of cryogenic loading operation in the experimental cryogenic loading system that has been developed at the Kennedy Space Center to test autonomous regimes of operation (Johnson, Notardonato, Currin, & Orozco-Smith, 2012). The KSC cryogenic testbed system consists of a 6,000 gallon storage tank is connected to a 2,000-gallon vehicle tank with pipes of different diameters, control and dump valves, pump and sensors to measure pressure and temperature along the transfer line. The liquid motion through the transfer line is driven by an elevated pressure in the storage tank, which at working conditions is designed to suppress potential boiling of liquid cryogen at the operating temperature. During the initial stages of the loading operation, when the transfer line is still at high temperatures a substantial part of the incoming nitrogen boils increasing the pressure in the transfer line and slowing down the cryogen liquid motion. A set of control valves allows liquid flow in the corresponding segments of the pipe and dumping valves are to be opened sequentially to maintain the liquid flow and to allow for a gradual chill down of the system as the hot gas is substituted sequentially by the cold vapor, the two-phase mixture, and the cryogenic liquid.

A set of the valve open/close positions together with the dynamics of the storage tank pressure constitutes the filling protocol, which depends on the design of the experiment. A set of the temperature (TT102, TT162, TT146, TT149, TT156, TT191) and pressure (PT104, PT161, PT145, PT148, PT152, PT158) sensors allows for control of the
conditions of cryogen flow (Table 2). The faults in the valves, mass and heat leaks, clogging in the pipes could cause the pressure and temperature deviate from the nominal values (see Table 2).

The total length of the transfer line is about 45 m. The diameter of the stainless steel pipe varies along the pipeline from 0.1524 m to 0.0254 m. The thickness of its walls is approximately 3 mm. Initially, the storage tank is full and the vehicle tank is empty with the flow path between the tanks blocked. An ullage pressure in both tanks equal to the atmospheric pressure. Then the storage tank is pressurized first, and the chilldown begins. The dump valves CV112, CV117 and CV120 regulates the cryogenic flow and their positions for nominal regime are shown at Fig.2. In this study we consider the list of faults presented in the Table 2. The deviation from the sensors data over the margin values was used as tests. For each sensor we had two tests that represented deviation above the nominal value and below the nominal value (Table 1).

We use the homogeneous moving front model (Hafiychuk, Foygel, Ponizovskaya, Smelyanskiy, Watson, Brown, B & Goodrich, 2014) to simulate both the nominal and the fault regimes. The homogeneous model describes the properties of the two-phase flow in terms of the mixture density ($\rho$) and the mixture enthalpy ($h$)

$$\begin{align*}
\rho &= \alpha \rho_v + (1 - \alpha) \rho_l; \\
\rho h &= \alpha \rho_v h_v + (1 - \alpha) \rho_l h_l; \\
h &= x h_v + (1 - x) h_l
\end{align*}$$

(1)

here $\alpha$ is a void fraction, $\rho_v$ and $\rho_l$ is the density of the saturated vapor and liquid, $h_v$ and $h_l$ is the enthalpy of the saturated vapor and liquid, $x$ is mass quality.

Assuming that phasic velocities for the gas and liquid are the same and equal to $u$, we can write the mass, momentum and energy conservation equation in the reduced form:

$$\begin{align*}
\rho \dot{\alpha} + (\rho u) \frac{\partial \alpha}{\partial z} &= 0, \\
(\rho u) \frac{\partial u}{\partial z} + (\rho u^2) \frac{\partial u}{\partial z} &= -p \frac{1}{A} (\tau_w l_w) + \rho g \sin \theta, \\
(\rho c) \frac{\partial \rho c}{\partial z} + (\rho c h) \frac{\partial \rho c h}{\partial z} &= \frac{1}{A} q_w l_w.
\end{align*}$$

(2)

Here $\tau_w$ is the friction losses, $l_w$ is the pipe length, $g$ – gravity, $\theta$ is the angle if the pipe, $q_w$ is the heat transfer from the pipe walls to the mixture and $A$ is the cross section area of the pipe.

All the interphase heat and mass transfer terms and interface friction terms cancel each other due to so-called jump conditions. The wall temperature ($T_w$) is determined by the reduced energy conservation equation in the form

$$c_w \rho_w A_w \frac{\partial T_w}{\partial t} = H_{fs} l(T - T_w) + H_{amb} l(T_{amb} - T_w)$$

(3)

Here $c_w$ is the specific heat for the pipe walls material, $\rho_w$ is the density of the pipe walls material, $A_w$ is the walls surface area, $H_{fs}$ is the heat transfer coefficient from the walls to the mixture and $H_{amb}$ is the heat transfer coefficient from the ambient to the walls.

Additional important simplification to speed up the calculations is to neglect inertia in the momentum equation, which reduces this equation to the following form
\[
\frac{1}{\rho A^2} \left[ \frac{x^2}{\alpha p_e} + \frac{(1-x)^2}{(1-\alpha) p_i} \right] \right|_t = -p_x - \left( \frac{f}{A} + K \right) \left( \frac{1}{2 \rho^2} \right) \frac{d^2}{dt^2} - \rho g \sin \theta,
\]

where \( f \) and \( K \) are frictional losses and minor losses estimated at the center of the cell on the staggered grid. The solution of the equations (1)-(4) is achieved using a two step Adams-Moulton scheme (Hairer, Norsett & Wanner, 1993), (Hafiychuk, Foygel, Ponizovskaya, Smelyanskiy, Watson, Brown, B & Goodrich, 2014). The set of equations for the mass and energy conservations (1st and 3rd equations in (2)) are solved to find new pressure and mass using of old velocities, then, new velocities are found by solving quasi-steady momenta equation (4).

The order of the steps may vary depending on the initial and boundary conditions. The solution of the energy conservation equation for the wall temperature is decoupled from the solution of the fluid equations and is performed in the end of each time step for both algorithms.

In the context of the model based fault diagnostics, it is important to ensure that models produce time accurate predictions for the cryogenic loading dynamics. For this purpose the model was verified and validated. The versions of the code developed for the cryogenic health management applications were tested using multiple flow conditions and verified by comparison of the model performance with the predictions of the baseline model of the cryogenic chilldown developed in SINDA/FLUINT (Kashani, Ponizovskaya, Luchinsky, Smelyanskiy, Sass, Brown, & Patterson-Hine, 2014).

The model was validated on the KSC cryogenic testbed experimental data. The Figure 2 shows the comparison between the simulated data (red line) and the data from the corresponding pressure and temperature sensors (black line). The model accurately capture the main pressure and temperature transients observed during chill down of the cryogen transfer line.

4. Generate D-Matrix from Cryogenic Model

For the demonstration purpose, we considered 7 faults and 24 tests (12 sensors) as shown in Tables 1&2. Failures correspond to valves. These valves can either stuck open or closed manifesting as failures in the system by affecting the liquid flow. Each sensor corresponds to two tests with maximum and minimum threshold limits, respectively.

Each failure mode has one corresponding supervised data file. Each file is simulated over 1600s from the computer-aided cryogenic simulation model. As the model is closer to real-time model, the simulations are as good as operational data (shown in Figure 2). The fault is injected at the start time of the file and is present throughout 1600s. Thus, there are 7 files in total. For this paper, we use these files to build D-matrix and then analyze for sensor optimization.

The cryogenic model operates at different system modes depending on the valve positions. The opening position of valves is shown in Figure 3. We find that the time plot is divided into 5 sections at 500, 700, 1100, 1400, and 1600 seconds. Each section determines a system mode. In D-matrix context, each failure is defined by the corresponding fault signature in terms of 0’s and 1’s for each test. But these failures behave differently with respect to test detectability depending on the system mode. They can have different fault signatures. Traditionally, in such a case, there are multiple D-matrices for each system mode. In this paper, we will construct a single aggregated D-matrix with each row corresponding to failure mode and system mode. That means, each failure mode has multiple representations with each system mode.

Another issue with the current model is that the sensor measurements are influenced by failures after a delay. This is the case with the temperature sensors because it takes some time for the temperatures to raise or drop due to failure. Generally, the knowledge about this delay is incorporated in the inference algorithm. But, as the current design analysis is off-line, we do not consider analyzing for these delays and use sampling for every 100s to offset the delay effect. We focus on the test design efficiency assuming that the delay is inevitable.

Here, we will enumerate the steps to generate D-matrix using the data simulated from the physics model.

1. Define the list of failure modes and tests. The failure modes are duplicated in each system mode.
2. Simulate data corresponding to each failure mode. At least one file is required for each failure mode.
3. Define test logic corresponding to each test. We employed simple threshold checks for each test. If the sensor measurement goes above or below 5% the simulated value, the corresponding test is failed.
4. Generate test outcomes (passed (0) or failed (1)) based on sensor measurements and test logic. This is done for all the supervised data files available.

5. As the tests are associated with delays to detect failures, the data is sampled at the rate of 100s. This is done to offset the delay effect. Thus, we have 16 time points in each data file for the data collected over 1600s.

6. Now, we start generating the fault-test relationship for each file sequentially. Ideally, after analyzing the data file for a fault, 5 fault signatures corresponding to each system mode are generated.

7. The D-matrix entry is 1 if the test fails at least once during the selected data points. That means, during the system mode, the test should at least fail once to be included as 1 in the entry for the corresponding fault and system mode.

8. The generated D-matrix for each failure mode in each system mode is listed in Appendix. Each failure mode is appended with underscore and the corresponding system mode number. But, for space convenience only the corresponding failure mode no. is appended with underscore and system mode no. The columns and rows with all zeros are removed.

5. ANALYSIS OF D-MATRIX WITH IDME TOOL

We started the analysis with the generated D-matrix via iDME Tool. In this paper, we aim at providing guidance to design extra tests to improve diagnosability. For this analysis, we did not consider to repair test logic. So, we analyze only D-matrix with the aid of simulated data. This analysis is very similar to the regular testability analysis, except for the fact that the data is used to validate the model. This will be helpful to assess the model's ability to withstand noise in sensor measurements. Simulations can be done with various noise levels and the resulting D-matrix can be analyzed for efficiency.

The generated D-matrix is partially observable; thus there are duplicate rows present. The duplicate failure modes are listed in Table 3. It is imperative that additional tests need to be developed to be able to isolate among duplicate faults (only when recovery action is different). There is no other way to differentiate among these faults. Similarly, there are duplicate (redundant) columns (see Table 4). They are left as they are because they can be useful for other set of failures not considered here.

As we can see that the faults corresponding to open and close functions of a valve are not duplicate, but their faulty behavior is similar to the faults in the open and close positions of other valves, respectively. During system mode 2, stuck close failures corresponding to cv112 and cv120 are duplicate. Similarly, during system modes 3, cv112, cv120, and cv117 stuck close have similar signatures. Faults 2, 4, 5, 6, and 7 are duplicative during system modes 4 and 5. These faults are only detected by TT191 sensor. This indicates that there should be additional tests to isolate among these faults. The new test can analyze the existing sensor TT191 with new test logic or can be a new sensor. In a similar way, we can analyze other duplicate faults to design appropriate tests. Generally, duplicate faults are grouped if the recovery action is similar. But, this is not the case here.

<table>
<thead>
<tr>
<th>Failure Mode 1</th>
<th>Failure Mode 2</th>
<th>Failure Mode 1</th>
<th>Failure Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault1_2</td>
<td>Fault4_2</td>
<td>Fault1_5</td>
<td>Fault2_5</td>
</tr>
<tr>
<td>Fault1_3</td>
<td>Fault2_3</td>
<td></td>
<td>Fault4_5</td>
</tr>
<tr>
<td></td>
<td>Fault4_3</td>
<td>Fault6_5</td>
<td></td>
</tr>
<tr>
<td>Fault1_4</td>
<td>Fault4_4</td>
<td>Fault7_4</td>
<td></td>
</tr>
<tr>
<td>Fault6_4</td>
<td>Fault3_1</td>
<td>Fault5_1</td>
<td></td>
</tr>
<tr>
<td>Fault2_1</td>
<td>Fault4_1</td>
<td></td>
<td>Fault5_2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fault5_3</td>
</tr>
<tr>
<td>Fault3_2</td>
<td>Fault3_3</td>
<td>Fault3_4</td>
<td>Fault5_4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Test Name</th>
<th>Test Name</th>
<th>Test Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT104High</td>
<td>PT161High, PT145High</td>
<td>TT104Low</td>
<td>TT146Low</td>
</tr>
<tr>
<td>PT104Low</td>
<td>PT161Low</td>
<td>TT104Low</td>
<td>TT149Low</td>
</tr>
<tr>
<td>TT104Low</td>
<td>TT162Low</td>
<td>TT146High</td>
<td>TT149High</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fault</th>
<th>Hidden faults</th>
<th>Fault</th>
<th>Hidden faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault1_3</td>
<td>Fault1_2</td>
<td>Fault2_2</td>
<td>Fault1_2</td>
</tr>
<tr>
<td></td>
<td>Fault1_4</td>
<td>Fault2_1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault1_5</td>
<td>Fault3_2</td>
<td>Fault3_4</td>
</tr>
<tr>
<td></td>
<td>Fault2_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault2_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault6_3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another important problem with partial observability is the existence of hidden and masking faults. This results in ambiguous groups during diagnosis. These groups are always hard to isolate without proper system design. This can be avoided by adding more tests. The list of hidden faults after grouping duplicate faults is given in Table 5. In this case, if the hidden fault is of the same failure mode but a different system mode, then there is no need to design a new test because the recovery action is similar. But, remember that there are duplicate faults for the faults listed in Table 5. So, by analyzing Tables 3 and 5 together, it is understood that the system failures in cv117, cv120, and cv112 are either duplicative or hidden during stuck open or close. But, another consideration we ignored in this paper is the delay after which the test fails for the corresponding failure. This could probably alleviate the ambiguity in the current model. This will be pursued in future research.

Thus, we need to carefully analyze the existing set of tests along with their logic and understand if it requires additional sensors or additional tests that analyze the existing sensors differently. In this paper, we will not include the follow-up strategy to find the placement for the additional required sensors.

6. INNOVATION

This research is en-route to establish a singular ISHM framework to communicate with systems engineering process. This research will result to provide a unified and simplistic view to ISHM process that can be easily interpreted by system engineers; thereby integrating it with systems engineering process. The proposed framework is neither a new method for better diagnostics nor a replacement to the existing model-based techniques, but is an integrated framework that works to better each of these models. This work can be viewed as a common platform that helps in evaluating design and reducing errors in each individual diagnostic model. This is done by providing a better correspondence and unified platform for different communities through a simplistic interpretable view via D-matrix. This will advance the field of ISHM to be cost-effective.

7. CONCLUSIONS AND FUTURE RESEARCH

The idea here is to promote D-matrix as the common framework to aid a simplified communication platform between system engineers and diagnostic modelers. Additionally, the knowledge transfer between different modeling techniques can be done via D-matrix. This will be instrumental to create a common model and also helps in improving each individual model. Also, this will help in achieving better diagnosis across all models by carefully choosing the best modeling technique, best representation of system design.

This paper focuses on initial steps in this process. The framework is laid out at the lower level. For this, we generated D-matrix from high fidelity physics model of cryogenic system. Then, the model is validated via iDME tool for effective diagnosis by proposing additional tests to tackle duplicate and hidden faults. In our future work, we will further analyze this system with more number of faults and tests. Also, the design of the physics model will be accordingly altered, thereby producing high efficiency diagnostics. We will also compare computational performance of D-matrix based inference algorithms to full-scale physics models. We will also consider propagation delays either as part of the model or inference algorithm.

Another key aspect of future research is to provide more information on what type of test needs to be designed and corresponding placement. We did not explore this field yet, but will be a good addition in our iDME framework. We further focuses on translating the analysis on D-matrix to original models, thereby making each model effective.

Single D-matrix may not be always sufficient to represent a system especially during transient state. Thus, it requires multiple D-matrices for system representation and inference. But, in this paper, we introduced to build a single D-matrix – aggregate of all multiple D-matrices corresponding to each system mode. Also, additional information may be required, for example couplings between faults. This extra information needs to be properly represented in addition to D-matrix for proper utilization across the board during inference. Our future research will focus on this aspect of how best to represent D-matrix and the additional information. We will also streamline this process for systems engineering. In summary, the goal of this proposed process is to make model-based diagnostic field cost-effective and ready for verification and validation during systems engineering process.

REFERENCES


**BIographies**

**Anuradha Kodali** received the Ph.D. degree in electrical and computer engineering from the University of Connecticut (UConn), Storrs, in 2012. Currently, she is Assistant Research Scientist at University of California Santa Cruz. She served as a Reviewer in several IEEE journals. Her research interests include data mining, pattern recognition, fault detection and diagnosis, and optimization theory.

**Ekaterina Ponizovskaya-Devine** Dr. Ekaterina Ponizovskaya-Devine is a research scientist in SGT since 2009. Her current work is related to integrated health management and control of cryogenic propellant loading systems and two-phase cryogenic flow modeling. She obtained her PhD at Moscow institute of Physics and Technology in Physics, working on stochastic resonance in non-equilibrium electron-hole plasma. She was a post-doc in Spanish National Research Council, working in photonics, nanotechnology, negative index materials. In 2005 she had Visiting Scholar position at HP Lab working in nanotechnology. Her research interests are physics based models, two-phase flow, nanotechnology, physics model based fault diagnostics, optimization and control.

**Dmitry G. Luchinsky** Dmitry G. Luchinsky is a senior research scientist in MCT Inc. He obtained his MSc and PhD in physics in Moscow working on nonlinear optics of semiconductors. He is an author of more than 100 publications. He has been on a number of occasions a Royal Society Visiting Fellow and a NASA visiting scientist. He worked as a senior scientific researcher in VNII for Metrological Service (Moscow, Russia) and as a Senior Research Fellow in Lancaster University (Lancaster, UK). His research interests include nonlinear optics, stochastic and chaotic nonlinear dynamics, dynamical inference, fluid dynamics, ionic motion. His research is currently focused on theory and CFD of gas dynamics and cryogenic flows.

**Michael Khasin** Dr Michael Khasin is a Senior Researcher in SGT Inc., working at NASA Ames Research Center. He holds B.Sc. in Physics (Honors Program), 2001, M.Sc. in physics, 2003, and PhD in chemical physics, 2008, from the Hebrew University of Jerusalem. As a postdoctoral researcher at Michigan State University, Massachusetts Institute of Technology, and University of Michigan, Michael worked on the theory of non-equilibrium systems...
and their control and transport in disordered systems. His research interests include applied physics, nonlinear and stochastic dynamics, large fluctuations and their control in non-equilibrium systems and efficient simulation of complex dynamics. He has been an invited speaker to many international and national conferences and workshops. Currently his research is focused on the theory of heat transfer, fluid dynamics, and fluid-structure interaction.

APPENDIX

<table>
<thead>
<tr>
<th>Fault</th>
<th>PT1 04H</th>
<th>PT1 04L</th>
<th>TT1 04H</th>
<th>TT1 04L</th>
<th>PT1 61H</th>
<th>TT1 61L</th>
<th>TT1 62H</th>
<th>TT1 45H</th>
<th>TT1 46H</th>
<th>TT1 46L</th>
<th>TT1 48H</th>
<th>TT1 49H</th>
<th>TT1 49L</th>
<th>TT1 56H</th>
<th>TT1 56L</th>
<th>TT1 91H</th>
<th>TT1 91L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3.1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Fault diagnostics and evaluation in cryogenic loading system using optimization algorithm

Ekaterina Ponizovskaya-Devine¹, Dmitry G Luchinsky², Anu Kodali³, Michael Khasin⁴, Dogan Timucin⁵, Jarred Sass⁶, Jose Perotti⁷, Barbara Brown⁸

¹,³,⁴ SGT, Inc., Greenbelt, MD, 20770, USA
ekaterina.v.ponizovskayedevine@nasa.gov
anupa.r.bajwa@nasa.gov
michael.khasin@nasa.gov

² Mission Critical Technologies, Inc., El Segundo, CA, 90245, USA
dmitry.g.luchinsky@nasa.gov

⁵ NASA Ames Research Center, Moffett Field, CA, 94035, USA
dogan.timucin@nasa.gov

⁶,⁷,⁸ Kennedy Space Center, Kennedy Space Center, FL, 32899, USA
jose.m.perotti@nasa.gov
jared.p.sass@nasa.gov
barbara.l.brown@nasa.gov

ABSTRACT

Physics-based approach to the cryogenic flow health management is presented. It is based on fast and time-accurate physics models of the cryogenic flow in the transfer line. We discuss main features of one of these models – the homogeneous moving front model – and presents results of its validation. The main steps of the approach including fault detection, identification, and evaluation are discussed. A few examples of faults are presented. It is shown that dynamic features of the faults naturally form a number of ambiguity groups. A D-matrix approach to optimized identification of these faults is briefly outlined. An example of discerning and evaluating faults within one ambiguity group using optimization algorithm is considered in more details. An application of this approach to the Integrated Health Management of cryogenic loading is discussed.

1. INTRODUCTION

Future of the space exploration requires development of the Integrated Health Management (IHM) of cryogenic systems on the ground and in space (Chato, 2008). Cryogenic propellant loading operations are some of the most complex, critical activities in launch operations (Johnson, Notardonato, Currin, & Orozco-Smith, 2012). Full potential of autonomous intelligent health management of such systems on the ground and in space can be achieved by using fast and time-accurate physics models of the two-phase cryogenic flow.

Predicting the behavior of the two-phase flows is a long-standing problem (Prosperetti & Tryggvason, n.d.). To solve this problem we developed a hierarchy of the models of two-phase cryogenic flows (Luchinsky, Smelyanskiy, & Brown, 2014a), verified and validated their performance (Luchinsky, Smelyanskiy, & Brown, 2014b; Hafiychuk et al., 2014, 2015). The models are fast and accurate enough to allow for on-line applications to the cryogenic health management, including on-line solution of the optimization problem.

In this paper we describe the progress in development of such physics based approach to fault isolation and recovery for cryogenic loading operation. We present moving front model of the two-phase cryogenic flow (Zhang & Zhang, 2006; Hafiychuk et al., 2014, 2015) and the results of its validation. Next we discuss the application of this model to the creation and extension of the digital library of faults related to the cryogenic loading operation. We demonstrate that these faults naturally form a number of ambiguity groups with similar dynamic features within each group. We propose to use D-matrix formalism (Sheppard & Simpson, 1996) to enhance
identification of the faults with multiple ambiguity groups. To optimize the performance of the D-matrix method we propose to use iDME tool (Singh et al., 2009). The focus of the current paper is on the discussion of the development of on- and off-line optimization tools that can be used to discern and evaluate faults within one of the ambiguity groups. We present the results of numerical test in which one of the faults (valve stack open) was identified within ambiguity group. We use optimization algorithms to discern this fault from other possible faults (another valve stack open and a gas leak). We verified the performance of three optimization algorithms (local unconstrained nonlinear optimization, direct search, and Markov Chain Monte Carlo (MCMC)). We demonstrated that direct search algorithm can correctly identify and evaluate the fault. Finally, we provide the conclusions and discuss future work.

2. Model

The homogeneous model of the two-phase flow describes the flow dynamics and thermodynamics in terms of conservation laws for the mass, momentum, and energy

\[ \rho_t + (\rho u)_z = 0 \]

\[ (\rho u)_t + (\rho u^2)_z = -p_z - \frac{1}{\lambda}(\tau w)_w \]

\[ (\rho e)_t + (\rho u h)_z = \frac{1}{\lambda} \bar{q}_w l_w \]

written for the mixture density \( \rho \) and enthalpy \( h \)

\[ \rho = \alpha \rho_v + (1 - \alpha) \rho_l; \quad h = x h_v + (1 - x) h_l \]

and coupled to the equation for the wall temperature

\[ \rho_w c_w d_w \frac{\partial T_w}{\partial t} = H_w (T - T_w) l_i + H_a (T_a - T_w) l_s. \]

The set of the model equations is closed by adding equation of state to the system

\[ \rho_{(g,t)} = \rho_{(g,t)} \left( p, h_{(g,t)} \right). \]

The moving front version of this model (Zhang & Zhang, 2006) allows one to incorporate up to three coexisting states of the fluid (vapor, liquid, and their mixture) within one control volume, therefore, increasing fidelity of the model.

In the particular version of the moving front model developed in (Hafiychuk et al., 2014, 2015) the momentum equation is solved in the quasi-steady approximation, neglecting inertia terms

\[ (A p w^2)_z + \frac{1}{\lambda} \tau w l_w = -p_z - \rho g \sin \theta. \]

In this approach calculations of the mass fluxes is decoupled from the integration of the conservation equations for the mass and energy. The resulting algorithm allows for very fast and time-accurate predictions of the cryogenic two-phase flow in transfer lines. The speed of the calculations can be further improved by using non-conservative linearized version of the mass and energy conservation equations.

2.1. Vapor region

The specific form of the linearized equations depends on the type of the flow. For example, using Taylor expansion for the speed of the calculations can be further improved by using non-conservative linearized version of the mass and energy conservation equations.

\[ \frac{d p}{dt} = \frac{\partial \rho}{\partial \rho} \frac{dp}{dt} + \frac{\partial \rho}{\partial \rho} \frac{dp}{dh} \]

we obtain for the pure vapor region the following set of equations

\[ \left( \frac{\partial \rho g}{\partial p} - 1 \right) \frac{dp}{dt} + \frac{\partial \rho g h}{\partial h} \frac{dh}{dt} = \frac{(\bar{m}_g - m_{out})}{\Delta z A} \]

For liquid control volume the equations are exactly the same on the substitution of subindex \( g \) (for a gas) on subindex \( l \) (for a liquid).

2.2. Two-phase region

In the case of the two-phase flow region we have (Zhang & Zhang, 2006; Hafiychuk et al., 2014, 2015)

\[ \left( \frac{\partial \rho g h}{\partial p} + (1 - \alpha) \frac{\partial \rho l}{\partial p} \right) + \left( \rho g - \rho l \right) \frac{d h}{dp} \frac{dp}{dt} \]

\[ + \frac{\partial \rho g h}{\partial h} \frac{d h}{dt} = \frac{1}{V} \left( \bar{m}_g - \bar{m}_{out} \right) \]

\[ \left( \frac{\partial \rho g h}{\partial p} - (1 - \alpha) \frac{\partial \rho h}{\partial p} - 1 \right) + \left( \rho g h - \rho h l \right) \frac{d h}{dp} \frac{dp}{dt} \]

\[ + \frac{\partial \rho g h}{\partial h} \frac{d h}{dt} = \frac{1}{V} \left( (\bar{m}_g h - \bar{m}_{out} h_{out}) + \Delta z H_{amb} D (T_w - T) \right) \]

The flow quality is assumed to change linearly with coordinate as follows

\[ x(z) = x_1 + (x_2 - x_1) (z - z_1) / (z_2 - z_1) \]
and the variable \( b(p) \) is defined using correlations (Woldesemayat & Ghajar, 2007)

\[
b(p) = \left( \frac{\rho_l}{\rho_g} \right)^{n_1} \cdot \left( \frac{\rho_g}{\mu_l} \right)^{n_2} - 1 > 0.
\]

### 2.3. Vapor and two-phase region in one control volume

As we mentioned above, to elevate fidelity of the model the mixed control volumes are allowed. Here we consider one example of such control volume filled with vapor and two-phase region. It is assumed that the location of the boundary between two regions \( L_2(t) \) is changing with time. In this case we have to solve two sets of two-dimensional equations for each region considered in two previous sections simultaneously with the equation for the moving boundary (Jensen, Jensen, & Tummescheit, 2002).

In the two-phase region the equations take the form

\[
(\alpha \frac{\partial \rho_\alpha}{\partial p} + (1 - \alpha) \frac{\partial \rho_n}{\partial p}) \frac{dp}{dt} + (\rho_g - \rho_l) \times \\
\left[ \left( \alpha - 1 \right) \frac{d\ln L_2}{dt} + \frac{\partial p}{\partial t} \right] = \frac{m_{in} - m_{g}}{V_2}
\]

and for the vapor region we have

\[
\frac{\partial \rho_3}{\partial p} \frac{dp}{dt} + \frac{\partial \rho_3}{\partial h_3} \frac{dh_3}{dt} + (\rho_3 - \rho_g) \frac{d\ln L_3}{dt} = \frac{m_{g}}{V_3}
\]

These four equations can be solved simultaneously for pressure, enthalpies, and location of the boundary once we assume that the pressure is the same in both regions. We can further speed up calculations if we consider liquid phase to be at saturated conditions in the two-phase region. Similarly, the pressure and enthalpy for the mixture and pure phases can be found in all other cases (Hafiyuch et al., 2014; Jensen et al., 2002).

Once pressure and enthalpies are found one can calculate mass fluxes between the volumes.

### 3. Algorithm

The algorithm is straightforward and consists of two steps. At the first step of the algorithm mass and energy conservation equations are solved for all control volumes once initial conditions for the mass fluxes are provided.

For example for the control volume filled with vapor explicit Euler scheme can be used to advance solution in one time step

\[
\begin{bmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22}
\end{bmatrix}
\begin{bmatrix}
  dp_L \\
  dh_{g,L}
\end{bmatrix}
= \begin{bmatrix}
  f_1 \\
  f_2
\end{bmatrix} dt.
\]

The matrix and vector coefficients in Eq. (3) are given by the following equations

\[
b_{11,L} = \frac{\partial \rho_{\alpha,L}}{\partial p},
\]

\[
b_{12,L} = \frac{\partial \rho_{\alpha,L}}{\partial h_{f,L}},
\]

\[
b_{21,L} = \left( \frac{\partial (\rho_{\alpha,L})}{\partial p} \right)_{\partial L} - 1
\]

\[
b_{22,L} = \left( \frac{\partial (\rho_{\alpha,L})}{\partial h_{f,L}} \right)_{\partial L}
\]

\[
f_{2,L} = \frac{(\bar{m}_{g,h}) - (\bar{m}_{g,h})_{j-1}}{V_L} + \frac{4}{V_L} \frac{\partial \rho_{g,L}}{\partial L} \left( T_{w,L} - T_{g,L} \right)
\]

The node notation convention is shown in Fig. 1. Similar equations hold for all other types of control volumes.

At the second step of the algorithm, the momentum equation is solved as follows

\[
\frac{n_2^2}{\gamma \rho_T} \left[ \frac{1}{V_n} \left( f_n \frac{\rho_{in}}{\gamma \rho_T} + \sum K_n \right) \right] = -\Delta p_j - \Delta x_j \rho_j g \sin \theta.
\]

This equation is applicable when pipe diameters on the both sides of the junction are the same. The subindex \( n \) for frictional factor \( f \) and minor losses \( K \) refers to the control volume on the left or right hand side of the junction.

The stability of this approximation (Thome, 2006) was enforced by the donor-like formulation of the frictional losses for the mass fluxes written in (2) in the following form

\[
\left( A \rho u^2 \right)_{j-1} = \frac{n_2^2}{2A_n} \left( \frac{x^2}{\gamma \rho_T} + (1 - \gamma) \rho_T \right)_{j-1}.
\]

The donor-like formulation assumes that \( n = K \) if the right-hand side of the Eq. (4) is positive and \( n = L \) otherwise.

### 4. Correlations

The thermodynamic and mechanical properties of the two-phase flow within this model are described using the following set of correlations.

The frictional losses coefficients \( f_{1,3} \) in Eq(5) were calculated

![Figure 1. The convention for the nodes and junction notations.](image-url)
using the Swami-Jain (Thome, 2006) approximation of the Colebrook equation:

\[
\frac{1}{\sqrt{f_{1,3}}} = -1.74 \ln \left( \frac{d_c}{3.7D} + \frac{1.26}{\text{Re}_{1,3} \sqrt{f_{1,3}}} \right)
\]

\[
f_{1,3} = 0.25 \left( \log \left( \frac{\epsilon}{3.7D} \right) + \frac{5.74}{\text{Re}^{0.9}} \right)^{-2}
\]

where \( \epsilon \) is roughness, \( D \) - pipe diameter, and \( \text{Re} \) is the Reynolds number determined by the following relation mass flow rate \( \dot{m} \) being mass flow rate.

For the two-phase flow the frictional loss coefficient was calculated according to the Mueller-Steinhagen and Heck correlation (Thome, 2006) is

\[
f_2 = \frac{\rho_f f_1}{\rho_l (x_f - x_l)} \left[ \frac{\beta}{\pi} x^4 - \frac{3}{4} (1 - x)^{4/3} \left[ 1 + \frac{3}{4} \left( \beta - 1 \right) (3 + 4x) \right] \right]^{3/2}.
\]

The Dittus-Boelter approximation (Thome, 2006) was used to calculate the heat transfer:

\[
H_{wf} = \alpha H_{wg} + (1 - \alpha) H_{wl}
\]

\[
H_{wv, wl} = \left( \frac{\kappa}{D} \right) N_{\text{u}} v_t
\]

\[
N_{\text{u}_l, g} = 0.023 \text{Re}_{l, g}^{4/5} Pr_{l, g}^{2/5}.
\]

The boiling enhancement factor was introduced the same way as in the Gungor-Winterton (1987) correlation (Thome, 2006)

\[
H_2 = \eta \text{H}_{0} \left( (1 - \alpha)^{4/5} E \right)
\]

where \( \eta = F_r^{0.1-2F_r} i F_r < 0.05 \) and \( \eta = 1 i F_r > 0.05 \). Here

\[
E(x) = 1 + 3000 \text{Bo}^{0.86} + 1.12 \left( \frac{x}{1-x} \right)^{0.75} \left( \frac{\rho_l}{\rho_g} \right)^{0.41}
\]

is the boiling enhancement factor, \( F_r = \left( \frac{\dot{m}}{\rho_l A} \right) \frac{1}{gD} \) is the Froude number, and \( \text{Bo} = \frac{H_2(T_{w} - T_{f}) A}{\dot{m} (h_l - h_i)} \) is the boiling number.

5. Verification and Validation

The moving front model was verified for cryogenic applications by comparison of the model performance with the results of simulations using SINDA/FLUINT code (Kashani et al., 2014). It was also validated by comparison with experimental data obtained at NIST (Brennan, Brentari, Smith, & Steward, 1966) and at KSC (Hafichuk et al., 2015).

An example of the model validation using experimental data obtained at KSC cryogenic testbed is shown in Fig. 2. Model predictions for the temperature and pressure (red lines) during chilldown in transfer line are shown in comparison with experimental data obtained for temperature and pressure (black lines) obtained in the testbed at KSC. It can be seen from the figure that despite the simplifications introduced to speed up integration, the model can reproduce accurately both pressure and temperature variations at different locations along the line.

The proposed integration scheme is very fast. The 2000 seconds of the real time chilldown in this example were integrated in less than one second of the CPU time on a laptop.

The fast and time-accurate predictions of the cryogenics two-phase flow obtained using this model open a possibility of development of the model-based approach to the integrated health management of cryogenic systems. Below we report on the progress in development of such an approach.

6. Physics Based Approach to the Fault Diagnostics and Evaluation

The online health management of cryogenic systems involves the following set of basic operations: (i) continuous monitoring of the fluid transfer and fault detection; (ii) fault isolation and identification; (iii) fault evaluation. A more advanced IHM can also propose and optimize fault recovery strategies.

In this paper we will discuss briefly how physics-based approach can enhance the performance of an IHM system at every step. We will demonstrate that faults during cryogenic fluid transfer naturally fall within several ambiguity groups. We discuss how physics models can help to improve fault identification using D-matrix formalism. We present an ap-
**6.1. Fault detection**

An event of fault detection is depicted in the Fig. 3. In this test the transfer line is opened for the fluid flow at around 500 sec and one of the bleed valves in the system is stack closed. This fault causes partial blocking of the upstream flow through the line and chilldown delay. As a result the fluid temperature along the line is higher than in nominal regime. The deviation of the temperature from the nominal dynamics can be detected by several sensors along the line as the crossing of the top margin by the temperature time trace. Once this crossing is observed the system signals fault detection.

Once fault is detected the IHM system should locate and identify it. This task is nontrivial because there are many possible faults in the system, their dynamics is complex, and different faults may often cause similar response of the system.

The model based approach to the IHM allows one to simplify the solution of this problem by building off-line a digital library of such faults and investigating their dynamic features. The dynamic features can be ultimately related to the flow properties via full set of the flow boiling correlations available within the hierarchy of models. This digital library can be extended and improved by continuous experimental validation of some of these faults and by learning model parameters. Here we lay down the foundation of this process and provide some preliminary results of the corresponding analysis.

**6.2. Identification**

Our example below is focused on the chilldown for two main reasons. Firstly, it is very desirable to detect faults in the cryogenic line at earlier stage before transfer has began. And fault detection at the chilldown stage provides such an opportunity. Secondly, chilldown analysis is very challenging and most of the functional fault models ignore this stage of operation. In this sense the model based approach allows one to fill the corresponding gap in the IHM of cryogenic systems.

The faults in the cryogenic transfer line are most frequently related to the flow blocking or leaks. The flow blocking may have multiple origin including e.g. clogging, valve stack closed, vapor lock.

Some of these faults have dynamical features similar to those shown in the Fig. 3. These faults correspond to the partial flow blocking and can be detected by several sensors. The faults with similar dynamical features that can not be easily identified are combined in so-called ambiguity groups. The numerical analysis reveals several such ambiguity groups for the faults in cryogenic flow. Identifying faults with similar dynamic features within one ambiguity group is a challenging problem.

Example of another ambiguity group is shown in Fig. 4. The top row demonstrates temperature transients in the system for the case when one of the bleed valves is stack open. This fault results in the increased fluid flow that speeds up chilldown dynamics. As a consequence, the temperature of the fluid in the pipe becomes lower than in nominal regime. The deviation of the flow temperature beyond low margin can be detected by several sensors.

Similar dynamic features are observed when a different bleed valve is stack open. The model predictions for the corresponding temperature time-series data are shown in the middle row.

The leak in the pipe also results in the increase of the flow rate and cooling upstream of the fault. The transient dynamics of the flow temperature detected by sensors in this case is shown in the bottom row of the Fig. 4. It follows very closely the dynamics observed in two previous cases making it difficult to identify of one of these three faults.
At the sensor locations. The cost function was chosen in the form
\[ S(\mathbf{c}_k) = \sum_{n=0}^{N} (F_{1,n} - p_n)^2 + (F_{2,n} - T_n)^2 \]

6.3. Dependency matrix of the cryogenic chilldown

An efficient approach to the solution of this problem can be developed within D-matrix formalism (Sheppard & Simpson, 1996). In this approach each fault is projected on a binary sequence of the test results. In this sequence 1 corresponds to the dependency relation between faults and tests and 0 corresponds to no such relation. Overall, the directed graph corresponding to the logical relationships between the set of tests (results of the sensor measurements) and the set of diagnoses (faults) can be represented in a bit-wise D-matrix (Sheppard & Simpson, 1996).

An fragment of the D-matrix developed for the fault diagnostics in the cryogenic transfer line during chilldown is show in the Table ??.

<table>
<thead>
<tr>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>t6</th>
<th>t7</th>
<th>t8</th>
<th>t9</th>
<th>t10</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>f2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>f3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>f5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. The fragment of the D-matrix build for two groups if faults using outcome of eight sensors. The two ambiguity groups are highlighted by pink and cyan. The predicted dependencies for two virtual sensors that can partially disambiguate the faults are highlighted in the last two columns by gray color.

7. Evaluation

Once fault is identified to belong to one of the ambiguity groups the parameter space for subsequent fault analysis and evaluation is substantially reduced. At this stage a number of optimization tools can be employed to disambiguate and estimate the value of the fault.

In the numerical test described below three possible fault were identified: (i) valve cv1 stack open, (ii) valve cv2 stack open, and (iii) leak in one of the pipes. Their dynamic features (see Fig. 4 and faults f2, f3, f5 in the Table ??) are very similar and additional analysis is required to disambiguate the fault. Three optimization tools were developed and tested for cryogenic chilldown model - unconstrained nonlinear optimization, the MCMC, and the direct search.

The unconstrained optimization is very efficient for the local search and fine tuning of the single parameter. In the case of search in the space of three parameters this algorithm tends to stack at multiple local minima. The MCMC algorithm with Metropolis-Hastings step was also dwelling on the local minima for prohibitively long time for on-line applications.

At the same time it was found that for three-dimensional parameter space the direct search algorithm is very efficient in locating the neighborhood of the global minimum and discerning between the faults within one ambiguity group. The results of application of this algorithm to the evaluation of the fault are shown in the Fig. 5.

The fault (valve cv2 stack open at 30%) was injected at 300 sec. It was almost immediately detected by one of the sensors and a short time later by two other sensors. We emphasize that only temperature sensors were able to detect the fault, while pressure sensors readings remained within the margins.

The identification procedure described in the previous section reduced the analysis of the fault root to three possible causes. To discern between the possible fault causes the following nonlinear curve fitting algorithm was employed. The model predictions for the temperature and pressure \( F_n(t_n, \mathbf{c}_k) \) were fitted to the measured values of \( p_n \) and \( T_n \) on the time interval \( t_0, ..., t_N \) (spanning from detection time to present time or the end of chilldown) at the sensor locations. The cost function was chosen in the form

\[ S(\mathbf{c}_k) = \sum_{n=0}^{N} \left( (F_{1,n} - p_n)^2 + (F_{2,n} - T_n)^2 \right) \]
Figure 5. The results of discerning and evaluating faults within one ambiguity group.

where the $c_k$ is the set of parameters of suspected faults.

By scanning through the parameter space it was possible to rule out two of the suspected faults and to estimate the value of the correctly identified remaining fault. In particular, it can be seen from the figure that the leak fault was ruled out after 10 iterations, and after 50 iterations it was possible to disambiguate between two suspected valve stack open faults. It can also be seen in the figure that the algorithm converges to the correct value of the fault in 60 iterations. Each iteration takes less than half a second of integration on a laptop.

The fault discerning and evaluation described above complete the example of application of the physics based approach to the integrated health management of cryogenic loading operation. It demonstrates that the physics based approach to the IHM of cryogenic systems may offer a number of significant extension and improvements of the IHM capabilities as compared to other techniques.

8. Conclusion

In conclusion we note that physics models develop within this project allow for very fast and time-accurate predictions of the pressure and temperature dynamics of cryogenic system during chilldown and loading operations. This capability opens a unique opportunity of developing model-based approach to the on-line health management of cryogenic systems.

We provided an example of application of the model-based approach to three basic steps of the integrated health management - fault detection, identification, and evaluation. We demonstrated that at every step the model-based approach can enhance and extend capabilities of the health management system. We emphasize that provided example refers to the chilldown operation, which is one of the most challenging regimes of loading from the point of view of model predictions and fault management.

We showed that model-based approach allows one to create and extend digital library of faults in cryogenic system that includes realistic dynamic features of the faults. It also allows one to create and optimize dependencies matrix between faults and sensors measurements that can be used for the fast on-line fault identification. We demonstrated that physics based analysis reveals several ambiguity groups for the faults in cryogenic system.

We developed a number of the model-based optimization tools that allows one to discern and evaluate faults within each ambiguity group. The development of these tools paves the way to a number of important applications of the model-based approach including machine learning of the system and flow parameters, optimization of the fault recovery strategies, loading regimes, and design of cryogenic systems. The results of the development of these applications will be discussed in details in our future work.

Acknowledgment

This work was supported by the Advanced Exploration Systems and Game Changing Development programs at NASA HQ.

Nomenclature

- $u$ velocity
- $T$ temperature
- $p$ pressure
- $e$ specific energy
- $h$ specific enthalpy
- $H$ heat transfer coefficient
- $g$ gravity
- $Re$ Reynolds number
- $Fr$ Froude number
- $Bo$ Boiling number
- $t$ time
- $\Delta t$ time step
- $E$ enhancement factor
- $A$ cross-sectional area
- $V$ volume of the control volume
- $f$ friction factor
- $l$ perimeter
- $z$ coordinate along the pipe
- $x$ mass fraction
- $y$ height of the control volume
- $\dot{m}$ mass flow rate
- $c$ specific heat
Greek
\(\alpha\) gas void fraction
\(\rho\) density
\(\tau\) wall shear stress
\(\mu\) viscosity

Subscript
\(g\) gas
\(3\) two-phase region
\(w\) wall
\(a\) ambient

REFERENCES

PROFILs
Ekaterina Ponizovskaya-Devine
Dr. Ekaterina Ponizovskaya-Devine is a research scientist in SGT since 2009. Her current work is related to integrated health management and control of cryogenic propellant loading systems and two-phase cryogenic flow modeling. She obtained her PhD at Moscow Institute of Physics and Technology in Physics, working on stochastic resonance in nonequilibrium electron-hole plasma. She was a post-doc in Spanish National Research Council, working in photonics, nanotechnology, negative index materials. In 2005 she had Visiting Scholar position at HP Lab working in nanotechnology. Her research interests are physics based models, two-phase flow, nanotechnology, physics model based fault diagnostics, optimization and control.

Dmitry G. Luchinsky
Dmitry G. Luchinsky is a senior research scientist in MCT Inc. He obtained his MSc and PhD in physics in Moscow working on nonlinear optics of semiconductors. He is an author of more than 100 publications. He has been on a number of occasions a Royal Society Visiting Fellow and a NASA visiting scientist. He worked as a senior scientific researcher in VNII for Metrological Service (Moscow, Russia) and as a Senior Research Fellow in Lancaster University (Lancaster, UK). His research interests include nonlinear optics, stochastic and chaotic nonlinear dynamics, dynamical inference, fluid dynamics, ionic motion. His research is
currently focused on theory and CFD of gas dynamics and cryogenic flows.

**Michael Khasin** Dr Michael Khasin is a Senior Researcher in SGT Inc., working at NASA Ames Research Center. He holds B.Sc. in Physics (Honors Program), 2001, M.Sc. in physics, 2003, and PhD in chemical physics, 2008, from the Hebrew University of Jerusalem. As a postdoctoral researcher at Michigan State University, Massachusetts Institute of Technology, and University of Michigan, Michael worked on the theory of non-equilibrium systems and their control and transport in disordered system. His research interests include applied physics, nonlinear and stochastic dynamics, large fluctuations and their control in non-equilibrium systems and efficient simulation of complex dynamics. He has been an invited speaker to many international and national conferences and workshops. Currently his research is focused on the theory of heat transfer, fluid dynamics, and fluid structure interaction.

**Dogan A. Timucin** is a Research Scientist in the Intelligent Systems Division at NASA Ames Research Center. He holds B.S. (1989), M.S. (1991), and Ph.D. (1994) degrees in electrical engineering, and has been with NASA since 1995. His research interests are in the general areas of classical electrodynamics and optics, statistical physics, nonlinear dynamics, Bayesian inference, and numerical optimization.
Interactive Multiple-Model Application for Hydraulic Servovalve Health Monitoring

Wlamir Olivares Loesch Vianna\textsuperscript{1} and Takashi Yoneyama\textsuperscript{2}

\textsuperscript{1}EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil
wlamir.vianna@embraer.com.br

\textsuperscript{2}ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil
takashi@ita.br

\section*{Abstract}
Hydraulic systems are widely used as power source for several different applications. Servovalves are critical components and often subjected to failures. Estimating degradations from these components requires dynamic analysis of their behavior and consequently advanced monitoring techniques. This article proposes an online monitoring method to estimate a degradation parameter of the servovalve using an interactive multiple-model technique considering a bank of Extended Kalman Filters that models not only the valve itself but also the degradation trend. A single failure mode was considered related to the nozzle line clogging. The degradation estimates and the likelihood of the correctness of each model were analyzed in order to evaluate the proposed method.

\section{Introduction}
Hydraulic servovalve health monitoring have been addressed in several works, including (Samadani., Kwuimy & Nataraj, 2014), (Borello, Vedova, Jacazio & Sorli, 2009), (Mussi & Góes 2009) and (Sepasi 2005). Most failure modes from these components require dynamic analysis of its behavior and consequently advanced monitoring techniques. One commonly used method is the Kalman filter applied as parameter identification and examples of application include (Hajiyev & Caliskan, 2003) and (Sepasi 2005). These applications consider an augmented state model including the variable of the model associated with the degradation. Eq. (1) and Eq. (2) give an example from (Hajiyev & Caliskan, 2003) where the parameter \( a \) of the system is the desired value to be estimated.

Linear system:

\begin{align*}
\begin{cases}
x_k = ax_{k-1} + w_{k-1} \\
z_k = x_k + v_k
\end{cases}
\end{align*}

(1)

Augmented system:

\begin{align*}
\begin{bmatrix} x_k \\ a_k \end{bmatrix} &= \begin{bmatrix} a_{k-1}x_{k-1} \\ a_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1} \\ 0 \end{bmatrix} \\
z_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ a_k \end{bmatrix} + v_k
\end{align*}

(2)

This augmented state-space model considers the parameter \( a \) constant and may not address properly its estimation if the system presents variations of \( a \) specially when submitted to abrupt degradation variations and when quick decisions are required such as applications in reconfiguration systems. Alternatives include modeling the dynamic of the parameter being estimated and including it at the augmented state model. An example is given in (Keong, Lim & Mbab, 2014) where a Helicopter tail gearbox bearing is monitored considering three possible degradation dynamics: stationary trend, linear trend and polynomial trend. Figure 1 illustrates these dynamics.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Degradation trends extracted from (Keong, Lim & Mbab, 2014)}
\end{figure}
Estimation of parameter $a$ may be useful when it is related to a degradation, for example a friction or orifice diameter in hydraulic line whose variation may indicate clogging of the line.

Each possible degradation dynamic model included at the augmented state model result in a different Kalman Filter (KF) and consequently an estimation of the degradation parameter as well as other state variables. The technique that combines different models using KF are called Switching Kalman Filter (SKF). The methods used in SKF applications include Autonomous Multiple Model (AMM), Generalized Pseudo-Bayesian of first-order (GPB1), Generalized Pseudo-Bayesian of second-order (GPB2), Interacting MM (IMM) among others. An implementation and comparison of several of these method is presented in (Pitre, 2004) with application in Target Tracking. From these examples, the most popular one is the IMM (Pitre, 2004), whose main advantage is the lower computation cost (Chze & Inseok, 2008), but by using more complex “mixing techniques”, it is more difficult to analyze its results (Chze & Inseok, 2008).

This paper proposes a method to monitor a hydraulic servovalve using an IMM algorithm combined with a bank of Extended Kalman Filter containing some augmented state-space models similar to Eq. (2), modeling not only the dynamics of the valve itself but also the dynamics of the degradation.

2. HYDRAULIC SERVOVALVE MODEL

This article considered a two stage servo valve as illustrated in Figure 2.

![Figure 2: Schematic of a two stage electro hydraulic servovalve with force feedback (Merrit, 1976).](image)

The first stage of the servo valve comprises the permanent magnet, pole piece, armature, flapper, nozzle, leaf type feedback spring and the spool. The equation relating the current input and the spool and flapper position is given by Eq. (3) (Merrit, 1976).

$$K_f \Delta i = J_a s^2 \frac{x_f}{r} + (r + b)K_f \left[ \frac{r}{r + b} \frac{x_f}{x_f} + x_s \right]$$

In which:
- $K_f$ is the torque constant of the torque motor;
- $\Delta i$ is the current input;
- $J_a$ is the inertia of armature;
- $x_f$ is the flapper position;
- $r$ is the distance between center of armature and flapper;
- $b$ is the distance between flapper and spool;
- $K_f$ is the spring constant feedback at free end;
- $x_s$ is the spool position.

The equation relating spool and flapper position is given by Eq. (4) (Merrit, 1976).

$$\frac{x_s}{x_f} = \frac{K_{qp}}{A_v \left( s^2 + \frac{2 \delta_{hp}}{\omega_{hp}} + 1 \right)}$$

In which:
- $K_{qp}$ is the flow gain of flapper valve;
- $A_v$ is the area of spool;
- $\omega_{hp}$ is the hydraulic natural frequency of pilot stage;
- $\delta_{hp}$ is the pilot stage damping ratio.

The parameters values used in this work are:

- $K_t = 0.025 \text{ in.lbs/ma}$
- $r = 0.015 \text{ in}$
- $b = 0.0012 \text{ in}$
- $A_v = 0.026 \text{ in}^2$
- $K_{qp} = 3.9 \text{ in/in.sec}$
- $K_f = 93 \text{ in.lbs/in}$

In order to simulate the time varying input current $\Delta i$, a sinusoidal wave form was adopted. The system response for this input is given in Figure 3.
3. Bank of State-Space Models

The first step to build the bank of filters is to obtain the state space model of the first stage servovalve equations given in last topic. To accomplish that, discrete time domain equations based on Eq. (3) and Eq.(4) and simplifications described previously are built using the Euler discretization method and then put in the state space model. Eq. (5) shows the resulting model.

\[
\begin{align*}
    x_k &= Ax_{k-1} + Bu_k + w_{k-1} \\
    z_k &= Hx_k + v_k
\end{align*}
\]

in which:

- \(x_k\) is the state vector:
  \[
  x_k = [x_f, x_v, \dot{x}_v, \dot{x}_f]^{T}
  \]

- \(u_k\) is the input:
  \[
  u_k = [\Delta i]^{T}
  \]

- \(z_k\) is the output:
  \[
  z_k = [x_f, v_f, \dot{x}_f, \dot{x}_v]^{T}
  \]

- \(A = \begin{bmatrix}
    0 & -r/(r+b) & 0 & 0 \\
    0 & 1 & ts/2 & 0 \\
    K_{qp} / A_v & 0 & 0 & 1
  \end{bmatrix}
  \]

- \(B = \begin{bmatrix}
    rK_f / (K_f(r+b)^2) & 0 & 0 & 0
  \end{bmatrix}
  \]

- \(H = \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0
  \end{bmatrix}
  \]

- \(ts\) is the sampling time;

- \(w_{k-1}, v_k\) are the process and measurement noise.

In order to present the proposed method, the procedures are illustrated using three augmented models similar to those in (Chze & Inseok, 2014) associated to stationary trend, linear trend and second order polynomial trend of the degradation parameter. The degradation parameter chosen to be evaluated is the flow gain of the flapper valve (\(K_{qp}\)) and its decrease relates to clogging of the nozzle line. In order to estimate this parameter, some augmented state space system are considered. Notice that by putting the degradation parameter in the state vector, the model becomes non linear, since this parameter multiplies a state parameter (\(x_f\)) requiring the implementation of a modified version of the Kalman Filter. To accomplish that, a bank of Extended Kalman Filter was implemented. The three augmented models are given in what follows:

1) Stationary degradation (\(\dot{K}_{qp_k} = 0\)):

\[
\begin{align*}
    x_{stat_k} &= [x_f, x_v, \dot{x}_v, K_{qp}]^{T} \\
    z_{stat_k} &= z_k
\end{align*}
\]

\[
A_{stat_k} = \begin{bmatrix}
    0 & -r/(r+b) & 0 & 0 \\
    0 & 1 & ts/2 & 0 \\
    K_{qp} / A_v & 0 & 0 & 1
  \end{bmatrix}
\]

\[
B_{stat_k} = \begin{bmatrix}
    rK_f / (K_f(r+b)^2) & 0 & 0 & 0
  \end{bmatrix}
\]

\[
H_{stat_k} = \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0
  \end{bmatrix}
\]

2) Linear degradation (\(\dot{K}_{qp_k} = \dot{K}_{qp,k-1}\)):

\[
\begin{align*}
    x_{lin_k} &= [x_f, x_v, \dot{x}_v, K_{qp}, \dot{K}_{qp}]^{T} \\
    z_{lin_k} &= z_k
\end{align*}
\]

\[
A_{lin_k} = \begin{bmatrix}
    0 & -r/(r+b) & 0 & 0 & 0 \\
    0 & 1 & ts/2 & 0 & 0 \\
    K_{qp} / A_v & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 1
  \end{bmatrix}
\]

\[
B_{lin_k} = \begin{bmatrix}
    rK_f / (K_f(r+b)^2) & 0 & 0 & 0 & 0
  \end{bmatrix}
\]

\[
H_{lin_k} = \begin{bmatrix}
    1 & 0 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 & 0
  \end{bmatrix}
\]

3) Polynomial degradation (\(\ddot{K}_{qp_k} = \dot{K}_{qp,k-1}\)):

\[
\begin{align*}
    x_{pol_k} &= [x_f, x_v, \dot{x}_v, K_{qp}, \dot{K}_{qp}, \ddot{K}_{qp}]^{T} \\
    z_{pol_k} &= z_k
\end{align*}
\]

\[
A_{pol_k} = \begin{bmatrix}
    0 & -r/(r+b) & 0 & 0 & 0 & 0 \\
    0 & 1 & ts/2 & 0 & 0 & 0 \\
    K_{qp} / A_v & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 0 & 0 & 1
  \end{bmatrix}
\]

\[
B_{pol_k} = \begin{bmatrix}
    rK_f / (K_f(r+b)^2) & 0 & 0 & 0 & 0 & 0
  \end{bmatrix}
\]
\[ H_{0pk} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \]

The Jacobian matrices (required for the Extended Kalman Filter process) containing the partial derivatives of \( A \) are given below.

\[
dA_{start} = \begin{bmatrix} 0 & -r & 0 & 0 \\ K_{q0} & 1 & \frac{ts}{2} x_{f0} & 0 \\ 2A_v & 0 & 0 & x_{f0} \\ A_v & 0 & 0 & 0 \end{bmatrix};
\]

\[
dA_{link} = \begin{bmatrix} 0 & -r & 0 & 0 & 0 \\ K_{q0} & 1 & \frac{ts}{2} x_{f0} & 0 \\ 2A_v & 0 & 0 & x_{f0} \\ A_v & 0 & 0 & 0 \end{bmatrix};
\]

\[
dA_{pol} = \begin{bmatrix} 0 & -r & 0 & 0 & 0 \\ K_{q0} & 1 & \frac{ts}{2} x_{f0} & 0 \\ 2A_v & 0 & 0 & x_{f0} \\ A_v & 0 & 0 & 0 \end{bmatrix};
\]

4. Interactive Multiple-Model Algorithm

The IMM algorithm (Blom & Bar-Shalom, 1988) reinitializes each model with a weighted sum of the updated estimates from every model based on probabilities estimations of each model. This process is called merging and it reduces its computational complexity to \( M \) where \( M \) is the number of models used in the algorithm, which in this case is 3. An illustration of the IMM model switching process is described in Figure 4. The interaction between the models depends on the switching probabilities and the likelihood of each of the model. The IMM result is a combined state vector that is the sum of the state vectors for each of the modes weighted by their model probabilities.

![Figure 4: IMM model switching process (Farmer, Hsu & Jain, 2002).](image)

The estimation of each switching probability and model likelihood is described below where a single cycle of the IMM algorithm is given (Eq.(6) to Eq.(14)). It consists of 4 steps: reinitialization where mixing estimates and variances are estimated for each model; the filtering process itself also for each model and considering the mixed estimates; probabilities and likelihood updates and finally estimate fusion resulting in a single state estimation.

1) Model-conditioned reinitialization (for \( i = 1,2,\ldots M \)):

1a. Predicted mode probability:

\[
\mu_{i\mid k-1} = \sum_{j=1}^{M} \pi_{i,j} \mu_{k-1}^{j}
\]

(6)

1b. Mixing weight

\[
\mu_{k-1}^{i,j} = \frac{\pi_{i,j} \mu_{k-1}^{j}}{\mu_{k-1}^{i}}
\]

(7)

1c. Mixing estimate:

\[
\tilde{x}_{k-1\mid k-1}^{i} = \sum_{j=1}^{M} \tilde{x}_{k-1\mid k-1}^{j} \mu_{k-1}^{i,j}
\]

(8)

1d. Mixing covariance:

\[
\tilde{P}_{k-1\mid k-1}^{i} = [ \sum_{j=1}^{M} P_{k-1\mid k-1}^{j} + (\tilde{x}_{k-1\mid k-1}^{i} - \tilde{x}_{k-1\mid k-1}^{j}) (\tilde{x}_{k-1\mid k-1}^{i} - \tilde{x}_{k-1\mid k-1}^{j}) ] \mu_{k-1}^{i,j}
\]

(9)

2) Model-conditioned filtering (for \( i = 1,2,\ldots M \)):

\[
(\tilde{x}_{k\mid k-1}^{i}, \tilde{P}_{k\mid k-1}^{i}) \rightarrow (\tilde{x}_{k\mid k}^{i}, \tilde{P}_{k\mid k}^{i}, \tilde{Z}_{k}^{i}, \tilde{S}_{k}^{i})
\]

(10)

3) Mode probability update:

3a. Model likelihood:
\[ L^i_k = N(\tilde{x}^i_k; 0, S^i_k) \]  
(11)

3b. Model probability
\[ \mu^i_k = \frac{\mu^i_{ik}L^i_k}{\sum_{i \in M} \mu^i_{ik}L^i_k} \]  
(12)

4) Estimate fusion:
4a. Overall estimate:
\[ \hat{x}_{1:k} = \sum_{i \in M} \hat{x}^i_{1:k} \mu^i_k \]  
(13)

4b. Overall covariance:
\[ P_{1:k} = \sum_{i \in M} [P^i_{1:k} + (\hat{x}^i_{1:k} - \hat{x}_{1:k})(\hat{x}^i_{1:k} - \hat{x}_{1:k})^T]\mu^i_k \]  
(14)

in which:
\( \pi^i_j \) is the model transition probability;
\( \mu_k \) is the probability of each model;
\( L_k \) is the likelihood of each model;
\( M \) is the number of models, which in this case is 3.

A summary of the IMM algorithm is illustrated in Figure 5.

Figure 5: IMM process

Eq. (10) represents the Extended Kalman filter detailed by the following process:

1) Time Update:
\[ \hat{x}^\text{aug}_k = A^\text{aug}_k \hat{x}^\text{aug}_{k-1} + B^\text{aug}_k u_k \]  
(15)

\[ P^\text{aug}_k = dA^\text{aug}_k P^\text{aug}_{k-1} dA^\text{aug}_k + Q^\text{aug}_k \]  
(16)

2) Measurement Update:
\[ S^\text{aug}_k = H^\text{aug}_k P^\text{aug}_k H^\text{aug}_k \]  
(17)

\[ K^\text{aug}_k = P^\text{aug}_k H^\text{aug}_k \left(S^\text{aug}_k \right)^{-1} \]  
(18)

\[ \hat{x}^\text{aug}_k = \hat{x}^\text{aug}_{k-1} + K^\text{aug}_k \left(z^\text{aug}_k - H^\text{aug}_k \hat{x}^\text{aug}_k \right) \]  
(19)

\[ P^\text{aug}_k = (I - K^\text{aug}_k H^\text{aug}_k) P^\text{aug}_k \]  
(20)

in which:
\( K^\text{aug}_k \) is the Kalman gain;
\( P^\text{aug}_k \) is the covariance matrix of the state estimates.

As an example, the estimations of the EKF states with stationary trend of the degradation parameter and the same sinusoidal input current in Figure 3 is given in Figure 6.

Figure 6: EKF estimations

5. RESULTS

In order to evaluate the IMM method, four degradation trends were evaluated, a stationary one, a linear one, a polynomial one and another containing a combination of the three last ones. Figure 7 shows these input trends.
All these trends were combined with the same sinusoidal current input from Figure 3 and submitted to all three filters (stationary, linear and polynomial) described previously using the conventional EKF as well as the IMM described in last topic. For all simulations the IMM transition probability matrix used is:

\[
\pi = \begin{bmatrix} 0.998 & 0.001 & 0.001 \\ 0.001 & 0.998 & 0.001 \\ 0.001 & 0.001 & 0.998 \end{bmatrix}
\]

The degradation parameter estimations are given in Figure 8, Figure 9 and Figure 10.

It is possible to see from these results how lower order models could not estimate properly degradations submitted to higher order variations (i.e. red dashed line).

The probability of each trend estimated in the IMM method are given in Figure 11, Figure 12, Figure 13 and Figure 14. As mentioned before these probabilities are used in the IMM fusion step in Eq. (13) and Eq. (14) as weighting factors to estimate the resulting states from all three models estimations.
Figure 13: Probabilities for polynomial degradation simulation

Figure 14: Probabilities for combined degradation simulation

Figure 11 (stationary simulation) shows as expected a predominance of the stationary model probability for stationary simulation. Figure 12 (linear degradation simulation) shows a predominance of the linear model, although during some small intervals the stationary model had higher probability. Figure 13 (2nd order degradation simulation) could predict correctly the increased probability in only some intervals at the 2nd half of the simulation. Figure 14 (combined trend) could predict correctly the stationary trend (higher probability for the initial interval) but could not distinguish properly between linear and polynomial trend for the 2nd half of the simulation. From all these results it is possible to observe that higher the order of the degradation trend, more difficult is to distinguish between them.

In order to compare the precision of all filters from all degradation trends, the mean square error (MSE) between the simulated degradation parameter and its estimation from all simulation frame were estimated. Table 1 shows the results, where each row contain the filtering model (stationary, linear, polynomial and the IMM respectively) and each column the simulation performed (stationary, linear, polynomial and the combination trend as in Figure 6).

As expected, the MSE corresponding to the stationary trend was lower for the stationary model, the MSE of the linear trend for the linear model and the MSE of the polynomial trend for the polynomial model. For the combined trend, the IMM had the lowest MSE proving its effectiveness to deal with multiple evolutionary degradation trends. Also it performed well for the other non-combined trends.

6. CONCLUSIONS

The present work showed an application of an Interactive Multiple Model for on-line degradation estimation of a single failure (nozzle clogging) of the first stage of a two stage flapper nozzle hydraulic servovalve. To accomplish that, three augmented states models were built from the valve model considering stationary, linear and polynomial trend of the degradation parameter. After building these models the IMM could be implemented.

The evaluation of the IMM was done considering four different degradation trends: stationary, linear, polynomial and combination of all previous ones. Together with the IMM, conventional EKF was applied to all simulations considering all three models. Results showed that the IMM had a better estimation for the combination trend while the stationary model for the stationary trend, linear model for the linear trend and the polynomial model for polynomial trend.

It is possible to conclude from this work that the IMM algorithm successfully estimated degradations from the servovalve model relating correctly the probabilities of each model, specially when dealing to a combination of different degradation trends.

The main benefit of using the method proposed in this paper is the possibility to have an on line health monitoring of the component with fast response to degradation variations. Applications may include systems that requires quick decisions for fast degradation evolutions such as reconfiguration systems for transmission lines power grids, flight controls reconfiguration systems and launch vehicle abort trigger.

Improvements in this work includes investigating this method with different components (i.e. actuators) as well as other failure modes, also evaluating other multiple models algorithms such as the Generalized Pseudo-Bayesian of second-order (GPB2) and applications using field data.
ACKNOWLEDGEMENT

The authors acknowledge the support of FINEP (grant 1498/07) and CNPq (grant 303450/2013-4).

REFERENCES


BIographies

Wlamir Olivares Loesch Vianna holds a bachelor’s degree on Mechanical Engineering (2005) from Universidade de Sào Paulo (USP), Brazil, and Master Degree on Aeronautical Engineering (2007) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), Sào José dos Campos, SP, Brazil, since 2007. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications in aeronautical systems.

Takashi Yoneyama is a Professor of Control Theory with the Electronic Engineering Department of ITA. He received the bachelor’s degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in medicine from Universidade de Taubaté, Brazil, and the Ph.D. degree in electrical engineering from the University of London, U.K. (1983). He has more than 250 published papers, has written four books, and has supervised more than 50 theses. His research is concerned mainly with stochastic optimal control theory. He served as the President of the Brazilian Automatics Society in the period 2004-2006.
Lessons Learned in Implementing a Practical Aircraft System Health Management (ASHM) System

Joshua Koelle¹, Matthew Smith², Peter Sulcs³, Greg Kacprzynski⁴ and Rhonda Walthall⁵

¹,²,³,⁴ Impact Technologies, a Sikorsky Innovations Company, Rochester, NY, 14623, USA
joshua.koelle@impact-tek.com
matthew.smith@impact-tek.com
peter.sulcs@impact-tek.com
greg.kacprzynski@impact-tek.com

⁵ UTC Aerospace Systems, San Diego, CA, 92123, USA
rhonda.walthall@utas.utc.com

ABSTRACT

Aircraft System Health Management (ASHM) is a web application used for Boeing 787® and Airbus A320® and A380® aircraft system monitoring by airlines and field engineers worldwide which also serves all existing Aircraft Condition Monitoring Function (ACMF) reports, Flight Deck Effects (FDE) records and aircraft metadata to UTC engineering teams to aid in efficient aftermarket support. This enables creation, testing and fielding of off-board diagnostics and prognostics modules of varying levels of sophistication, that convert this abundance of existing data into actionable and timely knowledge about a/c fleet health. ASHM encourages and promotes cross functional collaboration allowing those with the most subject matter expertise within the enterprise to access the field data they need to observe operational performance and to create, test and field modules that can actively diagnose and warn field service professionals of problems when and potentially before they arise. A practical case study related to monitoring of the novel Boeing 787® electromechanically driven distributed aircraft environmental systems is presented. This use case motivates a discussion of pragmatic lessons learned in the fielding of diagnostic and prognostics solutions.

1. INTRODUCTION

Modern commercial aircraft contain computerized maintenance systems that have replaced the dials, indicators, switches, and diagnostic read-outs of prior generations of aircraft. These systems in addition to performing on-board diagnostic functions also record high value parametric data that can be used for system and component health tracking, fleet data studies, and prognostics. By observing changes in component performance or recognizing abnormal response behaviors it is possible to observe incipient fault conditions before they grow into significant problems that are recognized by the on-board Built-in Test (BIT) checks which in the worst case may cause a delay or cancellation of service. Looking at data trends across a fleet of aircraft can identify outlier behaviors that are indicative of degraded health, determine the effect of usage factors on component life, and optimize maintenance practice. Prognostics goes beyond fault assessment to project remaining useful life, allowing advanced scheduling of maintenance procedures, proactive replacement part allocation, and enhanced fleet deployment decisions based upon the estimated progression of component life usage.

This aircraft data and diagnostic information is of high value to multiple groups who have interest in the current and future health states of these critical systems. First, the airline operators of these aircraft are looking for a change in aircraft health that can be used to optimize maintenance and prevent the occurrence of delays and cancellations. Second, the field service engineers who support the airlines benefit from information that allows them to better support maintenance troubleshooting and logistics. Finally, the system engineers who support each of the aircraft systems can obtain critical information to characterize product issues and develop enhancements to diagnostic capability. To serve each of these important user groups, UTC Aerospace Systems has developed and continues to improve and expand the Aircraft System Health Management (ASHM) Tool.

ASHM takes in ACMF reports for selected subsystems and components of each supported aircraft platform, parses and
processes the reported parameters against thresholds, computes estimated or expected values for some key parameters, and serves the report data and the processed results as part of a fleet view available to airline, maintenance, and engineering users. The application allows for the creation, integration, and execution of custom analytic modules that extract enhanced diagnostic and prognostic information from the raw report data. This information is also made available for visualization, trending, and alerting.

In this paper, the sub-systems of the Boeing 787 will be used as a case study of how this high value parametric data and derived diagnostic and prognostic information can be used to enhance commercial aircraft maintenance practice. The Boeing 787 is outfitted with a modern computerized maintenance system that records key data for each of these aircraft systems. The available on-board data is transformed by ASHM into actionable component and system fleet information to guide fleet troubleshooting, opportunistic maintenance, and logistics. In addition to the Boeing 787, the ASHM software is also being used to support the Airbus A320 and A380 platforms.

2. AIRCRAFT DATA SOURCES

The ASHM software tool collects, organizes, and stores aircraft data from two different sources. The first class of data is comprised of system status flags that report anomalous behavior or degraded performance as obtained from on-board BIT checks and diagnostic functions. These events which are commonly referred to as Flight Deck Effects (FDEs) are captured along with linked maintenance messages that capture the symptoms of the observed condition. The second class of data is recorded by the Airplane Condition Monitoring Function (ACMF) and is specifically targeted at long term analysis of aircraft health and usage (Ramohalli 1992).

The ACMF report data is of particular interest to the ASHM fleet monitoring software. These reports capture aircraft parametric data based upon triggering criteria and a format that have been established by aircraft system domain experts. The specific content in each report can be targeted at periods of operation sensor signals that are of particular interest for a given component or fault mode. Some reports trigger based upon entry into a given operating mode that is appropriate for system performance characterization or detection of anomalies. Other reports are triggered by the occurrence of specific events that are noteworthy, abnormal, or perturb the system in a way that makes performance or fault conditions more observable. Each report collects an assortment of sensor data, state information and contextual metadata that is of interest to the system or component monitored by the report. This data includes typical operating parameters such as: temperature, pressure, position, and speed. The reports in some cases record operational state values or calculated values that have been derived from the raw parametric data. Often, the aircraft system data is reported along with aircraft level data such as altitude, air temperature, and Mach number that provide important context about the operating conditions at the time of acquisition. The data may be acquired as a single snapshot in time, a set of statistical metrics such as mean or peak value, or as a time series history.

3. ASHM SOFTWARE ARCHITECTURE

ASHM is an enterprise application that monitors systems on multiple commercial aircraft platforms using ACMF reports, automated parameter alerts and notifications and diagnostic and prognostic custom analytic modules. The overall software architecture is depicted in Figure 1. ACMF reports and other information sources are generated by the on-board maintenance system. The reports are offloaded and automatically processed by ASHM using a fully automated workflow. The ASHM application parses and stores incoming reports to generate alerts and execute advanced algorithms that were developed or adapted using existing system models and data. The event driven architecture powers the real-time web portal where airlines, maintenance support and engineers can analyze reports and be notified in advance of potential issues. The portal also displays high level dashboards for easy information consumption and drill-downs, graphs and report viewers for detailed analysis.

Figure 1. ASHM Software Architecture

An extract, transform, load (ETL) engine is utilized in ASHM. As different report types arrive they are bucketed by type into directories for further processing. Each subsystem has one or more report types, and as ASHM grows to process more systems of each aircraft platform, and adds more aircraft platforms, the number of report types will grow accordingly. In the first parsing step for each
While the removal data aligns with the SI records in many situations, there are some cases where the component involved in the highest number of maintenance procedures.

The complex event processing portion of the ASHM software is shown in Figure 2. Data flows through RulePoint by report type, originating with a SQL Source database that acquires parameter instance data from the ASHM database and pushes it into a RulePoint® Topic. A Rule references one or more topics and may use data from those topics to 1) determine anomalous conditions, e.g. value out of range, 2) compute new values based on those parameters, 3) send those computed values or detected conditions to a Responder that is responsible for storing new data back to the same ASHM database. For the ASHM project automatic alert rule generation based on thresholds defined in the database is employed. This custom tool uses a Java API Adapter to 1) connect to the development RulePoint instance, 2) remove all previously generated (as opposed to hand entered) rules, and 3) generate a new set of rules based on those thresholds.

The ASHM application checks for out of range “alert” conditions on selected incoming report parameters, looking for warning or alarm conditions that are higher or lower than expected under normal operating conditions. Each “alertable” parameter has its own set of thresholds defined in the database for low and high warning and alarms. There are also mechanisms in place to define two additional criteria which are when the thresholds are to be ignored, say when some (the same or another) parameter’s value meets a certain conditional relationship with a fixed value, e.g. <= some value, = some value, or => some value. The parameter alert rules store parameter out of range conditions back to the database, where they are used to display those anomalous conditions to the end user in the web application. In addition to simple thresholds, the ASHM application can invoke a custom analytic that performs calculations on the report data. This functionality is used to run diagnostic and prognostic algorithms that extract refined system health information from the raw parametric data. The process for developing, vetting, and integrating these custom analytic modules is described in detail in the following section.

4. CUSTOM ANALYTIC DEVELOPMENT PROCESS

The development and use of custom analytic modules enables health monitoring capability that extends far beyond what is possible using only parameter alerting and trending of raw ACMF data. These modules use ACMF report data to extract subtle fault indicators that give advanced notice of impending failures at severity levels below that captured by the on-board diagnostics. The overall development process for the ASHM custom analytic modules is shown in Figure 3.

The first step in the development process is the establishment of need. When deciding where to allocate effort to create new diagnostic and prognostic capability, the first priority is given to components and fault modes that cause flight cancellations and delays. These service interruptions (SIs) have significant negative effects for all interested parties from the airlines, field service support, and product engineers. The primary objective of custom analytic development and ASHM in general is the minimization of SIs throughout the fleet. The leading causes of SIs are established by analyzing fleet reliability data compiled by the worldwide network of field service personnel. This data is summarized using Pareto chart analysis and the most significant contributors to SIs are given the highest priority when developing new custom analytics. After SI data, priority is given to the components and systems that are creating the highest load on aircraft maintainers. This is quantified by analyzing LRU removal data compiled by field service personnel to determine which components are involved in the highest number of maintenance procedures. While the removal data aligns with the SI records in many situations, there are some cases where the component
criticality or the complexity/time-intensiveness of a given maintenance procedure may affect priority for a given component or system. Lastly, there is the issue of component logistics. Replacement of large, expensive LRUs like Auxiliary Power Units (APUs) requires careful planning due to limited product availability and transportation time. Therefore these systems are prime candidate for development of PHM technologies that provide an assessment of component health, and forecast future maintenance needs.

The second step in the custom analytic development process is creation of the diagnostic or prognostic algorithm. In this step the PHM design experts collaborate with the subject matter experts to determine the available raw materials to support algorithm development, and how this information aligns with known system issues. When possible, model based approaches to diagnostics and prognostics are preferred. The raw ACMF report data represents only one snapshot in time of system operation. It can be difficult to know if a change in a parameter is due to fault or the normal variations that occur over the operating range of the system. The use of system and component models that simulate normal system response can be used to reduce the effect of typical system variations on health assessment performance. Modern engineering design makes extensive use of simulation during the product development phase. For many of the systems the component design models have been readily accessible for implementation into multiple custom analytic modules. However, a suitable model is not available in every case, and in some cases the data required to exercise the model is not present in the related ACMF reports. In those cases, an approach is used that acts directly on the raw report data. If there is a good understanding of the failure modes, either via documented field or laboratory failure data or an understanding of the underlying physics, a feature based approach can be developed to extract known fault indicators from the parametric data. If there is limited understanding of the nature of failure, or if a generalized anomaly detection capability is desired, an approach that is based purely on the known range of healthy data is used.

Regardless of the approach, the module must be designed so that it has sufficient robustness to accommodate variations in the report data that are not associated with system performance. The ASHM software package is independent of the on-board aircraft software and therefore must accommodate changes in report format and content as they occur. The nature of the report data may be affected by one or more component operating states, and these must be observed and tracked to ensure proper operation. The software must also be prepared to recognize on-board data collection irregularities and screen out the affected values so that spurious fault correlations are not reported.

If the current ACMF data is deemed insufficient for enhanced system monitoring, the team turns its focus to establishing how the report can be modified to obtain the highest value condition information. The reports were designed to provide the most important system condition information as understood at the time of implementation. On a new aircraft like the Boeing 787®, this means that these decisions were made based upon a theoretical understanding of the system or using the available test data. An examination of field data can identify opportunities to improve the available system configuration information based upon actual usage.

After the diagnostic or prognostic method has been created, it is subjected to a series of validation tests to ensure that it provides the desired system heath information with an acceptable level of performance. Generally the performance of diagnostic methods is evaluated by determining the correctness of fault detection results, and the accuracy of fault severity assessment metrics. Of particular interest are the rates at which two diagnostic results occur: false alarms, or when the diagnostic system detects a fault but the condition of the system is not significantly degraded, and missed detection when the diagnostic system does not indicate a fault when one is known to be present. To support this activity, again the field SI and removal data is used to obtain ground truth information about the health state of the fleet. Documented failures, particularly those with a conclusive root cause assessment are extremely valuable in establishing system response at degraded health states. The fleet service history not belonging to known fault cases can be used to establish baseline system performance. Laboratory test results can be used to supplement field data experience, particularly in cases where practical fault experience is limited. The validation of prognostic approaches is a far more complex topic and has a more significant set of input requirements (Byington, Roemer, Kalgren, & Vachtsevanos, 2005). Given the long timescale of component life, it is generally impractical to complete significant validation prior to algorithm deployment. However, it is of critical importance to establish the accuracy and uncertainty bounds of the models used to assess and predict system health progression.

Validation is difficult in cases where the fault mode does not result in a condition that requires documented maintenance actions. For example, the air flow pathways in environmental control systems or air management systems may become contaminated by foreign material that may become lodged in components such as heat exchangers (Najjar, Hare, D'Orlando, & Leaper 2013). This condition is problematic and requires a cleaning operation, but may not result in a removal or SI. Knowing when these events occur is essential to establishing health state ground truth for the related fault modes. This example illustrates the importance of communication between the operators and field support engineering in developing and validating effective PHM methods.
Validation may be performed at various points throughout the development cycle. It is preferable to conduct significant validation prior to deployment of the analytic to ASHM. This is more realistic for mature platforms that have been in service for a significant amount of time. For example, the ASHM team has created a custom analytic module to evaluate the progression of Airbus A320® APU health. There is a wealth of data for over a decade of operation for a very large fleet of aircraft, and significant validation was possible prior to analytic deployment. By comparison the Boeing 787® is a relatively new platform, with significantly less observed health progression and field maintenance issues throughout the fleet. While it is possible to confirm proper basic functionality and baseline response, it may not be possible to completely validate algorithm response to field failures prior to deployment. In these cases, it may be necessary to deploy a version of the algorithm for validation against new fleet data as it arrives.

When a custom analytic reaches a state that is mature enough for implementation in ASHM, it enters the production rollout phase. The algorithms as created by the PHM and system domain experts are translated into production software modules. The engineering and software teams work together to define a set of verification tests that evaluate all relevant logical paths within the software. These tests ensure that the production implementation matches exactly the approach that was validated during engineering sandbox development. The input and output data streams are established and any relevant contextual information is integrated as meta-data that is cataloged in the production database and made available to downstream processes that consume the custom analytic output. Finally, the module output is integrated into the downstream ASHM processes that will serve this enhanced system condition information to the users. This includes configuration of custom data visualizations, data plotting and trending, and definition of parameter alerts including warning and alarms.

Upon successful deployment of a custom analytic module, it enters the sustainment phase. The output of the module will be regularly inspected and compared to the documented SIs, field failures and maintenance actions. The report data will be monitored for format changes or other updates that require reconfiguration of the event processing configuration or custom analytic software. If new classes of failure or degradation are observed, or the understanding of a given fault mode changes, an alternative version of the custom analytic can be created and evaluated in the engineering sandbox as a software upgrade candidate.

5. GENERATION OF ACTIONABLE INFORMATION

ASHM aids the user in extracting actionable information for short term proactive fleet support from the raw data sources. It does this by raising visibility of event reports (system reports that are only generated when an anomaly is encountered), automatically interpreting various error codes generated by the monitored equipment and triggering ASHM alerts based on threshold exceedances on reported parameters.

Custom analytics provide the means of generating more sophisticated health indicators from the raw data. These health indicators provide actionable information in the following ways: diagnostic and or prognostics indicators augment the raw report within ASHM’s report viewer; alerts based on computed parameters are displayed alongside the ones based on raw reported parameters; and finally computed parameters can be included in the ASHM graphs pages to be monitored and observed visually for trends or anomalous behavior.

The final way that ASHM provides actionable short term information is by allowing the user to compare health indicator parameters by aircraft across the whole fleet thus focusing attention on the aircraft with health indicators that are abnormal compared to the fleet.

ASHM and the associated Data Analytics Tool also provide actionable information for a different audience with a longer term interest: the Engineering teams responsible for supporting fielded systems as well as new product design. ASHM aggregates operational field data that is invaluable in terms of closing the loop between Engineering and Field Support. It provides unprecedented visibility into how the systems that Engineering designs are operating in the field. This allows a closed loop refinement of all the assumptions made at design time, to both improve the current product offering and enable better design assumptions for the next generation of new products.

The goal of any deployed custom analytic is to provide the user of ASHM with actionable information. The following case study is a simple example of extracting invaluable information available by implementing a straightforward custom analytic.

6. CASE STUDY – ELECTROMECHANICAL SUBSYSTEMS

The transition from engine driven hydraulic subsystems to distributed electric motor driven subsystems has led to ACMF reports that capture characteristics from a wide array of systems despite being targeted to one component. A smart motor controller may now be responsible for 3 or more different tasks throughout the duration of a single flight. These tasks range from air management, to motor start procedures. The case study presented here will examine the complexities of the Boeing 787® motor controller system and its effect on the interpretation of ACMF data.

In previous generation civilian aircraft, the main engines generally provide four main sources of auxiliary power: electrical, pneumatic, hydraulic, and mechanical. The electrical system supplies power for avionics equipment, lighting, and in-flight entertainment. The pneumatic bleed
air system supplies power for cabin pressurization and wing anti-ice systems. The hydraulic system provides power for flight control systems and auxiliary systems, and the mechanical power is used within the engine for oil and fuel management.

The electrical subsystems incorporated on next generation electric aircraft combine some of these power systems into one, through the use of generators and smart power distribution systems. This reduces system hardware complexity, resulting in weight reduction and efficiency improvements. A secondary result of this change is an increase in electrical system complexity. There are multiple electrical power distribution systems and new FAA certified software accompanied with these systems (Wheeler & Bozhko 2014).

The Common Motor Start Controller (CMSC) ACMF reports contain parametric data on component power draw, fluid temperatures, power frequency, and active mode information. Among these reports, data is provided in two different formats; time series data and snapshot data. The time series data provides prognostic systems with valuable information regarding how a system reacts to applied power. With this data it is possible to measure spool up time for an engine, or when a starter generator is drawing maximum power. ACMF reports which contain time series data typically target vital aircraft procedures, like Main Engine start or APU start. The snapshot data, in contrast, allows maintainers to track and trend parameters and features over extended periods of time. ACMF reports, which contain snapshot data, capture data at key points throughout each portion of a flight, from Taxi to Landing. The snapshot data is generally divided into two capture methods, peak data value and average data value. Both of these data points are useful in application to component fault detection. While the ACMF data provides vital information about the functions of these subsystems, it is equally as important to understand the operational modes and connectivity of the CMSC subsystem. Without this understanding, the data loses its diagnostic and prognostic value. Figure 4 is a simplified representation of the CMSC component assignments. In this figure, each CMSC is connected to two unique components, Cabin Air Compressors (CAC) and Generators (G), which then provide power to critical systems.

An important part of establishing component connectivity is the operation mode flag. Each CMSC reports an operational flag associated with each data point, whether single point or time series. That operational flag indicates the CMSC’s active mode, which ties to one of the Boeing 787’s redundant system components. These operational flags have a significant bearing on the development of custom analytics for this system.

In the time series ACMF data, the operational mode flag can be used in determining how long the CMSC was driving a specific component and, in combination with the parametric data, provides information about the response of the controller and motor pair. The process involves filtering the data based on the operational mode flag. The time series ACMF reports contain data for a defined period of time after a trigger has occurred. This results in the ACMF report containing data from a CMSC driving multiple components over the course of that time series. When a CMSC is responsible for driving components of vastly different power requirements, it becomes clear that it is vital to filter the ACMF reports by the operational mode prior to applying statistical analyses to the data. Development of analytics which harness this time data can assist maintainers and engineers in analyzing and categorizing the effect of routine procedures on key aircraft components. Figure 5 illustrates the operational mode filtering utilized to select a targeted subset of data.
By contrast, the snapshot ACMF reports contain peak or average data from the CMSC at specific points in each flight leg. In this data, the operational mode flag is useful in plotting component trends over time. For example a motor which draws more power each flight, while performing the same duty cycle, might indicate that the motor or driven component is experiencing degradation. Pairing this information with removals, allows for the development of fault detection thresholds which can be alerted on in ASHM, notifying the maintainers that maintenance action should be taken to avoid future Service Interruptions.

It is not enough, however, to know that a CMSC is driving a specific type of component. For redundancy and distributed system operation, multiple instances of many components exist on the aircraft. A complex logic dictates which of these components is active, and which motor controller is driving each component. A calendar rotation is applied to certain redundant components, thus distributing the wear among each. In a second common arrangement, redundant components share the load on a common task. When one fails in this arrangement, the healthy component is responsible for providing the power to compensate for the failed component, usually at a reduced level of performance. Another common practice is for components to operate in a master slave arrangement where one of the components is present simply to provide a backup should the primary system fail.

The goal for these analytics is not only to identify issues in the motor, but issues in the intermediate components driven directly by the CMSC. The smart power distribution system is responsible for dynamically assigning tasks to each CMSC based on priority and system availability. This information can be useful in fault isolation. For example, if an issue is present with a starter generator, and that issue is prevalent when driven by multiple motor controllers, we now have evidence that the issue exists within the starter generator and not within the power generation system. An ACMF report is designed to capture data during a specific routine during a flight leg. Understanding which routine the ACMF report is designed to monitor provides important information regarding which CMSCs will be providing parametric data within that report. Combining this information with the CMSC operational mode flag will result in the isolation of data which was captured when a specific CMSC was driving a specific component. More importantly, this prevents the ASHM system from producing false alarms on irrelevant data. This method of data fusion was prevalent throughout the development of analytics for the CMSC subsystem on the Boeing 787. After filtering the data, simple statistical analyses were performed on the data to provide parametric data and binary flags. An example of these statistical analyses on a filtered data set, is shown in Figure 6. The parametric data derived from the ACMF reports resulted in the application of the following statistical features:

- Statistical Electrical Power Calculations
- Amount of time the CMSC powered a specific component
- Amount of time required for a component to reach a speed increment
- Oil Temperature Trending: Also useful for trending engine temperature

Enumerated flags were also generated from this filtered report content. The focus was on certain key operating routines/conditions which were highlighted for investigation. Utilizing the filtered data, enumerated flags were incorporated to track the frequency at which these key operating conditions were performed on the aircraft. The example shown in Figure 7: Sample Data Used for Development of Enumerated Flags highlights a flag which would identify when a specific component was operating at High, Limited, or Low power. The assumption was made that high power operation exposed the component to high load and thus more demanding operation.
Other detection logic was put in place to recognize maintenance events and environmental conditions that result in accelerated system life usage. These flags appear as calculated parameters and can be plotted in ASHM. This gives maintainers and engineers the ability to correlate observed usage patterns with field failures.

7. Conclusion

The ASHM software tool provides enhanced health monitoring capability for the commercial aircraft fleet. The development and implementation of custom analytic modules unlocks the full potential of the recorded ACMF data for diagnostics and prognostics. By employing a multi-step analytic development and validation strategy, software development is accelerated while ensuring the quality and accuracy of the actionable condition information provided to the fleet stakeholders. Care must be taken to create robust algorithms that recognize irregularities in the report data, selectively filter applicable data, and ignore any potentially spurious or errant output. Effective validation requires communication between all interested parties to ensure that the high value system health ground truth information is documented and included in the technology assessment.

A case study for the Boeing 787® Common Motor Start Controller subsystem illustrates how trending and alerting on raw data alone is not enough for effective aircraft system monitoring. A complete understanding of system connectivity and operation states is required. The analytics developed for the CMSC subsystem follow three basic steps. First, they filter and down select the data. Each analytic is designed to target a specific system component. This filtering is achieved through the use of the operational mode flags, system connectivity information, and ACMF report information. Second, statistical power features, oil monitoring, temperature monitoring, and speed monitoring parameters are calculated from the data. These calculated features provide a summary of the target component during the report time period. Third, enumerated flags are generated from reported parameters or calculated features. These flags act to communicate relevant events to the operator such as maintenance procedures, abnormal or damaging environmental conditions and differentiate these noteworthy events from standard operation.

These calculated features can be observed and plotted over time in ASHM to provide insight into fleet trends or individual aircraft trends. Maintainers and engineers can then assess this data to indict specific LRUs and then proactively plan maintenance avoiding service interruptions.

The ASHM software tool is a key enabling technology for condition based maintenance of commercial aircraft, and provides the capability needed to reduce the rate of service interruptions and improve field service logistics operations.

NOMENCLATURE

| ACMF | Aircraft Condition Monitoring Function |
| API  | Application Program Interface          |
| APU  | Auxiliary Power Unit                   |
| ASHM | Aircraft System Health Management      |
| BIT  | Built In Test                          |
| CAC  | Cabin Air Compressor                   |
| CMSC | Common Motor Start Controller          |
| FDE  | Flight Deck Effect                     |
| G    | Generator                               |
| LRU  | Line-replaceable Unit                  |
| PHM  | Prognostics and Health Management      |
| SI   | Service Interruption                   |
| SQL  | Structured Query Language              |
| UTC  | United Technologies Corporation        |

REFERENCES


Real-time Prognostics of a Rotary Valve Actuator

Matthew Daigle

NASA Ames Research Center, Moffett Field, CA, 94035, USA
matthew.j.daigle@nasa.gov

ABSTRACT

Valves are used in many domains and often have system-critical functions. As such, it is important to monitor the health of valves and their actuators and predict remaining useful life. In this work, we develop a model-based prognostics approach for a rotary valve actuator. Due to limited observability of the component with multiple failure modes, a lumped damage approach is proposed for estimation and prediction of damage progression. In order to support the goal of real-time prognostics, an approach to prediction is developed that does not require online simulation to compute remaining life, rather, a function mapping the damage state to remaining useful life is found offline so that predictions can be made quickly online with a single function evaluation. Simulation results demonstrate the overall methodology, validating the lumped damage approach and demonstrating real-time prognostics.

1. INTRODUCTION

Prognostics is a key technology in the area of systems health management. Failure prognostics specifically deals with the prediction of damage progression, end of useful life (EOL), and remaining useful life (RUL) of a component. Based on these predictions, maintenance can be optimized (Tian, Jin, Wu, & Ding, 2011; Camci, 2009) and/or loads can be reallocated to slow damage progression (Bole et al., 2010; Graham, Dixon, Hubbard, & Harrington, 2014). In cryogenic propellant loading systems (Barber, Johnston, & Daigle, 2013; Zeitlin, Clements, Schaefer, Fawcett, & Brown, 2013), most hardware faults are observed in the valves controlling the flow of propellant (Daigle & Goebel, 2011a), therefore, valve prognostics is a critical technology for safe and efficient cryogenic loading operations. Valve prognostics is critical in many other application domains as well.

Previous work in valve prognostics has focused on pneumatically-actuated valves (Daigle, Kulkarni, & Gorospe, 2014; Tao, Zhao, Zio, Li, & Sun, 2014), where the major fault mode is leaks of the pneumatic gas. As leaks grow over time, a significant performance degradation, as measured by valve opening and closing times, can be observed. Our previous work (Daigle et al., 2014; Kulkarni, Daigle, Gorospe, & Goebel, 2014, 2015) developed a model-based prognostics approach for pneumatic valves based on observation of these features.

In this paper, we investigate prognostics of motor-actuated valves, specifically, rotary-actuator quarter-turn valves. In these valves, there is no pneumatic system, and all actuation is electrical-based. Therefore, other damage modes will dominate, such as an increase in friction (Daigle & Goebel, 2011a) and electrical resistance over the life of the component. Further, while the valves in previous work were operated in a discrete open/close fashion, the valves considered here are actuated in a continuous fashion. In particular, they are used in a replenish operation of cryogenic loading, and so are controlled continuously to, in turn, control the flow of propellant in the vehicle tank to replace any propellant that has boiled off while waiting for launch. Thus, the usage of the valve is much more stochastic, and this presents additional challenges to the prognostics problem.

Here, we develop a model-based prognostics approach for rotary valve actuators in this usage context. In our application, only valve position is measured, and we find that friction and resistance faults cannot, as a result, be distinguished. So, a novel lumped-damage model (a concept familiar in structural mechanics (Marante & Flórez-López, 2003)) is used for damage estimation and failure prediction. We develop also new approaches for dealing with the future component usage in this kind of usage context. Further, our goals are for real-time prognostics, i.e., EOL/RUL predictions must be provided in real-time. To this end, we develop also an efficient model-based prediction based on offline model analysis, finding the functional mapping between valve state and RUL, thus avoiding the need for a computationally costly simulation. Although the goal here is to develop an efficient prognostics solution for the particular valve actuator under study, some of the methods developed in this paper can be applied on a more general level.
The paper is organized as follows. Section 2 formulates the prognostics problem and overviews the model-based prognostics approach followed in this paper. Section 3 presents the model of the rotary valve actuator. Section 4 describes the estimation approach, and Section 5 describes the prediction approach. Section 6 demonstrates the overall prognostics approach and presents some experimental results in simulation for validation. Section 7 concludes the paper.

2. MODEL-BASED PROGNOSTICS

In this section, we formulate the prognostics problem, using the framework presented in (Daigle, Sankararaman, & Kulakarni, 2015), which extends the concepts originally presented in (Orchard & Vachtsevanos, 2009; Daigle & Goebel, 2013; Saha & Goebel, 2009). We then provide a computational architecture for model-based prognostics that will be applied to the rotary valve actuator.

2.1. Problem Formulation

We assume the system model may be generally defined as

\[ x(k+1) = f(k, x(k), \theta(k), u(k), v(k)), \]

\[ y(k) = h(k, x(k), \theta(k), u(k), n(k)), \]

where \( k \) is the discrete time variable, \( x(k) \in \mathbb{R}^{n_x} \) is the state vector, \( \theta(k) \in \mathbb{R}^{n_{\theta}} \) is the unknown parameter vector, \( u(k) \in \mathbb{R}^{n_u} \) is the input vector, \( v(k) \in \mathbb{R}^{n_v} \) is the process noise vector, \( f \) is the state equation, \( y(k) \in \mathbb{R}^{n_y} \) is the output vector, \( n(k) \in \mathbb{R}^{n_n} \) is the measurement noise vector, and \( h \) is the output equation. The unknown parameter vector \( \theta(k) \) is used to capture explicit model parameters whose values are unknown and time-varying stochastically.

Prognostics is concerned with predicting the occurrence of some event \( E \) that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold \( T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_{\theta}} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R} \), where \( \mathbb{R} \triangleq \{0, 1\} \), changes from the value 0 to 1. That is, the time of the event \( k_E \) at some time of prediction \( k_P \) is defined as

\[ k_E(k_P) \triangleq \inf \{ k \in \mathbb{N} : k \geq k_P \land T_E(x(k), \theta(k), u(k)) = 1 \}. \] (3)

The time remaining until that event, \( \Delta k_E \), is defined as

\[ \Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \] (4)

In this paper, \( E \) specifically represents the end-of-life event. So, \( k_E \) is EOL and \( \Delta k_E \) is RUL.

2.2. Prognostics Architecture

We adopt a model-based prognostics architecture (Daigle & Goebel, 2013; Daigle & Sankararaman, 2013), in which there are two sequential problems, (i) the estimation problem, which requires determining a joint state-parameter estimate \( p(x(k), \theta(k)|y(k_P)) \) based on the history of observations up to time \( k \), \( Y(k_P) = [y(k_0) \ldots y(k_P)] \), and (ii) the prediction problem, which determines at \( k_P \), using the joint state-parameter estimate \( p(x(k), \theta(k)|y(k_P)) \), the future parameter trajectory \( p(\Theta_{k_P}) \), the future input trajectory \( p(U_{k_P}) \), and the future process noise trajectory \( p(V_{k_P}) \), a probability distribution \( p(k_E(k_P)|Y(k_P)) \).

The prognostics architecture is shown in Fig. 1. In discrete time \( k \), the system is provided with inputs \( u_k \) and provides measured outputs \( y_k \). The estimation module uses this information, along with the system model, to compute an estimate \( p(x(k), \theta(k)|Y(k)) \). The prediction module uses the joint state-parameter distribution and the system model, along with the distributions \( p(\Theta_{k_P}), p(U_{k_P}), \) and \( p(V_{k_P}) \), to compute the probability distribution \( p(k_E(k_P)|Y(k_P)) \). We describe an approach to solve the estimation problem in Section 4, and an approach for the prediction problem in Section 5.

3. VALVE MODELING

We consider here a rotary valve actuator (Flowserve Series 75 Actuator) controlled by a digital positioner (Flowserve DFP17). The actuator consists of a DC electric motor, and moves between 0 and 90°, i.e., it is for a quarter-turn valve. A 4–20 mA input signal is provided to command to a desired position, and the positioner outputs ±24 V to rotate the...
The state variables include the actuator position, \( \theta \), and the actuator velocity, \( \dot{\theta} \):

\begin{align}
\dot{\theta} &= \omega, \\
\dot{\omega} &= \alpha,
\end{align}

where \( \alpha \) is the angular acceleration, which is based on the torques on the actuator. The torques include the motor torque, \( \tau_m \), and the friction torque, \( \tau_f \):

\[ \alpha = \frac{1}{J} (\tau_m - \tau_f), \]

where \( J \) is the rotational inertia. The torques are defined by

\begin{align}
\tau_m &= Ki, \\
\tau_f &= b\omega,
\end{align}

where \( K \) is the motor constant, \( i \) is the motor current, and \( b \) is the friction coefficient.

The DC motor circuit is shown in Fig. 2. Ignoring the (fast) transients in the motor current (i.e., assuming steady-state for the inductance \( L \), in which the voltage drop is zero), it can be expressed as an algebraic function of the motor voltage \( V_m \) and back electromotive force (emf) \( E_b \):

\[ i = \frac{V_m - E_b}{R}, \]

where \( R \) is the motor electrical resistance. The back emf is described by:

\[ E_b = K\omega. \]

The motor voltage is controlled by a positioner. A 4–20 mA input signal, mapping to an angular position between 0 and \( \pi/2 \), is provided. If the motor position needs to increase, then a positive voltage is provided by the positioner to the motor, and if it needs to decrease, a negative voltage is provided. If the position error is within a small deadband, then zero voltage is given. The positioner is described by the following set of equations:

\begin{align}
\theta_d &= \left( \frac{u - 4}{16} \right) \frac{\pi}{2}, \\
e_\theta &= \theta_d - \theta,
\end{align}

where \( \theta_d \) is the desired position, \( u \) is the input signal (in mA), and \( e_\theta \) is the position error. The voltage is then determined using:

\[ V_m = \begin{cases} 
V_u, & \text{if } |e_\theta| > db \text{ and } e_\theta > 0, \\
-V_u, & \text{if } |e_\theta| > db \text{ and } e_\theta < 0, \\
0, & \text{otherwise},
\end{cases} \]

where \( V_u \) is the input voltage, and \( db \) is the deadband.

Here, only one sensor is available, which is the position:

\[ \theta^* = \theta \frac{180}{\pi}, \]

where \( \theta^* \) is the measured position in degrees.

In summary:

\[ x(t) = [\theta(t) \omega(t)]^T, \]

\[ u(t) = [u(t)], \]

\[ y(t) = [\theta^*(t)]. \]

### 3.2. Damage Modeling

Based on discussion with subject matter experts, we consider two distinct damage modes, an increase in internal friction (as captured by the friction coefficient \( b \)) and an increase in internal electrical resistance (\( R \)). We assume that the friction coefficient increases as a function of an unknown wear parameter, \( w_b \), the motor speed, \( \omega \), and the friction force, \( b \cdot \omega \), as described in (Daigle & Goebel, 2013):

\[ \dot{b} = w_b \cdot b \cdot \omega^2, \]

which is based on the basic wear equation (Hutchings, 1992). Note that friction damage only progresses when the valve is in motion.
Similarly, we assume that electrical resistance increases as a function of an unknown wear parameter \( w_R \) and the motor electrical power, \( V_m i \):

\[
\dot{R} = w_R |V_m i|, \tag{17}
\]

Note that resistance will only increase when there is power applied to the motor.

In the extended model, we now have:

\[
x(t) = [\theta(t) \omega(t) b(t) R(t)]^T, \\
\theta(t) = [w_b \ w_R], \\
u(t) = [u(t)], \\
y(t) = [\theta^*(t)].
\]

### 3.3. End of Life

In past valve prognostics applications, the valves were operated in a discrete open/close fashion, so the EOL threshold could be expressed based on required open/close times (Daigle & Goebel, 2011a, 2009). However, in this case, the valve position is controlled continuously, so the valve may never even go through an open/close cycle in actual operation. Thus, the EOL definition must be generalized.

Instead, we can measure degradation through the steady-state valve velocity, \( \omega_{ss} \), which is fundamentally equivalent to using open/close time thresholds. A major difference, however, is that \( \omega_{ss} \) cannot be directly measured.

In the steady state, \( \dot{\omega} = 0 \), and so angular acceleration is zero and the motor and friction torques must balance, i.e., \( \tau_m = \tau_f \) (by Eq. 7). Assuming \( w_{ss} > 0 \), i.e., \( V_m = V_u \), this means that (using substitutions from Eqs. 8, 9, 10 and 11):

\[
Ki = b \omega_{ss}, \\
K \left( \frac{V_u - E_b}{R} \right) = b \omega_{ss}, \\
K \left( \frac{V_u - K \omega_{ss}}{R} \right) = b \omega_{ss},
\]

and, so, solving for \( \omega_{ss} \), we have

\[
\omega_{ss} = \frac{V_u K}{K^2 + bR}. \tag{18}
\]

Now, we express EOL using a minimum steady-state velocity value, \( \omega_{ss}^- \):

\[
T_E = \omega_{ss} \leq \omega_{ss}^-, \tag{19}
\]

i.e., \( E \) (EOL) is reached when the steady-state velocity reaches its minimum value. Since \( \omega_{ss} \) is a function of the damage variables \( b \) and \( R \), given estimates of these variables an estimate of \( \omega_{ss} \) may be computed for the purposes of EOL prediction.

### 4. Estimation

The goal of the estimation step is to compute a joint state-parameter estimate based on the measured system outputs. As described in Section 3, here, only position is measured. However, there are two distinct damage modes that may occur. In fact, we cannot distinguish between these two faults based only on the position sensor. As shown in Fig. 3, the two damage progressions look the same. A zero \( w_H \) and nonzero \( w_b \) can produce observations that look like a nonzero \( w_R \) and a zero \( w_b \), as well as a nonzero \( w_H \) and nonzero \( w_b \). Even without sensor noise it is difficult to observe any difference, so with sensor noise, it will not be possible to distinguish them.

This lack of distinguishability is implied by the \( w_{ss} \) relation. In Eq. 18, both \( b \) and \( R \) appear in the denominator. For example, if \( b \) doubles, \( w_{ss} \) will look the same as if \( R \) doubles, or the product \( bR \) doubles. The damage progressions for \( b \) and \( R \) are similar enough that as they grow over time one can always be mistaken for the other, and for this reason, it will be very difficult to distinguish one from the other or some combination of effects. Fundamentally, this is an observability problem. We have two damage modes but only one sensor, and the estimation problem is under-constrained. In longer time horizons, Fig. 3 suggests that the ambiguity remains.

Since there is practically no hope in distinguishing the damage mode or combination of damage modes occurring, we can simplify the model and use a lumped damage approach. That is, we can estimate an equivalent single damage and damage progression rate, and make EOL predictions based on the lumped damage estimate. Since any combination of damage progressions looks like a friction damage progression, we just use the valve model minus the constraint describing the growth of the resistance parameter (Eq. 17), remove \( R \) from
x(t), and remove \( w_R \) from \( \theta(t) \).\(^2\) If we can still make accurate EOL predictions no matter the combination of damage modes, then this approximation is acceptable.

We use the unscented Kalman filter (UKF) (Julier & Uhlmann, 2004) for joint state-parameter estimation for the lumped damage model. Details on the filter can be found in (Julier & Uhlmann, 2004) and its application to diagnostics in (Daigle, Saha, & Goebel, 2012). To perform joint state-parameter estimation for the UKF, the state vector is augmented with the parameter vector, i.e., unknown parameters are treated as states. The parameters are assumed to evolve only via process noise terms.

It is well-known that the variance used for the process noise for the unknown parameters should be tuned online for optimal performance, and this has been addressed in the context of UKFs (Daigle et al., 2012; Daigle & Goebel, 2013) and particle filters (Orchard, Tobar, & Vachtsevanos, 2009; Liu & West, 2001; Daigle & Goebel, 2011b). We use here the approach developed in (Daigle & Goebel, 2013), in which a relative measure of spread on the unknown parameters (in this case, the wear parameter \( w_b \)) is controlled. A large variance is used initially to encourage convergence, and once convergence is reached, a small variance is used. The variance is increased or decreased proportionally to the error in the desired relative spread (e.g., 10%) from the actual spread currently estimated by the UKF. Details of the algorithm and pseudocode can be found in (Daigle & Goebel, 2013). Here, we initially control to a relative spread of 50% (as measured by relative standard deviation of the estimated wear parameter), and once convergence is achieved (determined by the initial relative spread being reached), then we control the relative spread to 15%.

5. Prediction

The goal of the prediction step is to, given the joint state-parameter estimate, predict EOL and RUL. However, in order to make a prediction, we require also an understanding of the uncertainty in the inputs to the prediction problem (Sankararaman, Daigle, Saxena, & Goebel, 2013). The inputs to the prediction problem include the state-parameter estimate, the future input trajectory, the future process noise trajectory, and the future parameter trajectory. For the purposes of this paper, we assume that process noise is negligible relative to the future input uncertainty, and that the parameters are constant (although uncertain at the time of prediction).

5.1. Future Input Uncertainty

Consideration of the future input uncertainty is critical to making accurate and useful predictions. Characterizing this uncertainty depends heavily on the application at hand. For the rotary valve actuator, the valve is used in a replenish mode for cryogenic propellant loading, in which the flow through the valve is controlled to replace any boil off in the vehicle tank due to heat exchange with the environment.

There are two important things to note about its usage. First, the valve is used only in a certain mode of operation. Thus, it makes sense only to report EOL/RUL in terms of usage time, not absolute time. This is because \( i \) we do not know how long the replenish operation will take, and \( ii \) we do not know how long the valve will be sitting unused between replenish operations.

Second, the valve can be in use but with no damage progressing. There must be motion of the actuator for the friction damage to increase (by Eq. 16) and current flowing through the actuator motor for the resistance damage to increase (by Eq. 17). If the actuator has reached its position setpoint within the deadband, then motor voltage and current are zero, and the valve is not moving, but the valve is still being used in the sense that it is receiving a command and responding to it. Thus, if we want to make a prediction in terms of usage hours, we must acknowledge the fact that for an (unknown) percentage of the time the valve is being used, it is not moving and damage is not progressing.

Given these considerations, a true future input trajectory will be interspersed with both nonusage time and usage time, and the usage time will be further divided into time in which the actuator is moving and damage is progressing, and time in which the actuator is not moving and damage is not progressing. If we only want a prediction in terms of usage hours, then we do not need to waste time simulating trajectories to EOL including nonusage time. We do not also need to waste time simulating trajectories in which there are intervals in which damage is not progressing; instead, we can predict assuming damage is always progressing, and then correct the predictions using statistical information on the percentage of time the actuator is actually moving.

So, to make an RUL prediction, we need only a subset of the model in which we compute \( T_E \) using \( V_m \) as an input, in which \( V_m \) is always 24 V. In this case, the actuator will continuously move and damage continuously progress. The needed submodel can be derived from the model equations using the general structural model decomposition framework presented in (Roychoudhury, Daigle, Bregon, & Pulido, 2013). Using the GenerateSubmodel algorithm in that work, we can compute the minimal subset of model equations needed to compute \( T_E \) using \( V_m \). In this case, we find that we require only Eqs. 6–11, 16, 18, and 19.

So, given a sample of the system state, we can simulate until \( T_E \) evaluates to 1 and obtain corresponding EOL and RUL predictions. In order to correct these predictions, we require

\(^2\)Since any combination of damage progressions also looks like a resistance damage progression, we could equivalently use that model to represent the lumped damage instead.
Algorithm 1 $k_E(k_P) \leftarrow \mathcal{P}(x(k_P), \Theta_{k_P}, U_{k_P}, V_{k_P})$

1: $k \leftarrow k_P$
2: $x(k) \leftarrow x(k_P)$
3: while $T_E(x(k), \Theta_{k_P}(k), U_{k_P}(k)) = 0$ do
4: \hspace{1em} $x(k + 1) \leftarrow f(k, x(k), \Theta_{k_P}(k), U_{k_P}(k), V_{k_P}(k))$
5: \hspace{1em} $k \leftarrow k + 1$
6: \hspace{1em} $x(k) \leftarrow x(k + 1)$
7: \hspace{0em} end while
8: $k_E(k_P) \leftarrow k$

Statistics on the percentage of the actuator usage in which it is actually moving. We assume that, while being used, the statistics of the past and future behavior in this context are equivalent. Thus, we use the history of past sensor measurements to keep track of when the actuator is moving and when it is not in order to compute these statistics. We have a binary distribution; either the actuator is moving or not. We define $f_m$ as the fraction of the time that the actuator is moving while in usage. The actuator is considered to be moving when the absolute value of the estimated velocity is greater than some threshold (i.e., 0.01 rad/s). Then, $f_m$ is computed as the amount of usage time this condition is satisfied over the total amount of usage time. We then compute the corrected RUL, $\Delta k_{E,f_m}$, based on the predicted RUL assuming full-time movement, $\Delta k_{E,100\%}$, as:

$$\Delta k_{E,f_m} = \frac{\Delta k_{E,100\%}}{f_m}. \quad (20)$$

The corrected EOL can then be computed based on corrected RUL:

$$k_{E,f_m} = \Delta k_{E,f_m} + k_P, \quad (21)$$

where $k_P$ is the time of prediction.

5.2. RUL Computation

Given the state estimate, we can sample from its distribution, simulate each sample to $E$, and obtain corresponding $k_E$ values using Algorithm 1 (Daigle & Sankaranarayanan, 2013). We can then correct these predictions using Eqs. 20 and 21 to obtain the desired EOL/RUL distribution. Recall, however, that we have the requirement of real-time prognostics, that is, we must compute the prediction for time $k$ before time $k + 1$. This is a difficult problem, considering that we must perform a simulation to $E$, and the amount of time the simulation takes depends on the rate of damage progression, which, in reality, is very small.

A solution to this problem is to move the simulations from online computation to offline, design-time computation. We may construct a lookup table for this purpose, in which different values of the state are simulated to EOL and the result stored, so that a mapping from states to RUL is established.

Lookup tables have been used previously in prognostics in the context of damage estimation (Teubert & Daigle, 2013, 2014; Daigle et al., 2014). However, the problem with a lookup table is that it has only a finite number of elements, so the precision and range of the table is finite. In reality, the wear rates can take a variety of values and it is difficult to capture a suitable domain. The granularity of the table will also determine the precision of the RUL predictions available.

So, instead of using a lookup table, here, we find the direct functional mapping from the actuator state to RUL, i.e., $\Delta k_E = g(x)$. In the lumped damage approximation, and assuming $f_m = 100\%$, only $b$ and $w_b$ will have an effect on the RUL prediction, so we have to find a function of only two inputs that computes RUL, i.e., $\Delta k_E = g(b, w_b)$. We first simulate to EOL for a range of states, and then use optimization methods to fit the function $g$ to the values.

To determine the structure of this equation, we first consider only a single value of $w_b$, so we have RUL as a function of only $b$. We find that in this case, a second order polynomial fits very well for any given $w_b$, as shown in Fig. 4:

$$p_{b,0} + p_{b,1}b + p_{b,2}b^2. \quad (22)$$

Now, we need to determine how the $p_b$ parameters change as a function of $w_b$. We find that these coefficients are propor-
We consider first a scenario in which only friction damage is present, and the desired position is constantly changing, so that the valve is always in motion (i.e., $f_m \approx 100\%$). Fig. 7 shows the lumped damage estimation, and Fig. 8 shows the wear parameter estimate, including the mean, minimum, and maximum values from the sigma points of the UKF. With the UKF and the properly tuned variance control algorithm, convergence happens relatively quickly, in about 1000 s (roughly 10% of the true $k_E$). Due to noise and the slow damage progression, the mean deviates, but the resulting predictions are fairly accurate, as shown in Fig. 9. Using a relative accuracy measure of $\alpha = 0.25$, we find that the mean $\Delta k_E$ prediction falls within the accuracy bounds for most of the time after estimation convergence. In this case, $f_m$ is estimated to be slightly less than 100%, so the corrected predictions are shifted up slightly. This error is due to the noise introduced in the $f_m$ computation as a result of the uncertain estimate of the actuator velocity and the use of the velocity threshold for determining if the valve is moving.

We consider next a similar scenario, except where only resis-
tance damage is present. Fig. 10 shows the lumped damage estimate, and here it is clear that its shape looks very similar to that when only friction damage is present (Fig. 7). As a result, the wear parameter estimate, shown in Fig. 11, converges, although it is not as steady as the estimate for only friction damage. The predictions, shown in Fig. 12, are fairly accurate once convergence of the wear parameter estimate occurs (which is relatively slower than in the previous scenario).

We consider next a scenario in which both friction and resistance damage modes are progressing, and the component usage is where $f_m \approx 75\%$. Here, the estimation performance is similar to the first scenario, and the wear parameter estimate converges relatively quickly, even though both damage modes are present, i.e., the lumped damage approximation works fairly well. As $kF$ is approached, the actual future $f_m$ is higher than estimated so the uncorrected predictions be-
come more accurate, i.e., the assumption that the future value of $f_m$ is the same as the past value is violated, resulting in inaccurate predictions. In practice, some uncertainty should be considered for $f_m$.

Finally, we consider a similar scenario to the last, except with $f_m \approx 50\%$. Estimation results are similar as in the previous scenario; the wear parameter estimate converges and remains approximately the same once convergence is achieved. The estimate for $f_m$ is shown in Fig. 14, and $\Delta k_E$ predictions are shown in Fig. 15. It takes some time for the $f_m$ estimate to converge. Early on, $f_m$ is higher than it will be in the future, and so the predictions are overly optimistic, falling outside the accuracy bounds. Convergence occurs at roughly 3500 s, at which point the predictions become much more accurate (here, fluctuations are due to those in the wear parameter estimate). After this point, the corrected predictions are much more accurate than the uncorrected ones, which are too pessimistic (overly conservative).

Overall, performance is quite good. Regarding the real-time performance, each second of real time takes only about 0.024 s of processing, which includes both estimation and prediction. Thus, the goal of real-time prognostics is easily achieved, being able to run over 40 times faster than real time.

7. Conclusions

In this paper, we applied the model-based prognostics approach to a rotary valve actuator. This case study presented several challenges for which new methods were developed to achieve prognostics. First, due to limited observability, a lumped damage approximation was used, which was found to be sufficiently accurate in this case. Note, however, that this will not apply in general; it works here due to the similar effect of the two independent damage modes on the single observable variable. Second, real-time prognostics was a requirement, so instead of online simulation to EOL, offline simulations were performed and an algebraic function was found mapping the damage space to RUL. Thus, for any system state, RUL could be computed extremely quickly, enabling the real-time performance requirement. This approach can be applied to any system, although in some cases it may be difficult to find such a function, especially for high-dimensional damage spaces. Third, we implemented an RUL correction procedure, since predictions were based on 100% movement of the actuator, when in reality during an actual usage, damage only progresses when the valve is moving, which, in our usage context, was stochastic.

In future work, some uncertainty regarding $f_m$ should be considered, as it was not generally true that $f_m$ computed over past values matched $f_m$ computed over future values. Second, accelerated damage progressions (high wear rate values) were considered here for the purposes of demonstration. In reality, wear rates will be much smaller, and this may require
the splitting of dynamics into fast- and slow-time dynamics, with different estimation methods for each (Luo, Pattipati, Qiao, & Chigusa, 2008).

ACKNOWLEDGMENT

This work was supported by the NASA’s Advanced Ground Systems Maintenance (AGSM) Project in the Ground Systems Development and Operations (GSDO) Program in the Human Exploration and Operations Mission Directorate.

REFERENCES


Tao, T., Zhao, W., Zio, E., Li, Y.-F., & Sun, J. (2014).
Condition-based component replacement of the pneumatic valve with the unscented particle filter. In *Progno-
nostics and System Health Management Conference* (pp. 290–296).


Model-based Prognostics of Hybrid Systems

Matthew Daigle¹, Indranil Roychoudhury², and Anibal Bregon³

¹ NASA Ames Research Center, Moffett Field, California, 94035, USA
matthew.j.daigle@nasa.gov

² Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, California, 94035, USA
indranil.roychoudhury@nasa.gov

³ Department of Computer Science, University of Valladolid, Valladolid, Spain
anibal@infor.uva.es

ABSTRACT

Model-based prognostics has become a popular approach to solving the prognostics problem. However, almost all work has focused on prognostics of systems with continuous dynamics. In this paper, we extend the model-based prognostics framework to hybrid systems models that combine both continuous and discrete dynamics. In general, most systems are hybrid in nature, including those that combine physical processes with software. We generalize the model-based prognostics formulation to hybrid systems, and describe the challenges involved. We present a general approach for modeling hybrid systems, and overview methods for solving estimation and prediction in hybrid systems. As a case study, we consider the problem of conflict (i.e., loss of separation) prediction in the National Airspace System, in which the aircraft models are hybrid dynamical systems.

1. INTRODUCTION

Prognostics deals with predicting the occurrence of some system event, so that decisions can be made either autonomously or by human operators to improve system performance in some way. For example, in failure prognostics, end-of-life (EOL) is predicted so that system usage can be modified to extend system life or maintenance planned (Camci, 2009; Tian, Jin, Wu, & Ding, 2011; Saha & Goebel, 2009; Orchard & Vachtsevanos, 2009). Engineering systems are becoming increasingly complex, and often consist of a tight integration between hardware and software components. As such, most real-world engineering systems exhibit hybrid dynamics, i.e., a mix between continuous and discrete dynamics. This feature of modern-day systems makes prognostics more complex.

A significant amount of research has been performed dealing with hybrid systems in the areas of modeling, verification, diagnosis, and control, among others. Several modeling paradigms have been developed to represent hybrid system dynamics, such as hybrid automata (Henzinger, 2000) and hybrid bond graphs (P. J. Mosterman & Biswas, 1998; P. Mosterman & Biswas, 2000). A significant amount of research exists on diagnosis of hybrid systems (Narasimhan & Biswas, 2007; Narasimhan & Brownston, 2007; McIlraith, 2000; Koutsoukos et al., 2003; Hofbaur & Williams, 2004; Bayoudh et al., 2008; Bregon et al., 2011; Cocquempot et al., 2004; Daigle et al., 2010), yet little work exists on prognosis of hybrid systems. Only recently have approaches for hybrid systems prognostics been investigated (Chanthery & Ribot, 2013; Zabi et al., 2013; Gaudel et al., 2014; Yu et al., 2011). In (Chanthery & Ribot, 2013) and (Zabi et al., 2013), hybrid automata models are used, and in (Gaudel et al., 2014), a new formalism, hybrid particle petri nets, is introduced. These works are focused mainly on the integration of diagnosis and prognosis for hybrid systems. The prognosis aspect is limited in that it is focused specifically on the subproblem of failure prognostics, and, further, aging laws specifically take the form of Weibull models. In (Yu et al., 2011), hybrid bond graphs are used, but the approach is similarly limited.

In contrast, our aim is to develop a general, model-based prognostics framework for hybrid systems. We adopt a compositional, component-based modeling approach (Daigle, Bregon, & Roychoudhury, 2015). The modeling approach is inspired by hybrid bond graphs, but does not restrict component dynamics to a fixed set as with HBGs. We advance the theory of model-based prognostics to the more general formulation for hybrid systems, and describe the complexities introduced for prognostics with hybrid system models. Al-
algorithms for estimation and prediction using hybrid system models are also discussed. To demonstrate the approach, we look at the problem of conflict (i.e., loss of separation) prediction within the National Airspace System (NAS) (Erzberger, Paielli, Isaacson, & Eshow, 1997; Tomlin, Pappas, & Sastry, 1998).

The paper is organized as follows. Section 2 develops the model-based prognostics framework for hybrid systems. Section 3 discusses hybrid system estimation, and Section 4 covers the prediction problem. Section 5 develops the case study and demonstrates the approach. Section 6 concludes the paper.

2. Model-based Prognostics

In this section, we first describe our hybrid systems modeling paradigm. We then formulate the prognostics problem for hybrid systems, and present a computational architecture.

2.1. Hybrid Systems Modeling

We define hybrid system dynamics in a general compositional way, where the system is made up of a set of components. Each component is defined by a set of discrete modes, with a different set of constraints describing the continuous dynamics of the component in each mode. Here, system-level modes are defined implicitly through the composition of the component-level modes.

At the basic level, the continuous dynamics of a component in each mode are modeled using a set of variables and a set of constraints. A constraint is defined as follows:

**Definition 1** (Constraint). A constraint $c$ is a tuple $(\varepsilon, V_c)$, where $\varepsilon$ is an equation involving variables $V_c$.

A component is defined by a set of constraints over a set of variables. The constraints are partitioned into different sets, one for each component mode. A component is then defined as follows:

**Definition 2** (Component). A component $\delta$ with $n$ discrete modes is a tuple $\delta = (V_\delta, C_\delta)$, where $V_\delta$ is a set of variables and $C_\delta$ is a set of constraint sets involving variables in $V_\delta$, where $C_\delta$ is defined as $C_\delta = \{C_{\delta_1}^1, C_{\delta_2}^2, \ldots, C_{\delta_n}^n\}$, with a constraint set, $C_{\delta_m}$, defined for each mode $m = \{1, \ldots, n\}$.

By composing a set of components, we can define a system model as follows:

**Definition 3** (Model). A model $\mathcal{M} = \{\delta_1, \delta_2, \ldots, \delta_d\}$ is a finite set of $d$ components for $d \in \mathbb{N}$.

Note that the set of variables for a model does not change with the mode, hence we need only a variable set in a component and not a set of variable sets as with constraints. The set of variables for a model, $V_M$, is simply the union of all the component variable sets, i.e., for $d$ components, $V_M = V_{\delta_1} \cup V_{\delta_2} \cup \ldots \cup V_{\delta_d}$. We say that two components are connected if they share a variable, i.e., components $\delta_i$ and $\delta_j$ are connected if $V_{\delta_i} \cap V_{\delta_j} \neq \emptyset$. $V_M$ consists of five disjoint sets, namely, the set of state variables, $X_M$; the set of parameters, $\Theta_M$; the set of inputs (variables not computed by any constraint), $U_M$; the set of outputs (variables not used to compute any other variables), $Y_M$; and the set of auxiliary variables, $A_M$. Parameters, $\Theta_M$, include explicit model parameters that are used in the model constraints (e.g., fault parameters). Auxiliary variables, $A_M$, are additional variables that are algebraically related to the state, parameter, and input variables, and are used to simplify the structure of the equations.

The model constraints, $C_M$, are a union of the component constraints over all modes, i.e., $C_M = C_{\delta_1} \cup C_{\delta_2} \cup \ldots \cup C_{\delta_d}$, where $C_{\delta_m} = C_{\delta_m}^1 \cup C_{\delta_m}^2 \cup \ldots \cup C_{\delta_m}^n$ for $n$ modes. Constraints are exclusive to components, that is, a constraint $c \in C_M$ belongs to exactly one $C_{\delta_m}$ for $\delta_m \in \mathcal{M}$.

To refer to a particular mode of a model we use the concept of a mode vector. A mode vector $\mathbf{m}$ specifies the current mode of each of the components of a model. So, the constraints for a mode $\mathbf{m}$ are denoted as $C_M^\mathbf{m}$. For shorthand, we will refer to the modes only of the components with multiple modes.

The switching behavior of each component can be defined using a finite state machine or a similar type of control specification. The state transitions may be attributed to controlled or autonomous events. However, for the purposes of this paper, we view the switching behavior as a black box where the mode change event is given, and refer the reader to many of the approaches already proposed in the literature for modeling the switching behavior (Henzinger, 2000; P. J. Mosterman & Biswas, 1998). We distinguish between two types of mode changes, controlled and autonomous, where controlled mode changes depend solely on the system inputs, and autonomous mode changes depend also on internal system variables.

2.2. Problem Formulation

The system model is constructed as a collection of components describing the system variables and the constraints describing their relationships (Defn. 3). By collecting all the constraints over each mode, the model variables and equations can be presented in a summarized, abstract form, consisting of the (continuous) state vector, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$; the mode (or, discrete state) vector, $\mathbf{m}(k) \in \mathbb{N}^{n_m}$; the unknown parameter vector, $\mathbf{\theta}(k) \in \mathbb{R}^{n_{\theta}}$; the input vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$; the process noise vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$; the output vector, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$; the measurement noise vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$; the state equation, $f$; the mode transition equation, $g$; and the
output equation, $\mathbf{h}$:

$$
\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{m}(k), \mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)),
$$

$$
\mathbf{m}(k+1) = \mathbf{g}(k, \mathbf{m}(k), \mathbf{x}(k), \mathbf{u}(k)),
$$

$$
\mathbf{y}(k) = \mathbf{h}(k, \mathbf{m}(k), \mathbf{x}(k), \mathbf{u}(k), \mathbf{n}(k)),
$$

where $k$ is the discrete time variable. Here, the full state of the hybrid system is defined by both the continuous state $\mathbf{x}$ and the mode $\mathbf{m}$. Note that the unknown parameter vector $\theta(k)$ is used to capture explicit model parameters whose values are unknown and time-varying stochastically. This presentation of the model is used here for problem formulation and does not necessarily represent implementation.

Prognostics is concerned with predicting the occurrence of some event $E$ that is defined with respect to the mode, states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold $T_E : \mathbb{N}^{n_m} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_{\theta}} \rightarrow \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$, changes from the value 0 to 1. That is, the time of the event $k_E$ at some time of prediction $k_P$ is defined as

$$
k_E(k_P) \triangleq \inf\{k \in \mathbb{N} : k \geq k_P \land T_E(\mathbf{m}(k), \mathbf{x}(k), \theta(k), \mathbf{u}(k)) = 1\}.
$$

The time remaining until that event, $\Delta k_E$, is defined as

$$
\Delta k_E(k_P) \triangleq k_E(k_P) - k_P.
$$

### 2.3. Prognostics Architecture

We adopt a model-based prognostics architecture (Daigle & Goebel, 2013; Daigle & Sankararaman, 2013), in which there are two sequential problems, (i) the estimation problem, which requires determining a joint state-parameter estimate $p(\mathbf{m}(k), \mathbf{x}(k), \theta(k) | \mathbf{Y}(k))$ based on the history of observations up to time $k$, $\mathbf{Y}(k) = [\mathbf{y}(k_0) \ldots \mathbf{y}(k)]$, and (ii) the prediction problem, which determines at $k_P$, using the joint state-parameter estimate $p(\mathbf{m}(k_P), \mathbf{x}(k_P), \theta(k_P) | \mathbf{Y}(k_P))$, the future parameter trajectory $p(\Theta_{k_P})$, the future input trajectory $p(\mathbf{U}_{k_P})$, and the future process noise trajectory $p(\mathbf{V}_{k_P})$, a probability distribution $p(k_E(k_P) | \mathbf{Y}(k_P))$.

The prognostics architecture is shown in Fig. 1. In discrete time $k$, the system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{m}(k), \mathbf{x}(k), \theta(k) | \mathbf{Y}(k))$. The prediction module uses the joint state-parameter distribution and the system model, along with the distributions $p(\Theta_{k_P})$, $p(\mathbf{U}_{k_P})$, and $p(\mathbf{V}_{k_P})$, to compute the probability distribution $p(k_E(k_P) | \mathbf{Y}(k_P))$. We describe the estimation problem in Section 3, and the prediction problem in Section 4.

### 3. Estimation

As we have already mentioned, the estimation problem in our approach deals with determining a joint state-parameter estimate $p(\mathbf{m}(k), \mathbf{x}(k), \theta(k) | \mathbf{Y}(k))$ based on the history of observations up to time $k$, $\mathbf{Y}(k)$. A general solution to the problem of joint state-parameter estimation is the unscented Kalman filter (UKF) (Julier & Uhlmann, 2004, 1997), which may be applied to nonlinear systems with Gaussian noise. Another solution highly used in the estimation literature is the particle filter (PF) (Arulampalam, Maskell, Gordon, & Clapp, 2002), which may be directly applied to nonlinear systems with non-Gaussian noise terms (Arulampalam et al., 2002). The main advantage of the PF against the UKF is the computational complexity, which is linear in the amount of samples, or particles, that are used to approximate the state distribution. The number of particles needed for joint state-parameter estimation is typically larger, and this number increases with the dimension of the state-parameter space.

For hybrid systems, the problem of joint state-parameter estimation becomes more complicated due to the mode changes. In the case the mode of the system is known, which happens when the system only has controlled mode changes, the problem reduces to continuous system state estimation, and the approaches mentioned above can be applied. A comparison of different filter techniques for the estimation problem in prognostics can be found in (Daigle, Saha, & Goebel, 2012). On the other hand, when autonomous mode changes can occur in the system, the estimation problem becomes more complex. Since the main focus of this work is the...
prediction problem of the prognostics architecture, we refer the reader to one of the many existing state estimation approaches in the literature (Riemmüller, Bayoudh, Hofbaur, & Travé-Massuyès, 2009; Benazera & Travé-Massuyès, 2009; Hofbaur & Williams, 2004; Koutsoukos et al., 2003; Blom & Bloem, 2004).

In (Hofbaur & Williams, 2004; Benazera & Travé-Massuyès, 2009) the authors propose an approach that requires to follow every possible state in the system at a particular instant, and then use search techniques from model-based reasoning in order to focus the estimation on the set of most likely modes, without missing symptoms that might be hidden among the system noise. Riemmüller et al. (Riemmüller et al., 2009) also use the state estimator proposed in (Hofbaur & Williams, 2004). In (Koutsoukos et al., 2003), the authors propose a particle filtering based estimation algorithm for a class of distributed hybrid systems. Finally, the approach in (Blom & Bloem, 2004) also uses particle filtering. Each one of this approach can be perfectly integrated within our prognostics framework to precede the prediction problem.

4. Prediction

The prediction problem is to find the time of some system event E and/or system variables at the time of that event. In the model-based paradigm, the approach is conceptually straightforward. Given the system model, an initial state, and future inputs, we simulate the model forward in time until the event occurs. Of course, in practice, this is difficult because the inputs to this problem are all uncertain. In this context, the prediction problem becomes one of uncertainty propagation (Sankararaman, Daigle, Saxena, & Goebel, 2013; Sankararaman, Daigle, & Goebel, 2014).

At prediction time \( k_P \), we consider four different inputs to the prediction problem: (i) the initial state estimate, \((m(k_P), x(k_P), \Theta_{k_P})\); (ii) the future parameter trajectory, \(\Theta_{k_P} \) (where \(\Theta(k_P)\) is the first value); (iii) the future input trajectory, \(U_{k_P}\); and (iv) the future process noise trajectory \(V_{k_P}\). The initial state estimate comes from the estimation module and is given as a probability distribution. The remaining uncertain inputs must be determined in some way either a priori or by using data from the system. For example, we can assume that the future inputs look like the past inputs over some time window with some uncertainty (Daigle, Saxena, & Goebel, 2012).

Underlying the overall prediction algorithm is the function \( \varpi \) (Algorithm 1) that, given a realization of all the prediction inputs, computes the corresponding value of \( k_E \). The algorithm simply simulates forward the model given the inputs. The version presented here extends the version for continuous systems (Daigle & Sankararaman, 2013) by including the discrete state, \( m(k) \), and the mode transition equation, \( g \) (Eq. 2). Further, it introduces a finite prediction horizon, specified by \( k_H \). If \( E \) is not reached by \( k_H \), then \( \infty \) is returned, meaning that \( E \) was not reached in the specified prediction horizon.

Prediction algorithms differ by how they make use of the \( \varpi \) function. In Monte Carlo sampling, inputs are sampled stochastically from the given distributions, and \( \varpi \) is called many times, once for each sample. Other algorithms like unscented transform sampling and the inverse first-order reliability method offer more complicated sampling or analytical schemes (Daigle & Sankararaman, 2013). For hybrid systems, these approaches become more complicated because there is a mix of discrete and continuous probability distributions. The presence of a discrete mode and the possibility of future mode changes means that the distribution for \( k_E \) will typically be multi-modal, which causes additional difficulties. The problem is present for both autonomous and controlled mode changes. Uncertainty in the initial state, process noise, and inputs means that, for some realizations, autonomous mode changes may occur, whereas for others it may not, and uncertainty in the future input trajectory means that controlled mode changes may occur for some realizations but not for others.

Although it can be inefficient due to the large number of sampling typically required, in this work we use Monte Carlo sampling, as it is the most straightforward to apply for hybrid systems; the algorithm can be directly used given distributions for all the prediction inputs. Since we consider a finite prediction horizon here, that limits the computational complexity of the algorithm. In the worst case, the complexity is given by the number of samples times the difference \( k_H - k_P \).

5. Case Study

The NAS consists of commercial flights, general aviation flights, air traffic controllers (ATCs), airports, etc. There are many prediction problems to consider within the NAS, in order to maintain safety and increase efficiency. We consider the problem of conflict prediction in the NAS. A conflict, or

\begin{algorithm}
\begin{algorithmic}[1]
\State \( k \leftarrow k_P \)
\State \( x(k) \leftarrow x(k_P) \)
\While {\( T_E(m(k), x(k), \Theta_{k_P}(k), U_{k_P}(k)) = 0 \quad \text{or} \quad k < k_H \)}
\State \( x(k + 1) \leftarrow f(k, m(k), x(k), \Theta_{k_P}(k), U_{k_P}(k), V_{k_P}(k)) \)
\State \( m(k + 1) \leftarrow g(k, m(k), x(k), \Theta_{k_P}(k), U_{k_P}(k)) \)
\State \( k \leftarrow k + 1 \)
\EndWhile
\If {\( k = k_H \)}
\State \( k_E(k_P) \leftarrow \infty \)
\Else
\State \( k_E(k_P) \leftarrow k \)
\EndIf
\end{algorithmic}
\end{algorithm}
loss of separation, is defined as an event in which two planes come within an unsafe distance from one another. In present-
day operations, conflicts are predicted within a 20 minute pre-
diction horizon (Erzberger et al., 1997). If a conflict is pre-
dicted, then an ATC must resolve it by issuing instructions to
the aircraft to turn, change speed, or change altitude. We want
to predict conflicts in order to determine if conflict-resolving
maneuvers need to be issued. Such predictions serve as inputs
to a decision-making problem. An estimate of uncertainty is
also required in order to maintain safety and avoid risks.

5.1. Modeling

In order to apply model-based prognostics to this problem,
we require a model of the system. In our modeling frame-
work, each aircraft with its controller defines a component.
The system is constructed as the set of all considered air-
craft. We follow the approach in (Bilmoria, Banavar, Chat-
terji, Sheth, & Grabbe, 2000; Chatterji, Sridhar, & Bilimo-
ria, 1996) for modeling the aircraft, using point-mass models
with simplified dynamics. For each aircraft, we consider two
components, one representing the aircraft dynamics, and one
for the control. The dynamics always remain the same, but
the control laws change depending on the flight phase (climb,
cruise, or descent). The component equations are summa-
rized in Table 1. While more complex and realistic equations
may be used to describe the dynamics and control, the simpli-
ified equations here are actually quite close to what is used in
practice, and suitable for demonstrating hybrid system prog-
nosis concepts.

We assume a point-mass model as in most trajectory predic-
tion methods in the literature (e.g., (Slattery & Zhao, 1997;
Erzberger et al., 1997; Chatterji et al., 1996)). The aircraft
position is defined by latitude λ, longitude τ, altitude h, head-
ing χ (the angle on the horizontal plane), and flight path angle
γ (the angle on the vertical plane). The velocity is described
by the airspeed \( V \) and the climb velocity \( V_c \). The ground-
speed \( V_g \) is computed based on an addition of the airspeed
and windspeed vectors. The windspeed vector is defined by
magnitude \( V_w \) and an angle \( \chi_w \). We assume there is no verti-
cal component to the wind. It can be decomposed into north
and east components, \( W_N \) and \( W_E \), respectively.

Latitude and longitude change based on the component of
the airspeed and windspeed in the horizontal plane and the radius
\( R \), which is computed as the radius of the earth \( R_e \) and
the altitude. Altitude changes directly based on the climb veloc-
ity. As in (Bilmoria et al., 2000), we assume simple dynam-
ics where the airspeed, climb velocity, and heading change
are based on abstracted first-order systems. For each, they
change based on the error between the current value and a
commanded value (\( c \) subscript) with some lumped inertia \( J \).

The control laws aim to point the aircraft to the desired
heading, and control the velocity according to the phase
of flight. The desired heading angle is computed as the
great-circle heading, \( \chi_{GC} \) with a wind correction term
\( \text{arcsin}(V_w/V_c \sin(\chi_w - \chi_q)) \) (Chatterji et al., 1996). The
desired climb velocity is zero in the cruise phase, and in the
climb and descent phases it is determined with a proportional
control law based on the altitude error. The commanded air-
speed is fixed for climb and descent (\( V_{climb}, V_{descent} \),
respectively), and for cruise it is determined based on the
distance to be traveled, \( D \), and the desired time to arrive
at the waypoint, \( t^* \). The desired waypoint is specified by
desired latitude, longitude, altitude, and time of arrival,
\( (\lambda^*, \tau^*, h^*, t^*) \).

The mode transition equation is implemented as follows. In
nominal operations, an aircraft goes from the climb mode, to
cruise mode, then descent mode for the final waypoint (i.e.,
the arrival runway), which are autonomous mode transitions.
The aircraft goes from climb or descent to cruise when it
reaches the desired altitude \( h^* \) within some error bounds. It
goes from cruise to climb or descent when the desired altitude
is lower than the current altitude within some error bounds.
Thus, in nominal operations, the aircraft follows the given
waypoints and switches between the control modes based on
these waypoints. If a conflict will occur, then an ATC action
can also cause the aircraft to switch modes. Predictions will
be made to determine if such actions need to take place.

The event of interest, \( E \), is a conflict, which is defined as two
aircraft being within 5 nautical miles and 1000 vertical feet of
one another. Since this event is defined over multiple aircraft,
we require a system-level perspective, where the system is
the NAS or a specific region within it (Daigle, Bregon, &
Roychoudhury, 2012).

The inputs to the system are the set of desired waypoints for
each aircraft and the wind. We assume that there is no uncer-
tainty in the desired waypoints. Without a model of how the
wind changes, we assume that the windspeed and direction
are steady within the 20 minute prediction horizon, and their
values fall within some assumed probability distribution.

5.2. Demonstration of Approach

We consider a scenario in which to demonstrate the overall
approach. Here, we have a region of the NAS in which five
aircraft are heading to different navigation waypoints. For
each aircraft, the probability of a conflict with another within
the next 20 minutes is computed, as well as the estimated
time of the conflict. Both the state and the wind are consid-
ered to be uncertain and captured through normal distribu-
tions. Process noise is ignored and parameters are assumed
to be known. Monte Carlo sampling is used for prediction
with 500 samples. Each sample is associated with a time of
conflict, from which the probability distribution of conflict
time is produced. The probability of a conflict is computed as
the number of samples with finite conflict times (i.e., 20 min-
utes).
The initial aircraft positions are shown in Fig. 2. Each aircraft is numbered, and their waypoints are drawn along with straight-line flight paths to their waypoints. The circle drawn around each aircraft represents the required separation distance (the aircraft are not drawn to scale). From the plot, it appears that conflicts may arise between aircraft 1 and aircraft 3, aircraft 2 and aircraft 3, and aircraft 4 and aircraft 5, as their intended flight paths cross. The predictions at $t = 0$ minutes are that aircraft 1 and aircraft 3 have a 76% probability of conflict, occurring between 13.3 and 15.7 minutes, that aircraft 2 and aircraft 3 have a 88% probability of conflict, occurring between 4.3 and 8.2 minutes, and that aircraft 4 and aircraft 5 have a 0.02% probability of conflict, occurring around 11 minutes.

Without intervention, the conflict between aircraft 2 and aircraft 3 appears at 6.6 minutes (see Fig. 3), so the original prediction captures the true conflict time. As the actual time of conflict is approached, the estimated probability of conflict increases, as shown in Fig. 4. After the actual conflict, the probability of conflict reduces. The drop to zero is not discontinuous, since there is uncertainty in the aircraft positions.

Without intervention, the conflict between aircraft 1 and aircraft 3 occurs at 14.6 minutes (see Fig. 5). The probability of conflict over time is shown in Fig. 6. Due to position uncertainty, the probability at the time of conflict is only around 50%, as the conflict is just on the border of not happening.

> Table 1. Components of the NAS

<table>
<thead>
<tr>
<th>Component</th>
<th>Mode</th>
<th>Constraints</th>
</tr>
</thead>
</table>
| Dynamics  | 1    | $W_N = V_w \sin(\chi_w)$  
|           |      | $W_E = V_w \cos(\chi_w)$  
|           |      | $\lambda = (V_t \cos \gamma \cos \chi + W_N)/R$  
|           |      | $\tau = (V_t \cos \gamma \sin \chi + W_E)/(R \cos \lambda)$  
|           |      | $h = V_h$  
|           |      | $R = R_e + h$  
|           |      | $\gamma = \arcsin(h/V_t)$  
|           |      | $V_t = (V_{e,c} - V_t)/J_t$  
|           |      | $V_h = (V_{h,e} - V_h)/J_h$  
|           |      | $\chi = (\chi_e - \chi)/J_\chi$  
|           |      | $V_{g,y} = V_{c} \cos \chi + V_0 \cos \chi_w$  
|           |      | $V_{g,y} = V_{c} \cos \chi + V_0 \cos \chi_w$  
|           |      | $V_g = V_{c} \cos \chi + V_0 \cos \chi_w$  
|           |      | $\chi_g = \arctan(V_{g,y}/V_{g,y})$  
|           |      | $h = \int_{h_c}^{h_e} \tau$  
|           |      | $\lambda = \int_{\lambda_c}^{\lambda_e} \chi$  
|           |      | $\tau = \int_{\tau_c}^{\tau_e} \chi$  
|           |      | $V_h = \int_{V_{h,c}}^{V_{h,e}} \chi$  
|           |      | $V_t = \int_{V_{t,c}}^{V_{t,e}} \chi$  

Control

| 1 | $\chi_{GC} = \arctan\left(\sin(\tau - \tau) \cos(\lambda^*)\right)$  
|   | $(\sin(\lambda^*) \cos(\lambda) - \sin(\lambda) \cos(\lambda^*) \cos(\tau^* - \tau))$  
|   | $V_{e,c} = V_{c,climb}$  
|   | $V_{h,c} = (h^* - h) P_{h,climb}$  
|   | $\chi_c = \chi_{GC} - \arcsin(V_w/V_t \sin(\chi_w - \chi_g))$  

| 2 | $\chi_{GC} = \arctan\left(\sin(\tau - \tau) \cos(\lambda^*)\right)$  
|   | $(\sin(\lambda^*) \cos(\lambda) - \sin(\lambda) \cos(\lambda^*) \cos(\tau^* - \tau))$  
|   | $D = \sqrt{(\lambda^* - \lambda)^2 R^2 + (\tau^* \cos(\lambda^*) - \tau \cos(\lambda))^2 R^2}$  
|   | $V_{e,c} = D/(t^* - t)$  
|   | $V_{h,c} = 0$  
|   | $\chi_c = \chi_{GC} - \arcsin(V_w/V_t \sin(\chi_w - \chi_g))$  

| 3 | $\chi_{GC} = \arctan\left(\sin(\tau - \tau) \cos(\lambda^*)\right)$  
|   | $(\sin(\lambda^*) \cos(\lambda) - \sin(\lambda) \cos(\lambda^*) \cos(\tau^* - \tau))$  
|   | $V_{e,c} = V_{c,descent}$  
|   | $V_{h,c} = (h^* - h) P_{h,descent}$  
|   | $\chi_c = \chi_{GC} - \arcsin(V_w/V_t \sin(\chi_w - \chi_g))$  

utes or less) over the total number of samples (which includes those for which no conflict occurred within 20 minutes, returned as $\infty$ by Algorithm 1).
For aircraft 4 and aircraft 5, the probability of conflict is pretty low, and does not occur where the intended flight paths are (see Fig. 5, where aircraft 5 reaches the point where the paths cross before aircraft 4 does). The estimated probability of conflict rises to 100% around 12 minutes, but this is for a conflict that is to occur at \( t = 30 \) minutes where the intended flight paths to the next waypoints cross.

Here, the hybrid dynamics do not have much effect on the predictions, since all the conflicts happen to arise when in the cruise mode. If a conflict is to happen on the border of climb and cruise, for example, then the resulting distribution may become bimodal, for example if just on the border of the vertical separation being violated.

In actual operations, if a conflict is predicted within the 20 minute prediction horizon, a resolving maneuver would be issued by ATC in order to prevent the conflict. In current practice, only deterministic trajectory prediction is available, and errors are added on afterwards on top of these predic-
tions in an ad hoc manner to obtain probabilistic predictions of conflicts (Erzberger et al., 1997). On the other hand, in our approach, uncertainties are added at the time of prediction, and are propagated throughout the prediction procedure in order to obtain conflict probabilities. The model-based prognostics approach is a more rigorous method compared to the currently used ad hoc methods for treatment of uncertainty, and allows for more robust and efficient operations and more informed decision-making.

6. Conclusions

We presented an extension of the model-based prognosis framework to hybrid systems. The presence of mixed discrete and continuous dynamics presents significant challenges to the estimation and prediction problems. However, the overall model-based prognostics approach is essentially the same, as there is an estimation step followed by a prediction step. The only difference is the multi-modal, hybrid models used, which requires more complex estimation and prediction algorithms.

The approach was demonstrated on the problem of conflict prediction in the NAS. We showed how the framework can be applied, and discussed also how the model-based prognostics framework offers a more systematic and robust approach, especially in the context of handling uncertainty, than the current state-of-the-art in that domain. In fact, the framework presented here has a much broader applicability. For example, given models of unsafe weather systems, “conflicts” with unsafe weather regions can also be predicted, and these predictions can be used within decision-making algorithms. Other safety events, such as low fuel and high congestion, can also be predicted within this type of architecture.

While much work has been done on state estimation for hybrid systems, relatively little has been done on prediction and uncertainty propagation using these types of models. Monte Carlo sampling is a simple but inefficient solution to this problem, and future work will address more efficient prediction algorithms for hybrid systems, e.g., the extension of the unscented transform sampling method.

Acknowledgments

This work has been funded by the NASA SMART-NAS project in the Airspace Operations and Safety Program of the Aeronautics Mission Directorate and the Spanish MINECO DPI2013-45414-R grant.

References


evaluation tool. In 3rd USA/Europe ATM R&D Seminar.


**Biographies**

**Matthew Daigle** received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively. From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems. Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE.

**Indranil Roychoudhury** received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems. Dr. Roychoudhury is a Senior Member of the IEEE and a member of the Prognostics and Health Management Society.

**Anibal Bregon** received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis of complex physical systems.
A Systems Approach To PHM

Thomas D. Mooney, PE

SOAR Engineering LLC
Gloucester, MA, USA
tmooney@soarengineeringllc.com

ABSTRACT

The operational objectives of today’s complex, software intensive high technology platforms requires taking a systems approach to PHM. A systems approach to PHM includes the complete techno-social system involved in the equipment’s operations and maintenance - its full cycle of care. It takes a broad view of health based on wellness and is focused on optimizing the performance of all elements of the system. It provides sustaining value by enabling enterprise wide enhancements in capability. By providing timely and relevant health information for all elements in the equipment cycle of care, a well-designed PHM system can become major facilitator in maximizing the performance and resilience of all elements in the system. This systems approach has major implications for PHM system design, architecture selection, data collection, sensor selection, control system software integration, algorithm development and analytics. PHM systems that encompass the health of the full socio-technical system in which the equipment is developed, operates and supported will maximize the effectiveness of the enterprise.

1. INTRODUCTION

To meet the needs of today’s challenging, ever changing and often-unknowable environments, modern high technology systems have become increasingly complex software intensive integrated platforms. These high technology interconnected platforms are expensive to develop, procure, operate and maintain. These systems require enhanced support capabilities to meet their operational goals. While significant progress has been made in making high technology equipment more reliable, maintainable and supportable, it has not been enough. Despite all of these advances, many modern systems fall short of meeting their business objectives.

Advances in diagnostic capability have been pursued for many years but Prognostics and Health Management (PHM) has clearly emerged the winning alternative to the fly-fix-fly and time-based maintenance philosophies of the past. PHM systems monitor the health of the equipment (usually in real-time) and predict its future health so that interventions can be taken to prevent failure and reduce unscheduled maintenance, equipment downtime and secondary damage.

PHM itself is an evolving field. It has progressed from single-issue diagnostic and prognostic capability for troublesome components to health management of major systems. However, many traditional PHM systems have failed to stay current with the ever-changing needs of today’s complex platforms. PHM systems that focus only on the physical electromechanical equipment, that take a limited view of health based solely on failure prevention and view value in terms of cost avoidance and cost reduction are often unable to provide the kind of enterprise wide enhancement needed to support today’s modern systems.

While the systems themselves are becoming more complex, budget pressures and limited resources are driving the need to keep equipment operating longer with fewer the resources for maintenance and support. The PHM systems needed to support these systems must be value-based, flexible and organic to keep up with these needs.

A new approach to PHM is needed to address these shortfalls. This new approach is based on a broader definition of the system that includes its full cycle of care, a new health model that is focused on optimizing performance of all the system elements and a new value model geared to sustaining continued operational and business improvements.

Advances in descriptive and predictive analytics coupled with the technology from smart connected products provide valuable insights that can be applied to PHM. The internet-of-things (IoT) centers on smart flexible approaches to asset-based management, value added services and collaborative business models. The many-to-many interconnection
architectures have applicability that can and should be applied to future PHM system communication networks.

This paper will explore the reasons why a systems engineering approach to PHM that focuses on an inclusive systems model, a broader definition of health based on wellness, and an enterprise wide value model can lead to systems that:

- Contribute to enterprise wide value
- Improve operational efficiency
- Increase in value over the life of the equipment
- Provide a greater return on investment
- Enable real time data driven decisions
- Enhance user experience
- Optimize products and services
- Enable development of more resilient products

2. BACKGROUND

An illustration of the concepts developed in the paper is taken from a vending machine company’s experience in applying the IoT model to their operations. Their experience is not exceptional but it provides a valuable illustration of the concepts further explored in this paper.

Vending machines are not among the typical high technology industries typically discussed at PHM conferences. However, one creative company in this industry was determined to reduce their high machine servicing cost using the IoT model. What they did illustrates the systems, health and value concepts presented in this paper. PHM system developers can directly benefit by incorporating the lessons learned from the example set by this company.

The vending company’s machines were being serviced by company technicians who, because they had no way of knowing product usage, would bring a full complement of product to each machine on a scheduled basis to service them. Only when the servicers opened the machine did they learn which products were needed and in what quantity. Sometimes the machines needed only a few items. At other times the machines were out of stock on popular items – a condition representing a missed sales opportunity for those items. Occasionally, the servicers would discover that a machine was broken and required reactive maintenance. The servicers manually recorded product replenishment information and cash collected. While these records provided some financial information they provided little insight into customer buying behavior or usage patterns.

The vending machine company reached out to an IoT provider with a request for help. Their challenge was to reduce their servicing costs and they believed an IoT approach could help them.

Good analytics starts with questions. With the IoT provider they deconstructed their problem into the following series of questions:

- Where and when should we send our servicers and what stock should they bring?
- How can we be sure the cash collected from the machine reflects the actual cash collected by the machines?
- How can we know when an out-of-stock situation occurs so that we can respond sooner?
- How can we know when a machine breaks down?

To answer these questions the IoT and vending machine team developed the following solution. They added a few sensors to the machine and connected the machines to the cloud via a wireless cellular network. They put analytics in the cloud to process the sensor data. The sensors were simple and inexpensive. They counted product quantity, usage and cash received. They monitored the machine internal temperature (a proxy for machine health) and its external temperature (a proxy for the local environment). They developed analytic algorithms to provide clear answers to their questions and put the analytics on a cloud so they could serve many machines and provide the needed flexibility to enable the system to evolve.

The analytics enabled them to service the machines intelligently so they could provide the machines with the products they needed when they needed them – significantly reducing the cost of servicing the machines.

This would have been a success story on its own but it is not the end of the story. The benefits spread across the enterprise. The finance department used the data on cash collected and product usage to detect and prevent fraud. The company found that initially they were able to respond faster to out of stock situations and machine breakdowns. They later learned they could proactively avoid out of stock situations and machine breakdowns using the same sensor set coupled with new predictive analytics. Marketing gained additional insight into customer buying patterns and used the information to improve product mix and placement. Inventory management used predictive analytics to prepare for predictable events (holidays) and weather conditions. The system continued to grow and learn as various entities across the enterprise continued to discover new uses for the data to support their operations.

This company’s experience is not exceptional but it provides a valuable microcosm of lessons learned that can be applied to the high technology products typically discussed at PHM.
conferences. They started by asking questions involving issues that had the most significant impact on their business. They wanted to understand how they could improve the health of their business. They did this by taking a comprehensive view of their system. They did not focus on the equipment alone but they included the environment and the people and organizations involved in their enterprise. Their value perspective was focused on the business improvements that could be achieved with data from connected products. They were not simply focused on operational cost reduction. While operational cost reduction was initially a key consideration, they looked beyond that to see how the new capability could create value across their organization so they could optimize all elements of their business.

This company leveraged advances in low cost sensor technology, data storage, bandwidth and wireless networks with cloud based descriptive and predictive analytics. They connected their products to their operations and services. They gained visibility into product usage and environments. They became proactive to developing issues and improved the effectiveness of several diverse departments in the company. Information on product usage became a key factor in differentiating their product in the marketplace and enabled their design team to develop more customized and resilient products.

This company looked at health as a measure of wellness and looked at value based on enabling business improvements across their enterprise. They took a system perspective that included the full socio-technical environment. These approaches are directly applicable and even more urgently needed for PHM systems for complex high tech systems where the stakes are even higher.

3. Health

The existential focus of health management is health. When prognosis is involved it encompasses both current health and expected future health.

Health, in a PHM context, is generally thought of in terms of its loss - such as a failure or degraded level of performance. A failure is often defined as the loss of an ability to function normally or the inability to accomplish an intended purpose. Health in the PHM context is generally considered as a loss of capability at the physical equipment level.

An approach to health management based only on failure prevention does not recognize the greater value that can be achieved when the health system is designed to get the most capability out of the equipment given its current circumstances and condition. Most operational equipment is not failed, but it is also not operating at its best either. In a manufacturing environment at Ford Motor Company, Edie, Kekedjian and Jalluri (2014) found that 98% of the equipment it was monitoring was not failed, but was not completely healthy either. Economic interventions can often be taken to address equipment that is not operating at its best. This is particularly true when the degraded equipment is impacting the larger system operation. Oskin (2014) recently discussed that monitoring Key Performance Indicators (KPIs) in real time could be used to enhance manufacturing system performance. Rather than focus on failure detection or impending failure prediction, measurements of various key performance indicators are taken in real time to keep factory operations at their best. Resources are applied not just to fix and prevent “failures” but to obtain the best performance possible for the system at all times. Health information is used at all levels to improve system performance – not simply to avoid failure.

The medical field defines health very broadly. The medical field views health as a system property - a property of the whole organic system. The World Health Organization (1948) views health as not merely the absence of disease or an infirmity, but as a state of complete physical, mental and social well being. In this context the health practitioner’s goal is not simply to help patients when they are ill or becoming ill, but to keep their patients well enabling them to perform at their best throughout their life.

The definition of health management systems in the medical field complements this broad definition of health. The World Health Organization (2010) defines a well functioning health system in the medical context as one that can respond in a balanced way to a population’s needs and expectations by improving the health of individuals and groups. It defends the population against threats to health, protects the population against the financial consequences of ill health and makes it possible for collaboration among people participating in decisions affecting their health and the health system itself. Porter and Teisberg (2006) argue that a well-functioning, value-based health system should be built around improving value to all participants in the health care network.

Applying this medical health model to equipment health implies that PHM systems should not merely be limited to detecting and isolating actual and impending faults in specific items of equipment, but it should have a goal of improving the performance all of the elements in the health care network. The goal should be to optimize the performance of all elements in the system. This more holistic approach to health care has major implications for the development of PHM systems. With this perspective on health care systems, the PHM systems would include the
physical system along with its operations, support and management. It would integrate health management over the full cycle of care from equipment development, to monitoring, to fault prevention, diagnosing, predicting, repairing, restoring and improving. It would include both the equipment and the support environment in which it operates.

A well functioning health system should be capable of responding in a balanced way to the needs of individual equipment users and the overall asset base by:

- Providing visibility into the health status of the equipment and asset base to users at all levels
- Enabling the projection of future health status for alternative operating scenarios and environments
- Providing information needed to improve the health status of the equipment and asset base
- Providing information needed to improve the maintenance support environment
- Enabling innovation and specialization across the operational and support environments
- Supporting maintenance decisions
- Supporting logistics decisions
- Identifying new and emerging issues
- Responding to changes in usage, environment and configuration
- Supporting financial projections
- Supporting logistics and repair facility planning and coordination
- Making it possible for all users to participate in decisions affecting health
- Enabling evidence-based decisions for operations, maintenance and product resiliency improvements
- Integrating with existing business systems such as engineering, logistics, sourcing, quality, finance, analytics and others.

Achieving these goals requires the adoption of an integrated systems approach to PHM where all participants in the value chain (equipment providers or OEMs, maintenance providers, repair facilities, sourcing organizations, logistics agencies and regulators) can redefine their strategies, operating practices and organizational structures to unlock sustaining improvement in value delivered. Measuring and reporting of results at each level enabled by the PHM system is critical to reforming the entire system.

Measuring outcomes for every piece of equipment, for every service provider, and for every condition over the full cycle of health is critical for improving first time yield. Knowing the actual capability of each equipment item and subsystem and the capability of each provider in the equipment health cycle enables optimal use of resources and encourages resources to go to the best performers.

When PHM systems are focused only on the electromechanical equipment and view health and value based on a loss reduction models they often fail to live up to the needs of the platforms they are designed to serve. PHM systems that include the full cycle of care and are designed to provide enterprise wide business enhancement correct that problem.

4. SYSTEM

Since health is a system property and health systems are designed to maintain and support the health of the system, the PHM system should be inclusive of all elements in the equipment cycle of care.

PHM systems that are focused only at the electromechanical equipment level severely limits PHM system effectiveness. Focusing on the physical equipment alone does not address the impact of the integrated system environment, the software interactions, information integration, the people, organizations training and leadership or equipment usage and environmental factors.

4.1 Integrated System Environment

The PHM systems should be defined to include all elements in the equipment cycle of care, including the complete social and technical environment. It should acquire, integrate and share information to enhance the value for all participants in the enterprise.

Many traditional PHM systems do not address the complex interactions found in high-technology software intensive systems. Complex operations, high dynamic loads and cascading effects from other systems create anomalies in sub-systems that can be far removed from the original source of the anomaly. Similarly, software interactions and information fusion from disparate systems must be addressed. PHM must be framed in an integrated system environment.

PHM developers often assume that a support system is well defined and deterministic. They assume that PHM support systems are deterministic - similar to software routines and control systems with well-defined inputs, outputs and processes. PHM is anything but. PHM systems are highly complex and dynamic. Operational and support environments involve complex interactions among the
people, organizations and services. Due to situations that are contingent, vague, ill defined or involve varied, distorted and conflicting patterns, these interactions cannot be scripted in advance. Optimal PHM systems are not fully human or fully autonomous, but hybrids where machine intelligence augments human decision-making.

A comprehensive PHM system model is needed for modern high tech systems. The system must include the full socio-technical environment -- including hardware and software as well as all of the operational support and management elements needed to support the operational system. A comprehensive PHM system will accommodate the ever-increasing equipment complexity and integration, changes in usage, environments, equipment and failure modes. A comprehensive PHM system must support human decision-making and collaboration.

4.2 Software and System Interactions

Traditional PHM systems based on simple cause and effect models often do not address the complex software driven interactions so important in complex operational systems. In a discussion of safety in complex systems Levison (2005) reported that accidents in complex systems are often due to unforeseen interactions among components, including the effects of software, rather than individual component failures. Ignoring software driven interactions can result in serious shortfalls in the health management system.

4.3 Information Integration

When a PHM system definition is limited to a specific electromechanical item, that item is often unable to utilize information from other elements in the larger system or information external to the system - even when that additional information could aid in health assessment. Similarly, information from fundamentally different data sources may not be available to other sub-systems where it could aid in those systems’ PHM performance. Occasionally, when the external information is available, it may not be sufficiently coded, defined or synchronized to be useful to other sub-systems. External information, such as the weather and weather forecasts or future usage planning information is also often available but inaccessible to equipment centric PHM systems, even when use of that information could assist with health assessment and health prognosis.

4.4 People, Organization, Training and Leadership

Traditional PHM systems are often developed as isolated, single issue systems that do not effectively communicate with the many other elements in the cycle of care. These systems often fail to consider the social nature of the operating and support environment in which they reside. This includes the organizations, training and leadership essential to the health of those systems. People, organizations and service providers (such as OEMs, maintenance and repair shops and logistics providers) are often not considered in the PHM system design. Furthermore, assumptions about the people, organizations and service providers that rely on the PHM information may not reflect their actual performance or needs.

Actionable intelligence requires getting the appropriate information to individuals in various roles in multiple organizations. Different people, organizations and service providers require different information from a PHM system to do their work effectively.

Many traditional PHM systems have a single user assumption. They provide a single output of health and failure prediction assuming that it can meet the needs of all users. These systems often fail to recognize the diverse needs of people and organizations that use health information or the variety of different ways the information is used. PHM systems must also address differences in user experience and levels of training.

Health information is required to make decisions impacting operations, maintenance and management. Each participant has unique information needs and decision-making responsibilities. For example, an operator needs to know if the equipment is working or not and, if not, what they can do to restore its functionality. Operators, when they encounter problems beyond their ability to fix, inform their maintenance staff that the operation has been compromised and needs repair. The maintenance staff then requires additional information to troubleshoot the problem and assemble the tools, materials and instructions needed to conduct the repairs. Management, at multiple levels, requires additional information on the anomaly so they can assess the impact on the larger operational goals. They must evaluate their options for mitigating the adverse impacts caused by the problem.

Wheeler (2013) explained that during complex negotiations people “act in real time on often incomplete and ambiguous information even when the stakes are high.” The same is true for PHM. People will make tough decisions based on the information provided by the PHM system. Sometimes that information is incomplete but people need to understand what it says and have sufficient confidence in the information provided to make their decisions.

When PHM systems are developed without input from its users, the developers make decisions about what they think the individuals involved with operations and maintenance
need to know. The actual individuals using the system may see things differently. For example, the decision on whether to provide a forecast could be based on best case, worse case or most likely case. The use of the information is highly dependent on circumstances that often cannot be determined in advance.

PHM is a revolutionary system developed to support the needs of complex products and systems. In his book on the revolution in military affairs, Boot 2006 explained that a technology revolution requires far more than revolutionary technology. “It also requires revolutions in organizations, doctrine, training and personnel.” The revolution in support concept needed for modern systems must include the people, organizations, training and leadership at all levels in the enterprise. The PHM system must recognize that people and organizations at different levels in an organization ask different questions about the current and expected future health. The PHM system should connect with all the elements in the business enterprise, especially the traditional ILS functions that interact with equipment information in various ways.

4.5 Usage and Environment

The PHM system should actively monitor and report on the actual usage of the equipment and the environment in which the equipment operates. Deviation from specification usage and environment must be continually monitored, so that the effects of any change on the equipment performance can be evaluated quickly. Usage and exposure that exceeds the equipment design limits is problematic but even prolonged exposure to harsh environments or abusive usage within design limits can cause premature failure and aging.

4.6 Validation and Verification

It is often difficult to validate and verify PHM systems particularly early in their design phase. PHM systems are designed to monitor the causes of failures and to detect actual failures. The initial fault library is typically based on the results of a FMECA or similar reliability analysis. The physics-based causes of failures (corrosion, overstress, etc.) and the causal warrant linking the failure cause to the failure effect are often still unknown and unproven during development. Failures and failure causes can be simulated and tested on the bench, but system performance in the real world remains unknowable until real failure conditions occur in the actual techno-social environment in which the equipment operates.

When PHM is based on simple physical models without including the system elements, failure progression estimation can be problematic. Prognosis models typically follow assumed trends or reasoning logic that may not reflect real world conditions. An effective PHM prognosis algorithm should include all relevant information from the system environment to quantify failure progression. This is needed to inform the decision makers of the true situation so that they can make evidence based decisions and take the necessary remedial actions.

5. Value

A well functioning health system should be centered on creating value for all participants in the value chain. In the IoT model, the value of interconnected products lies in the operational and business improvements possible when the health system information is incorporated into the business operational system.

Modern high tech systems are highly reliable and downtime is minimal. Failures rarely occur, but when they do they are of great concern. However, it is often difficult to justify adding costly sensors, enclosures and software to prevent failures when the conditions they are designed to prevent are believed to be rare and unexpected. Historical failure conditions experienced in previous designs are generally mitigated in upgraded or new design.

Traditional PHM systems are often justified based on a cost-benefit analysis focusing primarily on cost avoidance and cost reduction. The cost savings is typically based on maintenance cost avoidance, downtime reduction and secondary damage avoidance. It is difficult to justify funding for PHM systems to offset the effects of failures that the equipment designers hoped to have mitigated by design. These value models often exclude user inconvenience, customer impact and business impact caused by a loss of system capability. Rarely do they consider the loss of capability due to operating sub-optimally for long periods of time.

Porter and Happelmann (2014) argued that the IoT value model is based on a more positive value model where the real value of interconnected products is the enterprise wide enhancements made possible by knowing and improving the condition and usage of all elements in the system.

The cost of the PHM system should be justified based on a value model based on enterprise wide enhancement made possible by the system. Once the basic infrastructure is designed and built, additional costs can be allocated to enhance functionality as funds are allocated and circumstances allow. New analytics can be developed for
previously unforeseen uses of the data. Additionally, analytic software can be refined and re-purposed for other applications.

6. PHM System Design

The basic design principle of PHM implied by this paper is to start with the questions that the business needs to have answered. Then identify the information needed by the various participants in the system to make value-based evidence driven decision. Continue the PHM system design using a broad definition of the social and technical of the system, including the full cycle of care in which the system resides.

The basic system architecture and network infrastructure can flow from this. It can be designed and built during development, but the system should be allowed to grow organically to meet the needs of multiple participants in the cycle of care. Developers should plan for new and advanced descriptive and predictive analytics to meet the ever-evolving business needs.

Product designers can use the information from the PHM system to improve the product design. They can make the products more customer focused and more resilient. Financial managers can make evidence based budget decisions and projections based on real data and accurate forecasts. The maintenance shops can forecast upcoming demand for their services and resources and the logistics managers can forecast future parts demands.

6.1 PHM System Design Guidelines

The following guidelines summarize the recommended steps a PHM system designer should consider in developing a PHM solution based on the recommendations presented in this paper.

- Identify the questions with the most significant enterprise wide impact that the PHM system seeks to answer. Typically this involves assessing the present and future health of the equipment and service providers and assessing the performance and usage of all system elements.
- Identify the complete socio-technical system in which the equipment is operated, maintained and supported. Include all potential interactions (hardware and software) with other systems. Include components beyond the equipment’s physical boundaries that may impact the equipment. Include all elements in the full cycle of care. This should include elements of fault prevention, diagnostics, maintenance, repair and reuse.
- Identify the needs of the people and organizations across the enterprise involved in operations, maintenance and support of the equipment.
- Actively monitor the operating environment, product usage and failure modes. These are variables. Do not assume the equipment will be operated in accordance with the original equipment specifications.
- Select a PHM system architecture and network to maximize flexibility, learning and adaptation.
- Minimize on-equipment algorithms to only those needed for real time responses, such as operations and safety.
- Sensors are the prime source of basic information. Select them based on their value to the overall system.
- Exercise caution in leveraging or repurposing existing sensors, software and enclosures. They may not be suitable for PHM and may limit flexibility in the future.
- Maximize off-equipment (cloud based) PHM functionality to allow for growth and adaptation and enable data mining and analytics.
- In operation, provide timely and relevant health information to all participants in the value chain. Gather feedback - both good and bad.
- Utilize the results of the PHM system to make enterprise wide enhancements in all elements of the equipment cycle of care. Monitor progress and value created.
- Continue to improve the capability and performance of the product and the enterprise to establish a sustainable competitive advantage for your organization.

7. Conclusion

In their HBR paper on smart connected products, Porter and Happelmann (2014) claim that “smart, connected products offer exponentially expanding opportunities for new functionality, far greater reliability, much higher product utilization, and capabilities that cut across and transcend traditional product boundaries.” PHM can leverage the IoT example and take PHM for high technology system to the level of support needed for today’s modern equipment.

When properly developed and executed, the PHM system becomes a platform to build out an enterprise-based system with far reaching benefit to all participants in the value chain. A systems approach to PHM addresses the health of individual items, assets, operations and service providers. It is based on a broad definition of health based on wellness
and a value model that is based on enterprise wide improvements.

By providing timely and relevant information for all elements in the equipment cycle of care a well designed, the PHM system becomes a major facilitator in maximizing the performance and resilience of all elements in the business.

REFERENCES


BIOGRAPHY

Thomas D. Mooney, P.E., is founder, president and CEO of SOAR Engineering LLC, a PHM consulting company. At SOAR Engineering LLC he consults with industry and government agencies on the design and development of PHM systems. He has over 40 years experience in the aerospace industry. Prior to SOAR Engineering he was with GE Aircraft Engines for 37 years and Raytheon Company for 5 years. At GE Aircraft Engines he was a principal engineer in prognostics, health monitoring (PHM) for GE’s Advanced Technology Operation in Lynn MA and a founding member and later chairman of GE Aviation’s Design Board for Prognostics and Health Management (PHM). Mr. Mooney received a B.S. degree in Mechanical Engineering from the University of Massachusetts in 1971 and a M.S. in Engineering Management from Northeastern University in 1976. He received a Certificate in Aviation Safety from the University of Southern California. Mr. Mooney is a registered professional engineer in Massachusetts (1991). He is a member of the American Helicopter Society and the PHM Society. He also serves on the industry advisory panel for Tufts University, Tufts-Gordon Institute, a graduate program focused on developing leadership in engineering management and innovation.
Online Monitoring and Fault Diagnosis of Hybrid Systems Using Switched Dynamic Bayesian Networks

Gan Zhou¹, Gautam Biswas², Wenquan Feng³, Xiumei Guan⁴

¹,³,⁴School of Electronic and Information Engineering, Beihang University, Beijing, 100191, China
E-mail: gan.zhou@buaa.edu.cn
buafwq@buaa.edu.cn
guanxm@buaa.edu.cn

²Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37235, USA
E-mail: gautam.biswas@vanderbilt.edu

ABSTRACT

Modern real-world engineering systems typically have hybrid dynamic behaviors that can be modeled by continuous behaviors with discrete mode transitions. These complex systems present many significant challenges for online monitoring and diagnosis, including tracking system behavior, dealing with noisy measurements and disturbances, and diagnosing different types of faults. In this paper, we propose an integrated model-based diagnosis approach that extends the traditional Dynamic Bayesian Network-based particle filter approach for tracking continuous system dynamics. A novel mode diagnoser is presented that discriminates between residuals generated by inaccurate system tracking, discrete faults, and parametric faults. An extended quantitative fault isolation and identification scheme is combined with a qualitative fault isolation scheme to identify the abrupt parametric faults. We demonstrate the effectiveness of our approach by applying it to Reverse Osmosis (RO) subsystem of the Water Recovery System (WRS) developed at the NASA Johnson Space Center for long duration human missions.

1. INTRODUCTION

With the increasing complexity of modern engineering systems, there is a pressing need to guarantee their safety, reliability and efficient operation. Although these systems undergo rigorous testing and validation before deployment, degradation and faults in system components are inevitable due to their long and sustained operations. Therefore, online monitoring and diagnosis for these systems becomes particularly significant to avoid the catastrophic consequences of failures in the system.

Most of these systems are hybrid in nature, so accurate online monitoring of dynamic system behavior becomes a primary challenge. For hybrid systems, the interactions between discrete mode changes and continuous behavior evolution make system state estimation more complicated. Moreover, the uncertainty from modeling errors and sensor noise may degrade the estimation accuracy.

For fault isolation and identification, multiple types of faults need to be considered. Hybrid system faults can be classified into: (1) parametric faults and (2) discrete faults. Parametric faults cover partial failures and degradations in system components, and can be further characterized by different fault profiles: abrupt, incipient, and intermittent (Mosterman & Biswas, 1999). On the other hand, a discrete fault is a discrepancy between the actual and estimated hybrid system modes, and this generally occurs for discrete actuators associated with valves in hydraulic systems and relays in electrical systems. In this paper, our focus is on abrupt parametric faults and discrete faults.

Over these years, many researchers have proposed different methods for diagnosis of hybrid systems. An important class of approaches employs observers or filters based on the estimation of unknown variables. Hofbaur and Williams (2004) and Wang and Dearden (2009) predefine hybrid system behaviors into discrete nominal and fault modes. Extend Kalman Filters (EKF) and Particle Filters (PF) can be employed to track the system state. When discrete faults occur, the approach will converge to the fault mode as the system evolves. The work in (Narasimhan & Biswas, 2007) extended TRANSCEND (Mosterman & Biswas, 1999) from continuous system diagnosis to hybrid systems diagnosis. Soon after, Daigle et al. (2010a) incorporated discrete faults into the diagnosis framework, and demonstrated its effectiveness by applying it to spacecraft power distribution

Gan Zhou et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
systems (Daigle et al., 2010b). Another class of research approaches relies on analytical redundancy relations (ARRs), and the key idea is to eliminate unknown variables. Cocquempot et al. (2004) and Bayoudh et al. (2008) check the consistency of residuals to determine current system mode and detect and isolate faults. Arogeti et al. (2010) and Levy et al. (2014) introduce global ARRs (GARRs) into quantitative FDI to implement discrete mode tracking and identification.

In this paper, we propose an online monitoring and fault diagnosis framework for hybrid systems. We extend the observer based approach for continuous systems (Roychoudhury et al., 2008) to hybrid systems. For hybrid systems, a statistically significant non-zero residual can be the result of (1) inaccurate system behavior estimation, (2) discrete faults, and (3) parametric faults. It is well known that discrete faults can result in significant changes in system behavior. We propose a novel mode diagnoser algorithm to distinguish between these three situations when faults occur. Our approach first checks for inaccurate mode estimation and discrete faults, and once these are determined not to be presented, the fault isolation and identification (FII) module is invoked to analyze abrupt parametric faults. In this module, Qualitative fault isolation (Qual-FI) method by means of Hybrid TRANSCEND (Narasimhan & Biswas, 2007) and extended Dynamic Bayes Nets (DBN)-based PF method for Quantitative FII (Quant-FII) (Roychoudhury et al., 2008) are combined together to generate and refine possible parametric fault hypothesis and compute the fault magnitude.

This paper is organized as follows: A description of the case study is given in Section 2. Different models used for our diagnosis framework are shown in Section 3. Section 4 provides an overview of our common diagnosis framework for hybrid systems, and then discusses the details of three main function modules: system monitoring, mode diagnoser and parametric FII. Experimental results are presented in Section 5. Finally, Section 6 contains the discussion and conclusions.

2. DESCRIPTION OF THE APPLICATION

The Advanced Water Recovery System (AWRS), which was designed and built for long-duration manned missions at the NASA Johnson Space Center (JSC) (Bonasso et al., 2003), is a key unit of Advanced Life Support Systems (ALS). The AWRS is composed of four main components: Biological Waste Processor (BWP), Reverse Osmosis (RO) Subsystem, Air Evaporation Subsystem (AES), and Post Processing Subsystem (PPS). In this paper, we focus on the diagnosis of faults in the RO subsystem (see Figure 1), a hybrid dynamic system with discrete and continuous dynamics. More specifically, the RO subsystem cycles through three modes of operation, that is determined by a four-way multi-position valve. The feed pump is on in all modes, and it pulls the effluent from the BWP to provide a steady stream of input flow for the system. In the primary mode (M1), the input liquid mixes with the effluent from primary loop, as it flows through a tubular reservoir. The recirculation pump boosts the liquid pressure as it flows into the membrane. Clean water leaves the system on the other side of the membrane, and the remaining water with concentrated brine flows back through the return loop to be re-circulated.

As the time elapses, particulate matter collects on the membrane, increasing its resistance to the flow. Therefore, the outflow from the system decreases. After a time interval, the system transitions to the secondary mode (M2), where the length of the feedback loop is reduced, which causes the flow rate to increase, and the outflow of clean water to increase correspondingly. However, this also results in increasing brine concentration in the remaining effluent in the loop, and the collection of impurities in the membrane also increases at a faster rate. After some time, the concentration reaches a level where very little water can be pushed through membrane, and a transition is executed to the purge mode (PM). In this mode, the recirculation pump is turned off, and the valve position is set to allow the effluent with concentrated brine to flow into the AES, where the remaining water is recovered by an evaporation method. The above three modes constitute a whole operating cycle1, and then the system goes back to primary mode for the second period.

![Figure 1. Process diagram for RO subsystem](image)

As can be seen from Figure 1, five sensors are used to collect observation for our monitoring and diagnosis experiments: 1) the outflow from the feed pump, \( F_{in} \); 2) the pressure immediately after the recirculation pump, \( P_{pump} \); 3) the pressure of the permeate at the membrane, \( P_{memb} \); 4) & 5) the pressure and conductivity of the liquid in the return path of recirculation loop \( P_{back} \) and \( P_K \).

---

1 In reality, there is a fourth mode where clean water is pumped in the reverse direction through the membrane to remove particulate matter from its surface.
3. MODELING APPROACH
In this section, we formalize the basic definitions, concepts and notation of modeling approach for our fault diagnosis approach.

3.1. Hybrid Bond Graphs
Bond graphs (BGs) are a domain-independent topological-modeling language that captures energy-based interactions among the processes that make up a physical system (Karnopp et al., 2012). The nodes in BGs represent components of dynamic systems including energy storage elements (capacities, C and inertias, I), energy dissipation elements (resistors, R), energy sources (effort source, Se and flow source, Sf) and energy transformation elements (gyrators, GY and transformers, TF). Bonds show the energy exchange paths between bond graph elements, drawn as half arrows. Two junctions (0- and 1-junctions) model the equivalent of series and parallel topologies respectively.

Hybrid bond graphs (HBGs) extend BGs by introducing switched junctions to enable discrete changes in the system configuration (Mosterman and Biswas, 1998; Roychoudhury et al., 2011). The switched junctions may be dynamically switched on and off as system behavior evolves. When a switched junction is on, it behaves as normal junction. When off, 1-junction and 0-junction behave as source of zero flow and zero effort respectively. The dynamic behavior of a switched junction is implemented by a finite state machine (FSM) control specification (CSPEC). State transitions are defined by external control signals and internal variables crossing pre-specified threshold values. The output of a CSPEC determines whether the junction is on or off (Daigle et al., 2008). As a running example, the HBG for the RO subsystem is shown in Figure 2.

According to the definition of HBGs, a system mode is described by a unique state combination of all the switched junctions. Two typical switched junctions are common in hybrid physical systems: (1) A switched junction associated with a particular discrete component that can operate in multiple states. (e.g., a valve or a relay in hydraulic and electrical systems, respectively); and (2) the combination of the state of multiple switched junctions may define the mode of a system component. If this is the case, the faulty system component also should be defined by the combination of these junctions.

Previously, (Daigle et al., 2010a; b) have considered the first situation. This case is relatively simple. The discrete faults can be easily introduced into corresponding CSPEC definitions as unobservable fault events. The Qual-FI scheme yields fault signatures that can be used to generate and refine the set of possible discrete faults and parametric faults, and employs temporal causal graph (TCG) with extended fault signatures for diagnosis (Mosterman & Biswas, 1999; Narasimhan & Biswas, 2007). However, when the second situation occurs, since the faulty system component involves multiple switched junctions, the traditional method for defining and processing discrete faults cannot be applied.

![Figure 2. Hybrid bond graph of the RO subsystem](image_url)
For example, in the RO subsystem, described earlier in Section 2, a multi-position valve controls system configuration, and mode changes in this valve correspond to multiple coupled switched junctions. In primary mode, switched junctions l1, l2, l3, l4, and l5 are ON, and l6, l7, l8, l9, and l10 are OFF. In secondary mode, l1 and l2 become ON, and l3, l4, and l5 are OFF. Besides three nominal modes, discrete faults in RO subsystem can be classified into two categories: 1) stuck-at faults, i.e., though a valve may be commanded to transition into a new mode, it remains stuck at the current mode, and 2) uncommanded transitions, i.e., a valve may unexpectedly transition to a new mode without a trigger signal. Figure 3 shows the mode transition automaton for RO subsystem. As can be seen from this figure, the nominal mode transition is autonomous when the corresponding transition constraint function is satisfied. \( \tau_{ij} \) denotes the unobservable fault transition from nominal mode \( i \) to fault mode \( j \). If \( i \) is equal to \( j \), it is a stuck-at fault transition. Otherwise, the uncommanded transition occurs.

![Mode automaton for RO subsystem](image)

Figure 3. The mode automaton for RO subsystem

### 3.2. Dynamic Bayesian Networks (DBNs)

A DBN is a two-slice temporal Bayes net for modeling dynamic systems. It captures the uncertainty in estimating the values of the system variables any time slice \( t \) given the values at time slice \( t-1 \) (Murphy, 2002). A dynamic system consists of four different types of stochastic variables: the continuous state variables \( X_t \), other hidden variables \( Z_t \), input variables \( U_t \), and measured variables \( Y_t \). The relations between the state variables from time step \( t \) to time step \( t+1 \) can be generated using the discrete time form of the state space model. The hidden and output variable values at any time step \( t \) are related to the state variables at the same time step.

Since the TCG structure describes the causal and temporal constraints between system variables, the DBN can be easily constructed from TCG. In our work, we obtain the TCG systematically from the HBG model of RO subsystem for a given mode, and then generate the corresponding DBN. For lack of space, the DBN model construction process is not shown in this paper, but can be found in (Roychoudhury, et al., 2008, 2009).

![Nominal DBN for RO subsystem](image)

Figure 4. Nominal DBN for RO subsystem

### Table 1. The details of system variables for RO subsystem

<table>
<thead>
<tr>
<th>System Variables</th>
<th>Name</th>
<th>Description</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Variables</td>
<td>( f_{31} )</td>
<td>The outflow at the feed pump</td>
<td>All</td>
</tr>
<tr>
<td>( f_{28} )</td>
<td>The outflow at the recirculation pump</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>( e_{61} )</td>
<td>The pressure of the liquid in the tubular reservoir</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>( e_{13} )</td>
<td>The pressure of the liquid in the membrane</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>( e_{32} )</td>
<td>The pressure of the liquid in the pipe carrying brine</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>( e_{39} )</td>
<td>The conductivity of the liquid in the return path</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>Input Variables</td>
<td>( e_{1} )</td>
<td>The effort source value for the feed pump</td>
<td>All</td>
</tr>
<tr>
<td>( e_{25} )</td>
<td>The effort source value for the recirculation pump</td>
<td>M1/ M2</td>
<td></td>
</tr>
</tbody>
</table>

The nominal DBN model for the RO subsystem in primary mode is shown in Figure 4. The nodes at time step \( t \) include the following stochastic variables: \( X_t = \{ f_{31}, f_{28}, e_{61}, e_{13}, e_{32}, e_{39} \} \), and \( Y_t = \{ F_{f31}, P_{membrane}, P_{back}, P_{pump}, P_{KPM} \} \) respectively.
Input variables $U_i = \{e_{i1}, e_{i2}\}$ are deterministic, and no hidden variables $Z_i = \{\emptyset\}$ appear in the RO subsystem. Table 1 describes state and input variables in more details (The measured variables are shown in Section 2). The particular mode of operation that the variables may/may not be active is also indicated.

For each abrupt parametric fault candidate, we invoke a separate fault model by incorporating an extra state variable that denotes the fault parameter into the nominal DBN model. The fault DBN model for the RO subsystem in primary mode with an abrupt fault in $R_{fp}$ is shown in Figure 5. We replace every occurrence of constant $R_{fp}$ in the nominal model with the stochastic variables $R_{fp}(t)$, and add the corresponding links. Table 2 lists potential parametric faults in the RO subsystem.

4. Fault Diagnosis Method of Hybrid Systems

The computational architecture of our online monitoring and fault diagnosis methodology for isolation and identification of discrete mode and continuous parameter faults in hybrid systems consists of four main parts: (1) system monitoring module, (2) mode diagnoser module, (3) parametric FII module, and (4) online model transformations module, as shown in Figure 6.

Given system input $u(t)$ and system output $y(t)$ in time series form, the dynamic behavior evolution of the system is monitored by a hybrid observer using a switched DBN-based PF approach. The fault detection scheme continually compares observed output $y(t)$ and estimated output $\hat{y}(t)$ computed by the observer. Any statistically significant difference triggers the mode diagnoser module. The objective of mode diagnoser is to establish whether the discrepancies observed in system behavior tracking is attributed to a discrete fault or a parameter fault. If the possible fault is identified to be a parameter fault, the parametric FII module is activated to generate and refine parametric fault candidates, and compute fault magnitude using an estimation method.

In the online model transformation module, the HBG model is used to generate the BG and TCG corresponding to the current estimated mode of operation. Our online model transformation module provides the nominal DBN model constructed from the TCG to the hybrid observer. Similarly, the fault DBN model for Quant-FII scheme can also be derived by incorporating the fault parameter as a stochastic variable into the DBN. In addition, TCG is adopted by Qual-FI scheme to complete qualitative fault reasoning. The whole process of online model transformation should be rerun, if the HBG executes a mode change. In the following sections, we describe these schemes of our diagnosis methodology in greater detail.

### Table 1: Nominal RO Subsystem

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{fp}$</td>
<td>Increase in friction in the feed pump</td>
</tr>
<tr>
<td>$R_{rp}$</td>
<td>Increase in friction in the recirculation pump</td>
</tr>
<tr>
<td>$I_{fp}$</td>
<td>Decrease in inertia of the feed pump</td>
</tr>
<tr>
<td>$I_{rp}$</td>
<td>Decrease in inertia of the recirculation pump</td>
</tr>
</tbody>
</table>

### Table 2: Potential parametric faults of RO subsystem

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{fp}$</td>
<td>Decrease in the feed pump efficiency</td>
</tr>
<tr>
<td>$GY$</td>
<td>Decrease in the recirculation pump efficiency</td>
</tr>
<tr>
<td>$C_{res}$</td>
<td>Buildup of impurities in the tubular resistance</td>
</tr>
<tr>
<td>$C_{memb}$</td>
<td>Buildup of impurities in the membrane</td>
</tr>
<tr>
<td>$R_{pipe}$</td>
<td>Partial blockage in pipe carrying water to the membrane</td>
</tr>
<tr>
<td>$R_{brine}$</td>
<td>Partial blockage in the pipe carrying brine</td>
</tr>
<tr>
<td>$R_{drain}$</td>
<td>Partial blockage in the pipe to the AES</td>
</tr>
</tbody>
</table>

Figure 5. Fault DBN for RO subsystem in primary mode with abrupt fault in $R_{fp}$
parametric fault isolation and identification of signals assuming large, abrupt changes is also the estimated value of the system behavior. For hybrid systems, the deviated residual detected by statistically method is not necessarily an indication of a fault. A typical reason is inaccurate mode estimation resulting from autonomous mode changes under nominal system operation. Take the RO subsystem for example. Figure 7 shows the system monitoring results for $P_{\text{back}}$ and $P_{\text{mb}}$ signals assuming 40db additive noise. The top two figures describe the comparison of estimated and measurement variables, while the bottom figures represent the corresponding residuals. We can easily see that the abnormal residuals will be generated from secondary to purge mode in every cycle, because of the inaccurate autonomous mode transition and the large, abrupt changes in $P_{\text{back}}$ and $P_{\text{mb}}$ in short time. Apparently this erroneous estimation will affect the FDI process in a significant way.

A feasible solution is to trigger the traditional Qual-FI scheme and refine the fault hypothesis set until all of them are refuted. However, pruning parametric faults in Qual-FI scheme is complicated, because autonomous mode transitions need to be considered when a mismatch occurs between predicted fault signatures and observed in measured residuals (Narasimhan & Biswas, 2007). As a result, this approach will make the overall computations intractable.
In addition, hybrid systems diagnosis comprises of both parametric faults and discrete faults. Unlike parametric faults, all discrete faults introduce jumps in system variable values (Dressler & Struss, 1996), so we first analyze discrete faults during fault isolation and identification process. Moreover, processing discrete faults first postpones triggering of the Qual-FI scheme, which reduces the computational complexity of the diagnoser.

On the basis of the above considerations, this paper introduces the concept of mode diagnoser. The key idea is to find the possible nominal system modes and discrete faults using a quick roll-back process, and refine these mode hypotheses using a fast roll-forward process. The true discrete system mode will survive if inaccurate mode estimation or discrete faults occur. In contrast, all hypothesized discrete fault modes are refuted denotes the parametric faults.

The mode diagnoser algorithm using the combination of roll-back and roll-forward process is summarized as Algorithm 1. Once a statistically significant non-zero residual is determined, the mode diagnoser is invoked immediately. Considering that a discrete fault or a nominal mode transition may have occurred in a mode prior to the current mode hypothesized by the hybrid observer, the mode diagnoser algorithm reasons backwards, envisions past system modes, and hypothesizes that either a discrete fault mode or a nominal mode transition may have occurred in one of the past modes. After that, the roll-forward process applied for the refinement of discrete mode hypotheses is activated. Since the inaccurate mode estimation resulting in significant non-zero residuals is inevitable, we typically assume that the measurements estimated using real discrete mode will converge to the observed measurements within $s$ time steps from the possible fault occurrence time $t_f$. Ideally, after a predefined finite number of time steps, only the measurement estimated by correct discrete mode should converge to the observation.

### 4.3. Parametric Fault Isolation and Identification

Once the mode diagnoser rejects all the possible discrete mode faults, the parametric FII module combining Qual-FI scheme by means of Hybrid TRANSCEND methodology in (Narasimhan & Biswas, 2007) and extended Quant-FII scheme is activated to generate and refine parametric fault hypotheses, and calculate the fault magnitude.

#### 4.3.1. Qualitative Fault Isolation

Our qualitative fault isolation scheme is based on qualitative fault signatures describing the transient response of the dynamic system to abrupt parametric faults (Biswas, et al., 2003; Narasimhan & Biswas, 2007). A qualitative fault signature is expressed as the qualitative value of zeroth, first, and higher order time-derivative on a measurement residual.

![Figure 7. The system monitoring results using 40db noise](image-url)
The zeroth symbols are labeled as normal (0), above normal (+) and below normal (-). Similarly, derivatives are denoted as no change (0), increase (+) and decrease (-) from the nominal behavior, respectively. Ambiguity in a signature is represented by an *. Since the first change and slope value provide all the discriminatory evidence for qualitative fault isolation in dynamic system (Manders et al., 2000), we condense higher order signatures to magnitude change symbol and the first nonzero derivative change. Therefore, the formal definition of qualitative fault signature can be given as follow:

Definition 1 (Qualitative Fault Signature): Given a fault f, a measurement m and current system mode q, the qualitative fault signature is denoted as a tuple $QFS(f, m, q) = (s_o, s_1)$, where $s_o, s_1 \in \{0, +, -, *\}$ represents the magnitude and slope symbol respectively (Manders et al., 2000).

Figure 8 shows the schematic for qualitative fault isolation. The symbol generator models the residual deviation at fault detection time as zeroth fault signature. Hybrid hypothesis generation reasons backward to find all possible modes that fault may occur in prior to the current mode, and applies the back-propagation algorithm to generate fault hypotheses that are consistent with the observed deviation in individual mode. Starting with each fault hypothesis, hybrid signature generation employs a forward-propagation algorithm to yield fault signatures in corresponding discrete mode for abrupt fault hypothesis.

4.3.2. Quantitative Fault Isolation and Identification

The Quant-FI scheme is activated when Qual-FI scheme satisfies any of the following conditions: the fault candidates are refined to a pre-specified number k, Qual-FI scheme cannot prune the remaining fault candidates further, or a predefined time steps l have passed. We restrict the length of Qual-FI scheme as a predefined value, and assume that no autonomous change occurs during this period (Biswas, et al., 2003; Roychoudhury, et al., 2009).

We construct multiple faulty DBN model to track the system behavior from the possible fault occurrence time $t_{f_r}$. A separate PF algorithm is applied to estimate all stochastic variables including possible fault candidates. Ideally, only the estimated variables generated by true fault DBN model will gradually converge to the observed measurements after a finite number of time steps $l_f$. Therefore, a Z-test scheme is invoked to determine the statistically significant deviation between estimation and measurement from time step $l_f = t_{f_r} + l_f$. Moreover, since all the DBN models run in parallel with Qual-FI scheme, our fault isolation and identification scheme drops a fault candidate if Qual-FI scheme prunes the fault candidates or the significant difference in a fault DBN model is determined by Quant-FI scheme. Finally, a post processing step is required to calculate the bias term for true abrupt fault.

5. Experimental Results

To demonstrate the efficiency and accuracy of our comprehensive online monitoring and fault diagnosis framework for hybrid systems, we have conducted a set of simulation experiments on the RO subsystem (See Figure 1) for several fault scenarios. System behavior is simulated using Matlab Simulink for two full cycles of operation, which result in a simulation length of 32000 seconds. Zero mean and 40db Gaussian white noise was added to the measurements. In this paper, we illustrate two fault scenarios in more detail.

The first scenario comprises a stuck switch fault. The four-way multi-position valve should be transitioned to secondary mode at 27240s in the second operating cycle but remains stuck at the primary mode. Figure 9 shows the comparison of actual and estimated values of a set of measured variables for this fault scenario. First, due to the residuals generated by inaccurate mode estimation, mode diagnoser is triggered in the secondary mode of the first operating cycle at 16048s. Three different mode candidates: purge mode for nominal mode transition and the ‘stuck at’ discrete fault in the primary mode (S_M1) and purge mode (S_PM) are generated. At 16099s, the discrete fault ‘S_M1’ becomes inconsistent with the measurements and is pruned. In the remaining mode candidates, since the discrete fault ‘S_PM’ shows the same continuous behavior as nominal purge mode, these two modes cannot be distinguished if no
discrete mode transition occurs. In our work, we preserve
the nominal purge mode as the current real mode. (Even
though the discrete fault ‘S_PM’ is the true one, it could be
detected again shortly after the next mode transition). This
refined process is presented in Table 3. Soon after, fault
detection signals another deviation at 28938s. The mode
diagnoser rolls back to find the two possible fault
occurrence modes: primary mode and secondary mode, and
generates the possible mode transitions shown in Table 4.
As more observations are obtained, mode diagnoser prunes
this set of mode candidates. Finally, the discrete fault S_M1
is correctly identified as the fault.

![Figure 9. Estimated results if a discrete fault stuck at
primary mode](image)

Table 3. The refine process of nominal mode transition in
first operating cycle

<table>
<thead>
<tr>
<th>Mode Candidate</th>
<th>Previous Mode</th>
<th>Transition Occurrence Time</th>
<th>Refine Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM</td>
<td>M2</td>
<td>16048</td>
<td>Y</td>
</tr>
<tr>
<td>S_M1</td>
<td>M2</td>
<td>16048</td>
<td>16099</td>
</tr>
<tr>
<td>S_PM</td>
<td>M2</td>
<td>16048</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 4. The refine process of discrete fault valve stuck at
27240s

<table>
<thead>
<tr>
<th>Mode Candidate</th>
<th>Previous Mode</th>
<th>Transition Occurrence Time</th>
<th>Refine Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>M1</td>
<td>27244</td>
<td>28938</td>
</tr>
<tr>
<td>S_M1</td>
<td>M1</td>
<td>27244</td>
<td>27295</td>
</tr>
<tr>
<td>S_M3</td>
<td>M1</td>
<td>27244</td>
<td>27295</td>
</tr>
<tr>
<td>PM</td>
<td>M2</td>
<td>28938</td>
<td>28990</td>
</tr>
<tr>
<td>S_M1</td>
<td>M2</td>
<td>28938</td>
<td>29179</td>
</tr>
<tr>
<td>S_M3</td>
<td>M2</td>
<td>28938</td>
<td>28990</td>
</tr>
</tbody>
</table>

In the second fault scenario, a decrease in the recirculation
pump efficiency is introduced in gyrator GY as an abrupt
parametric fault with magnitude 10% at 20000s. Table 5
shows the fault diagnosis process for this scenario. The fault
detection scheme detects a significant negative deviation for
dump pressure $P_{pump}$ with 4 seconds delay. The mode
diagnoser module is triggered immediately to generate
possible mode candidates: M2, S_M2 and S_PM. At 20057s,
al the mode candidates are dropped, and it is determined
that the fault is a parametric fault. As a result, the Qual-FI
scheme is activated to generate the initial fault candidate set,
$F = \{GY^{-a}, R_{p}^{-a}, I_{fp}^{-a}, C_{res}^{-a}\}$. The next observed change, the
pressure at membrane $P_{memb}$ also shows a negative deviation
at 20096s. $I_{p}^{-a}$ and $C_{res}^{-a}$ conflicts with this observation.

Possible autonomous mode change is executed to create a
new trajectory for these three fault candidates (For lack of
space, we do not show this trajectory in detail). After that,
$P_{back}$ and $F_{fp}$ decrease successively, but these observations
cannot eliminate the remaining candidates set
$F = \{GY^{-a}, R_{p}^{-a}\}$. Further refinement in the fault can only
be done by Quant-FII scheme. We predefine 1000 as the
number of samples in Quant FII scheme. The fault DBN
model using $R_{p}^{-a}$ is eliminated at 20305s, and $GY^{-a}$
consistent with the observed measurements. The estimation
of fault models $GY^{-a}$ is presented in Figure 10. Therefore,
$GY^{-a}$ is isolated as the true fault. The PF correctly identifies
the fault magnitude to be about 10.1047% step decrease in
GY (See Figure 11), while the true fault magnitude is 10%.

Table 5. The diagnosis process for abrupt parametric fault

<table>
<thead>
<tr>
<th>FDI</th>
<th>Time</th>
<th>Symbolic</th>
<th>Candidate Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode Diagnoser</td>
<td>20004</td>
<td>M2,S_M2,S_PM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20056</td>
<td>M2,S_M2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20057</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Qual-FI</td>
<td>20004</td>
<td>$P_{pump} \times (-\cdot)$</td>
<td>$GY^{-a}, R_{p}^{-a}, I_{fp}^{-a}, C_{res}^{-a}$</td>
</tr>
<tr>
<td></td>
<td>20096</td>
<td>$P_{memb} \times (-\cdot)$</td>
<td>$GY^{-a}, R_{p}^{-a}$</td>
</tr>
<tr>
<td></td>
<td>20114</td>
<td>$P_{back} \times (-\cdot)$</td>
<td>$GY^{-a}, R_{p}^{-a}$</td>
</tr>
<tr>
<td></td>
<td>20308</td>
<td>$F_{fp} \times (-\cdot)$</td>
<td>$GY^{-a}, R_{p}^{-a}$</td>
</tr>
<tr>
<td>Quant-FI</td>
<td>20305</td>
<td>$GY^{-a}$</td>
<td></td>
</tr>
<tr>
<td>Parameter Estimation</td>
<td>Fault magnitude:</td>
<td>-0.101047</td>
<td></td>
</tr>
</tbody>
</table>
look-ahead technique to alleviate the problem of sample impoverishment, where there are not enough particles that can transition into a faulty mode with very low likelihood, they only track partial fault candidates with high probability. In contrast, we avoid sample impoverishment problem by constructing a separate fault model for all the fault candidates generated and refined by Qual-FI scheme, so all the possible fault candidates are considered in this paper. Second, a novel mode diagnoser is introduced to indentify real discrete system mode. Three different cases resulting in significantly residuals are taken into account. Our mode diagnoser analyzes inaccurate mode estimation and discrete faults prior to parametric faults, and if all the discrete mode candidates are rejected, parametric FII module will be triggered to isolate parametric faults. This strategy considers false-alarm situation resulted from inaccurate system tracking, and improves the computational efficiency of our previous work. Third, our comprehensive online monitoring and diagnosis methodology of hybrid system handles multiple forms of faults including discrete faults and abrupt parametric faults, and has been demonstrated its effectiveness by applying to a real-world system: RO subsystem of an ALS.

In future work, since the centralized model-based diagnosis scheme has more computational complexity and memory requirements and the actual systems are often large and complex, we intend to employ distributed diagnosis strategy to relieve the computational burden of our diagnosis methodology. In addition, extending our framework to deal with multiple faults to adapt real systems is also another research direction.

**ACKNOWLEDGEMENT**

This research was supported by China Scholarship Council under contract number 201306020068. The work was performed in Prof. Biswas’ lab at the Institute for Software Integrated Systems (ISIS), Vanderbilt University, USA.

**REFERENCES**


Biographies

Gan Zhou is currently a Ph.D. candidate at school of electronic and information engineering from Beihang University, Beijing, China. He has been a visiting Ph.D. researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, USA, from 2013 to 2014. His research interests include fault diagnosis of complex hybrid systems.

Gautam Biswas received the Ph.D. degree in computer science from Michigan State University, East Lansing. He is a professor of computer science and computer engineering in the department of Electrical Engineering and Computer Science, and a Senior Research Scientist at Institute for Software Integrated Systems at Vanderbilt University, Nashville, USA. His research interests include modeling and simulation of complex systems, and online distributed diagnostics and prognostics methods applied to these systems. His web page can be found at http://www.vuse.vanderbilt.edu/~biswas. He is a Fellow of the IEEE.

Wenquan Feng received Ph.D. degree in communication and information system from Beihang University, Beijing, China. He is a professor at the School of Electronic and Information Engineering, Beihang University. His current research interests include automated test and fault diagnosis for complex systems and satellite navigation.

Xiumei Guan is a researcher at school of electronic and information engineering from Beihang University, Beijing, China. Her current research interests include automated test for complex systems.
Characterization of Phase Space Topology Using Density: Application to Fault Diagnostics

Mohsen Samadani 1, Cedrick A. Kitio Kwuimy 2, and C. Nataraj 3

1, 2, 3 The Villanova Center for Analytics of Dynamic Systems (VCAD S)
Department of Mechanical Engineering, Villanova University, Villanova, PA, 19085, USA
msamadan@villanova.edu
cedrick.kwuimy@villanova.edu
nataraj@villanova.edu

ABSTRACT

Almost all engineered systems are nonlinear and show nonlinear phenomena that can only be predicted by nonlinear models. However, the application of model-based approaches for diagnostics has been constrained mostly to linearized or simplified models. This paper introduces a fundamental approach for characterization of nonlinear response of systems based on the topology of the phase space trajectory. The method uses the density distribution of the system states to quantify this topology and extracts features that can be used for system diagnostics. The proposed method has been employed to diagnose a multi degree of freedom system with various simultaneous defects.

1. INTRODUCTION

The ability to determine the state of the system and predict failures would greatly increase the safety and productivity of systems. The predictive maintenance, health monitoring and diagnostics methods have become the focus of many research projects in recent years (Rezvanizaniani, Dempsey, & Lee, 2014; Fekrmandi, 2015). Generally speaking, a defective system would have a different dynamical response from a healthy system. From the mathematical point of view, these dynamical changes can be caused by either alternations in the values of the system parameters or transformation of the structure of the model, which we would call parametric and structural defects, respectively. The development of a crack in a beam is an example of parametric defects which will result in a change in the stiffness of the beam. In contrast, structural defects cause change the structure of the mathematical model. A broken capacitor in an electrical circuit drops the first order derivative in the model and can be considered as a structural defect. Given the mathematical model of the system along its parameter values, one can easily obtain the response of the system using numerical integration techniques. In contrast, the diagnostics problem would be the inverse problem, where we have the dynamical response of the system and we want to identify and quantify changes in the mathematical model. The solution to this problem however, is not always a straight-forward task. In practice, all engineering systems are nonlinear and exhibit nonlinear phenomena that can only be predicted by nonlinear models. This includes periodic, multi-periodic, quasi-periodic and chaotic behaviour, limit cycles, bifurcations of the equilibrium points, etc. Many studies have reported the emergence of these complex nonlinear phenomena in machinery originating from defects or even due to their nonlinear nature in healthy conditions (Sankaravelu, Noah, & Burger, 1994; Mevel & Guyader, 1993; Kappaganthu & Nataraj, 2011). The prevailing estimation methods which are mostly based on optimization algorithms show poor performance coping with such complexities. In many cases, the models are linearized to simplify the estimation algorithms show poor performance coping with such complexities. In many cases, the models are linearized to simplify the estimation algorithms show poor performance coping with such complexities.

A phase space is a space in which all states of a system are represented and a phase portrait is a visual representation of the trajectory of this system. For the two-dimensional case, the phase space will turn into a phase plane. The phase space trajectory consists of a closed single loop for a periodic response and multiple loops for a multi-periodic behaviour. The topology of the phase space trajectory provides valuable information regarding the dynamics of a system in a qualitative fashion. While much work has been devoted to extract information from these topological patterns (Letellier et al., 1995; Carroll, 2015; Tufillaro et al., 1991), the concern here is to extract a set of features that can quantify the phase space topology in order to do the inverse problem.

This paper presents a novel method for characterization of the nonlinear response of the system based on the topology...
of its phase space. In an earlier work (Samadani, Kwimy, & Nataraj, 2015), we developed a method that we name here “Phase Space Topology (PST)” for characterizing the topology of the phase space trajectory with quantitative measures. This method which is based on the probability density distribution of time series was used to extract features from the phase plane response of a 1-DOF nonlinear pendulum in the periodic and multi-periodic domains and estimate two parameters of the system. The present paper is an extension to the previous work which generalizes the applicability of the method to multi degree of freedom systems with higher complexities. A 3-DOF nonlinear mass-spring-damper system with up to six simultaneous parametric defects has been used for the demonstration of the method. The robustness and sensitivity of the characterization method to various parameters including noise, time series length and time step and density estimation parameters have been considered as well.

The rest of this paper is organized as follows. Section 2 provides an overview of the method of PST and the computational approach which has been used in this study. In section 3, the case study and its mathematical model are introduced. Section 4 describes the feature extraction and diagnostics procedure and presents the results of its application to numerical and experimental data. The conclusion has been presented in Section 5.

2. Characterization of systems using the method of Phase Space Topology (PST)

PST quantifies the topology of these closed curves by computing the density distribution of points along each axis of the phase space. For simplicity of illustration, the examples here are presented in the two dimensional space; however, it can be extended to higher dimensions. For dimensions higher than three, even though the visualization of the phase space trajectory is not possible, the method is still applicable. In fact, the computations are performed individually and independently for each state of the system. The density of each state is computed by Kernel density estimator as described in the following

Kernel density estimation: Let \( X = (x_1, x_2, \ldots, x_n) \) be an independent and identically distributed sampled data drawn from a distribution with an unknown density function \( f \). The shape of this function can be estimated by its kernel density estimator.

\[
\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right)
\]

where, \( h > 0 \) is a smoothing parameter called bandwidth and \( K(.) \) is the kernel function which satisfies the following requirements.

\[
\int_{-\infty}^{\infty} K(u) du = 1
\]

\[
K(-u) = K(u) \text{ for all values of } u
\]

There are a range of kernel functions that can be used including uniform, triangular, biweight, triweight, Epanechnikov, normal, etc. Due to its conventional mathematical properties, we use the normal kernel function in our approach.

This density function can be computed and plotted for any state of the system. It turns out that the shape of the phase space trajectory which is a closed curve for periodic and multi-periodic behavior is in a direct relationship with the properties of the peaks in the density plots. Specifically speaking, as shown in (Samadani et al., 2015):

- At the end of the curve in the phase plane (where the curve turns back at the local maximum or minimum along each axis) the density of points is significantly higher than the other areas. This produces a sharp peak in the probability density plot of the corresponding axis. In other words, each loop in the phase plane portrait produces two sharp peaks in the density plot of each axis.
- The density of points on the curve at sharper ends (curvature with lower radius) is higher than density of points at rounder ends (bigger radius). Therefore, the sharper the end of the curve, the higher the corresponding peak will be in the probability density plot.
- A depression in the curvature of the phase plane causes a higher density of points in that region which is proportional to the extremeness of the depression and this creates a smooth peak in the probability density plot.

According to these empirical rules, the topology of the phase space trajectory can be characterized with the properties of the peaks in the density plot which themselves can be quantified with following measures:

- \( l_i \) : Location of the peaks
- \( h_i \) : Height of the peaks
- \( s_i \) : Sharpness of the peaks
  - Sharpness is defined as the difference between the left and right slopes at the peak

Therefore, the \( m \) dimensional trajectory of the phase space is mapped into \( m \) density plots, each of which containing several peaks whose properties can characterize the original trajectory. This mapping is unique for a specific set of estimation parameters; however, the inverse statement is not always true. In other words, there can be an unlimited number of phase space topologies for a given set of peak features. Let us now consider the phase plane portrait of a system response with \( n=30,000 \) sampled points shown in Fig. 1a. The data has been obtained by numerical integration of a second order ODE with time step \( \Delta t=0.001 \) sec. A closed curve with two
loops represents a bi-periodic behavior with two commensurable frequencies. The kernel density of the horizontal axis $x$ evaluated at $n_p=1000$ points is shown in Fig. 1b, when $h=5e-04$ has been used for the kernel bandwidth value. As can be seen, the ends of the curve along the horizontal axis $x$ has produced fours sharp peaks in the density plot. The curve has a lower radius in the left side than in the right side and therefore, the first peak in the density plot is higher and sharper than the last peak. The two peaks in the middle are also the result of the small loop in the middle of the phase plane. Similarly, due to the lower radius curvature in those turning points, their corresponding peaks are higher and sharper than the other two.

Figure 1. (a): A sample phase plane plot of a nonlinear second order system obtained from numerical integration (b): The distribution curve

3. CASE STUDY: DIAGNOSTICS OF A 3-DOF NONLINEAR OSCILLATOR WITH SIX PARAMETRIC DEFECTS

3.1. Experimental setup

A 3-DOF nonlinear mass-spring-damper system has been used in this study to demonstrate the implementation of the method. This system which is shown in Fig. 2a is a model 210-rectilinear plant manufactured by ECP and is designed to emulate a broad range of real-world applications including 1-DOF rigid bodies, flexibility in linear drives, gearing and belts, and other coupled discrete oscillatory systems. The mechanism consists of three mass carriages interconnected by springs. The mass carriages are mounted on anti-friction ball bearing type linear motions. Dashpots which provide adjustable viscous damping can be attached between the masses and the base plate. The position of all masses are measured by high resolution encoders. The masses, the springs stiffness and viscous damping of dashpots are adjustable and the reconfigurable design of the electromechanical apparatus allows the user to transform it into a variety of configurations which represent various important classes of real life systems. The rotary motion produced by a brushless DC servo motor is transformed to a linear excitation and is transmitted to the first carriage through a rack and pinion mechanism.

Model 210-rectilinear plant is originally a linear system. The system was transformed into a nonlinear mass-spring-damper system with mounting permanent magnetic discs on each mass carriage, as shown in Fig. 2b. With identical poles of magnets facing each other, a nonlinear repelling force is produced which is proportional to the inverse of the distance squared. In addition to this nonlinear force, a damping force is also produced by the magnets whose coefficient was found to be proportional to the inverse of the distance of magnets. The design of the system allows the user to adjust the initial distance of the magnets by rotating the screws on which the magnets are mounted.

3.2. Mathematical Model

The mathematical model of the system can be described by Eqn. 4. In this equation, $m_i, i = 1, 2, 3$ are the total mass of moving masses (The mass of carriages $m_c, i = 1, 2, 3$ plus the additional weights on each carriage), $k_i, i = 1, 2, 3$ are
the stiffness of springs, \( c_i, i = 1, 2, 3 \) are the viscous damping coefficients of each mass, \( c_m \) is the damping coefficient due to magnets, \( p \) is a constant coefficient associated with the force between magnets, \( r_{0i}, i = 1, 2, 3 \) are the initial distance of the magnets on each two masses, and \( \mu \) is the coulomb friction coefficient.

The rated values of the parameters representing the system in healthy conditions were estimated using system identification techniques. The value of \( p \) was estimated by measuring the static force between two repelling magnets at different distances. The values of \( m, k \) and \( c \) were estimated by fitting the numerical response to the experimental initial condition response of each mass using the method of nonlinear least squares. This procedure was done for several times, and the average values of each parameter were obtained. The magnets then were attached to the mass carriages and the additional damping coefficient value added to the system due to the magnets were estimated. The estimated parameters of the system are presented in Table 1. Note that the second and third mass carriages are similar; therefore, their masses and damping coefficients are identical. Also, the same springs and magnets were used for all masses of the system.

The magnitude of the input force is adjusted by parameter \( A \) on the experimental setup. The relation of \( A \) and \( f_0 \) was obtained by scaling the amplitude of a periodic numerical response to fit the experimental response of the system. This relation was found to be:

\[
f_0 = 6.4 A
\]  

(5)

3.3. System diagnostics using PST

3.3.1. Feature extraction

This section demonstrates the implementation of the method of PST in order to estimate up to six parameters of the system including the values of the masses and initial distance of magnets. Due to the complexity of the system, if the range of parameters is not bounded, a variety of different behaviours can be seen in the system response. However in practice, the parameters of a system stay in a limited range of defective conditions. Here we assume that \( m_i \in [0.8 1.0] \) kg and \( r_{0i} \in [0.019 0.023] \) m for \( i = 1, 2 \) and 3. Figure 3 shows three sample phase portraits of the first mass response (position \( x_1 \) vs. velocity \( x_2 \) ) along with the corresponding density plots for both \( x_1 \) and \( x_2 \), for system parameters presented in Table 2.

Let us now see how the density of \( x_1 \) and its peaks properties change based on the topology of the phase portraits. In Fig. 3a, we have a double-loop phase portrait which is a characteristics of a bi-periodic motion. The edges of the loops in the \( x_1 \) direction have produced four sharp peaks in the density plot of \( x_1 \). The two sharp peaks in the middle are higher than the other two due the lower radius of the phase plot curve in those parts. In addition, the depression of the phase plane curve in the middle has produced a smoother peak in the density plot. In Fig. 3b we have a similar topology; however, the depressed part has moved rightward. It can be seen from the corresponding density plot that the smooth peak has shifted rightward to the middle of the two sharp peaks as well. In Fig. 3c another loop has been evolved in the phase plane, representing a response with three frequencies. As a result, two more sharp peaks have been emerged in the density plot of \( x_1 \). The density plots of \( x_2 \) can also be explained in a similar way. Although here we characterized the topology of the phase portrait of the first mass response based on the density of its position and velocity, this distinction between the system states is not always easy or practical as they can be of different natures (e.g. position, velocity, electrical current, fluid flow, etc.). However, this process can be done for any state of the system independently; regardless of the nature or number of states. In other words, we do not even need to obtain any phase plots to do the analysis and the presented examples are just for the sake of illustration.

3.3.2. The inverse problem

With the features extracted from the states of the system, a machine learning tool can be used as a classifier to classify the faults or as a regressor to estimate the parameters of the system. An artificial neural network (ANN) has been used in this study to regress the computed features to the six parameters of the system. For this purpose, a two-layer feed-forward network with sigmoid hidden neurons and linear output neurons was developed. Due to the complexity of the system response and having three degrees of freedom, we will have a relatively large number of features for each sample of data. This requires a higher number of hidden neurons in order for the network to be trained with minimum regression error. The regression error was found to be minimum for twenty hidden neurons in this case. Depending on the number of the loops and the complexity of the curvature in the phase portrait, the number of peaks and therefore the number of inputs to the neural networks can vary. In this problem, we know that the maximum number of peaks is seven for the possible range of parameters. This knowledge can be obtained from bifurcation diagrams or by simulation of the system for many random parameter sets. We then build the matrix of inputs with respect to this maximum number. For example, in this problem, the response can produce a maximum of seven peaks. Three features are extracted from each peak and therefore, the number of inputs would be 21 for each state and 126 in total for all six states. For cases where the response of the system contains less number of peaks, since we need a constant number of inputs for the neural network, we choose to put zeros in the remaining columns, according to the procedure explained in (Samadani et al., 2015). The data was obtained by random selection of the values of parameters, simulation of the sys-
\[ x_1 = x_2 \]
\[ \dot{x}_2 = \frac{1}{m_1} \left( f_0 \sin(\omega t) + \frac{p_1^2}{(r_{01} + x_1)^2} - \frac{p_2^2}{(r_{02} + (x_3 - x_1))^2} + k_2 (x_3 - x_1) - k_1 x_1 - (c_1 + c_m) x_2 - \text{sign}(x_2) \mu m_1 g \right) \]
\[ \dot{x}_3 = x_4 \]
\[ \dot{x}_4 = \frac{1}{m_2} \left( \frac{p_1^2}{(r_{02} + x_3 - x_1)^2} - \frac{p_2^2}{(r_{03} + (x_5 - x_3))^2} + k_3 (x_5 - x_3) - k_2 (x_3 - x_1) - (c_2 + c_m) x_4 - \text{sign}(x_4) \mu m_2 g \right) \]
\[ \dot{x}_5 = x_6 \]
\[ \dot{x}_6 = \frac{1}{m_3} \left[ \frac{p_2^2}{(r_{03} + (x_5 - x_3))^2} - k_3 (x_5 - x_3) - (c_3 + c_m) x_6 - \text{sign}(x_6) \mu m_3 g \right] \]

Table 1. Nominal parameter values of the nonlinear oscillator

<table>
<thead>
<tr>
<th>( m_{c1} ) (kg)</th>
<th>( m_{c2} ) (kg)</th>
<th>( c_1 ) (Ns/m)</th>
<th>( c_2 ) (Ns/m)</th>
<th>( k_1 ) (N/m)</th>
<th>( p_1^2 ) (Nm^2)</th>
<th>( c_m ) (Ns/m)</th>
<th>( r_0 ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.880</td>
<td>0.397</td>
<td>2.5</td>
<td>1.175</td>
<td>368</td>
<td>0.052</td>
<td>0.014</td>
<td>0.022</td>
</tr>
</tbody>
</table>

Figure 3. Sample phase portraits of the first mass and the corresponding probability density plots for \( x_1 \) and \( \dot{x}_1 \) time series.

Figure 4a shows the learning curve of the ANN for training, validation and test sets. The mean square error (MSE) can be seen as an index of the performance of the feature extraction system and computation of the response features each time. A total number of \( N=200 \) sample data were used to train, and validate the neural network.
Table 2. Parameter values for three sample cases

<table>
<thead>
<tr>
<th></th>
<th>(m_1) (kg)</th>
<th>(m_2) (kg)</th>
<th>(m_3) (kg)</th>
<th>(r_{01}) (m)</th>
<th>(r_{02}) (m)</th>
<th>(r_{03}) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0.96</td>
<td>0.85</td>
<td>0.94</td>
<td>0.0207</td>
<td>0.0212</td>
<td>0.0220</td>
</tr>
<tr>
<td>(b)</td>
<td>0.98</td>
<td>0.85</td>
<td>0.91</td>
<td>0.0220</td>
<td>0.0200</td>
<td>0.0228</td>
</tr>
<tr>
<td>(c)</td>
<td>0.82</td>
<td>0.88</td>
<td>0.99</td>
<td>0.0223</td>
<td>0.0206</td>
<td>0.0211</td>
</tr>
</tbody>
</table>

Figure 4. Learning curve of the neural network for training, validation, and test sets

and response characterization as lower values of MSE represent higher precisions in mapping the response features to the parameters sets. The best validation performance is achieved at epoch 35 with mean square error (MSE) of \(1.3077 \times 10^{-4}\). The small values of MSE for all data sets are representatives of a perfect fit and effectiveness of the proposed method.

In real applications, some states of the system might not be available. For example in our system, we can only measure the position signals \(x_1\), \(x_3\) and \(x_5\). We are interested to see how effective the method of PST would be when the whole phase space is not available. A new ANN was developed using the same data set; but only based on the features extracted from these three states. Figure 4b shows the MSE of the new ANN for training, validation and test sets. Interestingly, it can be seen that even though the convergence has taken a bit longer here, the minimum MSEs achieved here for all data sets are lower compared to the case where all six states were used. This can be due to the fact that a less number of inputs are involved in the optimization process which makes it computationally more efficient. This example clearly shows that including all states of the system does not necessarily provide additional information and make the analysis more efficient and accurate.

**Estimation of Parameters**

Figure 5 shows the estimated values of each parameter versus their real values for seventy cases, where all six parameters have been chosen randomly. Red points on this plot represent the estimated values of the real parameters on the set-up computed with experimental data. In order to do so, \(x_1\), \(x_3\) and \(x_5\) were measured for these six random cases, and the corresponding features were extracted from them. As long as the obtained response is close enough to the response predicted by the mathematical model, the extracted features and the corresponding estimated parameters are expected to be close to their actual values. Figure 5 shows that the developed network can effectively estimate the values of \(m_1\), \(m_2\), and \(m_3\); whereas the estimation error for \(r_{01}\), \(r_{02}\) and especially \(r_{03}\) is relatively higher. Figure 6 shows the distribution of this estimation error for all six parameters. Note that in our analyses, all six parameters of the system were changed randomly at the same time, which rarely happens in real applications and makes the estimation problem more complicated.

**4. Conclusion**

A new method, namely the method of Phase Space Topology (PST) was employed in this paper to diagnose a multi degree of freedom system. The diagnostics was treated as a parameter estimation here, which is an accurate assumption for most real world defects. The features extracted from the nonlinear response of the system using PST are able to quantify the topology of the phase space trajectory. This is done by mapping the phase space trajectory into the density distribution plots of each state. The properties of the peaks in the density plots including their location, height and sharpness can describe this topology with quantitative measures.

The results show the effectiveness of the approach in characterizing the system behaviour and tracking the dynamical changes in the worst case scenario where six parameters were changed simultaneously. This is analogous to a system with six simultaneous defects which rarely happens in practice. The method shows superior capability in extracting useful information from the response in comparison to conventional
frequency, or time-frequency feature extraction methods which provide little information in the presence of nonlinear phenomena.

Some key aspects of the approach need to be addressed in the future work including the dependence of the method on density estimation parameters and robustness of the method to noise and other properties of data. The signals in this problem were fairly smooth and clean; whereas, in some real applications, the data is contaminated with noise and other uncertainties which makes the problem more complicated.

Acknowledgments

This work is supported by the US Office of Naval Research under the grant ONR N00014-13-1-0485. We deeply appreciate this support. Thanks are due to Mr Anthony Seman III of ONR.

REFERENCES


Rezvanizaniani, S. M., Dempsey, J., & Lee, J. (2014). An ef-
Figure 6. Distribution of estimation errors


**Biographies**

Moḥsen Samadani Moḥsen received his B.Sc and M.Sc in Mechanical Engineering from Isfahan University of Technology, Isfahan, Iran. He is currently a Ph.D. candidate at the Department of Mechanical Engineering at Villanova University. Moḥsen has been involved in various research topics including manufacturing technologies, control, vibrations, system dynamics, hydraulic systems and reliability analysis. His current research interests include data analysis, machine learning, nonlinear dynamics and vibrations with applications to machinery diagnostics and health management. He is a member of Sigma Xi and ASME and a recipient of Sigma Xi best poster award and PHM doctoral consortium travel award.

Cedrick Kwuimy Prior to joining the Department of Mechanical Engineering at Villanova University in Jan. 2011, Dr. Kwuimy worked in South Africa as a postdoctoral research associate at the African Institute for Mathematical Sciences (2009-2010) and as a Research and Teaching Assistant (2007-2009) at the Faculty of Science at the University of Yaounde, Cameroon. He has been involved in a wide range of research topics including vibration control, nonlinear dynamics of self-sustained electromechanical devices, synchronization, nonlinear analysis of butterfly valves, and chaos control and prediction in active magnetic bearings. He has over 30 peer-reviewed papers in international journals and conference proceedings and serves as a reviewer in high standard journals including Nonlinear Dynamics, Journal of Vibration and Control and Journal of Sound and Vibration. Dr. Kwuimy has supervised three graduate research theses at the African Institute for Mathematical Sciences and is the recipient of Victor Rothschild Fellowships at African Institute for Mathematical Sciences and Research.

C. Nataraj Dr. C. Nataraj holds the Mr. and Mrs. Robert F. Mortiz, Sr. Endowed Chair Professorship in Engineered Systems at Villanova University. He has a B.S. in Mechanical Engineering from Indian Institute of Technology, and M.S. and Ph.D. in Engineering Science from Arizona State University. After getting his Ph.D. in 1987, he worked for a year as a research engineer and a partner with Trumpler Associates, Inc. He is currently the Chairman of the Mechanical Engineering Department at Villanova University. Dr. Nataraj was also the founding director of the Center for Nonlinear Dynamics and Control in the College Of Engineering. He has worked on various research problems in nonlinear dynamic systems with applications to mobile robotics, unmanned vehicles, rotor dynamics, vibration, control, and electromagnetic bearings. His research has been funded by Office of Naval Research, National Science Foundation, National Institute of Health and many companies.
From Theory to Practice: Model-Based Diagnosis in Industrial Applications

Roxane Koitz1 and Franz Wotawa2

1,2 Institute for Software Technology, Graz, Styria, 8010, Austria
rkoitz@ist.tugraz.at
wotawa@ist.tugraz.at

ABSTRACT

Due to the increasing complexity of technical systems, accurate fault identification is crucial in order to reduce maintenance costs and system downtime. Model-based diagnosis has been proposed as an approach to improve fault localization. By utilizing a system model, possible causes, i.e. defects, for observable anomalies can be computed. Even though model-based diagnosis rests on solid theoretical background, it has not been widely adopted in practice. The reasons are twofold: on the one hand it requires an initial modeling effort and on the other hand a high computational complexity is associated with the diagnosis task in general. In this paper we address these issues by proposing a process for abductive model-based diagnosis in an industrial setting. Suitable models are created automatically from failure assessments available. Further, the compiled system descriptions reside within a tractable space of abductive diagnosis. In order to convey the feasibility of the approach we present results of an empirical evaluation based on several failure assessments.

1. INTRODUCTION

Fault diagnosis of technical systems has gained attention on account of safety and economic considerations in various fields such as artificial intelligence or fault detection and isolation (FDI). In order to improve diagnostic reasoning, the notion and foundations of model-based diagnosis have been investigated (Reiter, 1987; de Kleer & Williams, 1987). Model-based diagnosis as part of artificial intelligence rests on a formal description of the system to be diagnosed and derives root causes from observable anomalies. Within the last decades a solid theoretical background has been established with two approaches emerging: consistency-based and abductive diagnosis. The former depends on knowledge of the correct system behavior and infers diagnoses via inconsistencies (Reiter, 1987; de Kleer & Williams, 1987). In contrast, the abductive technique is based on models representing faults and their manifestations. It exploits the concept of entailment to compute abductive explanations for given observations (Console, Dupré, & Torasso, 1989). Although building upon different ideas, the close relationship between the two approaches has been proven (Console, Dupré, & Torasso, 1991). Cordier et al. (2004) have bridged the gap between the FDI and the consistency-based approach by investigating their relations and developing a unified framework.

Over the years there have been applications in various domains, such as space probes (Williams & Nayak, 1996), the automotive industry (Struss, Malik, & Sachenbacher, 1996), or environmental decision support systems (Wotawa, Rodríguez-Roda, & Comas, 2010). Several projects have been engaged in developing methods to integrate model-based diagnosis into industrial processes (Milde, Guckenbiehl, Malik, Neumann, & Struss, 2000; Fleischanderl, Havelka, Schreiner, Stumptner, & Wotawa, 2001). These efforts, however, mostly focus on the consistency-based method. In general, the model-based approach has not been accepted in practice, mainly due to the initial modeling and the computational complexity (Console & Dressler, 1999; Zoeteweij, Pietersma, Abreu, Feldman, & Van Gemund, 2008).

In this context we propose a process that relies first on Failure Mode Effect Analysis (FMEA) in order to develop system models while keeping knowledge acquisition affordable and second uses restricted logical formalisms where abduction is still tractable (Eiter & Gottlob, 1995; Nordh & Zanuttini, 2008). FMEA as a reliability analysis tool is growing in importance as it has been established as a mandatory task in certain industries, especially for systems that require a detailed safety assessment (Catelani, Ciani, & Luongo, 2010). An FMEA is a systematic analysis of possible component faults and the consequences said faults have on the system behavior and function (Hawkins & Woollons, 1998). Since it represents a clear causal dependency from specific fault modes...
to symptoms, an FMEA provides information needed for abductive reasoning (Wotawa, 2014). Model-based diagnosis research has been concerned with the automatic creation of FMEAs based on models (Price & Taylor, 2002; Struss & Fraracci, 2012), however, we are considering the reverse process, utilizing FMEAs already available to develop suitable system descriptions.

This paper describes a process for applying abductive model-based diagnosis to an industrial setting. The remainder is structured as follows. After defining the abduction problem and presenting one algorithm capable of computing abductive explanations, we outline our suggested process in Section 3. In particular, we focus on the modeling methodology from FMEAs, show that the generated models exhibit a certain topology resulting in a manageable computational complexity, and discuss possibilities to improve the initial diagnosis results. Section 4 covers an empirical evaluation of the abductive diagnosis algorithm for models derived from multiple FMEAs. Section 5 completes the paper and argues in favor of the process’ feasibility.

2. PRELIMINARIES

We assume standard definitions for propositional logic throughout this section (Chang & Lee, 2014). Abductive inference generates plausible explanations for a given set of observations by relying on the notion of entailment (Poole, Goebel, & Aleliunas, 1987). A set of premises $\psi$ logically entails a conclusion $\phi$ if and only if for any interpretation in which $\psi$ is true $\phi$ is also true. We write this relation as $\psi \models \phi$ and call $\phi$ a logical consequence of $\psi$. To utilize this type of reasoning, the abductive model-based diagnosis approach depends on a formalization of the relationship between faults and discoverable manifestations to derive causes for observed symptoms.

In general, abduction is an intractable problem, i.e. it cannot be solved by a polynomial-time algorithm. However, there are tractable subsets of logic, such as propositional definite Horn theory (Nordh & Zanuttini, 2008). We draw upon these findings and consider the propositional Horn clause abduction problem (PHCAP) as defined by Friedrich et al. (1990). A PHCAP links causes to effects via propositional Horn sentences. Let $HC$ be the set of Horn clauses. Along similar lines as Friedrich et al. (1990), we define a knowledge base as a set of Horn clauses from $HC$ over a finite set of propositional variables.

**Definition 1** A knowledge base (KB) is a tuple $(A, Hyp, Th)$ where $A$ denotes the set of propositional variables, $Hyp \subseteq A$ the set of hypotheses, and $Th \subseteq HC$ the set of Horn clause sentences over $A$.

The set of hypotheses denotes the propositional variables which are possible causes and that we can assume to either be true or false. Later in our modeling methodology these hypotheses refer to component-based fault modes. $Th$ represents the theory which contains rules describing the connections between hypotheses and their effects. In order to form an abduction problem, a set of observations, i.e. discovered effects, has to be considered for which explanations are to be computed.

**Definition 2** Given a knowledge base $(A, Hyp, Th)$ and a set of observations $Obs \subseteq A$ then the tuple $(A, Hyp, Th, Obs)$ forms a propositional Horn clause abduction problem (PHCAP).

**Definition 3** Given a PHCAP $(A, Hyp, Th, Obs)$. A set $\Delta \subseteq Hyp$ is a solution if and only if $\Delta \cup Th \models Obs$ and $\Delta \cup Th \not\models \bot$. A solution $\Delta$ is parsimonious or minimal if and only if no set $\Delta' \subset \Delta$ is a solution.

A solution to a PHCAP is a set of hypotheses which logically entails the observations together with the background theory, i.e. $\Delta \cup Th \models Obs$. In addition, we require $\Delta \cup Th$ to be consistent, as from inconsistencies anything can be inferred. Considering that a solution comprises a set of hypotheses explaining the observations it is equivalent to an abductive diagnosis. While Definition 3 does not impose the limitation on the diagnosis to be minimal, in most practical applications only parsimonious solutions are of interest. Therefore, if not specified otherwise, we refer to minimal diagnoses simply as diagnoses. Notice that finding solutions to a given PHCAP is an NP-complete problem. We refer the interested reader to Friedrich, Gottlob, and Nejdl (1990) for a proof.

While there are several abductive reasoning systems, such as Theorist (Poole et al., 1987), it is well known that Assumption-Based Truth Maintenance Systems (ATMS) (de Kleer, 1986a, 1986b) are capable of deriving abductive explanations as well. The ATMS employs a graph structure where hypotheses, observations, and contradiction are represented as nodes. Implications determine the directed edges in the graph. Each node has a label assigned which contains the set of hypotheses said node can be inferred from. By updating the labels, the ATMS retains consistency. In case a single effect is observed, the label of the corresponding proposition already contains the abductive explanations. To handle multiple observations, a single rule is added, comprising a conjunction of the observations on the left hand side and a new proposition on the right hand side, i.e. $o_1 \land o_2 \ldots \land o_n \rightarrow obs$. Every set contained in the label of $obs$ constitutes a solution to the particular PHCAP. Wotawa, Rodriguez-Roda, and Comas (2009) propose Algorithm 1 employing an ATMS and returning consistent abductive explanations.

3. PROCESS

Intercalating abductive diagnosis into real-world applications faces two major issues: constructing the domain model and the complexity of diagnosis. In this regard we define a pro-
Algorithm 1: abductiveExplanations

```
procedure ABDUCTIVEEXPLANATIONS (A, Hyp, Th, Obs)
    Add Th to ATMS
    Add \( \bigwedge_{o \in \text{Obs}} o \rightarrow \text{obs} \) to ATMS \( \triangleright \text{obs} \notin A \)
return the label of obs.
end procedure
```

cess to address these problems that takes advantage of information available and the structure of the resulting system descriptions. We divide the process into three main steps, as can be seen in Figure 1:

1. Model Development
2. Fault Detection
3. Fault Identification

We give a short overview of the stages and subsequently elaborate on certain parts in the following sections.

1. Model Development. As mentioned earlier abductive model-based diagnosis relies on an explicit description of the system behavior in presence of a fault. Our modeling methodology utilizes FMEAs available. As these assessments capture knowledge on failures and their symptoms, the mapping to a corresponding abductive knowledge base (KB), as defined in the previous section, is straightforward. Since abductive diagnosis depends on the premise of model completeness, we assume that all significant fault modes for each contributing part of the system are being considered in the analysis. Furthermore, our mapping approach expects consistent effect descriptions, i.e. a symptom is described in a uniform way throughout the FMEA. Since FMEAs usually consider single faults the resulting diagnostic system holds the single fault assumption. Note that the model can be compiled automatically offline.

2. Fault Detection. Abductive diagnosis derives possible explanations for observed anomalies, hence to initiate the diagnosis process, the presence of a fault has to be detected. Within our process, we assume the manifestation of a fault is discovered by a monitoring system and therefore do not consider the data acquisition or analysis in detail.

3. Fault Identification. Once the presence of a disturbance has been established, the possible causes associated with the observations are to be computed. Due to the knowledge represented in FMEAs, abductive diagnosis poses an intuitive approach for fault identification. We already discussed one possible algorithm capable of computing abductive diagnoses in the previous section. The process, however, is not limited to the use of this exact procedure (Koitz & Wotawa, 2015a).

In the course of this paper we explain further improvements to the initial set of solutions. In particular, we show a simple diagnoses ranking according to probability theory and how to determine the next probing point in order to diminish the number of solutions.

3.1. Model Development

The initial construction of the system description related to model-based diagnosis hinders a widespread industrial adoption. To automate this task, we propose a mapping function associating entries from an FMEA with propositional Horn clauses. Even though there are several possible representation languages suitable for diagnosis, logics provide a precise semantic of entailment necessary for abductive diagnosis.

Formally, we create a knowledge base \( KB \). To avoid some of the inefficiencies due to complexity, we focus on a subset of logics, namely definite propositional Horn clauses. This does not impose a restriction in our case, as this representation is specific enough to capture the information contained in the FMEAs.

3.1.1. Running Example

The converter of an industrial wind turbine will act as our running example to illustrate the modeling process. Since the converter allows operation at variable speed whilst con-
nnecting the turbine to a constant frequency grid, it is a funda-
mental part of a modern industrial wind turbine. Table 1
depicts a portion of the corresponding FMEA omitting all
parts concerned with reliability analysis, e.g. severity ratings.
Each record contains information on a component’s possible
fault mode and the failure’s effects. For example, $P_{\text{turbine}}$
refers to a deviation between expected and measured turbine
power output and $T_{\text{cabinet}}$ indicates a higher than predicted
temperature in the inverter cabinet (Gray, Koitz, Psutka, &
Wotawa, 2015). Notice that the effect descriptions in the third
column are consistent throughout the example.

We assume an FMEA comprises a set of components
$\text{COMP}$, their potential fault modes $\text{MODES}$, and the
set of effects which we define as a subset of the set of
propositional variables $\text{PROPS}$.

Definition 4 An FMEA is a set of tuples $(C, M, E)$ where
$C \in \text{COMP}$ is a component, $M \in \text{MODES}$ is a fault
mode, and $E \subseteq \text{PROPS}$ is a set of effects.

Since the FMEA already represents the relation between de-
fects and their manifestations the conversion to a suitable
abductive model is straightforward. The mapping function
$\mathfrak{M} : 2^{\text{FMEA}} \rightarrow HC$ generates corresponding propositional
Horn clauses for each entry of the FMEA, i.e. rules describ-
ing the connections between a component-based fault mode
and its effects.

Definition 5 Given an FMEA, the function $\mathfrak{M}$ is defined as
follows:

$$\mathfrak{M}(\text{FMEA}) =_{\text{def}} \bigcup_{t \in \text{FMEA}} \mathfrak{M}(t)$$ (1)

where

$$\mathfrak{M}(C, M, E) =_{\text{def}} \{ \text{mode}(C, M) \rightarrow e \mid e \in E \}$$ (2)

Hypotheses, hence all propositional variables allowed to
contribute as a cause, are represented as the proposition
$\text{mode}(C, M)$, where $C$ and $M$ relate to the corresponding
component and fault mode, respectively. Equation. (3) de-
defines $H_{\text{hyp}}$ in this modeling context.

$$H_{\text{hyp}} =_{\text{def}} \bigcup_{(C, M, E) \in \text{FMEA}} \{ \text{mode}(C, M) \}$$ (3)

Considering the running example. We would obtain the fol-
lowing elements for the set of hypotheses:

$$H_{\text{hyp}} = \{ \text{mode}(\text{Fan_Pin}, \text{Corrosion}), \ 
\text{mode}(\text{Fan_Bearing_Running_Surface}, \ 
\text{Thermo_mechanical_fatigue(TMF)}), \ldots \}$$

Equation (4) defines the set of propositional variables as the
union of all effects and hypotheses stored in the FMEA.

$$A =_{\text{def}} \bigcup_{(C, M, E) \in \text{FMEA}} E \cup \{ \text{mode}(C, M) \}$$ (4)

Continuing our converter example:

$$A = \{ \text{mode}(\text{Fan_Pin}, \text{Corrosion}), \ 
\text{mode}(\text{Fan_Bearing_Running_Surface}, \ 
\text{Thermo_mechanical_fatigue(TMF)}), \ldots \}$$

Applying $\mathfrak{M}$ results in the following theory $\text{Th}$ completing the $KB_{\text{Converter}}$:

$$\text{Th} = \{ \text{mode}(\text{Fan_Pin}, \text{Corrosion}) \rightarrow T_{\text{cabinet}}, \ 
\text{mode}(\text{Fan_Pin}, \text{Corrosion}) \rightarrow P_{\text{turbine}}, \ 
\text{mode}(\text{Fan_Bearing_Running_Surface}, \ 
\text{Thermo_mechanical_fatigue(TMF)}) \rightarrow T_{\text{cabinet}}, \ldots \}$$

It is worth noticing that $\text{Th}$ constructed from an FMEA
via $\mathfrak{M}$ features bijective definite Horn clauses. To en-
sure that contradicting observations are omitted during di-
agnosis, additional Horn clauses are created in $\text{Th}$, stating
that an effect and its complement cannot occur at the same
time, i.e. $e \land \neg e \rightarrow \bot$. For example, we
would include the rule $\text{Equivalent_series_resistance}(<) \land \text{Equivalent_series_resistance}($$>) \rightarrow \bot$ in the theory.

3.1.2. One Single Fault Diagnosis Property

The appropriateness of the models obtained from the FMEA
is yet to be examined. Due to the fact that abductive explana-
tions are consistent by definition and complete given an ex-
haustive search, suitability refers to the characteristic of the
model that given all necessary information a single diagnosis
can be computed. We refer to this feature as the One Single
Fault Diagnosis Property (OSFDP).

Definition 6 Given a $KB (A, H_{\text{hyp}}, \text{Th})$. $KB$ fulfills the

<table>
<thead>
<tr>
<th>Component</th>
<th>Fault Mode</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan - Pin</td>
<td>Corrosion</td>
<td>$T_{\text{cabinet}}, P_{\text{turbine}}$</td>
</tr>
<tr>
<td>Fan - Bearing Running Surface</td>
<td>Thermo-mechanical fatigue (TMF)</td>
<td>$T_{\text{cabinet}}, P_{\text{turbine}}$</td>
</tr>
<tr>
<td>Buck Boost - Electrolyte Capacitor</td>
<td>Electrical chemical aging</td>
<td>$T_{\text{power_cabinet}}, P_{\text{turbine}}, \text{Equivalent series resistance}($$&gt;)$</td>
</tr>
<tr>
<td>Buck Boost - Electrolyte Capacitor</td>
<td>Electrical chemical aging</td>
<td>$T_{\text{power_cabinet}}, P_{\text{turbine}}, \text{Alarm code (over voltage, link)}, \text{Equivalent series resistance}(&lt;)$, Electrolyte trace</td>
</tr>
<tr>
<td>IGBT - Wire Bonding</td>
<td>High-cycle fatigue (HCF)</td>
<td>$T_{\text{inverter_cabinet}}, T_{\text{nacelle}}, P_{\text{turbine}}$</td>
</tr>
</tbody>
</table>
The property can be checked by computing for each $h \in H_{yp}$ the set of propositions $\delta(h)$, such that $\{h\} \cup Th \models \delta(h)$. In case $\{h\} \cup Th$ leads to a contradiction, $\delta(h)$ equals $\emptyset$. If for any two hypotheses the derived propositions are the same, the OSFDP is not satisfactory. Besides determining whether single fault diagnoses can be computed, the absence of the property indicates that $KB$ is not complete, i.e. information is missing. In the case of FMEAs this can signal that internal variables or observations have not been contemplated during the analysis. A polynomial time algorithm for testing whether the property is satisfied can be found in Wotawa (2014).

A simple procedure to enforce the OSFDP treats indistinguishable faults as a unit. Hence, each set of indistinguishable hypotheses $\{h_1, h_2, \ldots, h_n\}$ is replaced by a new hypothesis $h'$. We proceed with these substitutions until the OSFDP is fulfilled. Algorithm 2 ensures that after termination the given $KB$ satisfies the property. It assumes that for each hypothesis in $H_{yp}$ the set $\delta(h)$ has already been computed. Due to the finite number of hypotheses as well as possible effects contained in $\delta(h)$, the procedure must halt. Further, the complexity of the algorithm is determined by the three nested loops, hence $O(|H_{yp}|^2 + |A - H_{yp}|)$.

Enforcing the OSFDP has a practical rational: Since the indistinguishable faults cannot be differentiated, all components have to be repaired or replaced in case they are part of the diagnosis. Thus, treating them as a single unit during diagnosis does not influence the result; however, it does have an effect on the computational effort because it reduces the number of possible hypotheses to consider.

Our running example of the converter does not fulfill the OSFDP, since $mode(Fan_Pin, Corrosion)$ and $mode(Fan_Bearing, Running_Surface, Thermo_mechanical_fatigue,(TMF))$ are not distinguishable. By removing both hypotheses and introducing $h' = mode((Fan_Pin, Fan_Bearing, Running_Surface), (Corrosion, Thermo_mechanical_fatigue,(TMF)))$ the property is fulfilled.}

### Algorithm 2 distinguishHypotheses

```algorithm
procedure distinguishHypotheses
   $KB(A, H_{yp}, Th)$
   $\Psi||H_{yp}|| \leftarrow H_{yp}$
   for all $h_1 \in \Psi$ do
      for all $h_2 \in \Psi$ do
         if $h_1 \neq h_2$ then
            if $\delta(h_1) = \delta(h_2)$ and $\delta(h_1) \neq \emptyset$ then
               Create new hypothesis $h'$
               $h' \notin H_{yp}$
               Add $h'$ to $\Psi$
               Add $h'$ to $A$
            for all $e \in \delta(h_1)$ do
               Add $(h' \rightarrow e)$ to $Th$
               Remove $(h_1 \rightarrow e)$ from $Th$
               Remove $(h_2 \rightarrow e)$ from $Th$
            end for
            Remove $h_1 \land h_2$ from $\Psi$
            Remove $h_1 \land h_2$ from $A$
         end if
      end for
   end for
   return $KB(A, \Psi, Th)$
end procedure
```

the intersection of the set of hypotheses and effects is empty. These features of the model all reduce the computation complexity in regard to the abduction problem. In particular, abductive diagnosis requires polynomial time in our case. For a more detailed discussion we refer the interested reader to Koitz and Wotawa (2015b).

### 3.2. Observation Discrimination

Generally, there might be an exponential number of diagnoses. In a real world context, however, a single solution is preferred. Probe selection has been proposed as a way to minimize the number of results. While other approaches assume an interleaved process between diagnosis and repair (Friedrich et al., 1990), Wotawa (2011) suggests computing all solutions and subsequently adding new symptoms which allow to discriminate diagnoses. A discriminating observation is a measurement not yet considered, which decreases the number of possible faults.

**Definition 7** Given a PHCAP $(A, Hyp, Th, Obs)$ and two diagnoses $\Delta_1$ and $\Delta_2$. A new observation $o \in A \setminus Obs$ discriminates two diagnoses if and only if $\Delta_1$ is a diagnosis for $(A, Hyp, Th, Obs \cup \{o\})$ but $\Delta_2$ is not.

According to information theory, the observation with the highest entropy $H(o)$ (Eq. (5)) provides the best probing point (de Kleer & Williams, 1987).

$$H(o) = -p(o) \cdot \log_2 p(o) - (1 - p(o)) \cdot \log_2 (1 - p(o)) \quad (5)$$

$p(o)$ denotes the probability of observation $o$ and is defined in Eq. (6).
Table 2. Features of the FMEAs and experimental results. For each component we conducted the experiment using the original model as well as a model fulfilling the OSFDP. The last three columns display the maximum number of single faults, double faults, and triple faults, respectively.

<table>
<thead>
<tr>
<th>Component</th>
<th>Model Structure</th>
<th>Runtime [in ms]</th>
<th>#Diagnoses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Hyp</td>
<td>#Effects</td>
<td>#Rules</td>
</tr>
<tr>
<td>Electrical circuit</td>
<td>Original</td>
<td>32</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>Ford connector</td>
<td>Original</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>system</td>
<td>OSFDP</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>HIFI - FPU</td>
<td>Original</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>MiTS1</td>
<td>Original</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>MiTS 2</td>
<td>Original</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>PCB</td>
<td>Original</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>ACD</td>
<td>Original</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>Inverter</td>
<td>Original</td>
<td>29</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>23</td>
<td>38</td>
</tr>
<tr>
<td>Rectifier</td>
<td>Original</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>Transformer</td>
<td>Original</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Backup components</td>
<td>Original</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>19</td>
<td>30</td>
</tr>
<tr>
<td>Main bearing</td>
<td>Original</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>OSFDP</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
p(o) = \frac{|\{\Delta | \Delta \in \Delta \text{-Set}, \Delta \cup Th = \{o\}\}|}{|\Delta \text{-Set}|} \tag{6}
\]

\(\Delta \text{-Set}\) is the set of diagnoses obtained as a solution to the PHCAP. Once the next best probing point has been selected and the additional measurements have been taken, the probing results are passed on to the diagnosis engine as observations and the fault identification process is restarted.

3.2.2. Fault Ranking

We assume independence amongst faults. Hence, the probability of a diagnosis \(\Delta\), derived from the knowledge base \(KB\) and given observations \(Obs\), can be computed by Eq. (7).

\[
p(\Delta) = \prod_{h \in \Delta} p(h) \prod_{h \notin \Delta} (1 - p(h)) \tag{7}
\]

\(p(h)\) represents the a-prior probability of the fault \(h\). We presume that the fault probabilities are known, e.g. from the manufacturer or fault history analysis. Given a \(PHCAP\) we compute \(p(\Delta)\) for all diagnoses in \(\Delta\)-Set and subsequently assign ranks correspondingly.

4. EMPIRICAL EVALUATION

In this section we report on our test scenarios and results. We obtained several publicly available as well as project internal FMEAs considering diverse technical systems and subsystems. Subsequently, we created corresponding abductive knowledge bases \(KB\) from the analyses via the mapping function \(\mathcal{M}\). The FMEAs cover electrical circuits, a connector system by Ford, the Focal Plane Unit (FPU) of the Herschel Space Observatory, printed circuit boards (PCB), the Anticoincidence Detector (ACD) mounted on the Large Area Telescope of the Fermi Gamma-ray Space Telescope, the Maritim ITStandard (MiTS), as well as rectifier, inverter, transformer, main bearing, and backup components of an industrial wind turbine. As can be seen from Table 2 these FMEAs vary in the number of components and faults (i.e. hypotheses), observations (i.e. effects), as well as connections between faults and effects (i.e. rules). Note that the numbers referred to in the table correspond to the underlying FMEAs and not to the abductive model. We tested each FMEA for the OSFDP and as Table 2 reveals none, except of the model resulting from the transformer’s failure assessment, of the original models satisfies the property. To generate models which fulfill the OSFDP, each set of indistinguishable hypotheses was exchanged with a new single hypothesis representing said set. Thus, the number of hypotheses and
rules diminishes for the adapted models. We do not report on the computation time of the mapping, as model generation is executed offline and the conversions we have computed so far took less than a second.

After model compilation, we examined the performance of a Java implementation of Algorithm 1 on the generated KBs. The evaluation was performed on an Intel Core i7-4700MQ processor (2.60 GHz) with 8 GB RAM on Windows 7 Enterprise (64-bit). Note that our implementation utilizes an unfocussed ATMS (Forbus & de Kleer, 1988). For each FMEA we ran the algorithm for $|\text{obs}|$ from 1 to maximum number of effects possible. The observation set was generated randomly, however, we utilized the same observations for the original as well as for the adapted model. The results reported in Table 2 have been obtained from 100 trials. Unsurprisingly, the runtime increases with the number of rules to consider. As the results show while there are maximum computation times of around five seconds, the median of the distributions is located around and below ten milliseconds. Comparing the original model to the OSFDP variant, we see a performance advantage of the latter for the majority of examples. It is worth noticing that even though the transformer example already satisfied the OSFDP, the runtimes deviate. These discrepancies can be attributed to the small unit of measurement in the millisecond range.

Figure 2 depicts the underlying statistical distribution of the performance for the original and the adapted models. In order to determine whether the adapted models are superior in regard to the diagnosis performance, we used an adaptation of the sign test as described by Stumptner and Wotawa (2001). Suppose paired runtime data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from the original and adapted models, respectively. We propose the hypothesis $H_0: mX = mY$, stating a median difference of zero. $H_1: m_d > 0$ is our alternative hypothesis, where $m_d$ denotes the median of $X - Y$. Let $Z$ be the sum of pairs, where $x_i > y_i$. Given $H_0$ is true, the test statistic $Z \sim B_{0.5,n}$ has to be binomial distributed. We refute $H_0$ and accept $H_1$ if the critical value $z_\alpha$ is smaller than the $z$ value from the sample. Since there is a large number of samples in our evaluation, the critical values for the sign test are not based directly on the binomial distribution, but rather on a normal approximation. For $\alpha = 0.05$ we accepted $H_1$, i.e. the runtime performance for the adapted models is superior to the original ones.

5. CONCLUSION

In the course of the presented research, we proposed a process to facilitate the adoption of abductive model-based diagnosis in industrial practice. FMEAs contain information suitable for abductive system descriptions and allow us to automatically generate models offline. Exploiting failure assessments is a feasible approach, as on the one hand this sort of analyses is becoming increasingly important and on the other hand the abduction problem corresponding to the contents of these documents is computationally feasible. We evaluated the resulting models on an implementation of an abductive diagnosis algorithm to identify corresponding performance trends. The results indicate that the computation times are on average under half a second. We argue that the automated modeling based on FMEAs allows for immediate reuse of information and thus provides a convenient way to employ abductive model-based diagnosis without the associated modeling effort.

ACKNOWLEDGMENT

The work presented in this paper has been supported by the FFG project Applied Model Based Reasoning (AMOR) under grant 842407. We would further like to express our gratitude to our industrial partner, Uptime Engineering GmbH.
NOMENCLATURE

α  significance level
A  set of propositional variables
ACD Assumption-Based Truth Maintenance System
COMP set of components
δ(h) propositions entailed by hypothesis h
Δ diagnosis
Δ-Set set containing all diagnoses
DF double fault
E set of effects
FDI fault detection and isolation
FMEA Failure Mode Effect Analysis
H entropy
HC set of Horn clauses
HIFI-FPU Focal Plane Unit of the Heterodyne Instrument for the Far Infrared built for the Herschel Space Observatory
Hyp set of hypotheses
KB Knowledge Base
m median value
M fault mode
M mapping function
MiTS Maritim ITStandard
MODES set of fault modes
obs set of observations
Obs set of all possible observations
OSFDP One Single Fault Diagnosis Property
p probability
PCB Printed Circuit Boards
PHCAP Propositional Horn Clause Abduction Problem
PROPS set of propositional variables
SF single fault
Th theory
TF triple fault
X, Y random variables
Z test statistic
zcs critical value

REFERENCES


studies from the India project. *AI Communications, 13*(2), 99–123.


Wotawa, F. (2011). On the use of abduction as an alternative to decision trees in environmental decision support systems. *International journal of agricultural and environmental information systems, 2*(1), 63–82.


102
Prognostics and Health Management of an Electro-Hydraulic Servo Actuator

Andrea Mornacchi¹, George Vachtsevanos² and Giovanni Jacazio³

¹,³ Politecnico di Torino - Department of Mechanical and Aerospace Engineering, Turin, 10129, Italy
andrea.mornacchi@polito.it
giovanni.jacazio@polito.it

² Georgia Institute of Technology, Atlanta, GA, 30332, USA
gjv@ece.gatech.edu

ABSTRACT

Electro-Hydraulic Servo Actuators (EHSA) is the principal technology used for primary flight control in new aircrafts and legacy platforms. The development of Prognostic and Health Management technologies and their application to EHSA systems is of great interest in both the aerospace industry and the air fleet operators.

This paper presents the results of an ongoing research activity focused on the development of a PHM system for fly-by-wire primary flight EHSA. One of the key features of the research is the implementation of a PHM system without the addition of new sensors, taking advantage of sensing and information already available. This choice allows extending the PHM capability to the EHSA systems of legacy platforms and not only to new aircrafts. The enabling technologies borrow from the area of Bayesian estimation theory and specifically particle filtering and the information acquired from EHSA during pre-flight check is processed by appropriate algorithms in order to obtain relevant features, detect the degradation and estimate the Remaining Useful Life (RUL). The results are evaluated through appropriate metrics in order to assess the performance and effectiveness of the implemented PHM system.

1. INTRODUCTION

Flight control systems and their associated flight control servoaetuators are one of the critical aircraft systems and belong to the top operational disruption contributors. Developing effective PHM algorithms for primary flight control actuators that can be integrated in a health monitoring system for the entire aircraft flight control system will lead to a valuable technological advancement.

The benefits achievable from developing an efficient health monitoring system able to anticipate the failures of the aircraft flight control system fall in two areas:

- Improvement of the aircraft operational reliability and dispatchability by avoiding:
  - Aircraft on ground immobilization
  - Takeoff delays and cancellations
  - Re-routing
  - In-flight turn back

- Reduction of direct maintenance costs by:
  - Performing maintenance operations of anticipated failures at an airline main base
  - Improving troubleshooting of failures
  - Reducing scheduled maintenance operations and rescheduling some recurring maintenance tasks

Costs related to unscheduled maintenance operation and to flight disruptions resulting from unexpected failures may vary in a relatively large range, depending on the type of aircraft and of its flight control system, on the operational environment, on the maintenance policies and on the aircraft usage. Though not easily quantifiable, these costs are at present a large fraction of the life cycle cost.

- Assuming an average cost related to unexpected failures equal to 30% of the system total life cycle cost, which is a conservative underestimate, the development of a health management system for the aircraft flight controls able to reduce to half that cost would provide a tremendous benefit to the aircraft operation.

- It is also generally accepted that the average cost for an aircraft downtime is equal to about US$ 10000 per
hour (Pohl, 2013), therefore, also a minor reduction of
the average aircraft downtime for an aircraft fleet
entails very large savings.

IATA projection for global spending in 2020 for
maintenance, repair and overhaul is US$ 65 billion (IATA,
2011). Although the spending for flight control actuators
will be only a fraction of that total figure, it is evident that
the contribution gained from the introduction of an effective
health monitoring system for aircraft flight control actuators
will still contribute to a large cost saving for maintenance
operations. Another large cost saving is obtained from the
reduction of flight disruptions and delays. A recent study
on integrated disruption management and flight planning
shows that suitable planning can mitigate the effects of
flight disruptions and lead to about 6% cost saving for the
airline (Marla, Vaaben, Barnhart, 2011). Though this study
did not specifically refer to health monitoring systems, it
provides an indication of the order of magnitude of the cost
savings that can be attained by reducing flight disruptions
and delays.

It is therefore easy to understand how the Prognostic and
Health Management (PHM) systems has found an intense
interest in the aerospace area over the past years. Primary
flight control systems are an engineering area where PHM
has found so far very limited interest, although they are one
of the critical aircraft systems. Some work has been
reported on PHM for electromechanical flight control
actuators, but almost very little or nothing for
electrohydraulic servo-actuators (EHSA) for primary flight
controls. However, although electromechanical actuators
(EMA) for primary flight control systems are long-term
objectives, sensitivity to certain single point of failures that
can lead to mechanical jams, results in a reluctance to adopt
EMAs for flight safety critical applications. EMAs for
primary flight controls have so far been limited to UAVs
(Jacazio, 2008). It should be pointed out that primary flight
control actuators for fly-by-wire commercial aircraft in
service and for aircraft under development are almost
invariably electrohydraulic servo-actuators; the only
exception are some electro-hydrostatic actuators (EHA)
used as a backup to conventional EHSAs in the flight
control systems of Airbus A380, A350 and Gulfstream
G650. “Electro-hydraulic servovalves (EHSV) are a critical
component of EHSAs, they are made up by a large number
of parts and can thus fail as a result of several causes. The
research work presented in this paper was therefore focused
on developing a PHM system able to identify the
progressive degradations of EHSV and alert of a
developing failure. The research activity will then continue
addressing the faults of the hydraulic linear actuator in
one of the few research papers focused on the hydraulic
actuators for aviation. The authors examine the possibility
developing a PHM system for the F/A-18 stabilizer
Electro-Hydraulic Servo-Valves (EHSV). The data-drive
approach developed uses neural network error-tracking
techniques, along with fuzzy logic classifiers, Kalman filter
state predictors, and feature fusion strategies. An interesting
work was presented by NASA Ames Research Center
(Narasimhan, Roychoudhury, Balaban & Saxena, 2010).
The paper proposed a combined model-based and feature-
driven diagnosis methodology that allows the detection of
the common EMAs fault modes. Brown et al. (2009a and
2009b) have shown the possibility of exploiting the particle
filter for the diagnostics and prognostics of EHAs.

The major objective of this contribution is to develop an
innovative fault diagnosis and failure prognosis framework
for critical aircraft components that integrates effectively
and mathematically rigorous and validated signal
processing, feature extraction, diagnostic and prognostic
algorithms with novel uncertainty representation and
management tools in a platform that is computationally
efficient and ready to be transitioned on-board an aircraft.

2. EHSA CONFIGURATION

The EHSA used in this research is a typical electrohydraulic
primary flight control actuator. It is composed of the
hydraulic and the control parts. The first consists of one
electrohydraulic servo-valve and a linear hydraulic actuator.
The servo-valve is of the flapper-nozzle type and it is made
up of two stages with the first stage receiving the current
command as the input and using the torque motor in order to
move the flapper thus creating a pressure drop at the ends
of the second stage spool, which controls the flow to the
hydraulic actuator. The control structure uses a position
linear sensor as the feedback sensor for closed loop position
control. The reference system for the EHSA is shown in
Figure 1. In order to ensure redundancy of the drives and,
consequently, greater safety, two actuators acting on the
same flight control surface are employed with the two
EHSA operating in an active-active, or active-standby
mode.
2.1. EHSA signals

The typical structure of an EHSA for legacy aircraft allows for the acquisition of three kinds of information:

- Position command: corresponding to the position request processed by the flight control computer.
- Real position: information acquired by means of the LVDT and used to close the control position loop.
- Servo-valve current: generated by the controller coincident with the compensated error. It is used to control the valve.

2.2. Servo-valve degradation

The types of faults that most commonly occur in the EHSA are well known, although no consolidated models for such failure modes exist which can be taken as a basis for predicting their fault progression. Although these fault growth models are not yet fully validated, their physical based approach ensures that the fault growth pattern is described correctly allowing for a virtual testing of the efficacy of health monitoring algorithms.

Possible degradation modes include:

- Reduction of the torque of the first stage torque motor. This can be the result of a shorting of adjacent coils of the torque motor due to the presence of metallic debris, or to a degradation of the magnetic properties of the materials. A progressively slower response of the servo-valve is obtained.
- Contamination of the first stage filter and nozzles. As dirt and debris accumulate in the first stage filter or in the nozzles, their hydraulic resistance increases which, in the end, leads to a slower response of the servo-valve.
- Stiffness variation of internal feedback spring, which is generally caused by yield in strength due to excessive loads or to normal aging of the component; involves hysteresis phenomena and instability.
- Increase of the backlash at the mechanical interface between the internal feedback spring and spool. This is the result of a wear due to the relative movement between these two parts giving rise to an increasing hysteresis in the servo-valve response, which leads to an instability.
- Variation of the friction force between spool and sleeve. This is due to a silting effect associated either with debris entrained by the hydraulic fluid or to the decay of the hydraulic fluid additives which tend to polymerize when the fluid is subjected to large shear stresses.
- Increase of the radial clearance between spool and sleeve and change of the shape of the corners of the spool lands due to wear between these two moving parts.

In the absence of consolidated degradation models, progression of a degradation provisionally assumed to be a function of either usage time, or amplitude / frequency of commands, or both. In the study of the occlusion of the first stage filter it is considered to be a function of the square of flight hours.

3. Prognostics and Health Management

We introduce an integrated framework for fault diagnosis and failure prognosis that relies on systems engineering principles and takes advantage of physics of failure models, Bayesian estimation methods and measurements acquired through seeded fault testing and/or on-board the aircraft. The proposed Bayesian estimation framework for diagnosis and prognosis for nonlinear, non-Gaussian systems begins with a systems engineering process to identify critical components and their failure models, sensing and monitoring requirements and processing algorithms. Fundamental to this approach is the development of physics-based failure or fatigue models and the optimum selection and extraction of features or Condition Indicators (CI’s) from raw data that form the characteristic signatures of specific fault modes. The latter are selected based on such criteria as sensitivity to particular fault modes and their correlation to ground truth data. The proposed framework employs a nonlinear state-space model of the plant, i.e. critical aircraft component, with unknown time-varying parameters and a Bayesian estimation algorithm called particle filtering to estimate the probability density function (PDF) of the state in real time (Orchard & Vachtsevanos, 2009). The state PDF is used to predict the evolution in time of the fault indicator, obtaining as a result the PDF of the RUL for the faulty component/system. A critical fault is detected and identified by calling on the particle filter-based module that expresses the fault growth dynamics. Prognosis has been called the Achilles’ heel of CBM due to major challenges arising from the inherent uncertainty in prediction. Prognosis may be understood as the result of the procedure where long-term (multi-step) predictions - describing the evolution in time of a fault indicator – are generated with the purpose of estimating the RUL of a failing component. The same particle filtering framework and nonlinear state model suggested above will be used to estimate the RUL (Roemer, Byington, Kacprzynski, Vachtsevanos & Goebel, 2011).

Particle filtering has a direct application in the arena of fault detection and identification (FDI) as well as prediction of the time to failure of a critical component. Indeed, once the current state of the system is known, it is natural to implement FDI procedures by comparing the process behavior with patterns regarding normal or faulty operating conditions. (Vachtsevanos, Lewis, Romer, Hess & Wu,
A fault diagnosis procedure involves the tasks of fault detection and identification (assessment of the severity of the fault). In this sense, the proposed particle-filter-based diagnosis framework aims to accomplish these tasks, under general assumptions of non-Gaussian noise structures and nonlinearities in process dynamic models, using a reduced particle population to represent the state pdf (Orchard, Kacprzynski, Goebel, Saha & Vachtsevanos, 2008). A compromise between model-based and data-driven techniques is accomplished by the use of a particle filter-based module built upon the nonlinear dynamic state model:

\[
\begin{align*}
\dot{x}_d(t + 1) &= f_b(x_d(t), n(t)) \\
\dot{x}_c(t + 1) &= f_t(x_d(t), x_c(t), \omega(t)) \\
\dot{f}_p(t) &= h_e(x_d(t), x_c(t), v(t))
\end{align*}
\]

where \(f_b, f_t, \) and \(h_e\) are non-linear mappings, \(x_d(t)\) is a collection of Boolean states associated with the presence of a particular operating condition in the system (normal operation, fault condition) \(x_c(t)\) is a set of continuous-valued states that describe the evolution of the system given those operating conditions, \(f_p(t)\) is a feature measurement, \(\omega(t)\) and \(v(t)\) are non-Gaussian distributions that characterize the process and feature noise signals respectively. For simplicity, \(n(t)\) may be assumed to be zero-mean i.i.d. uniform white noise. At any given instant of time, this framework provides an estimate of the probability masses associated with each fault mode, as well as a pdf estimate for meaningful physical variables in the system. Once this information is available within the FDI module, it is conveniently processed to generate proper fault alarms and to inform about the statistical confidence of the detection routine. Furthermore, pdf estimates for the system continuous-valued states (computed at the moment of fault detection) may be used as initial conditions in failure prognostic routines, giving an excellent insight about the inherent uncertainty in the prediction problem. As a result, a swift transition between the two modules (FDI and prognosis) may be performed, and reliable prognosis can be achieved within a few cycles of operation after the fault is declared. This characteristic is, in fact, one of the main advantages of the proposed particle-filter-based diagnosis framework.

4. PHM Strategy

One of the main difficulties for developing a prognostic system for EHSAs is the lack of knowledge regarding the loads acting on the wing surface and corresponding commands. An interesting solution, proposed from Jacazio, Dalla Vedova, Maggiore, Sorli (2010), Mornacchi, Vignolo (2014) and Jacazio, Mornacchi, Sorli (2015) , is to exploit the pre-flight time to carry out the prognostic analysis integrating this new procedure with the pre-flight checks.

This solution has two interesting advantages: the first is the possibility of stimulating the EHSA with any kind of command, offering the possibility of developing commands that maximize the effect of degradation on the extracted features while the second is related to the loads acting on the wing surface. With the aircraft on the ground, the aerodynamic force depends only on atmospheric wind; therefore, it is small and does not affect the response of the servo actuator.

The strategy implemented in this work provides the stimulus, during the preflight operations, of the EHSA with a ramp command with 33 mm/s ratio and a max amplitude equal to 50% of half-stroke of the actuator.

4.1. Operational Scenario

The behavior of an actuator is strongly dependent on external conditions and the temperature of the hydraulic fluid. In order to simulate the EHSA in conditions as close as possible to those encountered in flight, a possible operating scenario has been suggested. This includes a series of flights within the European network and, for every situation of pre-flight conditions identified, starting from real data, the oil temperature and the average velocity of the atmospheric wind are accounted. The data are shown in the graph of Figure 2.

![Figure 2. Example of oil temperature and wind speed](image)

5. Feature Extraction

Feature or Condition Indicator (Cis) selection and extraction constitute the cornerstone for accurate and reliable fault diagnosis. The classical image recognition and signal-processing paradigm of data→information→knowledge becomes most relevant and takes central stage in the fault diagnosis case, particularly since such operations must be performed on-line in a real-time environment.

Fault diagnosis depends mainly on extracting a set of features from sensor data that can distinguish between fault classes of interest, detect and isolate a particular fault at its
early initiation stages. The remainder of this section evaluates a feature derived from the Hilbert transform to identify asymmetries because of turn-to-turn winding insulation faults.

A significant step in the development of robust and accurate PHM algorithms involves the extraction and selection of appropriate features or condition indicators from raw data. In our case, features are extracted using only the actuator position data and the servo-valve current. The analysis of the data obtained from the simulations has identified the first stage filter occlusion as a key indicator affecting two observable quantities.

Occlusion of the filter of the first stage of the servo valve leads to a lower response of the servo-command causing an increase of the time required for the actuator to reach the commanded position, as shown in Figure 3, with a consequent growth of the error between the command and the real position. The increase of the error leads to enhancement of the servo-valve current generated by the controller (Figure 4). Until the current of servo-valve is below its saturation level, the controller can compensate for the growth of degradation thus limiting its effects on the system. After reaching the saturation threshold, the growth of the position error no longer results in an increase of the control current and, therefore, the system is no longer able to compensate for the growth of degradation.

The data analysis has led to the definition of four different features that can be extracted by combining the acquired information. The features identified were then evaluated by appropriate metrics and one was selected and used in the prognostic algorithms. The features extracted from the data include:

- Mean error between real position and command; this is evaluated in the time range $[0.15\ 0.35]$ s in order not to consider the initial and the end portion of the response. The feature is calculated as shown in following equation, where $x_c$ is the command and $x_r$ is the real position:

$$\text{MeanError} = \text{mean}(|x_c(t) - x_r(t)|)$$

- Mean speed, defined as the average speed of the actuator in the time range $[0.15\ 0.35]$.

- The correlation coefficient between the position error and the current. In nominal conditions, there is a linear correlation between the current generated by the controller and the position error; this is due to the structure of the control logic, which provides a proportional part prevailing over the integral part. The presence of a degradation due to the saturation of the current and, consequently, to an increase of the error.

- Current fall time, defined as the time required for the current to return to a value less than 5% of its maximum value.

The values of the features functions of the occlusion of the servo-valve’s first stage filter, as shown in Figure 5.

![Figure 3. Influence of degradation on EHSA position](image)

![Figure 4. Influence of degradation on current](image)

![Figure 5. Features function of degradation](image)

### 5.1. Feature performance

The features were evaluated by means of appropriate metrics, which lead to the definition and utility of those features that more accurately represent the state of the system. The metrics are:

- Accuracy measure: defined as the linear correlation between the occlusion of the first stage filter and the feature.
• Precision measure: the percent mean deviation of the feature with respect to the interpolation line used to describe the feature as a function of degradation:

\[ PMD(x) = \frac{\sum_{i=1}^{n}(x_i - \hat{x})}{n} \cdot 100 \]

Where: x is the real value, \( \hat{x} \) is the interpolated value and n is the sample number.

• Moving correlation: defined as the linear correlation between the degradation and the feature inside a moving window. The window size is 100 points with 99 points of overlap.

5.2. Feature selection

The choice of which feature should be used in prognostic algorithms was based on two main considerations: the metrics and previous (historical) knowledge. The metrics shown in Table 1 and Figure 6, exhibit an acceptable average error and are chosen for further processing.

At an operational level, the average error proves to be the best feature, since mean error is strictly related to the nature of the EHSAs. An increase of the average error between the actual position and the commanded one is easily linked to a degradation of the system. Other features like the current file time and mean rod speed are also physically linked to the behavior of the actuator but were not included in the targeted set.

The functional diagram of the model is shown in Figure 7, where inputs and outputs of the simulation model are highlighted with the latter coinciding with the signals available on the EHSA. The EHSA degradation that can be addressed by the virtual hardware include:

- EHSV feedback spring degradation (partial yielding, backlash increase)
- Increase of radial clearance between EHSV spool and sleeve
- EHSV spool friction increase
- Torque motor degradation
- Progressive clogging of an EHSV nozzle
- Contamination of the EHSV inlet filter
- Increase of the friction of the actuator spherical bearing
- Actuator seals damage
- Change of sensitivity of position sensor

The servo-valve torque motor is modelled using the Urata (2007a) magnetic circuit shown in Figure 8. Applying the proposed equations is possible to express the torque generated as a function of the magnetic flux density of each air-gap.

\[ T = \frac{L_0 A_g}{4 \mu_a} \sum B_i \quad (i = 1,2,3,4) \]  (2)

where \( B \) is the flux density in the air-gap, \( L_0 \) is the distance between the left and right pole, \( A_g \) is the cross-sectional area of air-gap, \( \mu_a \) is the permeability of air.

The model also takes into account the influence of unequal air-gap thickness in servo valve torque motors, this is achieved by expressing the reluctance of the air-gap as function of air-gap thickness

\[ R_i = \frac{L_0 \pm H \pm W \pm G \pm x_{arm}}{\mu_a A_g} \quad (i = 1,2,3,4) \]  (3)
In Eq. 3 \( l_0 \) is the nominal thickness and \( x_{arm} \) is the armature position and \( H,W,G \) are the coefficients that allow to express the misalignment of the armature, signs depend on the air-gap considered, for a more details see Urata (2007b).

The torque obtained from Eq. 2 combined with the dynamic equations of the flapper. The position of the flapper causes a variation of the flow from the two nozzles of hydraulic amplifier, and a consequent change in the pressure of the chambers placed at the ends of the spool. In the model the relationship between the position of the flapper and pressures at the ends of the spool is modeling diversity with the following equations

\[
\begin{align*}
P_A &= G_{PA} \times (x_T - G_{QA}x_S) \\
P_B &= G_{PB} \times (x_T - G_{QB}x_S)
\end{align*}
\]  

(4)

where \( P_A \) and \( P_B \) are the pressures in the chambers, \( G_{PA} \) and \( G_{PB} \) are pressure gains and \( G_{QA} \) and \( G_{QB} \) are flow gains. Varying the value of the gains is possible to simulate contamination of the first stage filter or occlusion of one of the two nozzles.

The pressures determined by the equation (4) are used in the dynamic equation of the spool in order to estimate the opening of the flow ports. The equations that describe the kinematic system take into account the influence of the feedback spring force, coulomb and viscous friction and structural stiffness and damping. Furthermore, each parameter can be modified in order to simulate the degradation of the components.

The resulting servo-valve control flows, for each port, from the difference of the contributions from the supply and the return, the mathematical model calculates the individual contribution by exploiting the electrical similitude. Each port is represented as a circuit composed of two variable resistances placed in series \( R_c \) the laminar resistance and \( R_A \) the turbulence resistance.

\[
\begin{align*}
R_c &= \begin{cases} 
\frac{12\mu_{oil}(ol - x_s)}{2.5 \times w_d \delta^2} & \text{for } ol \leq x_s \\
0 & \text{for } ol > x_s
\end{cases} \\
R_A &= \frac{\rho_{oil}Q^2}{2Cd^2A^2}
\end{align*}
\]  

(5)

where \( ol \) is the overlap, \( h_r \) is the spool radial gap and \( w_d \) is width of servo valve port. \( A \) is the area of the servo valve port. \( \rho_{oil} \) and \( \mu_{oil} \) are density and absolute viscosity, respectively. \( Q \) is the flow passing through the port and \( Cd \) is the discharge coefficient function of Reynolds number and of the ratio between corner radius and port opening.

Using the value of the resistance, the model estimates the flow from each port with the equation (6)

\[
Q = \frac{-R_c - \sqrt{R_c^2 - 4R_A \Delta P}}{2R_A} \times \text{sgn}(\Delta P)
\]  

(6)

where \( \Delta P \) is the pressure drop between the port.

A 3-DOF model describes the hydraulic linear actuator (Figure 9): the first two describe the rod, the surface position, and the last represents the deformation of the attachment point between the actuator and the fixed structure. The actuator coulomb friction is a function of the dynamic condition of the rod and of the geometrical and physical data of the seal as well as the pressures in the actuator chambers (Martini, L. J. 1984).

![Figure 9. Actuator mathematical model](image)

The mathematical model allows simulating the aerodynamic load acting on the wing surface, comprised of the sum of four components:

- Airplane velocity
- Atmospheric wind, obtained by a normally distributed random number
- A random number generator determines wind gust, whose amplitude and duration. Gusts occur in a random pattern.
- Turbulence, implemented using the Dryden model (Yeager, 2008).

Oil properties, such density, viscosity and bulk modulus, are computed using a set of equations that are functions of oil temperature.

A merit of the virtual hardware is a detailed physical representation of each EHSA component, enabling the rapid change of parameters and the evaluation of the corresponding changes of the EHSA performance, thus allowing the assessment of the effects of single and multiple degradations.

6.1. Validation
The model validation was carried out using data acquired through experimental testing and includes frequency responses of the EHSA and system responses to different stimuli. As shown by the example of Figure 10, the response of the mathematical model to a 2 Hz sinusoidal command, output of the model is very close to the actual behavior, concerning both the servo-valve and the actuator. The validation was carried out only for the hydraulic servo-system in nominal conditions.

7. DEGRADATION DETECTION
In the paper two different methodologies to detect the degradation; the first is a data-driven approach based on the data and the features extracted from the data while the second uses the particle filter in order to estimate the state of the system and identify the presence of degradation. In both approaches, the presence of the degradation is detected by comparing the curve of the feature in nominal conditions (baseline) with that obtained at the observation time.

7.1. Diagnostic performance requirement
Customer specifications are translated into acceptable margins for the type I error and type II errors in the detection routine:
- False alarm rate: defined as the probability of a false alarm. It coincides with type I error and equal to 5%
- Confidence: coincides with 100-Type II error [%] and it expresses the level of confidence with which a degradation is detected. This work is set to 95%.

The algorithm itself will indicate when the confidence level, defined as 100-Type II error [%], has increased to the desired level.

7.2. Data driven approach
The data-driven approach takes advantage of the data acquired during each stage of pre-flight. The histogram of the feature distribution that approximates the pdf curve is realized by using a moving window of 50 acquisitions with an overlap of 49. The baseline is achieved with 50 acquisitions made on the actuator in nominal conditions.

Figure 11 shows an example of degradation detection occurring after 698 flight hours and in the presence of an occlusion of 19% with an accuracy of 97%.

7.3. Particle filter approach
This approach exploits the particle filter framework to estimate the probability distribution of the extracted feature at each time instant for degradation detection purposes.

Applying the particle filter the system equations reduce to the form of Equation 7 allowing to estimate the occlusion of the first stage filter (coincident with continuous-valued states $x_c$) and the pdf curve of the feature mean error.
In equation 7: \( f_{\text{ob}}, f_{\text{c}}, \) and \( h_{\text{i}} \) are non-linear mappings, \( x_{d,1} \) and \( x_{d,2} \) are Boolean states that indicate normal and faulty conditions, respectively, \( x_c(t) \) is a set of continuous-valued states that describe the evolution of the degradation those operating conditions, \( dt \) is the interval between \( t \) and \( t+1 \), \( h \) is delta for numerical derivative and \( \omega(t) \) is noise describe like a normal distribution with zero mean. \( v(t) \) \( v(t) \) is a normal distribution noise with mean equal to zero and sigma equal to the accuracy of the acquisition system, estimated as the sum of the position transducer error, position transducer demodulator error and A/D converter error. The initial conditions of the equation system (Eq. 7) are \( x_c(0) = 0 \), \( x_{d,1} = 1 \) and \( x_{d,2}(0) = 0 \).

The non-linear mappings \( f_{\text{c}} \) and \( h_{\text{i}} \) are functions that express the occlusion of the filter of the servo valve as a function of flight hours and the feature as a function of degradation, respectively. The equations are obtained using a symbolic regression tool, which starts with a set of data to identify the best fitting approximation. The selected functions for both cases offer the best compromise between accuracy of fitting and simplicity of the model, reducing the calculation time of the implemented algorithms.

The function \( f_{\text{c}} \), obtained using the symbolic regression is:

\[
x_c(t) = a + b \cdot t^2 + c \cdot t^3 + d \cdot t^4
\]  

(8)

Where \( a, b, c, d \) are time-invariant coefficient and \( t \) is the flight hours. Figure 12 shows the comparison between the data and fitting curve, Table 2 shows the parameters related to the accuracy of the fitting.

![Figure 12. Occlusion model](image)

### Table 2: Occlusion model accuracy

<table>
<thead>
<tr>
<th>R2</th>
<th>0.99993</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation coefficient</td>
<td>0.99996</td>
</tr>
<tr>
<td>Mean squared error</td>
<td>0.0436</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.1725</td>
</tr>
</tbody>
</table>

The function that represents \( h_{\text{i}} \) expresses the feature as a function of the occlusion of the filter:

\[
x_c(t) = e + f \cdot x_c^2(t) + g \cdot x_c^3(t)
\]  

(9)

Where \( e, f, g \) are time-invariant coefficient.

The results of symbolic regression are reported in Figure 3 and in Table 3 where the parameters related to the accuracy of the fitting are shown.

<table>
<thead>
<tr>
<th>R2</th>
<th>0.99834</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation coefficient</td>
<td>0.99928</td>
</tr>
<tr>
<td>Mean squared error</td>
<td>0.0022</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.0319</td>
</tr>
</tbody>
</table>

(figure)

### Table 3: Feature model accuracy

7.4. Results

The detection algorithms were tested using ten simulated responses of the EHSA. The simulations were carried out by injecting diverse operational scenarios and different travel patterns within the European network; the EHSA behavior has been simulated with different environmental conditions and temperature profiles of the hydraulic fluid variables.
The average time to detection is 428 flight hours, which corresponds to an average occlusion equal to 19%.

The algorithm based on the particle filter exhibited better performance; it was able to identify the degradation in all conditions. Even the average time to detection is significantly lower; it is approximately 400 flight hours, coincident with about 16% of filter occlusion.

8. Prognostics

Prognosis is understood as the generation of long-term predictions describing the evolution in time of a particular signal of interest or fault indicator. In the work presented in this paper, predictions are based on an estimate of the evolution of the features’ mean error. Its evolution is predicted using the particle filter presented previously, in particular the system of equations (7). The approach employs the previous state estimate to generate the a priori state pdf estimate for the next time instant.

We define by $t_f$ the instant at which fault detection occurs, the particle filter then uses the pdf estimates for the system continuous-valued states $x_c(t_f)$, computed at the moment of fault detection, as initial conditions for the failure prognostic routines.

By using the state equation to represent the evolution of the fault dimension in time (Eq. 3), it is possible to generate a long-term prediction for the state pdf, in the absence of new measurements, then use the predicted states $x_c$ to estimate the resulting evolution of the feature.

The algorithm terminates the prediction when the estimated feature pdf for a given point in time completely surpasses the set threshold.

The limit, beyond which the component is considered failed, and thus needs to be replaced, has been estimated to be equal to a mean error of 4 mm.

Figure 15 shows example results of the RUL estimation; the end of life (EOL) of EHSA is 986 flight hours, which corresponds to a remaining useful life, defined as $RUL = EOL - t_d$, equal to 313 flight hours.

8.1. Performance metrics

The performance of the prognostic algorithm was evaluated using the metrics proposed by Saxena, Celaya, Balaban, Gobel, Saha B, Saha S, and Schwabacher (2008). In particular, the following metrics were used:

- Prognostic horizon ($H(i)$): defined as the difference between the current time index $i$ and the end of prediction (EOP) utilizing data accumulated up to the index $i$, provided the prediction meets desired specification. $H(i) = EOP - i$

- $\alpha$-$\lambda$ performance: which allows to verify that the prediction to a generic $\lambda$ instant has an accuracy $\alpha$. $(1 - \alpha) \ast r(t) \leq r_p(t) \leq (1 + \alpha) \ast r(t)$

Where $r_p$ is the predicted RUL at time $t$, $r$ is the real RUL, $\alpha$ is the accuracy and $i$ is defined as $t = P + \lambda(EOL - P)$

Where $P$ is the first prediction time instant and $\lambda$ is window modifier.
• Relative Accuracy (RA): relative prediction accuracy at a specific time instance. Perfect score is RA=1.

\[ RA(t) = 1 - \frac{|r(t) - r_p(t)|}{r(t)} \]

Where \( r_p \) is the predicted RUL at time \( t \), \( r \) is the real RUL and \( t = P + \lambda(EOL - P) \)

Where \( P \) is the first prediction time instant and \( \lambda \) is window modifier.
• Cumulative Relative Accuracy (CRA): normalized sum of the relative prediction accuracies. Perfect score is CRA=1.

\[ CRA = \frac{1}{EOL - P + 1} \sum_{t=0}^{EOL} RA(t) \]

8.2. Results

The prognostic algorithm has shown good results with all ten data sets used; in all cases, the metrics demonstrated the robustness and accuracy of the algorithms. The mean prognostic horizon of the algorithm is equal to 292 flight hours. Example results for the metric \( \alpha-\lambda \), shown in Figure , demonstrate the accuracy of the algorithm in the estimation of the RUL, with the estimated value always within the limits of 20% for all data sets.

Similarly, the RA and CRA metrics have demonstrated the accuracy of the algorithm in estimating the useful life: for all the data sets the 95% of the RA value is inside the range 0.98 to 0.88, the minimum value is equal to 0.85. Figure 17 exhibits the trend of the metric in the case of a single data set. The average value of CRA, for the ten data sets, is equal to 0.943. The best CRA is 0.948 while the worst is 0.941.

9. CONCLUSIONS

This paper presented a particle-filter based fault detector capable of detecting the occurrence of a major fault mode in its incipient stages for a safety critical aircraft actuation system. Furthermore, the same basic estimation method was adopted for prediction of the remaining useful life of the actuator. An overview of a generic PHM architecture was presented and applied to a particular EHSA fault mode based on a FMECA study. The primary fault mode was modeled using physics-of-failure mechanisms indicating the primary failure effect. A feature was derived using statistical analysis to quantify the primary failure effect. Then, simulation data were acquired to validate the model. Although a specific system, an EHSA, was selected as the test-platform with a specific fault mode, the overall PHM architecture can be applied to an entire range of systems and application domains. In fact, similar techniques, which allow for early fault detection with acceptable performance in the presence of faults, are being developed for a wide variety of system actuators in both manned and unmanned air vehicles. Therefore, the concept of using system health information (diagnosis and prognosis) is at the forefront of modern space and avionics applications requiring increasingly sophisticated diagnostic and prognostic systems that are robust, reliable, and relatively inexpensive.

ACKNOWLEDGEMENT

This work was supported by Lufthansa Technik AG; we appreciate their contribution to the development and validation of the mathematical model by providing to the research team actuators and technology applied to test benches.

REFERENCES


Pohl, T., (2013). Cost per hour of downtime per aircraft is ~ 10,000 USD+ more, in *SAP for Aerospace & Defense*


**Biographies**

**Mornacchi A.** is a mechanical engineer and a PhD student in mechanical engineering. His primary interest is in the areas of prognostic for aerospace servosystems and modelling and simulation of hydraulic and electromechanical servos. He is a member of the servosystems and mechatronics group of the Department of Mechanical and Aerospace Engineering.

**Jacazio G.** is professor of applied mechanics and of mechanical control systems. His main research activity is in the area of aerospace control and actuation systems and of prognostics and health management. He is a member of the SAE A-6 Committee on Aerospace Actuation Control and Fluid Power Systems, and a member of editorial board of the international society of prognostics and health management.
Abstract

Synthetic aperture radars are complex, data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com

1. INTRODUCTION

Synthetic aperture radars are complex data-generating systems that can form intrinsically detailed images of the current environment in which the radar system operates. The deployment and testing of these systems, specifically on an unmanned platform, can be costly and therefore it is imperative to determine if the system is operating as anticipated. If not, the mission may be at risk and therefore it is important to detect the system's operating state. It would be a significant waste of resources if an unmanned platform can be costly and therefore it is imperative to determine the system's operating state.

SYNTHETIC APERTURE RADAR (SAR) PAYLOADS

Application of Health Management and Diagnostics for Synthetic Aperture Radars

Gregory Bower, Curtis Wrable, Ross Bird, and Paul Woodford

QorTek Inc., Williamsport, PA, 17701

gbower@qorTek.com

cwrable@qorTek.com

rbird@qorTek.com

KEYW Corporation, 7763 Old Telegraph Road, Severn, MD 21145

pwoodford@keywcorp.com
is to define the model such that the model reflects changes occurring in the system itself due to degradation and usage.

In regards to the current application, the availability of vast amounts of radar phase history data is ideal training data for a data-driven approach. This paper focuses on the further development and deployment of the algorithm and contains results from flight-testing the system in real-time with the radar in operation. The objective was to demonstrate the system operating with the radar, obtaining measures of the quality of the radar data in near real time, and to not interfere with the radar’s primary function.

The paper begins with an introduction into Symbolic Analysis. This brief review can be augmented with our previous paper [Bower, 2014]. Section III describes how the SAR data is utilized in the algorithm described in Section II. Examples of training and test data are shown. Section IV investigates the training and testing of the algorithm on the RFI datasets provided to QorTek. Section V details the flight-testing including the hardware design and results obtained. The paper is concluded with Section 5 discussing the results and future work to be accomplished.

2. Problem Background

Expanding upon the work presented in [Bower, 2014], the objective was to design a system that was capable of integrating the developed software into the radar system. The software, Symbolic Analysis (SA), is briefly reviewed here.

2.1. Symbolic Analysis

Symbolic Analysis is a statistical pattern recognition tool based upon symbolic theory. Most work in the symbolic realm deals with the development of optimal models to determine the trajectory of modeled system states (Daw, Finney & Tracy, 2003). These methods are used to model complex and chaotic systems. The resultant optimal model, known as the ε machine, has a variable dimensional structure whose dimensions were constantly adjusted depending on the data collected over time. This variation in dimensionality made it difficult to determine deviations between models developed through system usage. In order to make meaningful comparisons between models, a machine was developed with a-priori fixed dimensional structure (Ray, 2004). This fixed dimensional machine allows for meaningful comparisons between statistical models defined at different temporal points in the system’s life at the cost of optimality. The process of SA is shown in the block diagram of Figure 1. Each process will be briefly described.

2.2. Data Capture

The data capture is an important step as it identifies data sources that are related to underlying degradation signatures. In addition to identifying relevant observables, the SA approach requires two assumptions: 1) the system does not undergo ‘self-healing’, and 2) that the underlying degradation dynamics can be separated from the system dynamics. Assumption 1 forces the system to undergo a monotonically increasing degradation state, which assists in predicting future failure. Assumption 2 is far more stringent in that the data captures must be sufficiently long enough to develop statistics but also not capture changes in the underlying dynamics of the system. In other words, it is assumed that the degradation dynamics evolve at different rates then the system operates. This simplifies into a separation of scales assumption of which the SAR indeed fulfills as data rates are significantly higher than degradation signatures.

2.3. Symbolization

The next step involves transforming the time series data into the symbolic domain. This step can be thought of as a general re-quantization of the original data resulting in a coarser distribution. Symbolization requires the determination of the number of partitions to be used as well as the type of partitioning. The two most common types of partitioning include uniform partitioning (UP) and maximum entropy (ME) partitioning. QorTek has devised a new partitioning approach which combines the advantages of both UP and ME which is called Mixed (MX) partitioning.
2.3.1. Partitioning

That partitioning scheme of the algorithm allows for the collected data to be converted into the symbolic space. As stated previously, there are three main approaches utilized for this work including UP, ME, and MX although the majority of this write-up will focus on MX partitioning.

Uniform partitioning divides the range of the time series data into equal sized regions where the total number of determined partitions are defined as the set \( P \). Given the range of the time series data as \( U \), the partition sizes are defined as \( U/P \) and the boundaries developed from the range \( U \). Each partition region \( P_i \) was mutually exclusive and exhaustive over the range of the data. The probabilities of the partition occurrence in the uniform case are not necessarily equal; however, the partitioning structure was equal.

The maximum entropy (ME) partitioning scheme was defined by the principle of entropy in determining the partition structures. Recall entropy as shown in Eq. 1.

\[
H(X) = - \sum_{i=1}^{n} p(x_i) \log_2 p(x_i)
\]  

(1)

The entropy can be maximized by setting \( p(x_i) = p(x_j) \), \( \forall i,j \). The logarithm to base 2 was used so that the unit of entropy is in bits. In the time series data, accomplishing maximization of entropy in the baseline case was necessary to make sure all partitions (or symbols) have equal probability of occurrence. The partition structure resulting from ME does not necessitate equal partitions as in the uniform case but does guarantee equal prior probabilities for the partitions in the baseline case. A feature of the ME partitioning scheme is that the partitions boundaries are closer in regions of the data where there are a dense number of data points. In regions where there are fewer date points, fewer partitions are generated in these areas.

Mixed partitioning was developed at QorTek to combined the sensitivity of the ME approach with the equal area distribution of the UP approach. In the work completed, the MX approach takes the desired number of partitions and divides them equally between those to be developed through ME and those to be developed under UP. The resultant MX partitioning approach very finely models regions of dense data and uniformly divides other regions to allow for evolution of the system. This became much more important when the partitioning was converted into two-dimensions. This also enables to algorithm to model slight changes in the data-dense regions while allowing for more significant system deviations through the UP partitions.

Once the partitions are defined each partition was labeled with a symbol from the alphabet \( S \). Given a time series \( X \) of length \( M \), if \( x_i \in P_i \), \( 0 \leq i \leq M \), then assign \( s_i \rightarrow x_i \), \( \forall i; s_i \in S \). By implementing the partition structure and assigning a unique symbol to each time series date point, the end result was called the symbol stream. This is the re-quantized time series data that is now transformed into the symbolic domain.

2.4. Statistical Analysis

Once the partitions have been developed and symbols assigned to each partition, the next step is to construct the statistical model based on the resultant symbol stream. This step is controlled by another parameter for the SA methodology, the depth parameter \( D \). The depth parameter controls the definition of model states from the symbol stream. States in the model are formed from \( D \)-length subsets of symbols. Therefore, the total number of states in the algorithm given the number of partitions \( P \) and the depth \( D \) is shown in Eq. (2).

\[
N_s = P^D
\]

(2)

Equation (2) holds true independent of the partitioning scheme utilized. As an example, assume a ternary partition scheme is implemented that results in three symbols; labeling them -1, 0, and 1. The methodology’s resultant statistical states depend on the number of symbols in the algorithm as well as the chosen depth. The parameter depth adjusts the memory of the resultant symbolic model, that is, the parameter controls the groupings of symbols into states. For instance, if \( D \) was unity, the resultant states are 0, 1, and -1. If \( D \) was two, the resultant states would be 00, 01, 10, 11, 0-1, (-1)0, (-1)(-1), 1(-1), and (-1)1 according to (2).

Shown in Figure 2 is an example model formation with the three partition symbolic system and with \( D \) being equal to two applied to a recorded sine wave of arbitrary amplitude. These parameter choices result in a model with three states. The example sine wave in the figure is divided into zero (0), one (1) or minus one (-1) by a set threshold (partition boundary, uniform in this example). The symbol sequence is the square wave in the figure.
With the symbol sequence $s_i$ completed, the next step is to form states out of the symbols or groups of symbols. The probabilities of the state occurrences can be calculated and tracked across each data capture. Counting state occurrences can then be converted into probabilities to generate what is known as the State Probability Vector (SPV): the probabilities are arranged in a $N_s \times 1$ vector, where $N_s$ represents the total number of states in the algorithm given by Eq. (2). In the case where depth of the algorithm is equal to unity, as it is in most cases, the total number of states is equal to the number of symbols used. Choosing $D$ equal to unity results in the smallest possible model for a given number of symbols, thereby reducing computational complexity of the approach.

Once the probabilities or counts are known, a distance type metric can be applied to the baseline case and future cases to develop an anomaly based on the current system operation.

2.5. Anomaly Quantification

Anomalies inherent to degradation in the system can be generated from the use of the SPV between the data captures. The metric quantifies the deviation between the known baseline, commonly known as the healthy state of the system, and a future system state. A measure commonly used to quantify an anomaly between captures is based on the Euclidean distance given in Eq. (3) for pulse $j$.

$$A_j = \sqrt{\sum_{i=1}^{N_s} (z_{i,\text{nominal}} - z_{i,j})^2}$$

In Eq. (3), $z_{i,\text{nominal}}$ is the nominal (baseline) SPV state $z_i$ and $z_{i,j}$ is the corresponding SPV state at iteration $j$. The Euclidean distance measure is used as it provides a straightforward means to measure the change between SPVs in an analysis. From this measure, it is possible to quantify anomalies present in the system and how they evolve over time and usage.

From this evolution of the anomaly, it is then possible to define a threshold of failure for the system. The threshold can then be implemented in a predictor to estimate remaining useful life of the system.

The anomaly can be used as a diagnostic measure to determine the amount of degradation the system has incurred over its lifetime or to be used as a prognostic measure. If training data exists for the system, the anomaly measure can then be used in a prognostic application to predict the remaining useful life of the system.

3. Phase II Focus and Development

The Phase I program investigated applying the algorithm to previously recorded good SAR data and testing the response of the algorithm to artificially inducing degradation effects such as adding in RFI. The results were promising which thus enable QorTek to further investigate the application of the algorithm.

Based on the results of the Phase I program, the Phase II objective was to further refine the approach and apply it to ‘real world’ radar data collections that could contain varying degrees of data degradation. In order to accomplish this task, QorTek teamed up with KEYW Corp., manufacturer and supplier of SAR radar systems for imaging applications. Utilizing this data, the algorithm was then tested with typical situations that a SAR radar would be involved with, namely RFI, scene variations, and hardware faults. These situations were used to train the algorithm and develop the necessary parameters that will be hardcoded into the algorithm for system implementation and operation.

An interesting aspect of radar data is that it can be contained within the frequency domain thereby providing both frequency and phase information. The original work solely focused on the frequency amplitude information ignoring the phase information. For this work, the phase information was also utilized and integrated into a two-dimensional implementation of SA where the space the algorithm operated on was now frequency magnitude and phase. This resulted in two-dimensional partitioning scheme as illustrated in Figure 3. The figure shows an example data distribution with the MX partitioning employed. Notice that there are ME partitions in the data dense regions of data with uniform partitions extending out from these ME partitions. There are an equal number of ME and UP partitions. The magnitude range of the ME partitions covers a single standard deviation of the phase history data. The remaining range was divided with UP. Note that the figure shows the I/Q representation of that data, that is:

$$d_i = a + jb$$

In (4), $d_i$ is the data point defined by the complex number formed from $a$ and $b$. Therefore, recall that the magnitude and phase of the data is given as:

$$|d_i| = \sqrt{a^2 + b^2}$$

$$\Phi_{d_i} = \tan^{-1} \left(\frac{b}{a}\right)$$

To obtain phase information, the regions were further divided either into two or four regions. The approach was limited to four regions in order to minimize the complexity of the model and to implement it efficiently into a GPU device.
This two-dimensional approach to partitioning was also implemented completely with ME and UP approaches as well. The UP case presents a unique situation. The definition of 'uniform' in this case can be ambiguous. For the present case, two approaches were developed. These cases consisted of the uniform partitions being created either through uniform radius or by uniform area in which each partition has equal area. Examples for the two types of UP are shown below. The first figure, Figure 4 shown below, demonstrates the UP approach involving uniform radius circles. As can be observed, each concentric ring is equidistant from each other.

The following figure, Figure 5, shows the results of UP when employing equal area circles. This results is a markedly different partitioning structure. This type of UP would group the majority of data within the first or second partition region. The remaining regions would be used to identify outlier data of which RFI is typically manifested.

3.1. Application to Radar

The choice of underlying partitioning depends, in part, on the degradation signatures involved in the process. In this work, the data processed by the algorithm is the phase history data of the radar. The phase history data is the frequency transformed time series sampled data of the radar returns. The resultant phase history data is truncated in the frequency domain to contain data within the bandwidth of interest.

Each pulse sent out by the radar results in a data stream related to the previously transmitted pulse. This results in a single pass containing many thousands of pulses containing significant amounts of data which can be utilized by the SA algorithm. An example distribution of the phase history data can be seen in Figure 3. An example of the collected magnitude of data is shown in Figure 6. The plot shows the magnitude of the received frequencies against each individual pulses. There is interesting RFI that can be observed in the plot.

The SA algorithm is implemented on a pulse-by-pulse basis with the first pulse acting as the baseline for the entire pass. Each processed pulse would result in an output from the SA algorithm producing a plot for the entire pass. The output can then be used to diagnose the pass to determine if the system was operating as expected.
For diagnosis of the radar, two areas were of importance. First, the algorithm was to detect if any hardware faults had occurred within the radar and secondly, to detect pulses or passes that were significantly degraded by radio frequency interference (RFI). RFI detection will be the focus of this paper as it was the most common issue involving the radar system.

4. Radio Frequency Interference Detection

RFI is a problem with SAR systems in that the final images can be corrupted due to this additional energy in certain frequency bands (Meyer, 2013)(Lord, 2005). Cell communications, TV broadcasts, and satellite transmission can all degrade the operation and final results of a SAR pass. The environment RFI can easily overwhelm the weaker SAR reflected signals. Our goal is to detect RFI issues that are contained within the phase history data without the need for image generation which can be computationally intensive. The algorithm also must not affect the operation of the radar and must minimize false positives in its output.

Detection of RFI is a challenge for numerous reasons. A straightforward example would be attempting to detect RFI by return magnitude while attempting to reject scene reflectivity changes. Datasets were given to QorTek by KEYW to explore and develop/train the algorithm on typical examples of RFI and scene reflectivity changes. A pass was completed in which the radar was pointed away from sources of RFI and then pointed towards the sources. This additional energy cause a degraded final processed image. Note that the cleaner pass still contains traces of some amount of RFI. This data was then utilized in the training of the algorithm with the results contained in the next section.

4.1. Algorithm Application to RFI

In the development of the algorithm, the data described previously was used to train the algorithm and determine the parameters (partitioning and depth) that would most efficiently be used to detect RFI. This data allowed us to further develop the two-dimensional partitioning and evaluate its capabilities beyond magnitude partitioning alone which was accomplished in [Bower 2014]. In this paper, the following results are generated utilizing the SPV and two-dimensional model formation.

The set of results utilizing the RFI corrupted data is given in Figure 7. The parameters used in the generation of the results utilized 32 magnitude partitions and four quadrant partitions. Implementing a depth of unity to minimize model complexity yields 128 model states. The baseline was chosen in the upper plot of Figure 7 used the second pulse. The partitioning approach taken was derived from the MX (Figure 3) partitioning approach. The top figure (a) relates to the pass with minimal RFI and the lower figure (b) is the pass with substantially more RFI.

Not the change in magnitude between the upper plot and the lower plot. The upper plot shows an average value around 0.04 whereas the lower plots average value is nearer to 0.01. There are some variations from pulse to pulse as would be expected and some variation from scene reflectivity although this is minimal. The larger variations observed in the lower figure are most likely due to changes in received RFI power.

Utilizing the exact same data, the results for UP and ME are shown in the following two figures. The partitioning parameters were also kept constant for the UP and ME results. Both instances utilized 32 partitions. The UP was implemented with equal area regions. Four quadrant partitioning was used and the depth parameter of the SA algorithm was set to unity.

Figure 8 demonstrates the results when equal area UP was implemented. Since UP results in a system that is not as sensitive to noise in the system, the output of the algorithm contains less variation then the output with MX partitioning. The lower figure’s trending also follows that obtained with MX partitioning although the spike around pulse 6,000 is absent. This is due to the sensitivity of the MX partitioning as compared to UP. The underlying trend though similar to that obtained with MX partitioning as well.
The results in Figure 9 were generated from ME partitioning again using the same parameters for comparisons between the other two partitioning methodologies with the same radar field data. As was stated in the SA background, ME results in the most data sensitive partitioning approach and this can be observed in the results. Note the significant increase in pulse-to-pulse variation as compared to UP and MX. We originally believed this increase in sensitivity would assist in detecting degradation early, but the sensitivity to noise and data variations make it a challenge to use in typical situations. This is the primary reason for the development of the MX partitioning strategy. Again, the spike observed in the lower figure around pulse 6,000 is visible as with the case of MX partitioning.

For example, changes in the scenery reflectivity can act as a false alarm of RFI as the algorithm begins to notice the change in input data. KEYW was gracious enough to supply QorTek with an example of this event. The dataset was obtained from a collected pass that transitioned into land. An example of this is the Webster Field pass of which the developed SAR image is shown in Figure 10.

The interesting scenery features that can cause the issue is with the radar transitioning from the bay/water onto land. Water and land have very different radar reflectivity, so the statistics of the collected radar data change significantly when the illuminated scene changes from mostly water to mostly land. Changes in land use, for example from rural to urban, can also generate false alarms. Figure 11 shows the results using the CBBT data and the ME partitioning approach. The figure shows some variation in the algorithm output with the occasional pulse outlier. An interesting strong signal is detected just before pulse 200,000. This is due to a strong reflector in the scene which was identified as a ship.
Figure 12 follows the ME results with the MX partitioning methodology. The results, as expected, are very similar to the ME results due to sensitivity of both approaches. This indicates that the data is mostly focused within the ME partitions within the MX partitioning. Again, the same strong reflector is observed in this figure as was observed in the previous figure. The output magnitude is also different due to the change in generated statistics between ME and MX partitioning. The final figure, Figure 13, shows the results utilizing the uniform partitioning approach with equal area regions.

The results in the figure show some interesting features. First, the strong reflection received in the pulses before pulse 200,000 is clearly visible within the figure. In addition, the output increases towards the end of the pass. This increase is due to the radar platform moving from sea to land with a general change in scenery reflectivity. This is an interesting result as the two previous approaches did not clearly show the change in reflectivity. The reduced sensitivity of the UP approach allows this detail to be more pronounced than in both the MX and ME partition implementations.

Observing the above three different results for each type of partitioning is the reason why the algorithm approach for the radar integration will utilize all three partitioning strategies. Each approach has its benefits and can detect different features as detected from the scene. This information can then be used to further diagnose issues with the data and/or radar payload. The question of whether it was scenery change or RFI can be determined by utilizing all three approaches. A strong reflector/RFI is detected by all three approaches whereas scenery changes are more clearly indicated by UP. To determine the difference between a strong reflector or RFI, the approach requires the use of the state transition matrix (Bower 2014) and frequency information (band) to determine if it is RFI or a strong reflector. Indeed, a scenery change is very gradual as compared to a strong reflector/RFI.

The above practical examples allowed us to refine the algorithm and prepare it for initial prototype testing on the radar system.

5. Prototype Flight Testing

A primary goal of the Phase II program was to develop the hardware and software systems needed to integrate the system onto a SAR platform. Given that the system would be involved in significant data processing, the hardware platform chosen for the process was GPU based to allow for a highly-threaded implementation of the algorithm.

A high-end gamer PC system was utilized for the work to achieve the necessary throughput needed to keep up the radar. The system consisted of a Core i7-4770k with 16GB system RAM and an NVidia GTX670 with 2GB of onboard GRAM. The entire system was inserted into a 2U-compatible 19" rackmount system (Figure 14). The SA algorithm was written in CUDA/C and the support routines written in C#.

Figure 13. Algorithm output using 32 UP equal area regions with four quadrants on CBBT dataset.
The system was designed to allow for expansion specifically in the area of storage in case it was needed. The power required was also carefully designed to be within specification of the available power onboard the aircraft. The most significant power draw was the GTX670 which at full power can consume 170W. The CPU load was minimized in order to reduce the total power required by the added hardware. To further reduce power and the chance of problems, a 256GB solid state drive was utilized as the main hard drive for the test setup.

The hardware was then installed into the radar rack system (Figure 15) and connected into the system via 1000BASE-T network. The gigabit connection was necessary in order to download and process the radar data as it was being recorded and to enable the system to keep up with the radar.

The goal of this flight test was to demonstrate that the system can be integrated into the radar payload without affecting its operation or causing any other atypical operation, and to also demonstrate the capabilities of the SA algorithm in real time. It was discovered that the instantaneous output of the SA could be used for RFI detection during flight.

The flight path was through the Naval Air Station Patuxent River area. The flight tests took place on July 1st – 2nd, 2014 and consisted of a couple of passes. The testing was carried out for a pre-mission systems check and verification by KEYW. QorTek joined the KEYW team to test out the diagnostic hardware and software.

The first day of flights went as expected with the diagnostic hardware without any major issues. The results shown below implement 32 UP regions utilizing equal area. This partition structure was chosen to be implemented in the first test run as it performed equally well in all situations during algorithm development. The next revision of the SA algorithm code will include the other partition types.

The first example of results is shown in Figure 16. This data was taken from pass 212616 from the L-band with VH polarization. The data connection that QorTek had access to was the horizontally-polarized receive portion of the data. All of the following results were thus taken from the *H available polarization channels.

The results show minimal issues in the data with the exception of a single pass outlier around pulse number 6,000. A more interesting example is shown in Figure 17 and Figure 18 taken from the second day of testing.
The results shown in Figure 17 were taken from the second day of testing and from the PHH channel of data. The majority of data is indicated as normal until around pulses 7,500 – 8,500. The algorithm output begins to increase and decrease back down to the expected value. This is indicative of RFI or some other corrupting signal present in the data. To determine if this RFI was limited to the P-band, the next figure contains the results for the L-band channel.

Figure 18’s results do not show the same behavior during those periods indicating a P-channel related RFI event that has occurred during the data collection during the pass. In this case, the event was isolated mostly to the P-band radar. The source of the RFI was unknown in this case but occurred at a frequency of 425MHz and was periodic. It was also confirmed to be absent from the L-band data.

The above two figures are an excellent example of RFI. Note that nearly all passes are going to contain some amount of RFI which is what introduces some of the pulse-to-pulse variability in the output.

The detection of RFI in one band and not in the other was an excellent indicator of algorithm performance for QorTek. Some issues that occurred during testing included long processing times needed to pull data from the data arrays on the radar. This problem is being addressed for the next flight test.

6. CONCLUSION

The SA algorithm has been shown to indicate possible issues with the operation of a Synthetic Aperture Radar (SAR) payload through statistical observation of the phase history data. Of importance to the program was to identify the type of degradation, whether it is internally based due to hardware degradation or externally, environmentally sourced. The data shown contained RFI issues, the most common problem with these systems. The algorithm was trained and tested on these example data sets as provided by KEYW and then implemented into a flight test.

The flight testing of the prototype was a success in that it was able to obtain and process the radar phase history data from the radar while it was in operation. The algorithm itself did not negatively impact the operation of the radar and the results produced interesting features.

The next goal of the program is to identify the RFI to assist in radar operation. In addition, the algorithm will continue to be used to also detect the possibility of hardware problems. KEYW has provided example data sets that contain hardware problems that were also trained and tested in the algorithm. Although much more uncommon, these degradation features will also be programmed into the final algorithm.
implementation. The final objective of the program is to complete a second flight test of the improved algorithm that decreases its execution time and reduces the size, weight, and power impact to the host payload.

ACKNOWLEDGEMENT

This work was support by an AFRL contract FA8650-13-C-2318. QorTek would also like to thank Thierry Pamphile AFRL Wright-Patterson for his support and guidance during this work. QorTek would like to acknowledge the support from NAVAIR for the permission to test the prototype on a flight and for the data presented in this work.

NOMENCLATURE

\begin{align*}
A & = \text{anomaly} \\
D & = \text{symbolic depth} \\
d_i & = \text{complex data point} \\
H(\cdot) & = \text{entropy} \\
MX & = \text{mixed partitioning} \\
ME & = \text{maximum entropy} \\
N_i & = \text{number of states} \\
p(\cdot) & = \text{probability} \\
P_i & = \text{\textit{i}th partition} \\
s_i & = \text{\textit{i}th symbol} \\
RFI & = \text{radio frequency interference} \\
U & = \text{time series data amplitude range} \\
UP & = \text{uniform partitioning} \\
X & = \text{time series data} \\
z & = \text{state probability vector}
\end{align*}

REFERENCES


BIographies

Gregory Bower is currently the CTO of QorTek Inc. in Williamsport, PA. He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering all from The Pennsylvania State University. Previously, he had worked as an assistant with the Applied Research Laboratory in State College, PA. His research interests include Prognostics and Health Management (PHM) of electronic systems, robust and optimal control theory, system identification, and power conversion.

Curtis Wrable is currently an Integrated Systems Engineer at QorTek Inc. in Williamsport, PA. He received his B.S. in Electronics Engineering from Pennsylvania College of Technology in 2009. Over the past six years, Curtis has worked on a number of different research areas over the course of his career including: analog and digital design, piezoelectrics, power electronics, and automated testing systems. His research interests include the areas of PCB layout and design, development of embedded systems, and generating software solutions to critical engineering problems.

Ross Bird is currently the President of QorTek, Inc. He received his B.S. in Electronics from Penn State in 2001, he then received his MSEE in 2003. Over the past decade Mr. Bird has worked extensively at the leading edge of power electronics design (holding several patents and additional pending in this technology); that incorporate advanced materials, design and digital control. He has extensive experience in the design and development of advanced power modules and has been lead developer of such power. His research interests include the areas of digital design and microcontrollers, digital implementation of complex numerical algorithms and mixed A/D PCB layout and design.

Paul Woodford is a Principal Research Engineer at KEYW Corporation in Severn, MD. He received the B.S. degree in electrical engineering from Bucknell University in 1989 and the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University in 1991 and 1995, respectively. He worked at Essex Corporation from
1995 to 2007, and at Northrop Grumman Corporation from 2007 to 2011. His interests are in the areas of synthetic aperture and moving target imaging and exploitation. He holds two U.S. patents for radar signal processing.
Variable Selection and Indices Proposal for the Determination of an Aeronautic Valve Degradation

Ananda S. Ribeiro¹, Takashi Yoneyama², Rodrigo F. Souto³, and Wallace Turcio⁴

¹,³,⁴ Embraer S.A., S.J. dos Campos, SP, 12227901, Brazil
ananda.ribeiro@embraer.com.br
rodrigo.souto@embraer.com.br
wturcio@embraer.com.br

² ITA – Instituto Tecnológico de Aeronáutica, S. J. dos Campos, SP, 12228900, Brazil
takashi@ita.br

ABSTRACT
This work presents a method to select parameters and use them in an index to identify the degradation level of an aeronautic valve of the type PRSOV (Pressure Regulator and Shutoff Valve), a valve of the bleed air system of aircrafts. This index function is to be used as an input to the valve controller reconfiguration as the valve presents degradation while ages. The monitoring of degradation rates can provide information to modify the controller parameters in order to increase the valve useful life, to allow maintenance services planning or even to prevent the system performance being meaningfully impacted by the valve aging. The sorts of degradation considered were: friction increase, charging orifice obstruction and venting orifice obstruction. The study was made based on the valve mathematical model, in which was possible to vary those degradations levels and verify how each one influences the model response. The hysteresis graphic morphology was analyzed and also the sensibility and robustness after disturbances in the inlet air pressure with the system in closed loop, where the objective was to identify a monitoring that could be realized with the aircraft in air. The results allowed the identification of the most recommended parameters to monitor and the evaluation of the valve degradation through them.

1. INTRODUCTION
The motivation for the development of this research was to monitor an actuator degradation caused by the time of use and operation conditions (actuator aging) in a way that through the controller reconfiguration it is possible to mitigate the degradation consequences. The impacts caused by the degradation are:

1. System performance deterioration, since the actuator will deviate from its nominal condition.

2. Actuator useful life reduction, causing an impact on the actuator replacement and on the logistics planning for maintenance.

These two consequences were considered. An actuator diagnosis that identifies the valve aging can send this information to reconfigurable controllers that will modify its control law. A convenient performance can be achieved, despite the degradation, or the remaining useful life can be increased. These two goals are usually conflicting.

Even when using controllers that are robust to changes in the actuators parameters, the degradation index monitoring would allow the issue of warnings before critical levels of damage are reached. Therefore, one difficulty is to define the limits for the actuator replacement.

In this context, this work proposes a methodology to measure the aging of an actuator. For this study it was selected the pneumatic valve PRSOV, which is a pressure regulator valve and it is part of the pneumatic system of aircrafts (Moir & Seabridge, 2008; Turcio, 2014). This system extracts air from the motors and provides pressurized air to the air conditioning, pressurization and anti-ice systems among others.

This subject is related to the field of study PHM (Prognostics and Health Management). Health can be defined as the extent of degradation or deviation from an expected normal condition. Prognostics is the prediction of the future state of health based on current and historical health conditions (Pecht, 2008). A system health management may provide several benefits such as the reduction of maintenance costs and safety increase (Gomes,
Ferreira, Cabral, Glavão, & Yoneyama, 2010). A major motivation in the use of more advanced diagnostic and prognostic requirements is that they are necessary to new logistic support concepts. The goals are both to maximize equipment up time and to minimize maintenance and operation costs (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006).

Other researches about the application of PHM techniques on a pneumatic valve were already made. Daigle and Goebel (2011) apply a particle filtering algorithm to a pneumatic valve from the Space Shuttle cryogenic refueling system. The binary classification algorithm Support Vector Machine (SVM) was used in a prognostics method applied to aircraft bleed valves by Moreira and Nascimento Júnior (2012). Gomes et al. (2010) applied a health management system to a pneumatic valve employed in pressure regulation systems. The method is based on the statistical analysis of deviations of the controlled pressure signal from a baseline behavior in the presence of valve degradation.

1.1. Object of study

The Figure 1 has a diagram of a typical PRSOV.

![Diagram of a pressure regulator valve PRSOV](image)

Figure 1. Diagram of a pressure regulator valve PRSOV Adapted from Turcio (2014).

The results achieved using the presented method depends on the valve architecture. The valve considered in this research is pneumatically actuated and electronically controlled through a torque motor. This valve contains a pressure reducer valve (PRV) and a pneumatic feedback (not included in the valve of the Figure 1) that connects the downstream pressure with its closing chamber.

The valve degradation is related to problems such as leakage, friction, orifices obstruction and torque motor magnetic hysteresis. The degradation can cause consequences as discomfort because of the high cabin pressure fluctuation rates, impact on the air conditioning machines, spurious alarms shown to the pilots and deactivation of the bleed air system.

1.2. Valve Model

A mathematical model of a PRSOV valve, implemented at Simulink®, was used to the development of this research. The model was created by the Environmental Systems Simulation team of Embraer S.A.. The Figure 2 shows the model.

![PRSOV valve Simulink® model](image)

Figure 2. PRSOV valve Simulink® model.

The main block of the Figure 2 has inputs to simulate the degraded behavior of the PRSOV. These inputs receive gains that are proportional to the degradation level and they can be manipulated to evaluate the influence of the degradation on the valve model.

The available inputs represent the friction, charging orifice obstruction and venting orifice obstruction. They are related to the gains FLG (friction load gain), COG (charging orifice gain) and VOG (venting orifice gain), respectively. The nominal value of the gains is unitary, that means without degradation. The degradation increases as the gain FLG increases and as the gains COG or VOG decreases. This study considered the effects of each type of degradation only separately.

2. Methodology and Objectives

This research considered that the software to diagnose the valve aging should be on board of the aircraft. A typical aircraft on board environment presents several implementation challenges, as a limited bandwidth available for on board message traffic, limited data storage capability and limited computing processor availability (Black et al., 2004). For this reason, the monitoring and the logic to identify the degradation must be simple and the data necessary must be as minimal as possible.

Two analyses were performed using the valve model: hysteresis curve analysis (without the system controller) and an analysis of the effect of disturbances while the system is
in closed loop. The second analysis verified the sensibility and robustness to changes in the inlet air pressure.

The objective was to find parameters that are simple to be measured, calculated and characterized based on the degradation level. The two analyses represent two different ways to select parameters to be used in a degradation index. The analyses also had the objective of defining the maximum acceptable limits for the degradation that could be tolerated. The selected parameters and the limits were used to propose the degradation index.

The degradation gains were modified from the condition where the valve is new, without degradation, until a condition with high degradation in order to evaluate the effects on the valve model. The impact on the output air pressure and on the current of the torque motor was observed.

2.1. Parameter Selection through the Hysteresis Curve

In this analysis it was evaluated the influence of degradation on the valve hysteresis curve. The curve was created applying a current profile in the torque motor as described in Figure 3. The inlet air pressure was constant and equal to 45 psig. In the following plots, the real current value in mA was not used. Its value was given in percentage, in order to protect this information.

![Figure 3. Regulated current (%) for the hysteresis curve.](image)

A curve of the output pressure varying with the current was generated. The parameters evaluated are represented in the Figure 4.

![Figure 4. Hysteresis curve parameters.](image)

The maximum output pressure is the maximum value after the hysteresis rising curve, when the current reaches the value 100%. The curve width is the difference between the current in the rising and descent curves, for a specific pressure value. The inferior intersection point is the current value where the descent and rising curves meet for pressure equal to zero. The superior intersection point is the current value where the descent and rising curves meet for maximal pressure. All parameters were calculated in percentage of the maximal current, with exception of the maximal pressure.

The parameters selection depended on their sensibility to the increase of the degradation and the curve characteristic.

The following criteria were used to define limits for the degradation gains, when the valve should be replaced. They are related to the difficulty to control and regulate the valve and its capacity to fully open and close.

- The inferior distance should represent at least 25% of the maximal current.
- The pressure curve width should be lower than 25% of the maximal current.
- The maximum pressure should be at least 40 psig.

The disadvantage of using the hysteresis curve to monitor degradation is that a specific test is necessary and it must occur on ground. This monitoring cannot be executed with the aircraft in air.

2.2. Parameter Selection through the Analysis of Effects of Disturbances when in Closed Loop

This analysis inserted disturbances at the system inputs and verified the effect in the outputs while the system was in closed loop configuration. The analysis was divided in sensibility and robustness analyses.

2.2.1. Procedure for the Sensibility Analysis

The inlet air pressure, inlet air temperature and reference output pressure were configured as shown in the Figure 5.

![Figure 5. Air input pressure, air input temperature and reference pressure.](image)
In the plots of Figure 5 there are three disturbances that affect the output pressure and the motor current and were observed. They occur in the following instants:

- T=120s: there is a rising ramp in the inlet air temperature and pressure.
- T=300s: there is a descent ramp in the inlet air temperature and pressure.
- T=500s: there is a step in the reference pressure.

In the sensibility analysis it was evaluated how certain parameters behave as the valve degradation increases. They were calculated based on the output pressure and current signals. A typical output signal is represented in Figure 6.

The parameters investigated are listed below:

- Overshoot – “MP” at Figure 6.
- Peak time – “Tp” at Figure 6.
- Settling time (2% criterion) – “Ts” at Figure 6.
- Undershoot.
- Number of cycles: times that the signal crosses its stationary value until stabilize.
- Area under cycles: area of the cycles between the output signal and its stationary value.
- Impact in the motor current: stabilized current value and maximum value.

The sensibility of each parameter as the degradation increases was observed as well as the characteristics of the curve.

### 2.2.2. Procedure for the Robustness Analysis

This analysis verifies how changes in the amplitude and in the slope of the inlet air pressure ramps influence the output pressure and the motor current. The same parameters of the sensibility analysis were analyzed. The parameter selected with the result of the two analyses must be sensible to the degradation but also robust to changes in the input pressure signal disturbances. This second characteristic will allow the monitoring to be realized with the aircraft in air, since no specific input pressure signal disturbance will be necessary.

The nominal inlet pressure signal, used in the sensibility analysis, has maximum amplitude of 230 psig and its ramps last 5s. To study the influence of changes in the ramps slope, three cases were tested: rising and descent ramps that last 4s, 6s and 8s, while the maximum amplitude was kept in 230 psig. To test amplitude changes, two cases were analyzed: the first had maximum amplitude of the input pressure 20% over the nominal and the second had the maximum amplitude 20% lower than the nominal. The slope was held as the nominal case. The input pressure values for each case are described in the Table 1.

<table>
<thead>
<tr>
<th>Case 1: nominal</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 2: Ramps last 4s</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>301</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3: Ramps last 6s</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>299</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 4: Ramps last 8s</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 5: Amplitude 20% over the nominal</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 6: Amplitude 20% under the nominal</th>
<th>Pressure (psig)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>123.6</td>
</tr>
<tr>
<td></td>
<td>230</td>
<td>301.4</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>1000</td>
</tr>
</tbody>
</table>

A criterion was created to define if the parameter was robust. The greatest difference between the curves generated for each case should be less than 10% of the total change of the parameter value with degradation.

One difficulty encountered on the analysis of effects of disturbances with the system in closed loop is that most of the parameters considered depend on the controller configuration. In this work the analyses were made considering a standard controller that does not adapt with the valve aging. In the case of one of the transitory parameters (overshoot for instance) be selected for the degradation monitoring, it will be necessary to map different curves for each possible controller configuration.

### 3. ANALYSES RESULTS

#### 3.1. Hysteresis Analysis

The Figure 7 represents hysteresis curves for different values of the gain FLG. When the friction increases, the curve width increases and the maximum pressure reduces.
Figure 7. Influence of the friction on the hysteresis curve.

The curve width exceeds the limit of 25% for a friction load gain equal to 3.9, as shown in Figure 8.

Figure 8. Influence of friction on the hysteresis curve width.

This parameter varies approximately linear with the degradation and has a significant sensibility. It can be used to identify FLG.

The maximum pressure reduces to values below the minimum. The limit of 40 psig is exceeded for FLG equal to 3.78, as shown in Figure 9. This result is tighter than the one found for the curve width and, then, is the gain maximum value. The inferior distance is always over 25%, this limit was not exceeded.

An equivalent analysis was made to the degradation caused by obstruction of the charging and venting orifices. The hysteresis plots are shown in the Figure 10 and Figure 11 and the result of the three analyses is presented in Table 2.

Figure 9. Influence of friction on the hysteresis maximum pressure.

Figure 10. Influence of the charging orifice obstruction on the hysteresis curve.

Figure 11. Influence of the venting orifice obstruction on the hysteresis curve.
Table 2. Hysteresis analysis results

<table>
<thead>
<tr>
<th>Gain</th>
<th>Limit</th>
<th>Selected Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friction (FLG)</td>
<td>Maximum = 3.78</td>
<td>Hysteresis curve width</td>
</tr>
<tr>
<td>Charging Orifice (COG)</td>
<td>Minimum = 0.41</td>
<td>Inferior intersection point</td>
</tr>
<tr>
<td>Venting Orifice (VOG)</td>
<td>Minimum = 0.66</td>
<td>Inferior intersection point</td>
</tr>
</tbody>
</table>

3.2. Disturbance effects analysis when in closed loop

3.2.1. Sensibility Analysis

The plot of the output pressure in time has the characteristic showed at Figure 12. In this figure it is possible to observe the effects of the disturbances inserted.

![Figure 12. Influence of the charging orifice gain on the output pressure.](image)

Through the plots of the parameters evaluated varying with the degradation, it was possible to notice that the friction influences the output pressure more than the other two kinds of degradation. No parameter calculated based on the output pressure was adequately sensible to be used to monitor valve degradation. The parameters with the best characteristics were the overshoot and the undershoot after a descent ramp. They can be seen at Figure 13 and Figure 14.

![Figure 13. Overshoot on the output pressure at descent ramp.](image)

![Figure 14. Undershoot on the output pressure at descent ramp.](image)

Despite the adequate characteristics, the parameters have too low sensibility. The undershoot has a greater sensibility and varies only 3.2%, what corresponds to 1.1 psig. These parameters could be used only with a high precision pressure sensor but, in this case, the monitoring could have spurious errors with pressure fluctuations.

The parameter overshoot at the rising ramp has a great sensibility. It is represented at Figure 15.
Because of the plot irregular characteristic it could not be considered to monitor valve aging. The Figure 16 shows the characteristic of the current variation in time.

In this analysis it could be noticed that the motor current is more influenced by the charging and venting orifice obstructions than the friction. The parameters evaluated were maximum current value and stabilized value in steady state. It was given preference to the use of the current stabilized value, since it is not transitory and, in this way, it does not depend on the controller configuration. The best parameters chosen are listed in Table 3. The plots of how these parameters vary with degradation are in Figure 17 and Figure 18.

The current final value had low sensibility with the friction load gain. However, this value is perceptible and this parameter can be used to monitor friction. The low sensibility can cause spurious messages. In order to avoid that behavior, it is possible to monitor the integrity of the input data as well as filter the raw data.

<table>
<thead>
<tr>
<th>Degradation</th>
<th>Selected Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friction</td>
<td>Current final value at rising ramp</td>
</tr>
<tr>
<td>Charging orifice</td>
<td>Current final value at rising ramp</td>
</tr>
<tr>
<td>obstruction</td>
<td>Current final value at descent ramp</td>
</tr>
<tr>
<td>Venting orifice</td>
<td>Current final value at rising ramp</td>
</tr>
<tr>
<td>obstruction</td>
<td>Current final value at descent ramp</td>
</tr>
</tbody>
</table>

Table 3. Sensibility analysis results

![Figure 15. Overshoot on the output pressure on rising ramp.](image1)

![Figure 16. Influence on the venting orifice gain on the motor current.](image2)

![Figure 17. Current stabilized value at rising ramp.](image3)

![Figure 18. Current stabilized value at descent ramp.](image4)
3.2.2. Robustness Analysis

This analysis was separated in two: influence of slope changes and influence of maximum amplitude changes. For each case, the parameters that were considered robust are listed in the Table 4.

<table>
<thead>
<tr>
<th>Degradation</th>
<th>Robustness to ramp slope changes</th>
<th>Robustness to ramp maximum amplitude changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friction</td>
<td>-</td>
<td>Overshoot on output pressure at descent ramp</td>
</tr>
<tr>
<td>Charging orifice obstruction</td>
<td>Maximum current value at rising ramp</td>
<td>Current final value at descent ramp</td>
</tr>
<tr>
<td></td>
<td>Current final value at rising ramp</td>
<td>Maximum current value at rising ramp</td>
</tr>
<tr>
<td>Venting orifice obstruction</td>
<td>Current final value at rising ramp</td>
<td>Current final value at descent ramp</td>
</tr>
<tr>
<td></td>
<td>Current final value at descent ramp</td>
<td>Maximum current value at rising ramp</td>
</tr>
</tbody>
</table>

The study of ramp slope changes for friction had as result no robust parameter. As an example of the robustness analysis, the plots of the parameter current final value at descent ramp for the venting orifice obstruction are shown in the Figure 19 and Figure 20.

When comparing the results of the sensibility analysis with the ones of robustness analysis, it is possible to notice that to determine FLG there is no parameter that satisfies all criteria. That means that to monitor this type of degradation it is necessary a specific signal of the inlet air pressure, with defined ramp slope and amplitude. The parameter selected to monitor the FLG was the current final value at rising ramp.

The obstruction of the charging and venting orifices can be determined by the current final value at descent ramp, since it is sensible to the degradation and robust according to the analyses. In this way, these degradations can be monitored with the aircraft in air after disturbances are inserted in the inlet air pressure, since the disturbance is within the limits considered in the robustness analysis: slope between 4 to 8s and amplitude 230 psig ± 20%.

4. DEGRADATION MONITORING PROPOSAL

With the results of the analysis, two possible solutions for the initial problem were proposed. The first one uses the result of the disturbance effects analysis when in closed loop and the second one is based on the hysteresis analysis result. The degradation is monitored through the following parameters in each case.

- Solution A:
  - FLG: current final value after a rising ramp in the inlet air pressure (CRR).
  - COG and VOG: current final value after a descent ramp in the inlet air pressure (CDR).

- Solution B:
  - FLG: hysteresis curve width (W).
  - COG and VOG: inferior distance of the hysteresis curve (IP).
4.1. Calculation of the Degradation Gains

With the data obtained in the simulation for the parameters selected for the solution A, equations that represent the degradation gains in relation to these parameters were determined. A curve adjustment was made for each case, the polynomial regression method (Chapra & Canale, 2008) was applied. The adjusted curves are shown in Figure 21, Figure 22 and Figure 23 with their respective equation.

![Figure 21. Determination of FLG through the current value after a rising ramp.](image)

\[
FLG = 2.915694 \times 10^{-2} \times CRR^2 - 11.18469 \times CRR + 1073.694
\]

(1)

![Figure 22. Determination of COG through the current value after a descent ramp.](image)

\[
COG = 1.3 \times 10^{-6} \times CDR^2 - 0.00783 \times CDR + 2.918
\]

(2)

![Figure 23. Determination of VOG through the current value after a descent ramp.](image)

\[
VOG = 2.97 \times 10^{-3} \times CDR^2 - 0.00701 \times CDR + 0.855
\]

(3)

The same procedure was repeated for the parameters of the solution B. The equations generated are listed below:

\[
FLG = -9.858 \times 10^{-3} \times W^2 + 0.057786 \times W - 0.8575
\]

(4)

\[
COG = 1.917 \times 10^{-5} \times IP^2 - 0.01343 \times IP + 2.576
\]

(5)

\[
VOG = 1.58 \times 10^{-3} \times IP^2 + 0.00277 \times IP + 0.226
\]

(6)

The standard error of the adjusted curves was always under 0.02 for both solutions, except for the friction load gain with respect to the current final value (Solution A) that had a standard error equal to 0.22. A linear adjustment of the data would increase the error, because of that the second order regression was chosen.

4.2. Degradation Indices for the PRSOV

With the values of FLG, COG and VOG calculated through solution A or solution B and with the limit defined for each gain, from the hysteresis analysis, the valve degradation (DV) in percentage can be determined through the following equations.

- In the presence of friction:
  \[
  DV_{FLG} = \frac{FLG - 1}{3.78 - 1} \times 100
  \]

(7)

- In the presence of charging orifice obstruction:
  \[
  DV_{COG} = \frac{1 - COG}{1 - 0.41} \times 100
  \]

(8)

- In the presence of venting orifice obstruction:
  \[
  DV_{VOG} = \frac{1 - VOG}{1 - 0.66} \times 100
  \]

(9)

DV equals to 0% represents a new valve and DV equal to 100% represents a valve on its useful life limit, when it is necessary to be replaced. As each degradation type effects were considered only separately, the indices are independent. The general valve aging is then represented by the highest one of the three indices, the one that is most close to the limit for the valve replacement. The general valve degradation index is represented by equation 10.

\[
DV = \max(DV_{FLG}, DV_{COG}, DV_{VOG})
\]

(10)

5. CONCLUSION

This research proposed two possible solutions for the problem of evaluating the aging of a PRSOV valve. The same method could be applied to other types of degradation not addressed in this study and can be adapted to other types of actuators.

One limitation of the solution proposed is that it was not considered the combined effects of the different types of degradation. And it was also not considered possible
disturbances and perturbations different from changes in the inlet air pressure. In order to apply this method, it will be necessary to obtain real valve aging data for comparison and validation.

This work contributed to parameters selection that can be used for degradation indices conception and aging identification proposal. That monitoring can influence the maintenance logistics or the development of controllers that can mitigate the degradation of components.

Some possible future works are: (i) calculate model and sensor uncertainty and the error associated to the degradation index; (ii) evaluation the effects of the valve aging when more than one type of degradation is present; (iii) development of an algorithm to the controller reconfiguration based on the degradation index; (iv) analysis of the valve aging based on historical data and comparison with the solution proposed based on the model.

ACKNOWLEDGEMENT
The authors acknowledge the support of Fundação Casimiro Montenegro Filho for this research.

NOMENCLATURE
FLG  Friction Load Gain
COG  Charging Orifice Gain
VOG  Venting Orifice Gain
PHM  Prognostics and Health Monitoring
PRSOV Pressure Regulator and Shutoff Valve
CRR  Current final value after a rising ramp
CDR  Current final value after a descent ramp
W    Hysteresis curve width
IP   Inferior intersection point

REFERENCES
doi:10.1109/AERO.2004.1368154


doi:10.1109/AERO.2012.6187377


BIOGRAPHIES

Ananda S. Ribeiro is a Product Development Engineer of flight control systems at Embraer S.A. since 2014. She holds a bachelor’s degree on Electrical Engineering from Universidade Federal da Bahia (UFBA, 2011), Brazil and participated in a bachelor sandwich program at Technische Universität Ilmenau, Germany in 2010. She holds a master degree on Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brazil.

Takashi Yoneyama is a professor of control theory with the Electronic Engineering Department of ITA. He received his bachelor’s degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in Medicine from Universidade de Taubate, Brazil, and the Ph.D. degree in Electrical Engineering from the Imperial College London, U.K. (1983). He has more than 350 published papers, has written four books, and has supervised more than 90 MSc and PhD Theses. His research is concerned mainly with stochastic optimal control theory. He served as the president of the Brazilian Automatics Society in the period 2004–2006.

Rodrigo F. Souto is a Product Development Engineer at Embraer since 2009. Rodrigo has worked with some projects dealing with modeling and simulation of flight controls systems, bleed systems and cargo handling systems. He is an inventor on a US patent for Embraer and won the 4th edition of the SAE BRAZIL Young Engineer Award. Rodrigo has BS and MS degrees in Controls Engineering from University of Brasilia.
Wallace Turcio is a Product Development Engineer at Embraer S.A. since 1998. Wallace has worked developing Air Management Systems (Bleed, Air Conditioning, Cabin Pressurization and Oxygen Systems) for several programs acting as a Product Engineer until 2005, since then he has supported the product development acting as a Simulation and Control Engineer being a member of Embraer System Simulation and CFD Team. Wallace receives the D.Eng. degree in Electronic Engineering (Systems and Control) from the Technological Institute of Aeronautics (ITA) in 2014. He has received the M.Eng and B.Eng degrees in Mechanical Engineering from Polytechnic School of the University of São Paulo respectively in 2003 and 1997.
Estimating Remaining Useful Life Using Actuarial Methods

Eric Bechhoefer¹, Rune Schlanbusch², and Tor Inge Waag³

¹, ², ³ Teknova AS, Kristiansand, Norway
eric@gpms-vt.com
rune.schlanbusch.2008@ieee.org
³ MHWirth AS, Kristiansand, Norway
tor.waag@mhwirth.com

Abstract

In many instances, condition monitoring equipment has not been installed on machinery. Yet, operators still need guidance as to when to perform maintenance that is better than what is offered by the equipment manufacturers. For these systems, running hours, counts, or some other measure of usage may be available. This data, along with failure rate data, can provide an expected time to failure, and the estimated remaining useful life. The failure rate (even small sample size) is used to estimate the shape and scale parameters for the Weibull distribution. Then the conditional expectation of the Weibull is used to estimate the time to failure. This is an actuarial technique to solve the conditional survival function problem of: given that the equipment has survived to time \( x \), what is that probability of the equipment surviving to time \( x + y \)? The inverse cumulative distribution of the truncated survival function can then be used to estimate the remaining useful life, that is: a time when the conditional likelihood of failure is small, such as 10%. The 90% confidence of the shape and scale parameters is then used to give a bound on the remaining useful life. This method is then tested on a real world bearing dataset.

1. Introduction

There is for many industrial operators, a point where the business conditions force them into reducing costs. When evaluating cost saving measures that impact productivity, condition monitoring is clearing one methodology that is attractive. For most operators, unscheduled maintenance events impact profitability. Unscheduled maintenance events can be impacted through design, condition monitoring, or a combination of both. Design efforts usually take time, and are costly to retrofit into existing platforms. This leaves condition monitoring techniques as one of the most cost effective means to reduce unscheduled maintenance, thus improving productivity and profitability.

There are a number of condition monitoring techniques that an operator can explore, including vibration, acoustic emissions, and oil particle/condition monitoring. These methods give the operator an indication of the damage/state of the machine under monitoring. Other methods, such as usage tracking, can be used as well. These “open loop” (indirect) methods, while not as powerful as the direct (closed loop) measure of damage, such as vibration, can still bring value to the operator in planning a maintenance event.

Consider the operators current maintenance paradigm. The equipment, such as a drilling machine, is designed with an operating life of 20 years. This life does not take into account a number of externalities, such as:

- Oil contamination
- Oil level low/oil starvation
- Unanticipated loads
- Variations in material quality
- Improper maintenance

When the operator experiences an unscheduled maintenance event, for large equipment in remote locations, the down time and loss of productivity can be costly.

However, many large and critical manufacturing equipment have programmable logic control (PLC) units that can record torque, rpm, current, and at the very least, operating hours. This level of detail, in addition to historic failure data, can be used to develop a prognostics health management (PHM) program. This in turns helps advice the operator when to perform maintenance, and reduces the chance of an unscheduled maintenance event. For an operator initiating a condition monitoring (CM) program, this is a powerful and low cost mean to approach this.
Actuarial science is a discipline within mathematics where statistical methods are applied to estimate future contingent events. The discipline has evolved with great pace during the last 30 years due to the development of modern computers, and is an indispensable tool in industries such as insurance, finance and healthcare. Although the discipline is well developed within the listed industries, the techniques are not widespread within the community of engineering.

In this paper, using historic data and PLC operating hours, an estimate of the remaining useful life (RUL) is calculated using a conditional survival function. This is an actuarial procedure to generate a 90% probability of surviving to some future time. This is tied to a Health Index (HI) concept. The HI is a single measure of the total health of the component or equipment under consideration. A HI of zero indicates that the component is “new”, i.e. operates perfectly according to specifications. A HI of one indicates that the component operates at the boundary of its specification envelope, in which the operator is advised to perform maintenance.

The RUL is the time from the current HI to an HI of 1, i.e. the estimated time from current time until the time when the component operates outside designated specification envelope. The HI itself is a combination of condition indicators (CIs) fused together, where each CI is a certain measure or statistic chosen to detect faults with minimum false alarm rate. Thus, the CIs should be able to differentiate between different faults and healthy state with maximum confidence.

While this HI concept has been proposed for gear health monitoring (Bechhoefer and He, 2012), the mapping of a conditioned probability of survival to an HI is a new concept.

2. Concept of Health and the RUL

To simplify presentation and knowledge creation for a user, a uniform meaning across all monitored machines should be developed. The measured CI statistics, e.g. probability density functions (PDFs), will be unique for each component or machine monitored due to different rates, materials, loads, etc. This means that the critical values (thresholds) will be different for each monitored component. By using the HI paradigm, one can normalize the CIs, such that the HI is independent of the component or machine. This facilitates the use of a “stop light” informational system by using nominal (green), warning (yellow) and alarm (red) levels. This paradigm also provides a common nomenclature for the HI, such that:

- The RUL forecasts the time when it is appropriate to do maintenance, not the time until failure.
- The HI ranges from 0 to 1. For vibration based CM, the threshold is set such that the probability of exceeding an HI of 0.5 is the probability of false alarm (PFA). For the conditional probability of survival model, the HI is scaled such that there is only a 0.1 probability of exceeding 1.2. For the conditional model, this was chosen such that the probability of failure at HI 1 is small.
- A warning alert is generated when the HI is greater than or equal to 0.75. Maintenance should be planned by estimating the RUL until the HI is 1.0.
- An alarm alert is generated when the HI is greater than or equal to 1.0. Continued operations could cause collateral damage.

Again, this nomenclature does not define a probability of failure for the component, or that the component fails when the HI is 1.0. Rather, it suggests a change in operator behavior to a proactive maintenance policy: perform maintenance prior to the generations of cascading faults. For example, by performing maintenance on a bearing prior the bearing shedding extensive material, costly gearbox replacement can be avoided.

3. The Weibull Probability Distribution Function

The Weibull distribution is attributed to Waloddi Weibull in 1951 (Abernaethy, 1996). Extensive research by the U.S. Air Force for fitting of life data suggests that the Weibull analysis is a leading method. Abernethy (1996) reported while working at Pratt & Whitney that the Weibull method worked with extremely small samples: even two or three failures gave good results. This characteristic is important in many industries where the cost of development/applications testing is high. The ability of the Weibull to give relatively good parameter estimates with small sample size, allows this distribution to be used with more advanced techniques, such as failure forecasting and prognostics.

The Weibull PDF is characterized as:

\[
f(x, \lambda, k) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, \quad x \geq 0
\]

where \(k\) is the shape parameter and \(\lambda\) is the scale parameter. It is interesting to note that with \(k = 1\), the PDF is exponential, which describes a memoryless process (e.g. where the failure rate is constant over time, or Markovian). For \(k = 2\), the PDF is the Rayleigh distribution, which is used extensively in radio frequency/radar models to describe receiver random energy.

While a number of different methods can be used for estimating the parameters \(k\) and \(\lambda\), the maximum likelihood estimator (MLE) (Cohen, 1965) is commonly used, because of its numerical stability.

In general, consider the likelihood of the joint density function of \(n\) random sample of \(f(x, \lambda, k)\), then

\[
L = \prod_{i=1}^{n} f(x, \lambda, k).
\]
For a well behaved function, the maximum likelihood of \( f(x, \lambda, k) \) is the solution of
\[
\frac{d \ln(L)}{d\theta} = 0. \tag{3}
\]
Differentiating with respect to \( k \) and solving:
\[
\frac{n}{k} + \sum_{i=1}^{n} \ln x_i - \frac{1}{\lambda} \sum_{i=1}^{n} x_i^k \ln x_i = 0. \tag{5}
\]
Differentiating and solving for \( \lambda \), gives:
\[
-n/k + -1/\lambda^2 \sum_{i=1}^{n} x_i^k = 0. \tag{6}
\]
Solving for \( k \) in terms of \( \lambda \) gives:
\[
n \sum_{i=1}^{n} x_i^k \ln x_i = \sum_{i=1}^{n} x_i^k / -1/\lambda E[x^k]. \tag{7}
\]
Eq. (7) is easily solved for \( k \) using the Newton-Raphson method, and then \( \lambda \) is found as \( \lambda = E[x^k] \).

### 3.1. The Conditional Probability of Survival

The Weibull PDF gives the unconditional probability function of the component under analysis. This allows the estimation of the component life from the first moment (e.g. the expected life), and from the second moment, the variance in the life. That said, operators are typically interested in a different question: Given that the component has survived until today, what is the probability that it will survive until tomorrow, or until the next maintenance period. This concept of survival is well established by actuarial models and is the basis for insurance products.

The cumulative distribution function \( F(x, \lambda, k) \), is defined as the integral of \( f(x, \lambda, k) \) from 0 to \( x \), or the probability of a component failing between time 0 and \( x \). This allows one to define the probability of surviving to time \( x \) as: \( 1 - F(x, \lambda, k) \). For simplicity, one can assume that the estimates of \( \lambda, k \), are established.

One can now conceptualize the conditional probability of component survival to time \( x \). This is a subset of the sample space of the random variable \( X \), i.e. those values of \( X \) that fail in excess of \( x \). This is the condition survival function, or formally
\[
Pr(X > x+n \mid X > x) = S(x+n \mid X > x). \tag{8}
\]
This is the probability that the age of failure will exceed \( x+n \), given that is does last until \( x \). This is the concept that the probability of survival to \( x+n \), given survival to \( x \). From Bayes, one then finds that: \( S(x+n \mid X > n) = S(x+n)/S(x) \). In the actuarial sciences, this is called the lower truncation of the distribution of \( X \); see (London, 1997).

The more general view of the truncated distribution is to consider the distribution in which \( X \) fails between times \( y \) and \( z \). This truncated distribution is then given as:
\[
S(x | y < x \leq z) = Pr(X > x | y < x \leq z) = Pr(x < x \leq z | y < x \leq z). \tag{9}
\]
If this condition probability is multiplied by the probability of obtaining the condition (which is \( S(y) - S(z) \)), then the unconditional probability for failure between \( x \) and \( z \), which is \( S(x) - S(z) \), is then
\[
S(x | y < x \leq z) = [S(x) - S(z)] / [S(y) - S(z)]. \tag{10}
\]
From this, one can derive the expectation of the age of failure, \( X \), of a component known to be functioning at time \( y \). By subtracting \( y \) from this expected age of failure, one obtains the expected future life of a component (in the actuarial sciences, this is denoted at the expectation of life at age \( y \)). Formally, this is expressed as
\[
E[X \mid X > y] = y. \tag{11}
\]
Since \( \int_{x}^{\infty} f(x|X > y)dx = 1 \), the expectation is written as:
\[
E[X \mid X > y] = y = \int_{y}^{\infty} \int_{t+y}^{\infty} tf(t + y | X > y)dt. \tag{12}
\]
Note that \( f(t+y \mid X > y) \) is the probability distribution function of \( (X-y)/X>y) \).

### 3.2. Some Complications and Deficitions

The health concept is based on the idea that the operator does maintenance when it is appropriate. With vibration/oil condition monitoring, there is feedback from measurements that give indications of wear and damage. In the actuarial model, one is given probabilities that relate to failure. The time of estimated failure is not the time when one wants to trigger a maintenance event. Failure causes an unscheduled maintenance event, which leads to higher cost.

This leads to a definitional problem: what should the target condition probability of survival be? While not entirely an ad hoc issue, this is a case where simulation results can be used to evaluate the definition process in a structured way.

- The RUL is the condition probability that, given the component has survived to time \( y \), it will survive until time \( x \).
- The RUL is a conservative value, such that the time \( x \), at which maintenance is performed is such that the reliability of the component is not significantly degraded. For plants such as an off shore oil platforms, it may be difficult to get a replacement component if it fails prior to the planned maintenance event, and downtime is extremely expensive.
- The operator needs a range/confidence in the RUL estimate. The RUL estimate range is taken by the: low probability of failure as the 0.1 estimate of the Weibull parameters, while the high probability of failure is the 0.9 estimate of the Weibull parameters.
Because one is not interested in the time until failure, but the time until it is appropriate to do maintenance, the expected time of a failure conditioned is given at a probability of exceeding that 83% of that time, with a confidence of 90%.

The expected time of failure is defined as $e$. The last bullet point above thus means that the expected time of failure is the inverse of the lower truncated cumulative distribution function (CDF) at 90%, divided by 1.2. Thus, $RUL = e - t$, where $t$ is the current time.

Because there is no closed form solution for the inverse lower truncated CDF, this was calculated numerically via the Newton-Raphson method, such that:

$$ S = 1 - F $$

where $F$ is the Weibull CDF for the current time $x$, $\lambda$, $k$, $Fe$ is the Weibull CDF for expected time of failure at time $e$, $\lambda$, $k$. Further, we define the probability of survival after time $e$ as

$$ S(e) = 1 - Fe, $$

and the confidence

$$ P = S(e)/S = 0.9 $$

Simulation used to evaluate the performance of this HI paradigm was developed using a shape parameter, $k = 6$, and a mean time to failure of 5.5 years. Then, $\lambda$ was calculated as:

$$ \lambda = \mu / \Gamma(1 + 1/k) $$

The Weibull random function was then called to simulate the time of failure of 5 components. Then, using these failure times, an estimate of Weibull parameters, a 0.1 and 0.9 confidence of the Weibull was estimated using Cohen’s method (Cohen, 1965). Example Matlab© code can be found in the appendix. Figure 1 shows the simulated example of a component that will fail after 4.42 years, where the experiment has run 3.19 years. The RUL is 1.43 years, with a lower limit (e.g. confidence of the RUL) of 0.48 years to 2.28 years.

### 4. RESULTS AND PERFORMANCE METRICS

Simulation was used to develop the analysis routine and then evaluate the performance of a notional component. Once the analysis engine was tested via simulation, it was applied to a real world fault data set.

#### 4.1. Simulation Results

Simulation is a powerful tool to evaluation the performance of algorithms. Given the expense and time required to study fielded components, the ability to test “what if” conditions requires the establishment of performance metrics to grade the quality of the analysis. Three metrics that were chosen for this study where:

- Because one is not interested in the time until failure, but the time until it is appropriate to do maintenance, the expected time of a failure conditioned is given at a probability of exceeding that 83% of that time, with a confidence of 90%.

The capital cost to replace the failed equipment will be set at $600K per day. The cost of money (for early replacement) is taken as 7%. It is also assumed that the time to replace the failed equipment is 14 days, or $8.4 million in opportunity cost. For this simulation, just 7.6% of the trials failed prior to recommended replacement, of which 0.4% failed within 2 weeks of the RUL estimate (see Figure 2). Thus, the mean opportunity cost of failure in using this model is $638K, or approximately 1 day. Without this replacement model and simply making replacements upon failure, the opportunity cost is, as noted, $8,400K.
Given the distribution of the safety margin (e.g., replacement of the equipment prior to the failure of the equipment, Figure 3), the future value of the money spent on replacing the equipment early is $278K. Because the future value is skewed, the median is somewhat less, at $268K (Figure 4).

The net benefit of replacement based on this actuarial model is: $8,400K – ($638K + $268K) = $7,494K. This is a large cost saving, which can be achieved solely on existing data from in-service failures.

The time until failure for high speed bearing with an axial crack was generously provided by the National Renewable Energy Laboratory (NREL), Gearbox Reliability Collaborative. The data set consists of bearing with short life (mean age of failure 2.09 years based on 23 examples) and longer life (mean age of failure 4.07 years based on 25 examples).

The experiment was conducted by randomly sampling 6 bearings out of the dataset to estimate the Weibull $\lambda$, $k$ parameters. The estimated RUL was compared to the actual life, in a process similar to the simulation study.

Entering assumptions for cost can be defined based on historic industry values. The cost of replacing a high speed bearing “up tower” (i.e. prior to failure and the associated collateral damaged associated with failure) is $50K, and two days of lost production. The cost associated with a “down tower” (i.e. after the failure occurs, during which there is the cost of replacing the gearbox, and a mobilization cost for the crane), is $400K and 30 days of lost production. The lost revenue for a day of power production will be taken as $1K.

4.3.1. Short Life Bearing Replacement Policy

Based on the random draw of the six bearing failure times, a $\lambda$ of 2.26 and $k$ of 13.15 was calculated from for the Weibull. The expected life the bearing, from the inverse lower truncated CDF at the start of the experiment is 1.58 years. At the time of replacement, the expected value of the days remaining until failure (Figure 5) was 0.48 years.

The net present value of replacing the bearing early was: $53.7K, including lost revenue for the maintenance. This compares to a cost of 430K for the current practice. The net benefit of actuarial model is: $430K - $53.7K = $376.3K
4.3.2. Long Life Bearing Replacement Policy

For the long bearing life experiment, the $\lambda$ was 4.27 and $k$ of 9.5, giving a life the bearing, from the inverse lower truncated CDF at the start of the experiment of 2.81 years. At the end of the experiment, the expected lost years of usage (i.e. the margin of safety) was 1.1142 years (Figure 6).

The mean cost of this policy was $57.9K. Given a similar cost structure for replacing the bearing when its failed, the net benefit of the method was $372.1K.

5. Conclusion

Actuarial methods provide a means to examine condition probabilities of survival. This in turn can be used to modify existing maintenance practices to replace equipment when it is appropriate, versus when the equipment has failed. From a human factors perspective, the usage and current health of the equipment is present to the operator in such a way that maintenance is planned when the health is at 0.75, and performed when the component/equipment is at an HI of 1 or greater.

This conservative process insures that the probability of unscheduled maintenance is small. For the simulated data, this results in a significant opportunity cost saving relative to the current “run to failure” model. Even taking into account the cost of money, and the rare cases in which this model fails, this paradigm resulted in a 90% cost saving over traditional maintenance models ($7.5 million savings on an $8.4 million dollar estimated cost).

For the real world bearing data, this resulted in a cost reduction of 6:1 (e.g. ~$55K cost of replacing early, vs. a $430K cost when failed). It can be argued that, perhaps the failed bearing does not require a “down tower” repair. Even under these circumstances, its likely that the cost benefit is at least the lost revenue due to lost production, or 30K.

While this model used time, reduction in system variance may be improved by using other metrics of usage, such as power hours or some other more direct measure of load or wear on the equipment.

The use of simulation of the actuarial method would allow optimization and minimization of opportunity costs. This could be achieved by adjusting the expected life to HI mapping. This model is based on certain assumptions in the cost of money, the opportunity cost due to lost productivity, and the cost of the equipment. These clearly can be argued and updated as needed.

Acknowledgement

We would like to thank Dr. Shaugnwen Sheng, and Dr. Jonathan Keller, of the National Renewable Energy Laboratory, the Gearbox Reliability Consortium, for there help on this paper.

References


Biographies

Eric Bechhoefer is the president of GPMS, Inc., a company focused on the development of low cost condition monitoring systems. Dr. Bechhoefer is the author of over 100+ juried papers on condition monitoring and prognostics health management, and holds 23 patents in the field of CBM.
Rune Schlanbusch received his MSc degree in Space Technology from Narvik University College (NUC), Norway in 2007, and PhD degree in Engineering Cybernetics from NTNU, Norway in 2012. He currently holds positions as senior researcher at Teknova, Norway and II associate professor at Department of Technology, NUC.

Tor Inge Waag is a Technical Advisor at MHWirth AS and holds a part time position as senior researcher at Teknova AS in Norway. He has received both his MSc and PhD degree in Physics from NTNU, Norway. His primary work focus is at present on condition monitoring, signal processing, and prognostics health management.

**APPENDIX**

```matlab
function [safety, error, fHI] = getRUL(lam, k)

safety = zeros(100,1);
error = zeros(100,1);
fHI = zeros(100,1);
for j = 1:500,
    if nargin == 0,
        mv = 5.5;
        k = 6;
        lam = mv/gamma(1+1/k);
        sampe = wblrnd(lam,k,1,5);
        actual = wblrnd(lam,k);
        [parmhat, parmbnd] = wblfit(sampe,1);
    k = parmhat(2);
    lam = parmhat(1);
    kh = parmbnd(1,2);
    lamhi = parmbnd(1,1);
    kdlw = parmbnd(2,2);
    lamLw = parmbnd(2,1);
    pr = 0;
end

y = linspace(0,actual,100);

rul = zeros(1,100);
rulHi = rul;
rULw = rul;
life = rul;
hi = rul;
hlw = rul;
hhl = rul;

for i = 1:100
    crt = y(i);
    e = invLowTrunCDF(crt,lam,k,1)/1.2;
    elw = invLowTrunCDF(crt,lamLw,kdlw,1)/1.2;
    ehi = invLowTrunCDF(crt,lamhi,kkhL,1)/1.2;
    rul(i) = e-crt;
    rulHi(i) = ehi-crt;
    rULw(i) = elw-crt;
    life(i) = e;
    hi(i) = crt/e;
    if hi(i) > 1,
        break;
    end
    hlw(i) = crt/elw;
    hhl(i) = crt/ehi;
    if pr == 1,
        plot([0 rul(i)],[hi(i) 1],[0 rULw(i)],[hlw(i) 1],',m','y(1+i)-crthi(1+i).actual-crt,1,*',[0 rulHi(i)],[hhl(i) 1],',m','LineWidth',2)
        ax([1 8 5 0 1])
        xlabel('Usage: RUL (Years)';'FontSize',14)
        ylabel('Component Health', 'FontSize',14)
        legend('Expected RUL','Bound on RUL','Current Usage','Actual Failure')
        pause(.1)
    end
end

safety(j) = actual - crt;
error(j) = rul(i);
fHI(j) = hi(i);
if hi(i) < 1,
    disp(['Failed prior to repair: rul = ' num2str(rul(i)) ', HI ' num2str(hi(i))])
end

figure(1)
hist(safety);
titl('Safety Factor')
figure(2)
hist(error)
titl('Error in RUL')
figure(3)
hist(fHI)
titl('Final HI')

function n = invLowTrunCDF(x,lam,k,p)

global kl;
global laml;
global pTarget;
global xl;
small = 1 -1e-5;
upper = wblinv(small,lam,k);
kL = k;
lamL = lam;
xL = x;
pTarget = 1-p;

n = fminbnd('setLowTurnCDF',x,upper);

function x = setLowTurnCDF(val)

global kl;
global lamL;
global pTarget;
global xl;

p = lowTrunCDF(xL,val,lamL,kL);
x = (p-pTarget)^2;

function p = lowTrunCDF(x,n,lam,k)

F = wblcdf(x,lam,k);

S = 1-F;
```

The code above calculates the reliability and failure prediction based on the given parameters and conditions.
\[ Sn = 1 - \text{wblcdf}(n, \lambda, k); \]
\[ p = Sn / S; \]
\[ p(p < 0) = 0; \]
Remain the useful life prediction through failure probability computation for condition-based prognostics

Shankar Sankararaman

1 SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA
shankar.sankararaman@nasa.gov

Abstract

The key goal in a prognostics is to predict the remaining useful life (RUL) of engineering systems in order to guide different types of decision-making activities, such as path planning, fault mitigation, etc. The remaining useful life of an engineering component/system is defined as the first future time-instant in which a set of threshold conditions are violated. The violation of these conditions may render the system inoperable or even lead to catastrophic failure. This paper develops a computational methodology to analyze the aforementioned set of safety threshold conditions, calculate the probability of failure, and in turn, proposes a new hypothesis to mathematically connect such probability to the remaining useful life prediction. A significant advantage of the proposed methodology is that it is possible to learn important properties of the remaining useful life, without simulating the system until the occurrence of failure; this feature renders the proposed approach unique in comparison with existing direct-RUL-prediction approaches. The methodology also provides a systematic way of treating the different sources of uncertainty that may arise from imprecisely known future operating conditions, inaccurate state-of-health state estimates, use of imperfect models, etc. The proposed approach is developed using a model-based framework for prognosis using principles of probability, and illustrated using a numerical example.

1. Motivation

1.1. Introduction

The prediction of remaining useful life (RUL) is, perhaps, the central component of a prognostics and health management (PHM) system (Vaidya & Rausand, 2011). In order to predict the RUL, it is necessary to thoroughly understand the functioning of the engineering system under consideration, estimate the state-of-health, analyze possible failure modes, predict damage growth using degradation models, and identify the future time-instant at which it is not possible to continue operating the system (Engel, Gilmarten, Bongort, & Hess, 2000). The aforementioned time-instant is referred to as the end-of-life (EOL), and it is possible to check whether EOL has been reached by evaluating a binary threshold constraint (referred to as the EOL-threshold function). Typically, safety constraints and serviceability constraints are used to formulate the EOL-threshold function.

1.2. Uncertainty in Prognostics

An important aspect of prognosis is that future prediction intrinsically needs to account for the various sources of uncertainty that affect the future behavior of the system (Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008). As a result, the EOL and RUL predictions become uncertain (Sankararaman & Goebel, 2013); in fact, at any future time-instant, there is a probability that the EOL-threshold is violated/satisfied.

To begin with, it is practically impossible to estimate the state of health because (1) it is rarely possible to measure health directly, and it may be necessary to infer health from system output measurements; and (2) such measurements are obtained from sensors that may not be accurate due to noise, gain, bias, etc. Techniques such as Kalman filtering (Swanson, 2001) and particle filtering (Zio & Peloni, 2011) are used to estimate the state-of-health. Starting from an arbitrary state of health, it is necessary to predict damage growth using a degradation model. Damage growth is a function of usage/operating conditions, loading conditions, etc., all of which may be uncertain, and therefore, render damage growth uncertain. The degradation model used for damage growth prediction may also uncertain, and model uncertainty is represented through uncertain model parameters and model form errors (usually, approximated using process noise). It is important to systematically account for these sources of uncertainty in prognosis, and estimate the overall uncertainty in the RUL prediction.
1.3. Previous Work

Previous work in this context has focused on posing RUL prediction as an uncertainty propagation problem (Sankararaman & Goebel, 2013).

The RUL is expressed as a “black-box function” of all other uncertain quantities; for every realization of these uncertain quantities, the future behavior of the system is simulated until EOL is reached. The aforementioned “black-box function” is a combination of (1) damage degradation model (Luo, Pattipati, Qiao, & Chigusa, 2008), usually expressed as a state-space model (Sun, Zuo, Wang, & Pecht, 2012); and (2) the EOL-threshold. Then, the uncertainty in the RUL prediction is computed by propagating the different sources of uncertainty through the so-called black-box function. Such propagation can be accomplished through a variety of sampling-based (Daigle, Saxena, & Goebel, 2012; Sankararaman, 2015) and analytical methods (Sankararaman, Daigle, & Goebel, 2014). These methods have been applied to a variety of applications such as pumps (Daigle & Goebel, 2013), valves (Daigle & Goebel, 2011), batteries (Chen & Rincon-Mora, 2006), structural crack growth damage prognosis (Farrar & Lieven, 2007), capacitors (Kulkarni, Celaya, Goebel, & Biswas, 2013) etc.

1.4. Proposed Approach

In general, the computation of the black-box function may be computationally intensive since it requires simulation until EOL is reached. The present paper explores an alternative method, where RUL can be predicted without simulating the system until EOL. This can be achieved by evaluating the likelihood of system failure, and the relationship between such likelihood and the RUL can be mathematically proved. The likelihood of system failure can be calculated analytically, based on methods developed by researchers in the field of “model-based reliability analysis”. It is important not to confuse this terminology with reliability-based life-prediction or testing-based life-prediction (Saxena, Sankararaman, & Goebel, 2014) that focus on fleet-wide prognostics, and mathematically defines the remaining useful life, which in turn is based on the definition of end-of-life threshold function.

Suppose that it is desired to perform prognostics and predict the RUL at a generic time-instant \( t_P \). Consider the architecture shown in Fig. 1, where the whole problem of prognostics can be considered to consist of the following three sub-problems:

1. Present state estimation
2. Future state prediction
3. RUL computation

2.1. State Estimation

The first step of estimating the state at \( t_P \) serves as the precursor to prognosis and RUL computation. Consider the state space model that is used to continuously predict the state of the system, as:

\[
\dot{x}(t) = f(t, x(t), \theta(t), u(t), v(t))
\]

where \( x(t) \in \mathbb{R}^n_x \) is the state vector, \( \theta(t) \in \mathbb{R}^n_\theta \) is the parameter vector, \( u(t) \in \mathbb{R}^n_u \) is the input vector, \( v(t) \in \mathbb{R}^n_v \) is the process noise vector, and \( f \) is the state equation.

As stated earlier, the state of the system uniquely defines the amount of damage in the system.

The state vector at time \( t_P \), i.e., \( x(t) \) (and the parameters \( \theta(t) \), if they are unknown) is (are) estimated using output data collected until \( t_P \). Let \( y(t) \in \mathbb{R}^n_y \), \( n(t) \in \mathbb{R}^n_n \), and \( h \) denote the output vector, measurement noise vector, and output

Figure 1. Model-Based Prognostics Architecture

--ANNEX--
equation respectively. Then,
\[ y(t) = h(t, x(t), \theta(t), u(t), n(t)) \]

Typically, filtering approaches such as Kalman filtering, particle filtering, etc. may be used for such state estimation.

### 2.2. Future State Prediction

Having estimated the state at time \( t_P \), the next step is to predict the future states of the component/system. Note that, since the focus is predicting future, no data is available, and it is necessary to completely rely and use Eq. 1 for this purpose. This differential equation can be discretized and used to predict the states at any future time instant \( t > t_P \), as a function of the states at time \( t_P \).

### 2.3. RUL Computation

RUL computation is concerned with the performance of the component that lies outside a given region of acceptable behavior. The desired performance is expressed through a set of constraints, \( C_{EOL} = \{c_i\}_{i=1}^n \), where \( c_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{B} \) maps a given point in the joint state-parameter space given the current inputs, \( (x(t), \theta(t), u(t)) \), to the Boolean domain \( \mathbb{B} \triangleq [0, 1] \), where \( c_i(x(t), \theta(t), u(t)) = 1 \) if the constraint is satisfied, and 0 otherwise (Daigle & Goebel, 2013).

These individual constraints may be combined into a single threshold function \( T_{EOL} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{B} \), defined as:

\[
T_{EOL}(x(t), \theta(t), u(t)) = \begin{cases} 
1, & 0 \in c_i(x(t), \theta(t), u(t)) \bigcup_{i=1}^n \\
0, & \text{otherwise.}
\end{cases}
\]

For the sake of simplicity, the above equation can be simply written as:

\[
T_{EOL}(t) = T_{EOL}(x(t), \theta(t), u(t))
\]

The above notation is valid and simpler to use because all the arguments of \( T_{EOL} \) in Eq. 3 can be calculated as a function of \( t \), and hence, \( T_{EOL}(t) \) can be easily represented as a function of only \( t \).

\( T_{EOL} \) is equal to 1 when EOL-threshold constraint is violated. Then, the End of Life (EOL, denoted by \( E \)) at any time instant \( t_P \) is then defined as the earliest time point at which the value of \( T_{EOL} \) becomes equal to one. Therefore,

\[
E(t_P) \triangleq \inf\{t \in \mathbb{R} : t \geq t_P \land T_{EOL}(t) = 1\}.
\]

The Remaining Useful Life (RUL, denoted by \( R \)) at time instant \( t_P \) is expressed as:

\[
R(t_P) \triangleq E(t_P) - t_P.
\]

Note that the output equation (Eq. 2) or output data \( y(t) \) is not used in the prediction stage, and EOL and RUL are dependent only on the state estimates at time \( t_P \); though these state estimates are obtained using the output data, the output data is not used for EOL/RUL calculation after state estimation.

For the purpose of implementation, \( f \) in Eq. 1 is transformed into the corresponding discrete-time version. Discrete time is indexed by \( k \), and there is a one-to-one relation between \( t \) and \( k \) depending on the discretization level. While the time at which prediction needs to be performed is denoted at \( t_P \), the corresponding index is denoted by \( k_p \). Similar let \( k_E \) denote the time index that corresponds to the end of life. Thus, it is clear that RUL predicted at time \( t_P \), i.e., \( R(t_P) \) depends on

1. Present state estimate \( (x(k_P)) \); using the present state estimate and the state space equations in Eq. 1, the future states \( (x(k_P), x(k_P+1), x(k_P+2), \ldots, x(k_E)) \) can be calculated.
2. Future loading \( (u(k_P), u(k_P+1), u(k_P+2), \ldots, u(k_E)) \); these values are needed to calculate the future state values using the state space equations.
3. Parameter values from time-index \( k_P \) until time-index \( k_E \) (denoted by \( \theta(k_P), \theta(k_P+1), \ldots, \theta(k_E) \)).
4. Process noise \( (v(k_P), v(k_P+1), v(k_P+2), \ldots, v(k_E)) \).

For the purpose of RUL prediction, all of the above quantities are independent quantities and hence, RUL becomes a dependent quantity. Let \( X = \{X_1, X_2, \ldots, X_i, \ldots, X_n\} \) denote the vector of all the above dependent quantities, where \( n \) is the length of the vector \( X \), and therefore the number of uncertain quantities that influence the RUL prediction. Then the calculation of RUL (denoted by \( R \)) can be expressed in terms of a function, as:

\[
R = G(X)
\]

The above functional relation in Eq. 7 can be graphically explained, as shown in Fig. 2. Knowing the values of \( X \), it is possible to compute the corresponding value of \( R \), using Fig. 2 that is equivalently represented by Eq. 7. The quantities contained in \( X \) are uncertain, and the focus in prognostics to compute their combined effect on the RUL prediction, and thereby compute the probability distribution of \( R \). The problem of estimating the uncertainty in \( R \) is equivalent to propagating the uncertainty in \( X \) through \( G \), and researchers have investigated different types of methods for this purpose. The most commonly used methodology for this purpose is Monte Carlo sampling, using which multiple realizations of \( R \) can be obtained from multiple realizations of uncertain quantities. This approach is referred to as the direct-RUL-prediction approach in this paper, and this is not pursued. An alternative statistical methodology is developed and the direct-RUL-prediction approach will be used as a benchmark to compare results from the newly proposed methodology.
3. Failure Probability and RUL

The proposed methodology is based on evaluating the likelihood of satisfying the EOL-threshold constraint, and relating this likelihood to the RUL prediction.

Consider the time of prediction \( t_P \), all time-instants \( t > t_P \). As per Eq. 3, the system is said to be safe and operable so long as \( T_{EOL}(t) = 0 \), and the first future time-instant \( t_P \) at which \( T_{EOL}(t_E) \) becomes equal to one is said to be equal to the EOL. For the sake of illustration and terminology description, assume that failure corresponds to \( t_{EOL} \). Consider a generic future time-instant \( t \).

Using the following probability that calculates the likelihood of failure:

\[
P_f(t|t_P) = P(T_{EOL}(t) = 1|t_P) \tag{8}
\]

Note that the above probability is calculated at the time of prediction \( t_P \), but for a generic future time-instant \( t \).

If it can be assumed the amount of damage/failure is non-decreasing, then it can be easily visualized that the function \( P_f(t|t_P) \) is non-decreasing with respect to time \( t \). While this does seem to be a reasonable assumption, it is not universally true. For example, in structural damage prognosis, crack closure is a widely studied phenomena, and can lead to an “improvement in the health state” (since crack closure can result in an increased stiffness). The rest of this paper only focuses on scenarios in which the amount of damage/fault is non-decreasing, and therefore, \( P_f(t|t_P) \) is non-decreasing with respect to time \( t \).

The hypothesis proposed in this paper, is that \( P_f(t|t_P) \) is exactly equal to the “probability that the end of life is less than or equal to time \( t \)”. In mathematical terms:

\[
P_f(t|t_P) = P(T_E \leq t|t_P) \tag{9}
\]

Note that the right hand side of the above equation is exactly equal to the cumulative distribution function of the End-of-Life.

Recall from Section 2 that the End-of-Life is an uncertain quantity and needs to be expressed using a probability distribution. If \( T_E \) denotes the random variable, and \( t_E \) an instance of this variables, then the probability density function (PDF) and the cumulative distribution function (CDF) of this variable are denoted by \( f_{T_E}(t_E) \) and \( F_{T_E}(t_E) \) respectively. Therefore, Eq. 9 can be extended as:

\[
P_f(t|t_P) = P(T_E \leq t|t_P) = F_{T_E}(t_E) \tag{10}
\]

In order to prove the above hypothesis, consider “\( N \)” different, random system paths starting from the time of prediction \( t_P \). Consider a generic future time-instant \( t \), by which “\( m \)” paths have already reached the end-of-life. Therefore, by mere definition of the CDF, it can be written that:

\[
P(T_E \leq t|t_P) = F_{T_E}(t_E) = \frac{m}{N} \tag{11}
\]

At the particular time \( t \), there are now a total of “\( N \)” states out of which “\( m \)” states “fall in” the zone of failure. Therefore, it also follows that:

\[
P_f(t|t_P) = \frac{m}{N} \tag{12}
\]

Therefore, by comparing Eq. 11, and Eq. 12, the proposed hypothesis is therefore proved.

Figure 2. Definition of \( G \)
This hypothesis provides a fundamentally different way of calculating EOL and therefore, the RUL. Once EOL is obtained, RUL can be easily calculated using Eq. 6. The major advantage of the proposed methodology is that, in order to calculate \( P_f(t|t_P) \), it is not necessary to simulate the system until failure; nevertheless, this probability, through Eq. 10, can provide the cumulative distribution function of the EOL. The CDF value of EOL is critical in assigning credible-intervals for the EOL, which are useful for decision-making, and hence, it is believed that the proposed methodology will be of immense value in this regard.

The computation of failure probability has been discussed by several researchers, particular in the field of model-based reliability analysis. The most simplest method (simple to build and code, but expensive to implement) is Monte Carlo sampling (Robert & Casella, 2004). There are several advanced sampling methods such as importance sampling (Glynn & Iglehart, 1989), adaptive sampling (Bucher, 1988), stratified sampling (Caflisch, 1998), etc., which can improve upon the efficiency of basic Monte Carlo sampling. Alternatively, there are also analytical methods developed by structural reliability engineers; these include the first-order reliability methods (Haldar & Mahadevan, 2000), second-order reliability method (Der Kiureghian, Lin, & Hwang, 1987), etc. The focus of the present paper is not on testing the applicability of these methods, but on developing and proving the hypothesis in Eq. 9, as applicable to condition-based prognostics. As this hypothesis has been mathemtically proved in this section, it is illustrated using a numerical example in the following section.

4. Numerical Example

In order to illustrate the proposed methodology, consider the problem of crack growth prognosis in a simple plate. This plate is subjected to cyclic, uniform uniaxial stress (\( S \)), and Paris’ law is used for crack growth analysis. Paris law calculates the increment in crack size per cycle of loading, in terms of crack growth parameters (\( C \) and \( n \)), threshold stress intensity factor (\( \Delta K_{th} \)), and load stress intensity factor (\( \Delta K \)):

\[
\frac{da}{dN} = C(\Delta K)^n(1 - \frac{\Delta K}{\Delta K_{th}})^p \tag{13}
\]

The stress intensity factor (\( \Delta K \)), for the sake of illustration, is assumed to be available in closed form, as:

\[
\Delta K = S\sqrt{\pi a} \tag{14}
\]

For the sake of this numerical example, “the 7075-T6” aluminum alloy is considered. The quantities \( S \sim LN(100, 40) \), \( C \sim LN(6.54 \times 10^{-13}, 4.0 \times 10^{-13}) \), and \( \Delta K_{th} \sim LN(5.66 \times 10^6, 0.268 \times 10^6) \) are chosen to be lognormal random variables. The quantities in parentheses above indicate the mean and standard deviation of the random variables (all numerical values are in SI units). The exponents “\( n \)” and “\( p \)” are assumed to 3.89 and 0.75 (no unit) respectively.

An important challenge in crack growth analysis, is that the initial crack size is not known. A rigorous approach to fatigue life would account for crack growth starting from the actual initial flaw, accounting for imperfections, voids and non-metallic inclusions. This procedure is cumbersome because small crack growth propagation is anomalous in nature and hence not completely understood. On the other hand, there are several crack growth models (Paris law (Pugno, Ciavarella, Cornetti, & Carpinteri, 2006), AFGROW (Harter, 1999), etc.) in the long crack regime which are used to study long crack growth behavior. Equivalent initial flaw size is a fictitious quantity which was introduced to bypass small crack growth calculations and make direct use of a long crack growth law in order to make fatigue life prediction; the EIFS must be chosen in such a way that when the long crack growth law is used with EIFS as the initial value, it yields crack growth results that match with observed crack growth data (Liu & Mahadevan, 2009). Since EIFS is fictitious and hence, not measurable, it needs to be estimated based on observed data on crack size. There have been several studies on how to estimate the equivalent initial flaw size (EIFS), and the value reported by Liu and Mahadevan (Liu & Mahadevan, 2009) is used in the analysis below; EIFS is assumed to follow a lognormal distribution with mean and standard deviation equal to 0.23mm and 0.05mm respectively. If average values are assumed for all the quantities (i.e., without any uncertainty in them), then the crack growth behavior can be obtained as shown in Fig. 3.

The failure threshold limit is set to be the time-instant when the crack size exceeds 0.75mm. It can be easily seen that, after this crack size, the rate of increase in crack size is phenomenally high.

Including the different sources of uncertainty, the probability density function of RUL is first computed using the direct-RUL-prediction approach. Monte Carlo sampling is used for the purpose of illustration, and the resulting probability density function is shown in Fig. 4. For this particular analysis,
1000 Monte Carlo samples were used, and the resultant RUL samples were converted into a probability density function using principles of kernel density estimation.

Note that Fig. 4 is calculated using “G” in Eq. 2, whereas the proposed approach only calculates $P(T_{EOL}(t) = 1)$, which is equivalent to $P(a > 0.75)$, where “a” represents the crack size in this numerical example. As per the proposed approach, $P_f(t \mid t_0)$ is plotted as a function of time $t$, where the time of prediction is denoted by $t_0$, the initial time, as shown in Fig. 5. Here, Monte Carlo analysis is used for the calculation of failure probability (note that Monte Carlo analysis was previously used for direct-RUL-prediction, and required simulation until failure), only for the purpose of illustration; other advanced failure probability computation methods will be investigated in future.

In order to demonstrate the proposed methodology, and illustrate the comparison in Eq. 9, it is necessary to compare (1) the cumulative distribution function corresponding to the PDF in Fig. 4; and (2) the “failure probability versus time” plot in Fig. 5. This comparison is shown in Fig. 6. Note that, in this example, the time of prediction is considered as “$t_P = 0$”, and therefore the EOL and RUL are identical.

It can be seen from Fig. 6 that the two plots compare very well, thereby illustrating the hypothesis proposed in Eq. 9. While the direct-RUL-prediction approach requires simulation every sample until failure to obtain the prediction of EOL, the proposed approach based on failure probability computation does not require this. In fact, the proposed approach requires a much easier computation in comparison with the direct-RUL-prediction approach, and also provides a systematic way of handling uncertainty, analogous to the direct-RUL-prediction approach.

5. CONCLUSION

This paper presented a new computational methodology for remaining useful life estimation in prognostics. While RUL has been traditionally calculated by forecasting the state-of-health degradation, the proposed approach calculates the likelihood of system failure and mathematically connects such likelihood to the remaining useful life. The major advantage of the proposed approach is that it is possible to learn about the properties of RUL even without simulating until failure or the end-of-life. The proposed approach was developed in the context of model-based, condition-based prognostics, using fundamentals of probabilistic analysis. This method is applicable to scenarios where the degradation is monotonically decreasing (there may be some scenarios such as crack closure where the state of health improves). Finally, the method was illustrated using a simple numerical example, consisting of component-level remaining useful life prediction.

It is further necessary to investigate the applicability of the proposed approach to complicated EOL-threshold functions and extend this approach to larger, practical engineering systems and applications.

ACKNOWLEDGMENT

The work reported herein was in part funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), and by the NASA Automated Cryogenic Loading Operations (ACLO) project under the Office of the Chief Technologist (OCT) of Advanced Exploration Systems (AES).
REFERENCES


152
**Biography**

Shankar Sankararaman received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007; and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the Non-Deterministic Approaches (NDA) technical committee at the American Institute of Aeronautics and Astronautics (AIAA), the Probabilistic Methods Technical Committee (PMC) at the American Society of Civil Engineers (ASCE), the Institute of Electrical and Electronics Engineers (IEEE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, Moffett Field, CA, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.
Damage Accumulation: A New Generalized Machine Health and Condition Indicator for Continuous Monitoring

Jacob Loverich, Ph.D.¹, Jeremy Frank, Ph.D.², David Kraige, M.S.³, Gary Koopmann, Ph.D.⁴, Ashok Belegundu, Ph.D.⁵ and Michael Grissom, Ph.D.⁶

¹,²,³,⁴,⁶ KCF Technologies, Inc., State College, PA 16801, USA
loverich@kcftech.com
jfrank@kcftech.com

⁵ The Pennsylvania State University, University Park, PA 16802, USA
adb3@psu.edu

ABSTRACT

This paper presents a new Damage Accumulation machine health indicator that trades precision for ease of use and broad applicability, effectively addressing the challenges of practically applying continuous monitoring to large numbers of assets such as pumps, motors, gearboxes, and fans. Damage Accumulation offers comprehensive early warning indication for a broad range of faults that can in turn be used to trigger detailed analysis. Damage Accumulation is a time series vibration analysis technique that estimates the rate at which damage is accumulated at a given location. This indicator accounts for time-varying symptoms in machines which are often overlooked by traditional vibration diagnostic frequency analysis or time series analysis. It also considers the contribution of repeated load reversal cycles to component damage and the nonlinearity in the relationship between damage and vibration amplitude. This paper presents the fatigue analysis foundation of Damage Accumulation and demonstrates its efficacy in a bench test and in a high pressure pumping field example.

1. INTRODUCTION

With recent advancements in low power electronics, M2M communication, and cloud resources, it is now possible to cost-effectively monitor the health of Balance of Plant (BoP) assets continuously. The BoP refers to equipment supporting primary assets such as turbines in power plants. BoP equipment includes fans, pumps, and motors which are found in most industrial facilities, chemical plants, and manufacturing plants. The health of these less critical assets is typically not monitored, or is monitored with infrequent, periodic route-based techniques. Continuous machine health monitoring has the potential to offer dramatic improvements over traditional reliability programs by virtue of the much larger amount of machine health evidence, near real-time results, and analysis across a range of operating conditions. A challenge remains in deriving practical value from continuous monitoring, considering the limited knowledge of operators and maintainers in the BoP, small budgets for setup and customization of data analysis, and the large volume of data that must be analyzed.

The Damage Accumulation indicator addresses these challenges by reducing the upfront installation and setup cost, simplifying analysis, and supporting existing route-based monitoring programs. Using the Damage Accumulation indicator in combination with continuous monitoring enables the requirement for absolute precision in health monitoring and forecasting to be relaxed and is justifiable for the BoP because asset cost is relatively low and scheduled downtime is frequent. In this paradigm, practical cost savings are derived mainly from moving unscheduled maintenance into scheduled time slots. This requires failure prediction with a time resolution that is shorter than the scheduled downtime interval. High probability fault identification on such time scales is achievable using the proposed Damage Accumulation analysis technique. This is in contrast to the higher precision needed to derive value from route monitoring where the inspection interval is typically much longer than the downtime interval. In route monitoring, the machine health evidence is only a brief snapshot versus the thorough evidence provided by continuous monitoring.

Additional value is derived from the proposed methods, based on the new ability to provide easy to understand and actionable machine health status in near real time to operators to avoid problematic conditions. Avoiding these conditions in industrial facilities is typically not well served by detailed diagnostic or prognostic tools and analysis since most of
these conditions occur unexpectedly, are temporary, and are difficult to model.

This paper presents the fatigue analysis foundation of Damage Accumulation and its practical applicability. The value and limitation of this indicator is demonstrated for unbalance, misalignment, and soft foot seeded faults on a test stand. An infiel field application of Damage Accumulation is described for a hydraulic pump. In these cases, Damage Accumulation is shown to identify the machine faults more comprehensively than traditional time series and frequency spectrum analysis techniques.

The analysis and examples in this paper show that Damage Accumulation offers the ability to not only identify traditional rotating machinery faults, but also identify adverse operation conditions that damage machines, leading to reduced component life and unexpected failures.

2. CONDITION BASED MAINTENANCE

Optimizing maintenance practices is essential to maximizing profitability in industrial markets. Maintenance affects profitability by influencing downtime, failure-related collateral damage, energy efficiency, component life and most importantly, safety of employees.

Traditional approaches to optimizing maintenance focuses on planning part replacement times based on historical distributions of failure times and expected part lives. In most industries, the distribution of failure times and types is wide due to a range of difficult to control variables in an industrial environment, including the quality and precision of the routine maintenance. This leads to overly conservative maintenance and unexpected failures.

Condition Base Maintenance (CBM) reduces overly conservative maintenance by measuring the health of a component or machine and then tailoring maintenance to fit the specific needs at a particular point in time, thus accommodating variation in usage conditions and operational load.

![Figure 1](image1.png)
Figure 1. Benefits of performing maintenance proactively using CMB and wireless sensors.

In practice, CBM is enabled by using 1) periodic route-based inspections including vibration monitoring, or 2) instrumenting of equipment with wired sensors and continuously monitoring operation parameters and vibration. Route monitoring is typically performed with hand-held devices and performed on a monthly basis. A typical facility may have hundreds of monitoring points.

In a typical application scenario, the onset and progression of faults such as unbalance and bearing wear are tracked. When fault indicators approach critical thresholds, specific maintenance is scheduled to prevent failure events and the resulting unscheduled maintenance. With this approach, the correct parts can be ordered in advance and are on hand to perform scheduled maintenance.

When applied to a typical pump, CBM offers $900 of savings per 100 hp. annually (Sullivan, et al., 2002) relative to reactive maintenance costs (shown in graph below). The US Department of Energy estimates that 55% of maintenance is performed using high cost reactive approaches (Piotrowski, 2001). Thus, it is evident that there are great opportunities to improve profitability by moving to a more enlightened approach to maintenance in general.

![Figure 2](image2.png)
Figure 2. Typical cost saving associated with reactive, predictive, and preventative maintenance for pumps.

3. LIMITATIONS OF CONVENTIONAL CBM TOOLS

Route-based monitoring in part addresses the need for broad application of condition monitoring, but it lacks capability to catch many failures for the following reasons: 1) route-monitoring is done under controlled conditions (generally fixed load and running speed) so that the data can accurately be compared to a bench mark taken in a similar operating scenario, 2) it is performed on an infrequent basis due to the cost of manually acquiring data, and 3) hard to reach locations are rarely monitored or not at all.

The vast majority of machine failures, 89% reported in one study, are unexpected and only 11% are related to traditional machine wear out (Nowlan & Heap, 1978). These unexpected failures occur due to improper installation, maintenance, environmental conditions, or operating loads and speeds. In practice, such conditions can change rapidly and do not follow traditional, Potential Failure -Failure...
commonly known a P-F curves. Severe damage may be intermittent and go unnoticed for months or years. For example, cavitation in a pumping system is often intermittent and nearly always overlooked by conventional, diagnostic, route-based (periodic inspections) monitoring.

Similarly, machine resonance during spin-up and spin-down is not recognized by conventional diagnostics even though significant damage can occur during these transient operating conditions. Excessive loading and most faults in reciprocating equipment are intermittent and rarely considered by most diagnostic methods. This is because the root cause of damage in many cases is not directly related to a particular fault, and, therefore, the conditions that cause the damage in the first place often do not match a fault indicator anyway. In such cases, fault conditions are treated as outliers or are filtered out by fault indicator data analysis. Using route-monitoring, these conditions generally aren’t observed at all. Only in certain cases do coincidently timed inspections identify such conditions and enable them to be addressed with conventional diagnostics.

Wired sensor (permanently installed) condition monitoring technology has proven to be highly valuable for enabling CBM in highly critical applications such as power generation turbines. However, BoP assets such as pumps and motors have not benefited from online monitoring owing to the high capital cost—and in particular the high installation cost—of wired monitoring systems. For example, a 10 point wired real-time vibration monitoring solution, with 50 meter cable lengths, costs $13,497 per sensor ($134,970 total) (Emerson Process, 2015). The benefits of continuous monitoring has been well established, but the cost has only been justifiable for rare cases of high capital cost equipment.

The data acquisition is only one challenge with applying CBM methodologies. In most cases, advanced diagnostics and prognostics are also problematic to apply practically in the BoP because they require some combination of large historical fault data sets and detailed machine failure analysis. In addition, both the training data and the fault analysis are specific to a given application. For example, pumps in certain paper manufacturing facilities fail due to circumstances such as overheating and starvation, while the same pump in a lube oil system in an aluminum rolling facility may fail due to bearing wear or water contamination of the lube oil. These requirements and practical considerations make such detailed implementations too expensive to justify application to the BoP. At present, the current state of the art in prognostics is only practical for high capital cost equipment such as helicopters, turbines, etc.

In fact, the research in this area has little hope for addressing the broader need for wide scale, automated, and broadly applicable diagnostics and prognostics because funding for such development is focused almost exclusively on high value assets, despite the fact that the BoP market dwarfs the rest of the industrial machinery market. As a point of reference, 10% of global electrical energy production is consumed by pumps, which are only one part of the BoP (ISO 10816, 2015).

4. IMPROVED CBM FOR THE BALANCE OF PLANT

SmartDiagnostics is a wireless vibration monitoring tool that is low cost and offers near real time monitoring, which in part enables continuous monitoring in the BoP. In particular, SmartDiagnostics provides near real-time monitoring at a cost point that is 10-20 times less than that of a wired system and a few orders of magnitude more data than a route-based approach.

![SmartDiagnostics Infrastructure Cost](image)

SmartDiagnostics infrastructure cost is low, based on innovation in low power wireless communication, M2M infrastructure, cloud computing, and simple web based user interfaces.

5. THE BIG DATA PROBLEM

To fully derive value from monitoring large numbers of equipment in near real time, the data measured using tools such as SmartDiagnostics, needs to be efficiently reduced and presented to the maintainer in a form that is easily usable. In a typical facility with 300 monitoring locations with sensors reporting 800 line vibration spectrum and time series on a 10 minute interval generates 69 million data points per day. Even if a diagnostic expert system with advanced vibration analysis algorithms is used, a host of indicators are required for each monitoring location. Monitoring the health results from 300 monitoring points with multiple indicators on a
frequent schedule is time intensive and impractical in most cases. For example, monitoring of 1500 indicators (5 per monitoring point) may be required as machines move through temporary conditions, like during start up, is a common scenario. This would generate hundreds of alarm email notifications, overwhelming users.

Customization of the analysis and health indicators can be done to help automate the diagnostics, using certain expert diagnostics solutions. However, this configuration exercise often results in a brittle diagnostic system which only works in certain cases. It also is time consuming to implement in most cases, which erodes the cost-value case for this technology.

In these cases, a generalized indicator is helpful for early warning indication of the health of the equipment. This reduces the number of early warning indicators that are required and simplifies monitoring. The early warning is then used to trigger in-depth analysis if a problem is identified. Peak and RMS indicators are often used for this purpose. In fact, these two simple time series indicators are used far more than any other machine health indicator in BoP monitoring due to the aforementioned practical CBM implementation challenges. However, Peak and RMS indicators lack capability to account for impulse behavior, frequency of load reversals cycles, and nonlinearity of the increase in damage relative to an increase in vibration amplitude.

6. DAMAGE ACCUMULATION: A NEW MACHINE HEALTH INDICATOR

The new Damage Accumulation indicator is a stand-alone composite indication of the “real” damage that a machine is encountering at a given point in time. This is analogous to the RealFeel temperature which takes into account humidity and sunshine in a temperature measurement. Damage Accumulation uniquely accounts for vibration changes that a number of other indicators identify, include Peak and RMS, while also accounting for impulse behavior, frequency of load reversals cycles, and nonlinearity in the damage to amplitude relationship.

Damage Accumulation not only is useful as a generalized diagnostic indicator, it addresses the need for real time operational monitoring. For example, Damage Accumulation can indicate changes in machine load which are often overlooked in detailed analysis such as alignment or bearing analysis.

Operational monitoring is important because load and speeds can be adjusted in many cases if its effect on the machine life is estimated and understood. This is a key value in real-time, persistent monitoring versus route-based monitoring because the critical data in this type of analysis occurs during unexpected conditions that are not known apriori and, therefore, not selected for analysis using route monitoring.

Operational monitoring is differentiated from machine diagnostics because precursors relating to the operating conditions that lead to the initiation of faults are monitored rather than the faults themselves. Such precursors are generally unexpected loading conditions that lead to damage.

Damage Accumulation leverages the idea that machines fail as a result of repetitive applied loading that slowly leads to fault growth. In other words, most machines fail at a fundamental level due to fatigue or weakening of a component’s material caused by repeatedly applied loads. For example, most bearing damage initiates in this way, unbalance and misalignment progresses due to intermittent excessive loading, weld joints and bolts fatigue and crack, and many other forms of wear are initiated by cyclic overloading. The nominal maximum stress that causes such damage may be much less than the ultimate or even yield strength of the material. In these cases, microscopic cracks form at the stress concentrations near the surface at persistent slip bands (PSBs) and grain interfaces (Manson & Halford, 2006). Eventually a crack will reach a critical size, and propagate until fracture. This general fatigue damage growth behavior links most machine faults and adverse operating conditions, thereby providing an opportunity to create a generalize indicator, Damage Accumulation, and use it universally, in a similar way to how Peak and RMS are used.

Damage Accumulation builds on this fundamental cyclic loading failure concept and extends it generally to mechanical systems. Although not all mechanical systems fail due to traditional fatigue, the vast majority of faults grow in a way that matches this underlying material degradation mechanism. This approach accepts the approximate nature of vibration-based, fatigue monitoring because often the objective of this monitoring approach is to address the adverse conditions that lead to failure rather than estimate the precise failure time.

The generalization and broadness of the applicability of damage growth monitoring is clear when considering a general form where $L$, the life given by the number of repeated stress cycles before failure is inversely related to the applied stress or load, $\tau$, raised to a power, $n$, which is a component or material property.

$$ L \sim \frac{1}{\tau^n} \quad (1) $$

This relation, when shown on log-log scale is straight line. For most materials, decreasing stress levels have a diminishing impact on life, which is accounted for in some cases by a fatigue stress limit ($\tau_0$), where the component life is given by:

$$ L \sim \frac{1}{(\tau-\tau_0)^n} \quad (2) $$

In these cases, the probability density function can be used to calculate the number of events which occur at a load, $\tau$, based on the number of load reversals, $N$.

$$ N \sim \int_{\tau_0}^{\tau} f(\tau) \, d\tau \quad (3) $$

where $f(\tau)$ is the probability density function of the load distribution. This can be used for the paper to identify the number of load reversals which occur at a given load level, $\tau$, and the number of load reversals which occur at a given number of cycles, $N$. This can be used to calculate the number of load reversals which occur at a given load level, $\tau$, and the number of load reversals which occur at a given number of cycles, $N$.
Notice that the damage is not proportional to the stress amplitude, rather it is proportional to amplitude raised to a power that typically depends on material properties. The exponent is important because small increases in loading can result in large increases in damage. For example, a few short impulse with a 10 g amplitude, which is often filtered out and ignored in frequency analysis, can be responsible for more damage than a 1 g continuous vibration that occurs over a few hours of operation.

The application of fatigue life relationships to machine component life is common and is exemplified by ISO Standard 281 for rolling bearings, which uses this same basic relationship. This standard specifies that bearing life is given by the following relationship (Zaretsky, 2010)

\[ L_{10} = \left( \frac{C_D}{P_{eq}} \right)^p \]  

(3)

In Eq. 2, \( L_{10} \) is the point at which a bearing has a 90% probability of survival or a 10% probability of failure, \( C_D \) is the dynamic load capacity or the theoretical load on the bearing that will result in an \( L_{10} \) life of one million inner-race revolutions, \( P_{eq} \) is the radial load on the bearing, and \( p \) is the load-life exponent.

Some vacuum-processed steels, such as AISI 52100, exhibit load-life exponents of 4 for ball bearings and 5 for roller bearings, instead of the typical value of 3 given in the ISO standard (Zaretsky, 2010).

Notice the similarities between Eq. 1 and 2, where the ratio of \( C_D \) to \( P_{eq} \) is a relative bearing load value (analogues to \( 1/\tau \)) and \( p \) is related to the bearing properties (analogues to \( n \) in Eq. 1). The underlying similarity to fatigue damage growth exists because bearing surface fatigue damage eventually leads to small cracks that in turn lead to pitting, excessive bearing wear, and eventual failure (ISO 281, 2007).

### 6.1. Vibration as a Relative Measure of Cyclic Stress

The Damage Accumulation formulation is derived from the same power relationship but uses the following high-cycle (materials behaving elastically) metal fatigue form, commonly known as the Basquin equation (Basquin, 1910).

\[ N^i = \left( \frac{s^i_1}{s_f} \right)^{\frac{i}{b}} \]  

(4)

In this relationship, \( N^i \) is the number of stress reversal cycles to failure for a given loading \( i \), \( s^i_1 \) is the alternating stress, \( b \) is the fatigue strength exponent, and \( s_f \) is the fatigue strength coefficient and is approximately equal to the true fracture stress. This is simplified by generalizing the relative load parameter to \( A = s_f^m \) and define a load factor \( m = -1/b \) in the following equation

\[ N^i \equiv A(s^i_1)^{-m} \]  

(5)

Similar to the simple form in Eq. 1, the exponent \( m \) is application dependent, greater and zero, and typically ranges between 1 and 6 for mechanical structures and systems (ISO 281, 2007; Ragan & Manuel, 2007; Rychlik, 2013; Giyanani, 2013).

<table>
<thead>
<tr>
<th>( m )</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>standard value for crack-growth-dominated process, applicable to sharply notched or welded structures(^{10,11})</td>
</tr>
<tr>
<td>5</td>
<td>valid for many engineering components</td>
</tr>
<tr>
<td>6</td>
<td>glass fiber(^12)</td>
</tr>
</tbody>
</table>

Considering Eq. 5, fatigue damage monitoring requires strain or stress measurements at the critical points where failure is expected to occur or at key locations that can be used to characterize the stress state of the component. Measuring strain sensors is impractical in an industrial setting because
of the high cost of installing and sustaining wired strain sensors. In certain cases, such as helicopter rotor monitoring, strain sensors have been placed at critical locations that can be used to determine the stress or load state of the component.

Using tools such as SmartDiagnostics, vibration can be much more easily and cost effectively monitored than strain or stress. Relating vibration measurements to localized stress at the essential target failure location requires a transfer function. To this end, the following simple linear relationship between acceleration and stress can be used in many cases,

\[ s_a^l \propto a^l \] (6)

where \( a^l \) is the vibration amplitude for a given cycle. For example, dynamic loading from shaft unbalance or misalignment can cause a machine to vibrate. The acceleration measured on the exterior of the machine is loosely linked to localized stress through Newton’s second law. This is valid if the machine components are massive, are elastically linear, and have low material damping. In most cases, the critical stress at the critical fault location is proportional to the inertial forces that in turn can be estimated based on acceleration measurements. For this to be valid, the vibration measurement needs to be directly in or near the transmission path of the dynamic loading that is causing the stress variations. This generalization for relating stress to vibration is necessary to simplify the application of Damage Accumulation.

If the accelerometer cannot be located near the dynamic loading transmission paths, then a more comprehensive modeling approach may be required. This is because the local vibration modes of the structure may create a frequency dependence in the relationship between acceleration and stress. In addition, if multiple load transmission paths to the critical stress location exist, then multiple vibration monitoring points may be required. The relationship between stress and acceleration at multiple points \( (a_1, a_2, \ldots) \) can then take on the following linear form

\[ s_a = \sum_k a_k H_k(f) \] (7)

where the transfer function \( H_k(f) \) is a frequency dependent parameter. Other models that account for nonlinear weighting of the contribution of the vibration measurements can also be used in certain cases. In the case where one acceleration input is used, the transfer function could be measured practically using a strain measurements and the accelerometer. Often, this is a challenge because the critical stress is in a difficult to reach location. If multiple acceleration measurements are used as inputs, determining the transfer functions is a more involved process. In complex structures with multiple failure modes and multiple critical failure locations, a damage factor for each one may be required.

In addition, depending on the structure between the sensor (acceleration measurement location) and the critical stress location, displacement or velocity may relate more closely to stress rather than acceleration. This can be determined by considering the basic architecture of the structure. Three scenarios for relating stress to acceleration, velocity and displacement are shown below.

![Conceptual model for associating vibration and stress](image)

Figure 7. Conceptual model for associating vibration and stress.

\[ s \sim F \sim m\ddot{x} \sim a \]
\[ \sim c\dot{x} \sim v \]
\[ \sim kx \sim d \] (8)

In most cases, the structure that the sensor is mounted to is massive relative to the mount stiffness and internal damping or mounting damping forces of the structure, and, therefore, acceleration is used to estimate Damage Accumulation (previously shown in Eq. 6). In certain other cases where damping forces are significant, such as in journal bearing and some soft foot cases, velocity is a better basis for calculating Damage Accumulation. In rare cases, the structural loading is driven by stiffness and displacement and is used in the Damage Accumulation calculation. This is generally only useful for low speed excitation and light-weight or truss structures.

The significance of high frequency content is an additional factor that can be used to help decide whether to use acceleration, velocity, or displacement. Acceleration emphasizes the contribution of high frequency vibration, while, conversely, displacement emphasizes the contribution of low frequency vibration.
6.2. Damage Model

This paper focuses on the case where the stress is proportional to acceleration as shown in Eq. 6. In this case, Eq. 6 can be substituted into Eq. 5. Lumping the machine specific characteristics in the constant $C$, which is equivalent to $A$ in Eq. 5, gives the following generalized estimate for component life.

$$N^i \cong C(a^i)^m$$  \hspace{1cm} (8)

In Eq. 8, consideration of the fatigue life limit or endurance limit is not included. One way to account for the fatigue life limit is to subtract a threshold acceleration amplitude, $a_t$ from each cycle. This method also mitigates the erroneous influence of noise because cycles below the endurance threshold $a_t$ do not contribute to the damage estimate estimation.

$$N^i \cong C(a^i - a_t)^m$$  \hspace{1cm} (9)

Selecting the endurance threshold $a_t$, requires engineering judgement unless the stress-acceleration transfer function is measured and valid for the specific application. Considering the effort required to determine $a_t$ and that it is not accounted for in bearing life calculations (ISO Standard 281), $a_t$ should be chosen to be small, but larger than the noise floor of the sensor.

This analysis does not consider the mean stress on a component because the measured DC acceleration signal is not necessarily referenced to a neutral stress state. Compensating for mean stress would shift the fatigue life curve in Figure 5 vertically. The shift direction depends on the sense of the stress. In this analysis, mean stress is not accounted for.

Borrowing again from fatigue analysis methods, Miner’s rule can be applied in conjunction with Eq. 9. Miner’s rule states that part life is decremented by a linear combination of cycles for each given load. This approach allows load cycles of different amplitudes to be treated independently and summed to provide a composite damage estimate as follows:

$$D = \sum_i D^i$$

where

$$D^i = \frac{n^i}{N^i}$$  \hspace{1cm} (10)

And substituting Eq. 9 into Eq. 10 gives

$$D \cong \frac{1}{C} \sum_i n^i(a^i - a_t)^m$$  \hspace{1cm} (11)

where $n^i$ is the number of cycles at a given load condition. Component failure occurs when $D$ approaches a value of 1.

Because this method presumes no absolute knowledge of the machine mechanics to determine the parameter $C$ and the prior loading and maintenance history is not known, the damage accumulation rate relative to a known, baseline damage rate $D$ is used for computing $R$, the Damage Accumulation, as

$$R \cong \frac{a}{\Delta t} \left( \frac{D}{a^m} \right)$$  \hspace{1cm} (12)

SmartDiagnostics captures a time series block of vibration data every few minutes. This intermittent nature of the acquisitions offers the opportunity to assess damage rate each time a new block of data is received. The damage and baseline rates are calculated as a sum of the loading cycles raised to the $m$ power divided by the sample duration $\Delta t$.

The Damage Accumulation metric is then given by

$$R \cong \frac{1}{\Delta t} \sum_i n^i(a^i - a_t)^m / B$$

where the baseline is calculated as

$$B = \frac{1}{\Delta t_b} \sum_i n^i(a^i - a_t)^m$$

The baseline $B$ may be averaged over multiple blocks of data to increase accuracy. Similarly, the rate $R$ can be averaged over a moving period that is specified to be less than the load or speed variation of the machine. This is an adjustment that can be set for each particular application. A Rainflow counting algorithm such as that specified in ASTM Standard E1049 -85 can be used for digesting a signal into cycles at each load amplitude (ASTM E1049, 2011).

One important distinction between Damage Accumulation and other time series analysis indicators (i.e., Peak and RMS) is its relative nature, which therefore necessitates use of a baseline. A baseline is important in most cases anyway because most installations vary considerably from their normal operations with prescribed acceptable vibration levels, making standards such as ISO Standard 10816 impractical to use.

The seemingly arbitrary selection of the endurance threshold, the transfer function between vibration and critical stress, and the baseline is a reality in the practical application of Damage Accumulation. This is acceptable because vibration diagnostics is probabilistic, and approximate since vibration
characteristics are symptoms that are not part of an exclusive causal relationship with most faults. In general, the approximate nature of the Damage Accumulation analysis is in line with the overarching diagnostics limitations of vibration monitoring.

### 7. Damage Accumulation in Practice

The usefulness of Damage Accumulation as a key indicator of machine faults or operational problems is clarified in the following test bed example and in a high pressure pump field application. The test bed shown in Figure 8 consists of a motor, gearbox, shaft and pulley driveline. It includes a controllable shaft unbalanced, a flexible shaft coupling with adjustable parallel shaft misalignment, and an adjustable soft foot.

![Sensor location](image)

**Figure 8.** Bench test bed for fault testing.

#### 7.1. Bench Demonstration

Vibration data was acquired using the SmartDiagnostics sensor located on a pillow block. The sensor acquired an 800 line spectrum and used a sampling frequency of 1024 Hz. An \( m \) (recall the load parameter in Eq. 5) of 3 was used in all demonstrations.

Damage Accumulation’s application for identifying and quantifying the severity of knocking or impulse behavior is considered first. This type of behavior is often observed in reciprocating equipment or machines with loose components. Traditional frequency domain and time series methods do not accurately characterize the damage created by impulse shock loading or severity of vibration of this type (ISO 10816, 2015). Impulse vibration cases are considered in this analysis to emphasize the broad applicability of Damage Accumulation compared to traditional vibration monitoring using Peak, RMS, or frequency spectrum analysis. Two cases were considered: one with an infrequent impulse (Case I) and one with a frequent periodic impulse (Case II).

![BASELINE](image)

**Figure 9.** Machine condition baseline and shock impulse fault cases.

The change in the RMS, Peak, 1X, and the Damage Accumulation amplitude relative to the baseline is shown in Table 2. The 1X value is found by performing a Fast Fourier Transform (FFT) of the time series using a Flat Top window, and evaluating the amplitude in a narrow frequency band corresponding to the fundamental (1X) running speed of the machine. Damage Accumulation is calculated using Eq. 13.
Table 2. Comparison of Damage Accumulation to Peak, RMS, and 1X for fault Case I, II.

<table>
<thead>
<tr>
<th>Increase Relative to Baseline</th>
<th>Case I</th>
<th>Case II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage Accumulation</td>
<td>186%</td>
<td>884%</td>
</tr>
<tr>
<td>RMS</td>
<td>15%</td>
<td>68%</td>
</tr>
<tr>
<td>Peak</td>
<td>354%</td>
<td>318%</td>
</tr>
<tr>
<td>1X</td>
<td>1%</td>
<td>19%</td>
</tr>
</tbody>
</table>

In these example cases, the value in Damage Accumulation as a generalized indicator is demonstrated relative to traditional metrics including RMS, Peak, and 1X, based on the following observations:

1) Damage Accumulation increases as a result of short duration impulses, thereby characterizing the damaging impulse loading.

2) The higher Damage Accumulation for Case I than Case II accurately captures the higher level of damage caused by frequent impulses. Note that the Peak measurement doesn’t capture this difference.

3) The small 15% change in the RMS value for Case I indicates that it is a poor metric for identifying and monitoring impulse behavior.

4) Frequency spectrum analysis (1X amplitude in this case) is only useful when the impulses occur frequently or on a consistent basis where envelope analysis can be used.

Damage Accumulation is also valuable for monitoring common machine faults such as unbalance, misalignment, and soft foot. Frequency spectrum and key indicators are shown in Figure 10 for these three faults. In this analysis, the sampling frequency was set to 256 Hz with 800 lines of spectral resolution.

Any number of indicators can be used to identify the presence of these vibration faults including Peak, RMS, vibration spectral bands (1X, 2X, and 3X). The 3X band, rather than the more typical 2X band, shows elevated levels for the misalignment case due to the particular design of the 3-finger flexible shaft coupler. The frequency band analysis is performed as previously mentioned for fault Cases I and II. However, the results in Figure 10 emphasized that Damage Accumulation identifies the presence of impulsive behavior as well as common faults, which enables broad applicability of Damage Accumulation for a diverse range of machines and machine failure modes. This is not surprising considering the general nature of Damage Accumulation’s fatigue life foundation.

Figure 10. Induced fault results for unbalance, misalignment, and soft-foot cases.
7.2. Field Example

The efficacy of Damage Accumulation was further evaluated for identifying and quantifying the severity of adverse operating conditions in high pressure pumping applications. In particular, it was applied to hydraulic fracturing in oil and natural gas well completion where pump components rapidly wear and contribute to high downtime costs and safety incidents. Pump fluid ends exhibit valve wear, plunger seal (packing) failure, and in certain cases, fluid manifold cracking. In addition, fluid-induced mechanical resonance in the high pressure piping can cause large transients where the pipes vibrate violently. This leads to excessively high bending loads in unions and mounts, accelerated fatigue, and ultimately high-pressure leaks or catastrophic mechanical failure.

Hydraulic fracturing pump and, in particular, fluid end wear are caused by complex combinations of low suction pressure, excessive outlet pressure, excessively high or low pumping speeds, valve wear, packing failure, debris, damper malfunction, and interdependent adjacent pump interaction. One of the main underlying effects of these problems is intermittent cavitation and resulting high pressure impulses in the fluid, causing high wear and fatigue rates. Such impulses dramatically shorten pump fluid end life from an expected life of 1000 hrs. to as little as 350 hrs.

In this particular application, the pump speeds and pressures are only controlled within predefined ranges for the purpose of achieving the intended hydraulic fracturing outcome in the well. Machine and component wear is rarely considered and or even evaluated as an input to help define how to operate the pumps. Damage Accumulation is useful in this application because it enables a useful estimation of the severity of impulse pumping behavior. This information is used in real time to fine tune the pump speeds and pressures, diagnose problematic components (damper malfunction, blocked suction lines, etc.) and avoid high severity conditions.

Hundreds of SmartDiagnostics accelerometers have been used to monitor hydraulic fracturing pumping operations. The following plot shows a typical impulse as measured using a SmartDiagnostics wireless accelerometer mounted on the outside of the pump fluid end. The Baseline or normal pump acceleration time series is shown in the lower plot. The upper, problematic time series has a Damage Accumulation of 16.2 (1620%), which is relative to the baseline of 1.0 for the lower plot. This high level indicator value for this particular data set accurately shows the true severity and expected damage that these pumps are incurring. An $m$ value of 3 was used in this evaluation. This particular impulse occurred at the start of a fracturing stage where the pumps were being brought up to their full operational speed. This impulse behavior can be minimized or avoided by tailoring certain pump conditions at startup. Without this type of monitoring, the wear caused by this type of impulse goes unnoticed.

Large savings are realized by using Damage Accumulation monitoring in these applications because fluid ends cost on an order of $70,000 and downtime typically costs $20,000/hr. Over a 6 month period, SmartDiagnostics and Damage Accumulation were shown to reduce Non-Productive Time (NPT) by 2 hours per day, saving $40,000 per day in certain applications.
8. CONCLUSION

SmartDiagnostics wireless continuous monitoring and its complementary Damage Accumulation analysis technique were shown to address challenges in applying continuous health monitoring to the Balance of Plant. Damage Accumulation efficiently reduces large volumes of machine health data and offers a single parameter indication of machine health status to the maintainer, which is essential to realize the high value associated with Balance of Plant health monitoring. This Damage Accumulation monitoring approach trades precision and the ability to classify faults for universal applicability and ease of use, which is an essential technique to practically monitoring the Balance of Plant. Damage Accumulation avoids, for example, the brittleness of expert diagnostic systems, and the setup and user complexity overhead associated with customized machine analysis implementation. In these cases, Damage Accumulation is used to trigger a detailed inspection, fault classification, and diagnosis.

Damage Accumulation is broadly applicable because its foundation is rooted in the fundamental and universal idea that most machines fail as a result of repetitive applied loading that slowly leads to fault growth. In other words, most machines fail, at a fundamental level, due to fatigue or weakening of a component’s material caused by repeatedly applied loads. Relative to traditional vibration analysis, Damage Accumulation uniquely accounts for the contribution of repeated load reversal cycles to component damage and the nonlinearity in the relationship between damage and vibration amplitude.

In this paper, Damage Accumulation was shown to indicate the presence of unbalance, misalignment, and soft foot faults on a test stand. This indicator was further evaluated on a high pressure hydraulic pump. In these cases, it more comprehensively captured the machine faults than traditional time series and frequency spectrum analysis techniques, thereby showing that Damage Accumulation simplifies the application of continuous monitoring. In these cases, it also was shown to recognize commonly overlooked time varying symptoms in machines which are not properly handled by traditional vibration diagnostic frequency analysis or time series analysis.

It is worth noting that Damage Accumulation is a simple time domain calculation that can be performed using low power microprocessors.

REFERENCES

ASTM E1049 -85 Standard practices for cycle counting in fatigue analysis (Reapproved 2011).
Harris Tedric A Rolling Bearing Analysis. John Wiley and Sons, New York 1984
IEA Report 2009, IEA Statistics, CO2 Emissions from fuel combustion
Battery Capacity Anomaly Detection and Data Fusion

John Weddington¹, Wuzhao Yan¹, Wanchun Dou², and Bin Zhang¹

¹Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA
weddingj@email.sc.edu
wyan@email.sc.edu
zhangbin@cec.sc.edu

²Department of Computer Science & Technology, State Key Lab. For Novel Software Technology, Nanjing University, Nanjing, Jiangsu, 210093, P.R. China
douwc@nju.edu.cn

ABSTRACT
Anomaly detection is a critical enabling technique of PHM, especially in safety critical applications. In order for the PHM system to begin prediction of remaining useful life of a given system or component, the fault must be detected. This paper presents an integrated anomaly detection system for state-of-health of lithium-ion batteries. Two algorithms for state estimation and anomaly detection are used: the extended Kalman filter and the particle filter. A Dempster-Shafer Theory-based fusion approach is implemented to reduce the uncertainty of detection. The results on battery data show that the fusion improves the detection results significantly.

1. INTRODUCTION
There is an increasing need in engineered systems of all types for detection and handling of faults. Traditional engineering includes receiving a product or system specification, the design process, analyzing the design for accuracy, redesigning components that are inaccurate, testing the design to standard, final redesign for production, and validation and verification. In the ideal scenario, the engineering design process filters out discrepancies, incorporates all known data and techniques in an accurate and robust manner, considers all possible failure modes, and allows engineers to reasonably predict the total life of the engineered product or system. However, no matter how well the system is designed, manufactured, and maintained, faults may occur in any given system due to harsh environment, manufacturing flaws, and excessive operational conditions. Current industry trends continue to drive the need for advanced understanding of engineered systems, and appropriate warning when the functionality of these systems deteriorates. Even if the engineering process has incorporated fault-tolerant design practices, it is important to implement fault-tolerant design practices, and detect faults in the system for diagnostics and handling of faults.

Diagnosis consists of three parts: detecting, isolating, and identifying faults in order for the end user to make the best decision for the intended application. There are both model-based and data-driven diagnostic approaches. First-principles models require comprehensive system analysis to develop an accurate model, which is difficult for complex systems. Data-driven approaches, on the other hand, rely on large amounts of data that are generated with fault modes of interest and are statistically sufficient to design diagnostic systems (Wu, Zhu, Ge, & Zhao, 2015). Neuro-fuzzy methods have become popular in recent years for modeling systems, but the limitation for many applications is that no sufficient fault data is available for modeling (Viharos & Kis, 2014). This paper addresses the detection piece of diagnosis, and does not attempt to isolate or identify the fault type. The algorithms presented flag the user when an anomaly is detected.

On-board lithium-ion (Li-ion) battery detection involves the use of state estimation techniques (Orchard, Hevia-Koch, Zhang, & Tang, 2013; Zhang, Tang, DeCastro, Roemer, & Goebel, 2014). In the framework of Bayesian theory, state estimators are used to make one-step predictions of the current fault dimension, which is then filtered when the new measurement becomes available to obtain the posteriori fault dimension distribution. The posteriori distribution is then compared with a baseline distribution (established from the data of healthy system) to calculate the probability of detection. There are a number of different state estimators including wavelet analysis (Kim & Cho, 2015), Kalman filter (KF) (Gadsden & Habibi, 2011), multiple model adaptive estimation (Singh, Izadian, & Anwar, 2013; Wu, et al., 2015), extended Kalman filter (EKF) (Sidhu, Izadian, & Anwar, 2015), particle filter (PF) (Chen, Zhang, Vachtsevanos, Orchard, 2011; Chen, Brown, Sconyers, Zhang,
Vachtsevanos, Orchard, 2012; Goebel, Saha, Saxena, Celaya, & Christophersen, 2008), autoregressive integrated moving average (Box, Jenkins, & Reinsel, 2008; Saha, Goebel, & Christophersen, 2009).

After a fault is detected, prognosis is executed to estimate the time to failure. Each estimator differs from the other by offering better performance in different aspects. For example, the EKF is founded upon the efficient, optimal KF, which is able to accurately produce the underlying state for linear systems. The PF is a complex sampling algorithm that discovers the underlying state through many calculations, but it is well-suited for nonlinear systems. In order to utilize the best from each algorithm their results must be fused in some manner. Uncertainty management is a major consideration in real-world, online systems, especially in safety-critical applications. Each state estimation technique rolls into its estimate an undesirable level of uncertainty, and if fused improperly, the uncertainty could undermine the usefulness of the estimate.

This paper expands upon these concepts by combining the results from multiple state estimators and applying anomaly detection (AD) to the fused battery capacity estimation. First a fault dynamic model is developed. Using this model, EKF linearization equations can be created and integrated into the framework, along with the EKF and the PF estimation algorithms. As the data is passed into the algorithm, it passes separately through the EKF and the PF. Probability density functions (pdfs) are generated for each state estimation and are each fed into the Dempster-Shafer (DS) fusion algorithm. DS theory is well-suited to combine multiple sources in a neat framework. As will be demonstrated, the fusion of each state estimation results in tighter confidence bounds and reduced uncertainty, which is desirable in safety-critical applications.

The paper is organized as follows: The approach is developed from prior literature in section II. Section III details the implementation and experiments. Sections IV and V summarize the results and suggest expansion to the implementations described in this paper, respectively.

2. APPROACH DEVELOPMENT

Batteries are temporary power sources with limited lifespan. Over time, they experience degraded performance as a result of normal use, as well as accelerated degradation under strained operating conditions. This section details a method for AD of Li-ion batteries by combining the EKF and the PF to reduce the uncertainty inherent in each state estimator. Figure 1 demonstrates the flow chart used in designing the proposed AD system. Online fault detection is performed in a recursive manner by streaming in each sample, running each sample through the EKF and the PF separately, and then fusing the results from each algorithm. AD is performed on the fused result and a decision is made declaring either a healthy or a faulty battery. The AD algorithm used is a standard detection algorithm that flags anomalies when the baseline distribution and real-time fault state distribution are deviated a specified percentage. This paper will focus on three key portions of the AD design: modeling, algorithm development (EKF and PF), and data fusion.

![Flowchart for fused estimation and AD](image)

**2.1. Extended Kalman Filter**

For state estimation, the KF is known as a linear quadratic optimal filter. This algorithm is a linear, discrete time, finite dimensional time-varying state estimator that minimizes the mean-squared error (MSE) (Ribeiro, 2004). Capacity degradation of Li-ion batteries is a non-linear process, for which KF is insufficient and necessitates the use of EKF. The EKF expands the KF to incorporate non-linear dynamics, by linearizing around the current state, as described by the mean and covariance. The nonlinear process model (from cycle $k$ to cycle $k+1$) is described as a hidden Markov model (HMM) by

$$x_{k+1} = f(x_k, u_k) + w_k$$

where $x_k$ and $x_{k+1}$ are the features (vector) at the current time instant, $k$, and the next time instant, $k+1$, $f$ is a fault growth model, $u_k$ is the operating condition, and $w_k$ is a zero-mean Gaussian noise. The observation model is

$$z_k = h(x_k) + v_k$$

where $z$ is the observation vector, $h$ is a nonlinear observation function, and $v_k$ is the zero-mean Gaussian noise (Huang, 2010). EKF is evaluated in two stages: prediction and update. Both the prediction and update steps require the calculation
of the partial derivatives (the Jacobian) of \( f(x) \) and \( h(H) \), as shown below:

\[
\text{State Jacobian: } F_k = \frac{\partial f}{\partial x} \bigg|_{x_k, u_k} \\
\text{Observation Jacobian: } H_k = \frac{\partial h}{\partial x} \bigg|_{x_k}
\]

The final equations for the EKF prediction and update steps after derivation are

\[
\begin{align*}
\text{State Prediction: } \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_{k+1}) \quad (5) \\
\text{Prediction Covariance: } P_{k+1|k} &= F_k P_{k+1|k} F_k^T + Q_k \quad (6) \\
\text{State Update: } \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_k \left( z_k - h(\hat{x}_{k+1|k}) \right) \quad (7) \\
\text{Innovation Covariance: } P_{k+1|k+1} &= (I - K_k H_k) P_{k+1|k} \quad (8) \\
\text{Kalman Gain: } K_k &= P_{k+1|k} H_k^T \left( H_k P_{k+1|k} H_k^T + R_k \right)^{-1} \quad (9)
\end{align*}
\]

**2.2. Particle Filter**

One disadvantage of the EKF is that the desired pdf is estimated by a Gaussian, and as the system under study is not normally distributed, the estimation could deviate from the actual distribution and diverge (Goebel, et al., 2008; Zhang, Khawaja, Patrick, & Vachtsevanos, 2010; Zhang, Sconyers, Byington, Patrick, Orchard, & Vachtsevanos, 2011). The PF is developed as a solution for nonlinear random non-Gaussian systems. The algorithm assumes the process state equations can be effectively modeled as a first-order nonlinear Markov process in Eqs. (1) and (2). The state \( x \) and observation \( z \) for cycles 1 to \( k \) are defined as

\[
x_{0:k} \triangleq \{x_0, x_1, \ldots, x_k\}, \quad z_{0:k} \triangleq \{z_0, z_1, \ldots, z_k\}
\]

The PF is also known as a sequential Monte Carlo method for state-space inference. PF is able to accommodate nonlinearities easily, provided enough particles are used. The particle filter uses a set of weighted particles to estimate the current and future state based upon a nonlinear fault dynamic model. The algorithm begins with a set of \( N \) particles available at cycle \( k-1 \) sampled from the target distribution \( \pi_k \), as defined below,

\[
\{x_{0:k-1}^{(i)}\}_{i=1, \ldots, N} \sim \pi_k(x_{0:k}) \quad (11)
\]

where the target distribution is defined as the \textit{a posteriori} distribution of \( x_{0:k} \) in Bayesian filtering,

\[
\pi_k(x_{0:k}) = p(x_{0:k} | z_{1:k}) \quad (12)
\]

The objective of filtering is to obtain a set of \( N \) new particles and this set of particles is distributed according to the target distribution at cycle \( k \). To obtain these new particles, a known stationary distribution is chosen by the user to be the proposal or importance distribution. A set of \( N \) particles are selected from the proposal distribution, as shown,

\[
\{x_{0:k-1}^{(i)}\}_{i=1, \ldots, N} \sim q_k(x_{0:k-1}^{(i)}) \quad (13)
\]

and compared against the target distribution. The true distribution is approximated by a set of \( N \) weighted particles,

\[
\sum_{i=1}^{N} w_k^{(i)} \phi_k \left( x_{0:k}^{(i)} \right) \rightarrow \int \phi_k \left( x_{0:k} \right) \pi_k \left( x_{0:k} \right) dx_k \quad (14)
\]

The sum of these weights is equal to 1. To get the consistent estimate of posterior distribution, importance sampling corrects the difference between \( q_k \) and \( \pi_k \), by setting weighting factors for each particle, which is given by:

\[
\bar{w}_k \left( x_{0:k}^{(i)} \right) = \frac{p(x_{0:k}^{(i)} | z_{1:k})}{q_k \left( x_{0:k}^{(i)} \right)} \quad (15)
\]

and is normalized as

\[
w_k \left( x_{0:k}^{(i)} \right) = \frac{\bar{w}_k \left( x_{0:k}^{(i)} \right)}{\sum_{j=1}^{N} \bar{w}_k \left( x_{0:k}^{(j)} \right)} \quad (16)
\]

With this new set of weights, the target distribution can be approximated as:

\[
\pi(x_{0:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta \left( x_{0:k} - x_{0:k}^{(i)} \right) \quad (17)
\]

In a simple case of particle filter, Bootstrap filter, the importance density function is set as the \textit{a priori} pdf,

\[
q_k \left( x_{0:k} \mid x_{0:k-1} \right) = p(x_k \mid x_{k-1}) \quad (18)
\]

In this setting, the weights for the newly generated particles are proportional to the likelihood of new observations, i.e.

\[
w_k^{(i)} = w_k^{(i)} \cdot p(z_{1:k} \mid x_{0:k}^{(i)}) = w_k^{(i)} \cdot p \left( z_k \mid x_{k}^{(i)} \right) \quad (19)
\]
In particle filters, degeneracy is a problem that must be addressed. Degeneracy can be described as the decreasing number of more heavily weighted particles as sampling continues to be performed. This leads to a dominance of particles with small weights describing the distribution. In practice, this results in inaccurate estimation of the actual state. Degeneracy is addressed by resampling. Resampling effectively replaces the smaller-weighted particles with larger-weighted particles, so as to describe to true distribution with higher veracity (Arulampalam, Maskell, Gordon, & Clapp, 2002). Sequential Importance Resampling (SIR) is the PF implementation chosen for this application due to its robustness. The variance of the weighted particles generated by the Bootstrap filter is calculated using an effective sample size:

\[ \hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}(w_i^{(t)})^2} \]  

When \( \hat{N}_{\text{eff}} < N_{\text{threshold}} \), the particles are resampled to eliminate particles with small weights. The steps in SIR are included in Figure 2.

**Sequential Importance Resampling (SIR) Steps**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compute ( N^v = \left\lfloor N \cdot w^v \right\rfloor ), this is, for each particle take the integrate part of the product. This identifies the particles with good weights and the number of times it should be kept in new samples.</td>
</tr>
<tr>
<td>2</td>
<td>Compute ( N = N - \sum N^v ), which indicates the number of particles to be resampled.</td>
</tr>
<tr>
<td>3</td>
<td>Compute ( N_{\text{res}} = \frac{N \cdot w^v - N^v}{N} ) and compute its cumulative sum.</td>
</tr>
<tr>
<td>4</td>
<td>Generate ( N ) random numbers within ([0,1]) and the range of these numbers that belong to ( N_{\text{res}} ).</td>
</tr>
<tr>
<td>5</td>
<td>The index of the range corresponds to the particle to be created.</td>
</tr>
<tr>
<td>6</td>
<td>Obtain the resampling index and resample the particles from ( q_k(x_{0:k}</td>
</tr>
</tbody>
</table>

**Figure 2. Steps in SIR for PF.**

### 2.3. Data Fusion

Data fusion is the process of combining the reasoning from different algorithms to achieve a result with high reliability. There are four prevalent methods for decision fusion: voting logic, possibility theory (fuzzy logic inference), probability theory (Bayesian fusion), and belief theory (DS-based fusion). Voting fusion is used heavily in industry with redundant instrumentation. Typical methods for voting fusion are threshold, majority, median, adaptive, inexact, and others (Parhami, 2005). Voting fusion is mostly used with crisp numbers, but its application has expanded into numerous domains, including dependable system design with unreliable components (Parhami, 2005).

Fuzzy logic inference expands voting into the domain of “gradual decision making” (Blank, Föhlst, & Berns, 2010). Fuzzy logic is also called soft voting, because the fusion rules are not crisp, but allow for many different scenarios. The task of fusion using fuzzy methods was demonstrated in (Russo & Ramponi, 1994). Depending upon the complexity of the problem, fuzzy logic may require extensive understanding of system dynamics to perform effectively.

Bayesian theory is based on probability theory or Bayes’ theorem. The theory relates evidence and belief using prior and current information. Being a proven method, it maintains a larger contingent of supported applications of the theory. This is due, in part, to its simple formulation, which enables it to be better understood (Challa & Koks, 2004).

DS builds upon Bayes’ theory by incorporating unknowns and confidence measures into its calculations. These “degrees of belief” lead to classifying DS under the heading “Belief Theory” or “Evidence Theory” (Carl, 2001). The DS has certain advantages over the Bayesian method under certain circumstance (Shafer, 1976; Shafer, 1990). It is based on the concept of assigning a degree of belief or confidence to certain propositions on the basis of combining all the available evidence. DS accommodates uncertainty by mapping a proposition into an internal value in \([0,1]\). The end points of the interval can be interpreted as the lower and upper probability functions. The fundamental equation in DS for belief is

\[ (m_i \oplus m_j)(C) = \frac{\sum_{A \cap B = C} m_i(A) m_j(B)}{\sum_{A \cap B \neq \emptyset} m_i(A) m_j(B)} \]  

where \( C \) is the output, \( A \) and \( B \) are hypotheses, and \( m_i \) and \( m_j \) are the masses for the hypotheses (Wu, Siegel, Stiefelhagen, & Yang, 2002). The sum of the masses in the numerator can be known as the belief measure in the hypothesis, while the denominator is called the plausibility measure. The belief and the plausibility provide confidence bounds, which, when combined as indicated in Eq. (21), supply degrees of belief for the output. The DS method for data fusion is selected for this paper due to its ability to incorporate unknown system dynamics in a flexible framework.

### 3. Implementation

In this section, the proposed approach will be demonstrated in a case study of the capacity degradation of a Li-ion battery. The battery is a safety-critical component that provides power to system functions including command, control, communications, computers, and intelligence. Li-ion batteries are widely used due to the advantages in higher...
energy density, longer cycle life, no memory effect, and lower weight (Saha & Goebel, 2009). Since the life and state of the batteries are not directly measurable, state estimation techniques play an important role in estimating the battery state-of-health and state-of-charge.

In this implementation, the state-of-health of a Li-ion battery with rated capacity of 1.1Ah is used to verify the proposed approach. The charge-discharge cycle of the battery is conducted with the Arbin BT2000 system under room temperature at a discharge current of 1.1A. The charging and discharging of the battery are halted at the given cutoff voltage. The capacity degradation curve versus charge-discharge cycle is obtained by Coulomb counting.

3.1. Model Development

The data used for model development and testing were drawn from (He, Williard, Osterman, & Pecht, 2011). As battery charge-discharge cycle count increases, the battery capacity slowly degrades from baseline in an exponential form. At some point, the fade-rate increases rapidly. Figure 3 shows the capacity degradation of four Li-ion batteries.

![Figure 3. Battery capacity test data, along with preliminary models and final model.](image)

The model used in Figure 3 was developed based upon the 4 data sets. The model was manually tuned using an iterative process. The fault dynamic model has the following form:

\[
\beta(k) = p_1 (p_2 + p_3 k + p_4 k^{p_5} + p_6 k^{p_7} + p_8 k^{p_9})^{p_{10}}
\]

where \( p = \{p_1, p_2, p_3, \ldots, p_{10}\} = [2.7e-3, 1e-7, 1e-4, 8e-8, 2.2, 8e-11, 3.25, 0.9, -0.11, 2.7] \) is the parameter vector, \( k \) is the time-index, and \( \beta \) is the model.

There are a variety of thresholds used in flag the current state as healthy or faulty. The baseline size is selected to be 50, and is the number of points in the feature that are considered standard, or in the case of battery health, the cycles that indicate a healthy battery. The false alarm rate is selected to be 5% in this implementation.

3.2. EKF Implementation

In order to implement the EKF, a proper model must be developed. The above section selects a basic polynomial regression model. This is suitable for our purposes and closely follows the fault mode of the battery capacity degradation. The following model for detection is augmented from the fault dynamic model in Eq. (22).

\[
\begin{bmatrix}
  x_{d,1}(k + 1) \\
  x_{d,2}(k + 1)
\end{bmatrix}
= f_k \begin{bmatrix}
  x_{d,3}(k) \\
  x_{d,4}(k) \\
  x_{d,5}(k) \\
  x_{d,6}(k)
\end{bmatrix} + n(k)
\]

\[
x_1(k + 1) = x_1(k) + \beta(k) \cdot x_{d,1}(k) + \omega(k)
\]

\[
y(k) = x_1(k) + v(k)
\]

\[
f_k(x) = \begin{cases}
  [1 \\ 0]^T, & \text{if } \|x - [0 \\ 1]^T\| \leq \|x - [0 \\ 1]^T\| \\
  [0 \\ 1]^T, & \text{else}
\end{cases}
\]

where \( f_k \) is a nonlinear mapping, \( x_{d,1} \) and \( x_{d,2} \) are Boolean states that indicate normal and faulty conditions, respectively. \( y(k) \) is the battery capacity from Coulomb counting, \( \omega(k) \) and \( v(k) \) are noise signals, and \( n(k) \) is i.i.d. uniform white noise. The Jacobian is then calculated for each iteration of new data. As each new data point becomes available, it is analyzed using the EKF. The covariance parameters \( Q \) in Eq. (6) and \( R \) in Eq. (9) are used to adjust the dependence of the state estimate upon the model or upon the measurements. The state estimation pdf for EKF has a wide base, which translates to lower confidence in the predicted state. Figure 4 shows the pdf for the EKF algorithm at cycle 274. Cycle 274 is the latest cycle for AD without fusion, which will be discussed later. EKF is closer to an optimal filter because of its basis on the KF, which allows it to be a steadier filtering state estimator. The baseline pdf is shown in green. The yellow line is the AD threshold, and represents the 95% of the baseline pdf. Once 95% of the estimation pdf crosses the AD threshold, the algorithm flags the detection of a fault.

3.3. PF Implementation

As with the EKF, the PF implementation also requires the use of a model. Surrounding the model is the use of various parameters such as the number of particles used to estimate the true state. The larger the number of particles used in PF, the better the estimation. The choice of particle number, also known as sampling size, is a trade-off between computational efficiency and accuracy. Clearly, in offline applications,
computational efficiency is not as critical as it is to online systems. Our primary interest in implementation in embedded, decentralized architecture, so a trial was run to help select the min particles that can be used to achieve a high-performing PF. Figure 5 shows the integrated absolute error (IAE) for each iteration of particle number. The elbow point on the graph lies roughly at about 100 particles, which should provide sufficient performance and reduce the computational complexity.

Figure 5. Demonstration of the relationship between error and particles used for state estimation.

Figure 6 shows the state estimate pdf for the PF algorithm. The confidence bounds are tighter than EKF, but the algorithm produces more sporadic state estimates, which tend to jump around. This effect is a result of degeneracy that must be corrected for. It is clear in Figure 6 that the PF estimation is more sensitive to shifts in the mean due to the tighter confidence bounds.

Figure 4. EKF state estimate pdf at cycle 274.

Figure 6. PF state estimate pdf at cycle 274.

3.4. DS Fusion Implementation

DS fusion is implemented to integrate the mass density of EKF and PF state estimations. In Bayesian-based fusion techniques, typically two possible hypotheses are considered, which are that a fault is present ($\mu_1$) or a fault is not present ($\mu_2$). DS theory includes a third hypothesis which is that a fault is either present or not present ($\mu_3$), which signifies the uncertainty. The combination table for mass function is shown in Table 1. Using the baseline pdf, certain threshold values are calculated to assign mass functions for EKF ($m_1$) and PF ($m_2$) to each hypothesis. The detection threshold for capacity depletion, based upon the baseline data, is selected to be 0.9448 ± 0.0054 Ah. This correlates to healthy data above 0.9502 Ah (20% of the baseline pdf), faulty data below 0.9394 Ah (5% of the baseline pdf), and uncertain data in between.

Table 1. DS Mass Function Combination Rule.

<table>
<thead>
<tr>
<th></th>
<th>$m_1(\mu_1)$</th>
<th>$m_1(\mu_2)$</th>
<th>$m_1(\mu_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>$m_2(\mu_1)$</td>
<td>$m_2(\mu_2)$</td>
<td>$m_2(\mu_3)$</td>
</tr>
<tr>
<td>K</td>
<td>$m_3(\mu_1)$</td>
<td>$m_3(\mu_2)$</td>
<td>$m_3(\mu_3)$</td>
</tr>
<tr>
<td>F</td>
<td>$m_4(\mu_1)$</td>
<td>$m_4(\mu_2)$</td>
<td>$m_4(\mu_3)$</td>
</tr>
</tbody>
</table>

The combined probability of fault is computed using Eq. (21) by summing the mass function combinations that indicate a fault is present (shaded in yellow), and dividing by the conflicting hypotheses chosen to be the normalizing factor (shaded in blue). The uncertainty in the result is computed by summing the mass function combinations that indicate either hypothesis is possible (shaded in green), and dividing by the normalizing factor. The algorithm compares the masses of the PF and the EKF state estimation techniques.
and uses the DS to calculate what the actual state is. Table 3 shows the probability of fault (PoF) and uncertainty (PoFU) for cycle 39 and cycle 80.

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Cycle 39</th>
<th>Cycle 80</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PoF</td>
<td>PoFU</td>
</tr>
<tr>
<td>PF</td>
<td>0.1101</td>
<td>0.2350</td>
</tr>
<tr>
<td>EKF</td>
<td>0.1795</td>
<td>0.4939</td>
</tr>
<tr>
<td>Fused</td>
<td>0.1374</td>
<td>0.1371</td>
</tr>
</tbody>
</table>

The fault detection system includes a parameter to determine the number of consecutive ADs before the program will declare that it has detected a fault. The number used in this implementation is three. While this is used to help mitigate false alarms, it also slightly reduces the sensitivity of the algorithm. As noted above, cycle 274 is the latest cycle where the algorithm detects a fault when DS fusion is not implemented. As shown in Table 3, the fusion algorithm consistently detects a fault with a high degree of confidence well before the AD performance algorithm. The time to failure (TTF) is the number of cycles from point of detection to total failure, when the capacity drops below 0.5 Ah. This attribute of DS fusion is an important finding from the experiment.

<table>
<thead>
<tr>
<th>Battery Data Set</th>
<th>TTF w/o Fusion</th>
<th>TTF w/ Fusion</th>
<th>Early Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS2_35</td>
<td>655</td>
<td>734</td>
<td>79</td>
</tr>
<tr>
<td>CS2_36</td>
<td>494</td>
<td>659</td>
<td>165</td>
</tr>
<tr>
<td>CS2_37</td>
<td>653</td>
<td>802</td>
<td>149</td>
</tr>
<tr>
<td>CS2_38</td>
<td>771</td>
<td>851</td>
<td>80</td>
</tr>
</tbody>
</table>

Runtime testing was performed to then demonstrate visually the effect of the fusion algorithm on the final state estimation pdf. Figure 7 shows how much tighter the confidence bounds are for DS fusion than for either PF or EKF. The reliability of the measurement is dramatically increased by fusing the results from both the EKF and the PF.

4. CONCLUSION

EKF is implemented to accommodate for nonlinear battery dynamics, whereas the PF is a Monte Carlo sampling method for state-space inference. By fusing the state estimations from each of these techniques, the AD algorithm was able to detect degradation more quickly and with a higher degree of confidence. Using the DS method for fusion successfully accounted for unknowns and performed very well over the entire feature set. Due to a limited feature set, this paper only demonstrates a model which was tested and trained on the same data. For better results, models should be trained on one set of data, cross validated on another set of data, and finally tested on a third set of data, to ensure that the model generalizes well to data it has never before encountered. The IAE metric helped to determine that using more particles in the PF implementation clearly results in higher degree of veracity in state estimation. However, IAE does not account for the fact that by definition, filtering algorithms are attempting to estimate the true state underlying the measured state. Measurements contain unknown variation and noise, and it is therefore not desirable for the algorithm to perfectly mimic the measurement, but instead to accurately trend about the measurement distribution mean.

5. FUTURE WORK

For future implementations, the issue of diagnosis-triggered prognosis should be tackled, using the fused result as its basis. Using an expanded dataset should allow algorithms to be both trained and cross-validated prior to final proof-testing. A better model should also be developed. Rather than strictly relying on an estimated battery capacity data-driven approach, it is advisable to incorporate battery physics into the model (Saha & Goebel, 2009). The algorithm is constructed to incorporate input and operating mode changes, such as those mentioned above, but data was unavailable to be incorporated for this application. Also, PF approaches suffer from high computational load when utilizing large particle sets for estimation. To alleviate the need for large particle sets, one might also consider the value of incorporating each state estimation’s “historically-estimated correctness” (Wu, et al., 2002) into the DS fusion algorithm. This would mean that as part of DS fusion of PF and EKF, prior performance of the algorithm should be incorporated in order to dynamically adjust the weight the fusion algorithm places on either the PF or the EKF estimation. The algorithm would consider how well PF or EKF has been tracking the true state, and using this data the fusion algorithm would place more trust in one or the other. This should enable a
smaller particle set to be used, while maintaining high integrity fusion.

ACKNOWLEDGEMENTS
The project is sponsored by the ASPIRE grant program at the University of South Carolina and the Woodrow W. Everett, Jr. SCEEE Development Fund in cooperation with the Southeastern Association of Electrical Engineering Department Heads.

REFERENCES
Parameters Adaption of Lebesgue Sampling-based Diagnosis and Prognosis for Li-ion Batteries

Wuzhao Yan1, Wanchun Dou2, Datong Liu3, Yu Peng3, and Bin Zhang1

1 Department of Electrical Engineering, University of South Carolina, Columbia, SC, 29208, USA
wyean@email.sc.edu
zhangbin@cec.sc.edu

2 Department of Computer Science & Technology, State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, 210093, P.R. China
douwc@nju.edu.cn

3 Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, Heilongjiang, 150080, P.R. China
Liudatong@hit.edu.cn
Pengyu@hit.edu.cn

ABSTRACT

Traditional fault diagnosis and prognosis (FDP) approaches are based on Riemann sampling (RS), in which samples are taken and algorithms are executed in a periodic time interval. With the increase of system complexity, the real-time implementation of this Riemann sampling-based FDP (RS-FDP) becomes a bottleneck, especially for distributed applications. To overcome this problem, a Lebesgue sampling-based FDP (LS-FDP) is proposed. LS-FDP takes samples on the fault dimension axis and provides a need-based diagnostic philosophy in which the algorithm is executed only when necessary. In previous Lebesgue sampling-based FDP, the Lebesgue length is a constant. To accommodate the change of fault dynamics, it is desirable to execute FDP algorithm more frequently when the fault growth is fast while less frequently when fault growth is slow. This requires to change the Lebesgue length adaptively. The goal of this paper is to deliver an improved LS-FDP method with varying Lebesgue length, which enables the FDP to be executed according to needs. The design and implementation of varying Lebesgue length LS-FDP based on a particle filtering algorithm are illustrated with experimental results on Li-ion batteries to verify the performances of the proposed approach. The experimental results show that the new varying LS-FDP is accurate and time-efficient on long term prognosis and also keeps a closer monitoring on the fast increase of fault size.

1. INTRODUCTION

The utilization of embedded systems are increasing in modern vehicle and other complex system design. There are more than 70 distributed microcontrollers (also known as electronic control units-ECUs) in one high-end car (Pattipati, Wang, Zhang, Howell, & Salman, 2011). The ECUs in modern cars perform variety of functions such as stability control, cruise control, oil and coolant monitoring, energy-efficient propulsion, turbo on-and-off, navigation with real-time traffic, and even autonomous driving. To ensure these functions, diagnosis and prognosis are needed to monitor and predict the health condition of safety critical components, such as engine, battery, and transmission.

With the increase of components and subsystems in a complex system, more and more diagnostic and prognostic algorithms are deployed on local processors and embedded systems to alleviate the requirements on the communication bandwidth, power, and computation, thus to improve the reliability of the whole system (Schwabacher & Goebel, 2007; Zhang et al., 2011; Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006). However, these local processors and embedded systems have very limited computational resources. It is difficult or even impossible for traditional fault diagnosis and prognosis (FDP) algorithms to be deployed.

To overcome this bottleneck that prevents the distributed
FDP in complex systems, Lebesgue sampling-based FDP (LS-FDP) algorithms are developed (Zhang & Wang, 2014), in which a novel FDP philosophy is employed on an “as-needed” basis. It can efficiently reduce the computational cost compared with the traditional RS-FDP. Different from RS-FDP, LS-FDP divides the state axis by a number of pre-defined states (also called Lebesgue states). The FDP will be triggered when the feature value changes from one Lebesgue state to another, or an event happens. After the fault is detected, the prognosis is executed to estimate a distribution of operating time for the fault state reaching each Lebesgue state.

With the characteristic of “execute only when necessary” the computation demands are significantly reduced by eliminating unnecessary computation. When the fault growth is slow, the FDP is executed in a low frequency. While when the fault growth is fast, the FDP is executed more frequently. In previous LS-FDP, the Lebesgue length is constant and fixed, this is not an optimal solution for most fault dynamics that the fault growth is nonlinear. To accommodate the non-linearity of faulty dynamics, it is desirable to adjust the Lebesgue length adaptively and optimally.

Since the Lebesgue states in LS-FDP are selected to ensure the performance for the fastest fault growth scenario, the LS-FDP algorithm can be executed more frequently than necessary when the fault growth is slow. This results in significant over-provisioning of the real-time system hardware. In practice, the system may have multiple fault modes, and different faults have different growth speeds. The computational resources can be dynamically distributed among different FDPs to optimize the performance of microcontroller.

To achieve this goal, the fault growth speed estimated from previous Lebesgue states is used to optimize current Lebesgue state length, from which a new set of Lebesgue states is achieved. Compared with the initial Lebesgue state lengths, the new Lebesgue state lengths are shrunk or stretched, and is re-adjusted every time when prognosis is executed. The prognosis algorithm then predicts the distributions of operating time on these updated Lebesgue states.

The paper is organized as follows: Section 2 provides an overview of the framework of varying Lebesgue length sampling FDP (VLS-FDP). Section 3 develops a particle filtering based VLS-FDP method. A case study based on lithium ion battery is presented to demonstrate the advantages of VLS-FDP in Section 4. Conclusions and future research topics are given in Section 5.

2. VLS-FDP

2.1. Fault Growth Modeling

Assume that the fault growth model can be described by a continuous-time differential equation:

\[ \dot{a} = F(a, u) \]  (1)

where \( a \) is the fault dimension, \( u \) is the system control inputs which can influence the fault growth, and \( F(\cdot) \) is a nonlinear function that describes the fault growth under current operating scenario. The feature, denoted by \( y \), is extracted from testing data and serves as the measurement for the FDP algorithm. To simplify the problem, the mapping between \( y \) and \( a \) is described as \( y = a \), and \( y \) is employed as the indicator of the fault state, which is the measured real-time battery capacity in the case of battery life prediction.

Lebesgue sampling-based model for the fault growth dynamics in discrete-time can be described as:

\[ \hat{a}(t_{k+1}) = \hat{a}(t_k) + f(t, \hat{a}(t_k)) \]  (2)

where \( a(t_k) \) is the Lebesgue state of fault dimension, \( t_k \) is the \( k \)th sampling instant, and \( D \) is the Lebesgue state length.

2.2. Lebesgue Sampling based Diagnosis

The LS-based diagnosis is described as follows:

1. Divide the state range into a series of Lebesgue states \( \{F_1, F_2, ..., F_f\} \);
2. If the feature value \( y \) causes a transition from one Lebesgue state to another (Astrom & Bernhardsson, 2002; McCann & Le, 2008), i.e., an event happens, the diagnostic algorithm is executed. Otherwise update the time stamp and wait;
3. Calculate the current fault state distribution and determine whether a fault is detected.

2.3. Lebesgue Sampling based Prognosis

For LS-FDP, the prognosis is conducted along the state axis to calculate the time to failure (TTF) or remaining useful life (RUL) distributions of operation time to reach the defined Lebesgue states. The model for LS-based prognosis is given as:

\[ t_{k+1} = t_k + g(D, \hat{a}(t_k)) \]  (3)

This function describes the time distribution of the fault state to arrive at each Lebesgue states. The prediction horizon is the number of Lebesgue states from the current Lebesgue state to the state defined as failure threshold. Compared to RS-based prognosis, this prediction horizon is usually small and will significantly reduce the computation cost.

Figure 1 shows the flow chart of Lebesgue sampling-based prognosis. Suppose at state \( F_d \), the diagnosis algorithm de-
tects a fault and prognosis needs to be implemented to calculate the distributions of operating time when the fault size reaches Lebesgue states \( \{F_{d+1}, ..., F_f\} \).

The initial condition for prognosis is the time distribution to reach the current Lebesgue state \( F_d \). In a particle filtering-based algorithm, this time distribution can be obtained by conducting prediction based on the fault growth model in Equation (2) for all particles with \( F_d \) being set as the threshold.

### 2.4. The concept of VLS-FDP

This section will develop the complete VLS-FDP framework with an overview of the proposed solutions to overcome the limitations of the fixed Lebesgue length LS-FDP. The innovation of this method is that Lebesgue length of the diagnosis and prognosis algorithm is no longer fixed. Instead, Lebesgue length is online adjusted adaptively to accommodate the non-linearity of the fault dynamics. The concept is illustrated in Figure 2, the battery degradation data is from (He, Williard, Osterman, & Pecht, 2011). In LS-FDP method, as shown in Figure 2(a), the Lebesgue states are equally distributed on the state axis, which means the state changes are the same, no matter the fault grows fast or slowly. In VLS-FDP, the Lebesgue state length are changed based on the fault growth speed. The fault growth speed is characterized as the ratio between the Lebesgue state length of two successive Lebesgue states and the operation time between them. It is clear in Figure 2 (b) that the Lebesgue states are nonuniform.

### 3. ALGORITHM DESIGN

In this section, a particle filtering-based algorithm is designed in the framework of varying length Lebesgue sampling.

![Figure 1. Flow chart of Lebesgue sampling-based prognosis](image)

![Figure 2. Illustration of VLS-FDP. (a) LS-FDP with fixed Lebesgue state length; (b) VLS-FDP with varying Lebesgue state length](image)

#### 3.1. Particle filter for VLS-based diagnosis

An unobserved fault process \( X \) is assumed to be a Markov process characterized by initial distribution \( p(x_0) \) and the transition probability \( p(x_k | x_{k-1}) \) defined by \( x_k = f_k(x_{k-1}, w_k) \) with \( w_k \) being the process noise. The subscript \( k \) represents the \( k \)th events. The observation is assumed to be conditionally independent on \( X \). The distribution of \( Y_k \) given \( X_k \) (\( Y_k | X_k \)), is defined by \( y_k = h_k(x_k, v_k) \) and \( v_k \) is the measurement noise. Let \( x_{0:k} = \{x_0, ..., x_k\} \) and \( y_{1:k} = \{y_1, ..., y_k\} \) denote the states and measurements from beginning to the \( k \)th event. The objective is to estimate the posterior distribution \( p(x_{0:k} | y_{1:k}) \), which is the framework of Bayesian theory is achieved by prediction and filtering.

In nonlinear cases, most of the analytical solutions do not exist. Sequential Monte Carlo (SMC) methods, such as particle filter, are widely used to provide approximate solution. The particle filter approach approximates the posterior distribution at the \((k−1)\)th event by a set of \( N \) particles \((w^{(i)}_{k−1}, x^{(i)}_{0:k−1})\), where superscript \( i \) denotes the \( i \)th particle, \( w^{(i)}_{k−1} \) and \( x^{(i)}_{0:k−1} \) are the weight and location of the particles at the \((k−1)\)th event, respectively. The purpose is to achieve a new set of \( N \) particles \((w^{(i)}_{k}, x^{(i)}_{0:k})\) to approximate the distribution \( \pi_k(x_{0:k}) \), where \( x^{(i)}_{0:k} \) is the location of new particles. In SMC framework, a Monte Carlo approximation can be obtained as:

\[
\pi_k(x_{0:k}) = \sum_{i=1}^{N} w^{(i)}_{k} \delta(x_{0:k} - x^{(i)}_{0:k})
\]  

(4)

where \( \delta \) denotes the Dirac-Delta function, \( \sum_{i=1}^{N} w^{(i)}_{k} = 1 \). The weight of the particle is updated by a recursive method:

\[
w^{(i)}_{k} = \frac{w^{(i)}_{k−1} h_k(y_{1:k} | x^{(i)}_{0:k})}{\sum_{i=1}^{N} w^{(i)}_{k−1} h_k(y_{1:k} | x^{(i)}_{0:k})}
\]

(5)

An LS-based diagnostic model is used to implement the par-
particle filtering based fault diagnosis. The model is augmented from (2) and is given as:

\[
\begin{align*}
    x_d(t_k + 1) &= f_b(x_d(t_k) + n(t_k)) \\
    \dot{a}(t_k+1) &= \dot{a}(t_k) + f_1(D, \dot{a}(t_k)) \cdot x_d(t_k) + w_a(t_k) \\
    y(t_k) &= \dot{a}(t_k) + v(t_k)
\end{align*}
\]

where the nonlinear mapping \( f_b(x) \) is given by:

\[
f_b(x) = \begin{cases} 
    1, & \text{if } ||x - 1|| \leq ||x - 0|| \\
    0, & \text{otherwise}
\end{cases}
\]

where \( x_d \) is a boolean value that indicates the normal (0) or faulty (1), respectively, \( \dot{a} \) is the Lebesgue state that indicates the fault size, \( w_a \) and \( v \) are process and measurement noises, \( n \) is an independent and identically distributed uniform white noise, the initial condition is given as \( x_d = 0 \), indicating that there is no fault initially.

In this model, \( t_k \) is the event stamp indicating there is a state transition, the measurement \( y(t_k) \) is directly mapped from the fault size \( \dot{a}(t_k) \), for battery case, it is the capacity measured by Coulomb counting method.

During the diagnostic process, the algorithm will be executed only when two successive measurements trigger a transition of Lebesgue states.

3.2. Particle filter for VLS-based prognosis

In LS-FDP framework, the prediction horizon is reduced to the Lebesgue state number from the current Lebesgue state \( F_d \) to the state that indicates a failure threshold \( F_d \). Due to the switching of prediction horizon from the time axis to the state axis, the prediction horizon is greatly reduced, which results in reduction of a large mount of computation and the accumulation of uncertainties.

The prognostic model for LS-FDP is given as:

\[
t_{k+1} = t_k + g(D, \dot{a}(t_k)) + w(t_k)
\]

where \( D \) is the Lebesgue state length and \( w(t_k) \) is the model noise.

With this model, the particle algorithm estimates the distribution on the time axis. Note that the output of diagnosis is a fault state distribution defined on the state axis, which cannot be used in LS-based prognosis. The reason is that LS-based prognosis needs an initial condition of time distributions on the current Lebesgue state. To address this problem, a new set of \( M \) particles \((w_L^{(i)}, t_L^{(i)})\) is adopted to initialize the prognostic process, where the subscript \( L \) denotes the Lebesgue state, \( w_L^{(i)} \) and \( t_L^{(i)} \) are particle weights and particles on the time axis, respectively. The initial weights of the particles are uniform \( (w_L^{(i)}, t_L^{(i)}) = 1/M \). The RUL pdf is calculated under the condition of Lebesgue state \( L = F_f \) by (7).

The difference between LS-FDP and VLS-FDP is that the Lebesgue state length \( D \) in VLS-FDP is not a constant, but changed based on the diagnostic result. The whole process is illustrated in Figure 3 and is described as follows:

The procedure is described as follows:

1. With a new measurement, check if an event happens. If an event happens, run diagnosis algorithm to detect the fault with the initial set of Lebesgue states and Lebesgue state length;
2. When the fault is detected at the Lebesgue state \( F_d \), the prognostic algorithm is executed with an initial Lebesgue state length \( D_0 \) and the Lebesgue states are \((F_1, \ldots, F_{d-1})\). The time interval between Lebesgue state \( F_{d-1} \) and \( F_d \) is \( N_0 \) (in the diagnostic process), the slope is calculated \( S_0 = D_0/N_0 \);
3. Lebesgue state length for prognosis is \( D_0 \) when the prognosis is initialized on Lebesgue state \( F_d \). The time interval to reach \( F_{d+1} \) by (7) is \( N_1 = TTF_{F_1} - T_c \) for the prognosis on Lebesgue state \( F_d \), where \( TTF_{F_1} \) is the mean of the predicted time to failure (TTF), \( T_c \) is the current time instant. The Lebesgue state length between Lebesgue state \( F_d \) and \( F_{d+1} \) is updated by \( D_1 = S_0 \times N_1 \), and the TTF and time interval are also updated to \( TTF_{F_1} \) and \( N_1 \), and the new slope is \( S_1 = D_1/N_1 \);
4. The prognosis at Lebesgue state \( F_{d+1} \) starts with Lebesgue state length \( D_1 \) and slope \( S_1 \), time interval \( N_2 \) is also calculated by (7). Lebesgue state length \( D_2 \) is given as \( D_2 = S_1 \times N_2 \), then update the time interval \( N_2 \) to calculate the slope \( S_2 = D_2/N_2 \);
5. Repeat the step 3) and 4) until the Lebesgue state reaches the prognosis threshold \( F_f \). This step yields the TTF and RUL distribution;
6. At the next time instant, when a new measurement becomes available, repeat step 1) to 5). Note that the Lebesgue state lengths for diagnosis are also updated based on the calculation result in the previous prognostic process.
process. When a new event happens, the time interval is updated by the ground truth of cycle life to \( \tilde{N}_1 \), the slope \( \tilde{S}_1 = D_1 / \tilde{N}_1 \) and \( D_2 \) are used as the initial condition for the new recursive prognostic loop.

Note that, after the prognostic process, the length of each Lebesgue state is changed. The new Lebesgue states are used for the following prognostic process as the initial set of Lebesgue states. So the whole FDP process will be executed with a different frequency, which is determined by the fault growth speed.

4. Experimental results

In this section, the proposed VLS-FDP scheme will be demonstrated with a particle filtering algorithm with an application to the prediction of the capacity degradation of a Li-ion battery. Battery is a safety critical component that provides power for most autonomous systems, such as computers, robots, electrical vehicles, and unmanned aircraft (Saha, Goebel, Poll, & Christophersen, 2009; Zhang, Tang, DeCastro, Roemer, & Goebel, 2014). Since the life and state of the batteries are not directly observable, diagnosis and prognosis are critical for estimating the battery state (K. Goebel, Saha, & Saxena, 2008; K. F. Goebel et al., 2006; Sidhu, Izadian, & Anwar, 2015), such as state-of-health (SOH), state-of-charge (SOC), and remaining useful life (RUL).

In this experiment, the SOH of a Li-ion battery with 1.1 Ah rated capacity is used to verify the proposed VLS-FDP algorithm, which is compared with RS-FDP and LS-FDP algorithms. The degradation of the capacity is obtained from charge-discharge cycle tests carried by an Arbin BT2000 battery test system under room temperature at a discharge current of 1.1 A. The charge-discharge cycle is cut off at predetermined cut-off voltages. The failure threshold for the SOH is set to be 0.25Ah and the battery capacity reaches this threshold at 854th cycle.

4.1. RS-FDP

To implement diagnosis and prognosis, a fault growth model for RS-FDP is developed by model fitting:

\[
C(t+1) = C(t) - \alpha \cdot p_1 \cdot p_2 + p_3 \cdot t + p_4 \cdot t^2 + p_5 + w(t) \tag{8}
\]

where \( C \) is battery capacity, \( t \) is the time index which is cycle number in this experiment, \( p = [5e^{-5}, -225, 5.6, -0.0135, 0.5] \) are parameters, \( \alpha \) is a hyper model parameter with mean of \( 3.8e^{-3} \) and variance of \( 5e^{-5} \), and \( w \) is a model noise.

For RS-based diagnosis, a set of 500 particles are used in the algorithm based on our computational resource. Figure 4 shows the diagnostic results at the 472nd cycle. The mean of capacity estimation is 0.87414 and the 95% confidence interval is [0.8497, 0.8945]. The upper sub-figure of this figure is the measurement compared with the filtered estimation. The bottom sub-figure shows the comparison of initial baseline pdf compared with the real-time estimated pdf at the 472nd cycle. Note that the diagnostic algorithm is executed 472 times, i.e., every time when a new measurement becomes available.

![Figure 4. Experimental result of RS-based diagnosis.](image)

With an estimation of the current battery capacity as the initial condition, the prognosis is executed to conduct the long-term prediction and estimation of RUL. Figure 5 shows the expected value, upper and lower bounds of 95% confidence interval of the battery capacity pdf at each future cycle. Note that only 20 particles are used for prognosis because of a large prediction horizon of 525 cycles. The battery capacity pdf at each cycle is compared against the failure threshold to obtain the RUL pdf, as shown in the histogram on the horizontal axis. The law of total probabilities is used in this process.

In this figure, the predicted result of the mean of the failure time is at the 748th cycle and the RUL life is 276 cycles. The distance between the prediction and ground truth is 106 cycles. The 95% confidence interval of the RUL pdf is [628, 987], which means that the uncertainty accumulated along the prediction horizon is very large.

4.2. LS-FDP

In LS-FDP, the input of the algorithm is the feature, which is divided into a series of Lebesgue states. If the new measurement causes a transition of Lebesgue state, i.e., an event happens, the diagnostic algorithm is executed. The time instant when an event occurs is called the “event stamp”. The sequence of the event stamps formulates a time series that is used as the input of real time diagnostic algorithms. The output of fault diagnosis is the fault state distribution, which is used to calculate the probability to declare a fault. The test is
performed between the estimate pdf of fault and the baseline pdf from healthy condition.

To implement LS-FDP for the battery capacity degradation, 40 Lebesgue states are defined based on the battery’s full capacity of 1.1Ah and the computation resource. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another, i.e., an event happens. The model for diagnosis is given by:

\[ C(t_{k+1}) = C(t_k) - p_d \cdot D \cdot sgn(C(t_k) - C(t_{k-1})) + w_C(t_k) \]  

where \( p_d \) is the model parameter, \( t_k \) is the event stamp indexes, \( sgn(\cdot) \) gives the sign, and \( w_C \) is the model noise.

Different from the diagnosis that yields fault state distribution at each time instant when an event occurs, prognosis estimate the time distribution on fault state reaching each Lebesgue state. The output of diagnosis is a capacity distribution at current time instant. It cannot be used for prognosis directly and has to be transformed into the operation time distribution. To implement prognosis in LS-FDP framework, the operation time distribution is achieved by predicting all the particles to the current Lebesgue state.

LS-based prognosis is conducted on fault dimension axis to predict the time-to-Lebesgue-state directly. The diagnostic model (9) cannot be used in prognosis. A new model for prognosis is proposed as:

\[ t_{k+1} = t_k + p_p \cdot D \cdot \exp(-\dot{C}(t_k)) + w_k(t_k) \]  

where \( p_p \) is the model parameter and \( w_k \) is the model noise. Figure 7 shows the prognostic results with 500 particles at the 472nd cycle. Compared to RS-based prognosis with large horizon (525 cycles) and small number of particles (20), the LS-based prognosis only has a prognostic horizon of 24 Lebesgue states and can afford 500 particles. The reduction of computation time is \( (2.822281-0.011541)/2.822281=99.59\% \) and the computation is about 244 times faster. Note that in RS-based prognosis, only 20 particles (at the cost of performance) are used to make real-time implementation possible. Note also that LS-FDP offers better performance than RS-FDP in terms of TTF prediction due to short prognostic horizon.
4.3. VLS-FDP

VLS-FDP is developed based on LS-FDP, the battery’s full capacity range is divided into 40 Lebesgue state as in LS-FDP. When fault is firstly detected, this Lebesgue state length is the initial Lebesgue state length in VLS-FDP. The model for diagnosis is the same as LS-FDP in model (9).

Figure 8. VLS-based diagnosis for battery at the 472nd cycle.

Figure 8 shows the diagnostic results at the 472nd cycle. The upper sub-figure is the battery capacity measurement by Coulomb counting method (blue curve) compared with the filtered capacity estimation (magenta). Note that the length for each Lebesgue state has been changed after the fault being detected compared with the case before fault being detected, which is an indicator of a closer monitoring on fault growth.

The lower sub-figure shows the fault distribution at the 472nd cycle, in which the black histogram is the baseline distribution, and the magenta one is the real-time battery capacity distribution from diagnosis.

The VLS-FDP prognosis is conducted on the state axis to predict the time-to-Lebesgue-state, the diagnostic model is the same as (9).

Figure 9 shows the prognostic result at the 472nd cycle. In this figure, the prediction horizon is 50, which is a little bigger than the LS-FDP with uniform Lebesgue length. This is the trade-off for closer monitoring of SOH when the capacity degradation becomes faster.

To make the figure clear, only the time distribution at selected Lebesgue states are plotted. As shown in Figure 9, the predicted TTF for this battery is 846.9 and the RUL is 374.9 cycles. The 95% confidence interval of the TTF is [819.6 861.9]. Compared with the ground truth TTF of 854, the difference is 7.1 cycles. Note that Lebesgue states for VLS-based prognosis are distributed unequally along the state axis, which is different from the case of LS-based prognosis. When the growth speed is fast, the prognosis is executed with a higher frequency and the length of the neighboring execution is reduced.

![VLS-based prognosis at the 472nd cycle.](image)

Figure 9. VLS-based prognosis at the 472nd cycle.

4.4. Comparison of RS-FDP, LS-FDP, and VLS-FDP

Diagnostic and prognostic results of RS-based, LS-based, and VLS-based algorithms are compared in Table 1 with the same benchmark. Compared with RS-based prognosis with a horizon of 525 cycles and small number (20) of particles at the 472th cycle, the LS-based and VLS-based prognosis have a horizon of 24 and 50 Lebesgue states, and can afford 500 particles. The computation time for LS-based and VLS-based prognosis are only 0.41% and 1.36% of the RS-based prognosis, respectively. Compared to LS-based prognosis, VLS-based prognosis is a little computational expensive. The reason is that VLS-based prognosis keeps a closer monitoring on the SOH after the 472nd cycles by shrinking the Lebesgue state length dynamically to accommodate an accelerated degradation speed.

Table 1. Comparison of Traditional RS-FDP, LS-FDP and VLS-FDP for Battery

<table>
<thead>
<tr>
<th></th>
<th>RS-FDP</th>
<th>LS-FDP</th>
<th>VLS-FDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnosis particles</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Capacity expectation</td>
<td>0.8744</td>
<td>0.8854</td>
<td>0.8854</td>
</tr>
<tr>
<td>Capacity 95% CI</td>
<td>[0.8469 0.8986]</td>
<td>[0.8285 0.9422]</td>
<td>[0.8290 0.9432]</td>
</tr>
<tr>
<td>Execution numbers</td>
<td>472 (100%)</td>
<td>76 (16.1%)</td>
<td>78 (16.5%)</td>
</tr>
<tr>
<td>Prognosis particles</td>
<td>20</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>True TTF</td>
<td>854</td>
<td>854</td>
<td>854</td>
</tr>
<tr>
<td>Estimate TTF</td>
<td>748</td>
<td>831.8</td>
<td>846.9</td>
</tr>
<tr>
<td>95% CI of TTF</td>
<td>[628 987]</td>
<td>[795.3 850.3]</td>
<td>[819.6 861.9]</td>
</tr>
<tr>
<td>Prognostic horizon</td>
<td>25</td>
<td>24</td>
<td>50</td>
</tr>
<tr>
<td>Computation time</td>
<td>2.822281 (100%)</td>
<td>0.011254 (0.41%)</td>
<td>0.038498 (1.36%)</td>
</tr>
</tbody>
</table>

Accuracy is one of the most important properties in FDP. In order to compare the accuracy of three FDP methods, $\alpha - \lambda$ matrix is introduces in (Saxena, Celaya, Saha, Saha, & Goebel, 2010), as shown in Figure 10 with $\alpha = 0.3$. The
matrix is defined as:
\[ [1 - \alpha] \cdot r(t_k) \leq r^*(t_k) \leq [1 + \alpha] \cdot r(t_k) \quad (11) \]
where \( r^* \) is the predicted RUL at the \( l \)th time instant, \( r_1 \) is the ground truth TTF, \( \alpha \) is the accuracy modifier (Saxena et al., 2010).

Because of the short prediction horizon and small uncertainty accumulation, the prediction accuracy for LS-FDP and VLS-FDP are higher than that of RS-FDP, as shown in Figure 10. The TTF of LS-FDP and VLS-FDP reach the accuracy zone quickly, and the prediction results are stable. The result of RS-FDP exceeds the accuracy limits sometimes, which means the estimation of TTF is not uniformly accurate. The high prediction accuracy of LS-FDP and VLS-FDP is achieved with much lower computation cost compared with RS-FDP.

The advantages of varying Lebesgue length in diagnosis and prognosis are obvious by the comparison in Table 1. For diagnosis, these three methods have comparable performances. In terms of prognosis, the VLS-FDP shows better performances in different aspects. First, VLS-FDP has better accuracy and precision than RS-FDP by comparing the confidence interval (CI) of the TTF. It reduces the computation resource greatly with similar diagnostic performance. RS-FDP has a large prediction horizon, which requires more computation time and resources. Second, VLS-FDP also avoids the large amount of uncertainty accumulation during the long-horizon prediction. Third, VLS-FDP dynamically distributes the limited computation resources between different fault growth stages, which is an improvement of LS-FDP.

The application of Lebesgue sampling method in FDP provides a natural solution for real-time FDP implementation, especially for those systems with limited computation resources. The improved Lebesgue sampling varying length method distributes the computation resources dynamically within a system. Similar to LS-FDP, the prediction horizon of VLS-FDP is very small compared with that of RS-FDP, which is beneficial for managing the uncertainty in prognosis.

5. CONCLUSIONS AND FUTURE WORKS

Many diagnostic and prognostic methods were developed based on traditional Riemann sampling framework with great success in the past decades. The application of RS-FDP on distributed FDP is limited because of its high computation cost. A new FDP methodology is introduced with a philosophy of “execution when needed” to reduce the computation cost, which makes the long-term online prognosis possible to be computed on an embedded system, such as the microcontrollers in a car and a mobile phone. However, the close monitoring on the fault size is sacrificed to some extent, especially when the fault growth is accelerated. This paper proposes varying Lebesgue state length in the LS-FDP framework. A particle filtering-based algorithm is developed with an application to the diagnosis and prognosis of Li-ion battery SOH.

In the VLS-FDP, models for diagnosis and prognosis are designed separately because diagnosis and prognosis are carried on state and time axis, respectively. The Lebesgue length for the diagnostic and prognostic models changes according to the past fault growth speed. Experimental results for RS-FDP, LS-FDP, and VLS-FDP on a Li-ion battery SOH are presented to demonstrate the effectiveness of the proposed algorithms.

In the future work, data collected from partial charge/discharge cycle and non-constant current charge/discharge with different current will be used to test our methods, new parameter adaption methods will be adopted to accommodate the real vehicle scenarios.

ACKNOWLEDGMENT

The project is sponsored by the ASPIRE grant program at the University of South Carolina and the Woodrow W. Everett, Jr. SCEEE Development Fund in cooperation with the Southeastern Association of Electrical Engineering Department Heads.

REFERENCES


Goebel, K. F., Yan, W., Eklund, N. H. W., Hu, X.,
doi: 10.1117/12.659704


**Biographies**

Wuzhao Yan received his B.A and Ph.D in Department of Physics in University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. After working in Amperex Technology Limited, Dongguan, China, for 3 years, he went to Department of Electrical Engineering in University of South Carolina. Now he focuses on the algorithms of diagnosis and prognosis for lithium ion battery.

Wanchun Dou received the PhD degree in mechanical and electronic engineering from Nanjing University of Science and Technology, China, in 2001. From April 2001 to December 2002, he did his postdoctoral research in the Department of Computer Science and Technology, Nanjing University, China. He is now a Full Professor at the State Key Laboratory for Novel Software Technology, Nanjing University, China. His research interests include workflow, cloud computing and service computing.

Datong Liu received the B.Sc. and M.Sc. degrees in Department of Automatic Test and Control from Harbin Institute of Technology (HIT), Harbin, China in 2003 and 2005, respectively. During 2001 to 2003, he also minored in the Computer Science and Technology in HIT. He received the Ph.D. degree in major of measurement and instrumentation from HIT in 2010. He is now an associate professor in the Department of Automatic Test and Control, HIT. His research interests include automatic test, machine learning and data mining for test data processing of complex system, data-driven prognostics and health management, lithium-ion battery prognostics, system health management for aerospace, etc.

Yu Peng received the B.S. degree in Measurement Technology and Instrumentation, Harbin Institute of Technology (HIT), Harbin, China, in 1996, and M.Sc. and Ph.D. degrees in Instrumentation Science and Technology, HIT, in 1998 and 2004, respectively. He is a full Professor with the Department of Automatic Test and Control, School of Electrical Engineering and Automation, HIT. His research interests include automatic test technologies, virtual instruments, system health management, reconfigurable computing, etc.

Bin Zhang received the B.E. and M.E. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. Before that, he was with R&D, General Motors, Detroit, MI, USA, with Impact Technologies, Rochester, NY, USA, and with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests are prognostics and health management and intelligent systems.
Lithium-Ion Battery End-of-Discharge Time Estimation and Prognosis based on Bayesian Algorithms and Outer Feedback Correction Loops: A Comparative Analysis

Carlos Tampier¹, Aramis Pérez¹, Francisco Jaramillo¹, Vanessa Quintero¹, Marcos E. Orchard¹ and Jorge F. Silva¹

¹Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile
c tampier@ing.uchile.cl, aramis.perez@ing.uchile.cl, vquintero@ing.uchile.cl, francisco.jaramillo@ing.uchile.cl, mochard@ing.uchile.cl, josilva@ing.uchile.cl

ABSTRACT

Battery energy systems are currently one of the most common power sources found in mobile electromechanical devices. In all these equipment, assuring the autonomy of the system requires to determine the battery state-of-charge (SOC) and predicting the end-of-discharge time with a high degree of accuracy. In this regard, this paper presents a comparative analysis of two well-known Bayesian estimation algorithms (Particle filter and Unscented Kalman filter) when used in combination with Outer Feedback Correction Loops (OFCLs) to estimate the SOC and prognosticate the discharge time of lithium-ion batteries. Results show that, on the one hand, a PF-based estimation and prognosis scheme is the method of choice if the model for the dynamic system is inexact to some extent; providing reasonable results regardless if used with or without OFCLs. On the other hand, if a reliable model for the dynamic system is available, a combination of an Unscented Kalman Filter (UKF) with OFCLs outperforms a scheme that combines PF and OFCLs.

1. INTRODUCTION

The main focus of this research is to establish a comparative analysis of two well-known Bayesian estimation algorithms, particle-filter (PF) (Arulampalam, Maskell, Gordon & Clapp) and UKF (Partovibakhsh & Guangjun, 2015), when used in combination with OFCLs (Orchard, Kacprzynski, Goebel, Saha & Vachtsevanos, 2008), (Orchard, 2007) to estimate the SOC and prognosticate the end-of-discharge (EoD) time of lithium-ion batteries.

The proposed case study, which is related to the problem of autonomy assessment in electromechanical devices, is selected due to its importance in decision-making processes that are related to mission reformulation based on condition monitoring, where the availability of real-time information is critical for optimal performance. Even though many manufacturers provide detailed information for batteries operating at constant temperatures and/or discharge currents, in practice this information is insufficient to avoid considerable errors on the autonomy estimates of the devices under time-varying power demands.

Numerous research efforts (Pola, Navarrete, Orchard, Rabie, Cerda, Olivares, Silva, Espinoza & Perez, 2015) have identified advantages associated with the implementation of Bayesian estimation techniques such as PF or UKF to characterize process and measurement uncertainty in the aforementioned problem. However, the incorporation of OFCLs has not been sufficiently discussed. In this regard, this article intends to present scientific evidence that could help future researchers to assess the real value behind the implementation of these schemes to characterize the uncertainty associated with the state estimates; which in turn define all initial conditions for online prognosis modules.

The structure of the article is as follows. Section 2 focuses on describing the theoretical framework that is required to understand the research performed. Section 3 presents the manner in which PF, UKF, and OFCLs algorithms were implemented to solve the SOC estimation problem. Section 4 presents the obtained results in terms of state estimation and EoD prognosis stages. Section 5 focuses on providing a performance comparison in terms of adequate
measures, and finally Section 6 summarizes the main conclusions of this research effort.

2. THEORETICAL FRAMEWORK

2.1. Outer Feedback Correction Loops

OFCLs play an important role within the structure of online prognosis modules, since they are capable of assuring increased precision and accuracy of remaining useful life (RUL) estimates model (Orchard, 2008). Typically, they measure the prediction capability offered by the process model (Orchard, 2008), (Orchard, 2007) through the analysis of short term prediction errors, improving the performance of the prognosis algorithm by either modifying the structure of the model that is used during the filtering stage (Orchard, Tobar, Vachtsevanos, 2009) or updating the hyper-parameters that define the process or observation noises (Orchard, et al., 2008), (Orchard, 2007), (Cruse, 2004).

One example of OFCLs is found in (Orchard, et al. 2009), where the authors propose a method that modifies the process noise variance depending if the prediction error over a horizon \( s \) starting from a time \( t \), \( e^s(t) \), is bigger or smaller than a determined threshold \( e^{th} \). Equation (1) shows the rule of decision, where \( [p, q] \) are such that, \( 0 < p < 1 \) and \( 1 < q \). As a result, the process variance related to the artificial evolution equation (Orchard, 2009) will increase if the prediction quality over the short-term horizon is poor, or it will decrease otherwise.

\[
\text{var}(w(t)) = \begin{cases} 
  p \cdot \text{var}(w(t)), & |e^t(t)| \leq e^{th} \\
  q \cdot \text{var}(w(t)), & |e^t(t)| > e^{th} 
\end{cases}
\] (1)

2.2. Prognosis Performance Indices

The evaluation of an algorithm capacity to predict the time-of-failure (ToF), which in this case would be equivalent to the EoD time, can be done considering different characteristics such as accuracy, precision or steadiness of results in time. The accuracy is related to the estimation bias and can be defined as a measure of proximity between the average estimation result and the ground truth value, while the precision measures the degree of concordance between different realizations obtained under similar circumstances.

2.2.1. Accuracy Index

Considers the relative width of the 95% confidence interval for the EoD estimate at time \( t \) \((C_{EoD})\), when compared to its conditional expectation \((E_{t}[EoD])\) \([30]\). Equation (2) quantifies the concept of “the more the amount of data, the more accurate the estimation results”.

\[
I_1(t) = e^{\left(\frac{\text{sup}(C_{EoD}) - \text{inf}(C_{EoD})}{E_{t}[EoD] - t}\right)}
\] (2)

\[0 < I_1(t) \leq 1, \forall t \in [1, E_{t}[EoD]), t \in \mathbb{N}\]

Accurate prognosis results are associated to values of \( I_1(t) \approx 1 \).

2.2.2. Accuracy-precision Index

Represents the amount of bias on the estimation of the EoD time, relative to the width of the corresponding 95% confidence interval, and penalizes the fact that the estimated expected value is greater than the real failure time (ground truth) (Orchard, et al. 2009).

\[
I_2(t) = e^{\left(\frac{\text{GroundTruth}[EoD] - E_{t}[EoD]}{\text{sup}(C_{EoD}) - \text{inf}(C_{EoD})}\right)}
\] (3)

\[0 < I_2(t), \forall t \in [1, E_{t}[EoD]), t \in \mathbb{N}\]

Good results of this index are associated to values such that \( 0 \leq 1 - I_2(t) < \varepsilon \), where \( \varepsilon \) is a small positive constant.

2.2.3. On-line Steadiness Index

Corresponds to the capacity of the algorithm to deliver prognosis results that are consistent in time. The evolution in time of the EoD conditional expected value is considered, and quantifies the concept “the more amount of data, the more stable the prognosis result should be” (Orchard, et al. 2009).

\[
I_3(t) = \sqrt{\text{Var}(E_{t}[EoD])}
\] (4)

\[0 \leq I_3(t), \forall t \in \mathbb{N}\]

Steady results are associated with small values of this index.

2.3. Characterization of the State-of-Charge

One of the main difficulties when estimating the SOC is that this parameter cannot be measured directly, and its value has to be obtained indirectly by measuring other parameters (Pattipati, Sankavaram, Pattipati, 2011), (Qingsheng, Chenghui, Naxin, Xiaoping, 2010), (Cadar, Petreus, Orian, 2009), (Di, Yan, Quin-Wen, 2011). Also, when estimating the SOC parameters such as temperature, rate of charge/discharge, hysteresis, age of the battery and self-discharge effect (Pattipati, et al., 2011). Chemical models for the SOC require many precise measurements for the different model variables (Pattipati, et al., 2011), (Charkhard & Farrokhi, 2011) and for this reason other methods are preferred. In this sense, the most popular methods are the Ampere-hour counter, internal impedance measurement and the open circuit voltage measurement (OCV) (Pattipati, et al., 2011), (Charkhard & Farrokhi, 2011), (Ran, Junfeng, Haiying & Gechen, 2010), (Qingsheng, et al., 2010), (Di, et al., 2011), (Saha, Goebel, Poll & Christophersen, 2009), (Tang, Mao, Lin & Koch, 2011).
The Ampere-hour counter estimates the battery capacity by the integration of the current during the charge/discharge cycle. This method has the advantage that can be implemented on-line. However, it has disadvantages, perhaps the main one is that it only is able to give good results for short periods of time, which leads to a low acceptance (Pattipati, et al., 2011), (Ranjbar, Banaei, Fahimi, 2012). Other disadvantages include the requirement of accurate measurements, the no consideration of the internal impedance losses, and the need to reference a SOC in order to compare the results (typically its maximum nominal capacity), among others (Pattipati, et al., 2011), (Charkhard & Farrokhi, 2011), (Cadar, et al., 2009), (Di, et al., 2011), (Tang, et al., 2011).

The OCV method has the advantage that it doesn’t need information prior to the measurements and that it has a direct relation with the SOC: the higher the OCV, the higher SOC (Tang, et al., 2011). Unfortunately, in order to realize this measurement the battery must have a prolonged period of rest (no current circulating) which makes difficult to use in systems where this time is not enough, and makes it hard to use on-line (Pattipati, et al., 2011), (Charkhard & Farrokhi, 2011), (Di, et al., 2011), (Tang, et al., 2011).

More recently, in (Pola et al., 2015) and (Cerda et al., 2012), the battery state model is obtained using an empirical scheme considering parts of the electric equivalents and a curve fitting of the voltage discharge curve, with good results obtained. The model of (Pola, et al., 2015) shown in Eq. (5) considers a two state vector \((x_1, x_2)\) where the first variable corresponds to the internal impedance of the battery and the second represents the state of charge in terms relative to its nominal capacity \(E_{crit}\). The observation equation \(v(t)\) characterizes the voltage measured during the discharge of the battery, and it is expressed as a function of the parameters \(v_0, v_L, \alpha, \beta\) and \(\gamma\). These parameters must be estimated off-line in order to obtain good results. The processes noises \((\omega_1, \omega_2)\) and observation noise \(\eta\) are assumed Gaussian. It is important to mention that \(\omega_2\) is correlated to \(\eta\), since the evolution in time of \(x_2\) depends of the voltage measurements.

\[
\begin{align*}
(x_1(t + 1) &= x_1(t) + \omega_1(t) \\
(x_2(t + 1) &= x_2(t) - v(t) \cdot i(t) \cdot \Delta t \cdot E_{crit}^{-1} + \omega_2(t) \\
v(t) &= v_L + \left[ (v_0 - v_L) e^{\gamma(x_2(t) - 1)} + \alpha v_L (x_2(t) - 1) - \cdots - (1 - \alpha) v_L \left( e^{-\beta} - e^{-\beta \sqrt{\overline{\gamma}}} \right) \right] - i(t) x_1(t) + \eta(t)
\end{align*}
\]

(5)

2.4. SOC Estimation and Prognosis

Sequential Monte Carlo methods such as the PF, offer good performance when used in the implementation of estimation and prognosis modules for nonlinear, non-Gaussian systems (Orchard & Vachtsevanos, 2009). There are studies where these techniques are applied to monitor the SOC and State-of-Health (SOH) of batteries in (Pola, et al., 2015), (Saha, et al., 2009), (Saha & Goebel, 2009), (Dalal, Ma, He, 2011), (Orchard, Tang, Saha, Goebel & Vachtsevanos, 2010) and (He, Williard, Osterman & Pecht, 2011). An alternative to the PF is the UKF, which has also been applied to the same problem (Bole, Daigle, Gorospe & Goebel, 2014). The UKF outstands for its good performance when nonlinear equations are present and its capacity to be implemented computationally in an efficient way (Van Der Merwe & Wan, 2001). Another type of techniques that becomes complimentary to the mentioned algorithms are the OFCLs, since they have been applied to estimation and prognosis problems (Orchard, et al., 2008), (Orchard, 2007) (Orchard, et al. 2009), hence it becomes interesting to analyze its impact.

3. IMPLEMENTATION OF SOC ESTIMATION SCHEMES BASED ON OUTER FEEDBACK CORRECTION LOOPS

3.1. Database description

Voltage and current data used in all experiments correspond to the discharge of a lithium-ion cell, identical to the ones described in (Pola, et al., 2015), and illustrated in the Figure 1. Data correspond to the characterization of usage of an electric vehicle in the city, specifically the Federal Urban Driving Schedule (FUDS), properly scaled for just one battery cell.

![Figure 1 a). Discharge current profile](image)

![Figure 1 b). Discharge voltage profile](image)
Table 1 shows the values for the parameters of the evolution of the state model. The process and observation noises are assumed Gaussian with a zero mean value.

Table 1. SOC Model Parameter values.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{crit}$</td>
<td>Battery model parameter</td>
<td>46858</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Battery model parameter</td>
<td>1</td>
</tr>
<tr>
<td>$v_L$</td>
<td>Battery model parameter</td>
<td>3.9974</td>
</tr>
<tr>
<td>$v_0$</td>
<td>Battery model parameter</td>
<td>4.144</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Battery model parameter</td>
<td>0.1469</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Battery model parameter</td>
<td>17</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Battery model parameter</td>
<td>10.4954</td>
</tr>
<tr>
<td>$R_{ww}$</td>
<td>Process noise covariance matrix</td>
<td>$\begin{bmatrix} 0.0015^2 &amp; 0 \ 0 &amp; 0.0055^2 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R_{vv}$</td>
<td>Observation noise covariance</td>
<td>0.067</td>
</tr>
<tr>
<td>$r_0$</td>
<td>Experimental internal resistance</td>
<td>0.1</td>
</tr>
</tbody>
</table>

### 3.2. Classical Particle Filter implementation

The base case used for comparison purposes corresponds to the classical PF-based implementation developed in (Pola et al., 2015). It uses a total of 40 particles and a basic design for an OFCL. The model described in Eq. (5) is used for this scheme.

The basic OFCL implemented in (Pola et al., 2015) considers a reduction of the process noise associated with the evolution of SOC in time, starting at a fixed time instant and considering a lower bound for the variance. In other words, if $\omega_2(t)$ is the process noise associated to the evolution of the battery SOC in time, then the OFCL is:

$$\text{if: } t > 200 \quad \text{then: } \text{std}(\omega_2(t + 1)) = \max(\text{std}(\omega_2(t))/1.01, 2 \cdot 10^{-4})$$

In this case, $\text{std}(\omega_2(t))$ is the standard deviation of the process noise at time $t$. This belongs to a basic correction loop since it does not measure the prediction capability of the model when using the output of the PF algorithm as the initial condition for prognosis. Instead, it opts to reduce the noise variance under the assumption that there is less uncertainty associated with the state estimation process since the filter has received more information. The PF estimates iteratively the SOC as new measurements of voltage and current are acquired. However, the complete scheme also includes the prognosis of the discharge. By applying the state transition equations, it is possible to characterize probabilistically the moment in which the battery is fully discharged (when the SOC falls down under a certain threshold or within a hazard zone). Nevertheless, it is necessary to know the value of the current that will be demanded in the future. To solve this issue, the work done in (Pola et al., 2015) proposes a two-state Markov Chain that emulates usage profiles with low and a high discharge currents. These two values, as well as the transition probabilities, are determined from historic measurements of the power demand. A more detailed description can be found in (Cerda et al., 2012). In (Pola et al., 2015), 25 realizations of discharge current profiles are used for prognosis purposes, hence the discharge time estimate computed at a determined moment corresponds to the weighted sum of 25 empirical distributions (Law of Total Probabilities), where each distribution is computed accordingly to Eq. (6), where $H_{lb}$ and $H_{ub}$ are, respectively, the lower and upper bounds of the hazard zone.

$$\Pr(EoD) = \sum_{i=1}^{N_p} \Pr[H_{lb} \leq x_2(EoD) \leq H_{ub}] W_i(EoD) \quad (6)$$

The discharge zone of the cell is defined in terms of a uniformly distributed hazard zone between 5.5% and 4.5% of remaining charge, becoming more critical when particles come near the lower bound. When calculating the distribution of the ToF of the prognosis scheme, the weight of each particle in Eq. (6) is modified as:

$$W'_i(EoD) = W_i(EoD) \cdot \min \left(1, \frac{0.055 - |\hat{x}_{i,2}(ToF)|}{0.055 - 0.045} \right) \quad (7)$$

where $\hat{x}_{i,2}$ corresponds to the estimated value for the second state (SOC) of the $i^{th}$ particle.

### 3.3. Battery Model

The discharge equations of a lithium-ion cell shown in Eq. (5) have a small inconsistency when compared to a traditional space state model: the evolution of the second state depends on the output of the system. Since the model output is a function of the state and the input, the right manner to implement the battery model is by replacing $v(t)$ by its prior estimate, as shown in Eq. (8). The reason why the model of Eq. (5) is used in (Pola et al., 2015) is simply because it is computationally less expensive, since it directly uses the acquired measurement instead of calculating the whole expression for each particle. In this approach, the algorithms are developed using the following model in order to describe the evolution of the states:

$$x_1(t + 1) = x_1(t) + \omega_1(t) \quad \text{and} \quad x_2(t + 1) = x_2(t) - i(t)x_1(t)$$

$$v(t) = v_L + (v_o - v_L)e^{T_x(x_2(t)-1)} + \alpha v_L(x_2(t) - 1) + \ldots + (1 - \alpha) v_L \left( e^{-\beta} - e^{-\beta x_2(t)} \right)$$

$$\text{where } \alpha = \frac{(1 - \alpha) v_L \left( e^{-\beta} - e^{-\beta x_2(t)} \right) - i(t)x_1(t) + \eta(t)}{(1 - \alpha)v_L e^{T_x(x_2(t)-1)} + \alpha v_L(x_2(t) - 1) + \ldots} \quad (8)$$
3.4. Unscented Kalman Filter

The UKF that is implemented corresponds to the classic version of the algorithm with the exception that the square root of the covariance matrix is replaced by its Cholesky factor, since the calculation is much simpler computationally speaking. Additionally, an outer correction loop is incorporated. The specific values of the UKF can be seen in Table 2.

Table 2. UKF Parameter values.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Battery model parameter</td>
<td>46858</td>
</tr>
<tr>
<td>α</td>
<td>Battery model parameter</td>
<td>1</td>
</tr>
<tr>
<td>β</td>
<td>Battery model parameter</td>
<td>3.9974</td>
</tr>
<tr>
<td>κ</td>
<td>Battery model parameter</td>
<td>4.144</td>
</tr>
</tbody>
</table>

It is important to mention that in the prognosis stage, the original structure described in (Pola, et al., 2015), which is based on empirical distributions, is maintained. Then, to prognosticate the EoD time it is necessary to sample the Gaussian probability density function (PDF) related to the state estimate. This is achieved by generating a sampling from the multidimensional Gaussian obtained by the UKF to represent the probability density distribution of the state, where each sample corresponds to the position of a particle and the weight is equal to all of the particles.

3.5. Outer Feedback Correction Loops

The implementation of OFCL aims to improve the performance of the estimation module, regardless of the main algorithm that is used for this purpose: UKF or PF. The OFCL designed for this case study affects the standard deviation of the process noise, which is assumed as Gaussian with a mean value of zero. This particular OFCL, though, is not based on short-term prediction results, but on the accumulated observation error instead. By observing the database, the voltage in the battery does not have considerable variations in small intervals of time (less than 30 seconds) during almost all the discharge cycle. Even more, the typical voltage drop that the battery undergoes during small time intervals, due to changes in the SOC, is comparable to the observation noise. In this regard, short-term predictions are not enough to evaluate the performance of the model. Increasing the prediction horizon is not a practical answer to this issue, since this generates algorithms lags and requires more memory. The use of the accumulated observation error solves the problem related to the required memory space; and also allows to evaluate the model performance, since it is able to detect inconsistencies between measurements and estimations of the output in previous time horizons. Thus, the proposed OFCL results:

\[
\text{If: } t > t_{\text{min}} \text{ then:}
\]

\[
e_{\text{accum}} = e_{\text{accum}} + |e_{\text{obs}}|
\]

\[
\text{If: } e_{\text{accum}} \leq e_{\text{rh}}
\]

\[
\text{std}(\omega_1(t)) = \max \left( p_1 \cdot \text{std}(\omega_1(t)), \text{std}_1 \right)
\]

\[
\text{std}(\omega_2(t)) = \max \left( p_2 \cdot \text{std}(\omega_2(t)), \text{std}_2 \right)
\]

\[
\text{elseif:}
\]

\[
e_{\text{accum}} = 0
\]

\[
\text{std}(\omega_1(t)) = q_1 \cdot \text{std}(\omega_1(t))
\]

\[
\text{std}(\omega_2(t)) = q_2 \cdot \text{std}(\omega_2(t))
\]

In this case, \(t_{\text{min}}\) corresponds to the instant in which the OFCL starts operating; \(e_{\text{obs}}\) is the observation error (the difference between the acquired measurement for the output and the one expected by the estimation algorithm); \(e_{\text{accum}}\) is a variable that accumulates the past observation errors with initial value of zero; \(e_{\text{rh}}\) is the decision threshold to modify the process noise. In other words, if it is lower than the threshold, the standard deviation of the process noise is reduced, but if it is larger than the threshold, it increases. Also \(p_1\) and \(p_2\) are constants with values between 0 and 1, while \(q_1\) and \(q_2\) are constants bigger than 1. Finally, \(\text{std}_1\) and \(\text{std}_2\) are the lower bounds which indicate the minimum standard deviation accepted value.

It is important to mention that the decision to increase the process noise includes a reset of the accumulated error, in order to allow the algorithm to have a time interval to correct its estimation before continuing to increase the uncertainty. In case that the observations do not meet the likelihood requirements, the accumulated error will become bigger than the threshold and the OFCL will increase the process noises. On the other hand, if small observation errors, accumulated during a prolonged time horizon, are able to surpass the activation threshold, the augmentation of the noise will only be done one time on that time horizon, and its effect will not be determinant on the performance of the method.

Table 3 summarizes the values of the parameters for the correction loops. The numeric differences for both methods are because the nature of each algorithm, basically the PF sensitivity to adjust its estimation, since the particles move quickly towards a zone with more likelihood with the observation. The reason because \(q_1\) has a bigger value than \(q_2\) is the decision of penalizing a higher uncertainty of the internal impedance estimation, since there is not available a good transition model for it.
Table 3. Parameter values for the OFCL in UKF and FP.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\min}$</td>
<td>5</td>
<td>200</td>
</tr>
<tr>
<td>$c_{1h}$</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>$p_1$</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>$p_2$</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>$q_1$</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>$q_2$</td>
<td>1.01</td>
<td>1.01</td>
</tr>
<tr>
<td>$std_1$</td>
<td>$10^{-5}$</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>$std_2$</td>
<td>$10^{-5}$</td>
<td>$10^{-4}$</td>
</tr>
</tbody>
</table>

3.6. Prognosis Performance

To evaluate the prognosis capability offered by the model and the outcomes of the estimation stage, the indices mentioned in Section 2 are used. Since these indices are functions of time, every value (at each time instant) requires to compute the output of the prognostic routine, conditional to the available information, until the end of the prediction horizon. To lower computational costs, one EoD prognosis result is computed every 10 iterations of the estimation module. When using the UKF within the estimation module, only one execution of the code is required (since it is a deterministic algorithm). However, in the case of PF algorithms, all results consider an average of 30 realizations of the code.

4. RESULTS

This research effort presents a comparison between filtering stages based on either PF or UKF, using OFCLs, and measuring the impact on the subsequent prognosis stage. For completeness purposes, and to measure the impact of OFCLs on filtering stages, we have also included results where classical version of the aforementioned filters are used during the estimation stage. Analysis is focused on estimation and prognosis of battery internal impedance, voltage and SOC. Experimental data were obtained from fully-charged cells (initial SOC is 100%), although initial condition always assumed 85% for the cell SOC to incorporate the effect of incorrect initial conditions.

4.1. PF-based Estimation and Prognosis

To establish a comparison between different estimation algorithms, it becomes convenient to establish a base scenario, which in this case corresponds to a classical implementation of PF-based estimation and prognosis modules. Since one execution of the PF code corresponds to a realization of a stochastic process, all conclusions require to analyze several realizations of the code. Figures show only one particular realization of the algorithm.

4.1.1. PF-based Estimation Results

Figure 2 shows PF-based estimates of the SOC, internal impedance, and voltage of the lithium-ion cell. The initial SOC of the battery is 100%, while for the PF the initial condition assigned is a uniform random sample between 76.5% and 93.5% (mean value of 85%) to evaluate if the algorithm is capable of correcting errors in the initial conditions. The initial condition assigned to the internal impedance is a Gaussian distribution sample of mean value of 0.1 and a variance of 2.5e-5. These values were determined experimentally in (Pola, et al., 2015), as shown in Table 1. The set of points plotted around the solid lines correspond to realizations associated with each particle, previous normalization of its weights through resampling.
4.1.2. PF-based Prognosis Results

Figure 3 shows one execution of the PF estimation and prognosis routine for a complete discharge cycle. On Figure 3 a), the filtered impedance value can be observed, as well as the value for each particle during the first stage. Later, the prognosis stage assumes that the impedance value is constant, while the 95% confidence level (thinner lines) increases in time. On Figure 3 b), it becomes notorious that the predicted voltage becomes fully discharged before the real data, hence a bad adjustment of the model towards the end of the discharge. Figure 3 c) shows the estimation and prognosis as well as the actual SOC value. Also, the discharge zone and the exact point when the battery is fully discharged (ground truth EoD). Finally, Figure 3 d) presents the probability density distribution for the time of failure or discharge, with a 95% confidence interval.

It is possible to notice that the procedure allows to implement a satisfactory prognosis scheme, in which no overestimation of the EoD time occurs. Moreover, the uncertainty is characterized in an adequate manner, which translates into a conservative approach.

4.2. Estimation and Prognosis results based on a combination of PF and OFCL

This section presents the results obtained when combining Outer Feedback Correction Loop (OFCLs) with the classical PF implementation. Once again, and since this is a stochastic algorithm, different results are obtained at each realization. Figures illustrate the average performance of the method, without perjury of realizations with better or worse results.

4.2.1. PF+OFCLs Estimation Results

Figure 4 shows the estimation using a PF+OFCLs when the initial SOC is of the battery is 100% and the assumed initial value is 85%. On Figure 4 a) the internal impedance module is shown. Here the dispersion of the particles is smaller, which implicates a smoother behavior.

Also, it is possible to notice on Figure 4 b) the status of the OFCL, and when it switches from “off” to “on”. Finally, Figure 4 c) shows the filtered and the offline SOC. It is possible to notice that the OFCL is able to quickly correct the initial condition, and correctly estimate the SOC ground truth.
4.2.2. PF+OFCL Prognosis Results

The results obtained for this approach are similar when the OFCL was not included. The main difference is shown in Figure 5a), since there is a reduction of the particle dispersion during the estimation stage, translated in a smaller 95% confidence intervals when doing prognosis. Figure 5b) shows the OFCL action. This action is defined as a two possible numbers: a number 1 indicates an increase of the standard deviation of the process noise, and a number 0 indicates a decrease of the same concept due to the good estimation performance. It is possible to note, that the prognosis of the discharge time is more accurate than the previous case. In other words, the distribution of the EoD time is closer to the ground truth.

4.3. UKF Estimation Module

Similarly to the previous case, results for the estimation and prognosis of the internal impedance, voltage, and battery SOC based on UKF schemes are now described. To measure the impact of OFCL on filtering stages, we have also included results where those correction loops were not activated. It is important to note that the UKF is used just for the estimation stage, since the prognostics are obtained using a PF-based scheme. In other words the estimation stage is performed using a UKF-based module, while the prognosis follows a classic PF-based implementation.
4.3.1. UKF Estimation Results

When using the UKF in the estimation module, the initial value of the state vector is characterized through a Gaussian distribution. This is intended to create similar conditions as the ones determined on the PF base scheme. Figure 6 shows the estimation realized with a UKF during one single discharge cycle. The dotted lines correspond to a 95% confidence interval, while the solid line indicates the estimation of the internal impedance and SOC. Figure 6c) shows that even though the UKF estimation quickly converges to actual SOC value during early stages of the algorithm execution, then eventually the filter diverges.

![Figure 6 a). UKF Internal impedance estimation](image)

![Figure 6 b). UKF Voltage measurement and estimation](image)

![Figure 6 c). UKF SOC estimation results](image)

It is important to mention that this poor performance condition coincides with periods in which larger currents values are demanded from the battery. In this situation, non-modeled dynamics of the battery affect measurements more evidently, reflecting on larger discrepancies for the results of the prognosis module. This fact is also reflected on large variances of the state vector.

4.4. Estimation and Prognosis based on a combination of UKF and OFCLs

4.4.1. UKF+OFCLs Estimation Results

The same procedure as before is applied to this new scheme, in which the OFCL is combined with an UKF-based estimation module. Figure 7 shows the results for this framework.

![Figure 7 a). UKF+OFCLs Internal impedance estimation](image)

![Figure 7 b). UKF+OFCLs Voltage measurement and estimation](image)

![Figure 7 c). UKF+OFCLs SOC estimation results](image)

The addition of the OFCLs improves considerably the performance, thus achieving more accurate SOC estimates. The reason for which the combination of UKF+OFCLs provides good results is that the empirical model obtained for the Li-Ion cell describes in a good way the real behavior during a large part of the discharge cycle. In this regard, the diminishment of the process noise is a result of the addition of the OFCLs, which allows the UKF to have a bigger robustness to measurements errors and certain flexibility to adapt when the observations do not match the one-step ahead predictions.
Another factor to consider is the larger variance on the estimation of the state when obtaining voltage measurements that are not similar to the model prediction. This effect can be reduced with a smaller covariance matrix, combined with a smaller process noise associated with the SOC evolution in time, considering the risk that estimates may be biased, since the assumed initial conditions are dissimilar to the actual conditions in the battery.

### 4.4.2. UKF+OFCL Prognosis Results

Figures 9a) to 9d) show the results for one realization of the prognosis module when using the UKF+OFCLs scheme during the filtering stage. It is possible to note an adequate performance according to what is expected. The results are similar to the ones obtained with the PF and the PF+OFCLs schemes, with the benefit that the accuracy of determining the discharge time is higher, associated to a good previous estimate of the battery SOC.
5. PERFORMANCE COMPARISON

Although graphic information is helpful to understand the performance and effectiveness of the different algorithms, it is not enough to evaluate the specific performance from a numeric point of view, so objective comparisons cannot be made. Even more, in the case of PF, one realization is not able to capture its real behavior, making it necessary to use the average of different realizations in order to establish an adequate characterization.

The following results correspond to three estimation experiments. Experiment #1 corresponds to the one shown on the previous figures, where the mean value of the initial condition is 85% of the SOC, while the real value is 100%. Similarly, Experiments #2 and #3 correspond to a SOC of 65% and 50%. Not all these results are shown in this article, since results from Experiments #2 and #3 exhibited similar performance as Experiment #1. The UKF without the OFCLs is left out of the experiments due to its poor performance. For the PF-based algorithms the average of 50 realizations is considered. The measurements were made at four time instants of the discharge period: near the beginning (200 seconds), two at the central area (1200 and 2700 seconds) and near the end (4100 seconds).

Additionally, a prognosis experiment is made where the performance indices explained before are accounted. These indices are time functions, so they require long term predictions at every instant during the whole discharge. To decrease the computational requirements, the predictions are made every 10 iterations. Also, since the computational cost is elevated, the numbers of realizations for the PF are reduced to 30.

5.1. Estimation Stage: 85% SOC initial charge assumed

The Tables 4, 5, 6 show the results for the different schemes. The SOC error is presented with a 95% confidence interval.

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>T=200</th>
<th>T=1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>[0.1110 0.9299]</td>
<td>[0.1115 0.7105]</td>
</tr>
<tr>
<td>Covariance (10^-4)</td>
<td>[0.0424 0.1425] [0.7900]</td>
<td>[0.3920 0.1423] [0.7519]</td>
</tr>
<tr>
<td>SOC error</td>
<td>0.0200 ± 0.0210</td>
<td>0.0004 ± 0.0203</td>
</tr>
<tr>
<td>Time</td>
<td>T=2700</td>
<td>T=4100</td>
</tr>
<tr>
<td>Mean</td>
<td>[0.1163 0.3692]</td>
<td>[0.1052 0.0735]</td>
</tr>
<tr>
<td>Covariance (10^-4)</td>
<td>[0.4081 0.0676] [0.2723]</td>
<td>[0.3320 0.0877] [0.1937]</td>
</tr>
<tr>
<td>SOC error</td>
<td>0.0105 ± 0.0206</td>
<td>0.0101 ± 0.0219</td>
</tr>
</tbody>
</table>

Table 5. Experiment #1: PF+OFCLs (average of 50 realizations).

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>T=200</th>
<th>T=1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>[0.1171 0.3695]</td>
<td>[0.1076 0.0775]</td>
</tr>
<tr>
<td>Covariance (10^-4)</td>
<td>[0.0275 0.0213] [0.1242]</td>
<td>[0.0318 0.0301] [0.1131]</td>
</tr>
<tr>
<td>SOC error</td>
<td>0.0102 ± 0.0184</td>
<td>0.0062 ± 0.0154</td>
</tr>
</tbody>
</table>

Table 6. Experiment #1: UKF+OFCLs.

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>T=200</th>
<th>T=1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>[0.1112 0.9299]</td>
<td>[0.1105 0.7154]</td>
</tr>
<tr>
<td>Covariance (10^-4)</td>
<td>[0.0404 0.1327] [0.7700]</td>
<td>[0.0643 0.1864] [0.8127]</td>
</tr>
<tr>
<td>SOC error</td>
<td>0.0195 ± 0.0190</td>
<td>−0.0045 ± 0.0090</td>
</tr>
<tr>
<td>Time (s)</td>
<td>T=2700</td>
<td>T=4100</td>
</tr>
<tr>
<td>Mean</td>
<td>[0.1171 0.3695]</td>
<td>[0.1076 0.0775]</td>
</tr>
<tr>
<td>Covariance (10^-4)</td>
<td>[0.0275 0.0213] [0.1242]</td>
<td>[0.0318 0.0301] [0.1131]</td>
</tr>
<tr>
<td>SOC error</td>
<td>0.0102 ± 0.0184</td>
<td>0.0062 ± 0.0154</td>
</tr>
</tbody>
</table>

Table 7. Experiment #1: UKF+OFCLs.

It is possible to note that the use of the OFCLs generates more accurate (smaller error) and more precise (smaller variance) estimations than the base PF used for comparison. In particular, the UKF+OFCLs is the algorithm with the highest accuracy, although its inability to represent multimodal distributions as the ones observed on the results. It is important to mention the considerable reduction of the variance of the internal impedance estimation module.

5.2. Performance measures

This section presents results obtained in terms of the evaluation of performance indices such as: precision, accuracy-precision, and on-line steadiness for prognosis. Figures 10a) to 10c) show the obtained values of the aforementioned performance indices, for the following cases: Base PF, PF+OFCLs and UKF+OFCLs.
6. CONCLUSIONS

Even though the UKF has a Square-Root version, which is reported as computationally more stable, and more efficient in times, this variant is not convenient for the treated problem. The realized implementations showed a more elevated execution time, given the model dimensionality. That is to say, for a two state characterization it is more efficient to calculate at each iteration of the UKF the square root of a matrix, which can be done with a Cholesky decomposition or in the analytical way for case of $2 \times 2$ matrices.

The effectiveness of the programmed algorithms (performance in estimation and prognosis) is improved when the OFCLs are incorporated in all cases of study. The UKF without the OFCL has a poor performance, and is not recommended, but if the OFCL is added, the performance is even better than the PF schemes, as long as there is a reliable model. This means that the process and ideally the observation noises have to be small enough or be able to allow its diminishment thorough OFCLs.

The results of the UKF are favored since the observation model and the state transition have a mainly linear behavior during the intermediate part of the discharge.

The PF schemes, with or without the OFCLs have acceptable results with SOC estimation errors that are below a 4% of the real value, except when the assumed and real initial condition are very different. The proposed structure of OFCLs allows an improvement on the performance of the different estimation algorithms. This means that the accumulated observation error is a useful index to make decisions of how to modify the model hyper parameters. This means that when facing a SOC estimation problem, it is highly recommended to start the study with a PF scheme to verify that the model is able to describe the phenomenology of the battery. If good results are obtained, the implementation of an UKF+OFCLs can help improve the consistency and quality of the results, and even the execution time depending on the platform that is implemented.

ACKNOWLEDGEMENTS

This work has been partially supported by FONDECYT Chile Grant Nr. 1140774, and the Advanced Center for Electrical and Electronic Engineering, Basal Project FB0008. The work of Jorge Silva was supported by CONICYT-Chile, FONDECYT Chile Grant Nr. 1140840. The work of Aramis Pérez was supported by the University of Costa Rica (Grant for Doctoral Studies) and CONICYT-PCHA/Doctorado Nacional/2015-21150121. The work of Vanessa Quintero was supported by the Universidad Tecnologica de Panama and IFARHU (Grant for Doctoral Studies). The work of Francisco Jaramillo was supported by CONICYT-PCHA/Doctorado Nacional/2014-21140201.
REFERENCES


Ran, L. J. (2010). Prediction of state of charge of lithium-ion rechargeable battery with electrochemical impedance spectroscopy theory. the 5th IEEE Conference on Industrial Electronics and Applications.


BIographies

B. Sc. Carlos Tampier received the B.Sc. degree in Electrical Engineering from Universidad de Chile, Santiago, Chile, in 2015. Currently he is a researcher at the Advanced Mining Technology Center (AMTC) of the University of Chile. His research interests include control systems, data analysis and machine learning with applications to robotics.

M. Sc. Aramis Pérez is a Research Assistant at the Lithium Innovation Center (Santiago, Chile) and Professor at the School of Electrical Engineering at the University of Costa Rica. He received his B.Sc. degree (2002) and Licentiate degree (2005) in Electrical Engineering from the University of Costa Rica. He received his M.Sc. degree in Business Administration with a General Management Major (2008) and also he is a M.Sc. candidate in Industrial Engineering from the same
Current research interests include parametric/non-parametric modeling, system identification, data analysis, machine learning and manufacturing processes.

B. Sc. Vanessa Quintero received her B.Sc degree from Electronics and Telecommunication Engineering at the Universidad Tecnologica de Panama (2007). Currently she is a doctorate student at the University of Chile. Her research interests include estimation, prognostics with applications to battery and protocols design.

B. Sc. Francisco Jaramillo received the B.Sc. degree in Electronics Engineering from Universidad de La Frontera, Temuco, Chile, in 2009. Currently he is a doctorate student at the Department of Electrical Engineering at the University of Chile under Dr. Marcos E. Orchard supervision. His research interests include machine learning, control systems, and estimation and prognosis based on Bayesian algorithms with applications to nitrogen removal in pilot-scale Sequencing Batch Reactors for Wastewater Treatment Plants.

Dr. Marcos E. Orchard is Associate Professor with the Department of Electrical Engineering at Universidad de Chile and was part of the Intelligent Control Systems Laboratory at The Georgia Institute of Technology. His current research interest is the design, implementation and testing of real-time frameworks for fault diagnosis and failure prognosis, with applications to battery management systems, mining industry, and finance. His fields of expertise include statistical process monitoring, parametric/non-parametric modeling, and system identification. His research work at the Georgia Institute of Technology was the foundation of novel real-time fault diagnosis and failure prognosis approaches based on particle filtering algorithms. He received his Ph.D. and M.S. degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He received his B.S. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile. Dr. Orchard has published more than 50 papers in his areas of expertise.

Dr. Jorge F. Silva is Associate Professor at the Department of Electrical Engineering, University of Chile, Santiago, Chile. He received the Master of Science (2005) and Ph.D. (2008) in Electrical Engineering from the University of Southern California (USC). He is IEEE member of the Signal Processing and Information Theory Societies and he has participated as a reviewer in various IEEE journals on Signal Processing. Jorge F. Silva is recipient of the Outstanding Thesis Award 2009 for Theoretical Research of the Viterbi School of Engineering, the Viterbi Doctoral Fellowship 2007-2008 and Simon Ramo Scholarship 2007-2008 at USC. His research interests include: non-parametric learning; sparse signal representations, statistical learning; universal source coding; sequential decision and estimation; distributive learning and sensor networks.
Towards a Methodology for Design of Prognostic Systems

Jose Ignacio Aizpurua and Victoria M. Catterson

Institute for Energy and Environment, University of Strathclyde, Glasgow, United Kingdom

Abstract

An effective implementation of prognostic technology can reduce costs and increase availability of assets. As a result of the rapidly growing interest in prognostics, researchers have independently developed a number of applications for asset-specific modeling and prediction. Consequently, there is some inconsistency in the understanding of key concepts for designing prognostic systems. This further complicates the already-challenging design of new prognostic systems. In order to progress from application-specific solutions towards structured and efficient prognostic implementations, the development of a comprehensive and pragmatic methodology is essential. Prognostic algorithm selection is a key activity to achieve consistency throughout the design process. In this paper we present a design decision framework which guides the designer towards a prognostic algorithm through a cause-effect flowchart. Failure modes, application characteristics, and qualitative and quantitative metrics are used to determine an appropriate approach for the stated problem. The application of the methodology can reduce the time and effort required to develop a prognostic system, ensure that all the possible design options have been considered, and provide a means to compare different prognostic algorithms consistently. The framework has been applied to different prognostic problems within the power industry to illuminate its effectiveness. Case studies are presented to show how the framework guides designers through the choice of prognostic algorithm according to system requirements. The results demonstrate the applicability of the methodology to the design of prognostic systems which consistently meet the established requirements.

1. Introduction

Successful implementations of prognostic techniques provide benefits for maintenance planning which result in cost-effective operation of assets (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2007). Traditional approaches to design of prognostic systems have been focused on applying prognostic techniques on a case-by-case basis to create a fit-for-purpose solution for each application. These solutions are not easily transferable to other domains, and therefore impede the adoption of prognostics applications in industrial fields.

In order to generalize the adoption of prognostics techniques a clear and consistent justification of the use of prognostic algorithms is needed. This justification should provide mechanisms for prognostics model selection so as to integrate this criteria into the design flow. Accordingly, in this paper we present a generally applicable methodology to design prognostics applications systematically. The main goal of the methodology is to choose a priori an adequate prognostics algorithm that meets the system requirements. This requires shifting from taxonomy and classification of prognostics approaches towards a design framework for the systematic selection and design of prognostic applications based on strategic decision points.

The main contribution of this paper is the development of a design decision framework which integrates the knowledge needed to design prognostic applications. As a proof-of-concept, we have analyzed its usability in different applications within the power industry. The successful implementation of this framework can (1) reduce the time and effort required to develop a prognostic system; (2) ensure that all the possible design options have been considered; and (3) provide a means to compare different prognostic algorithms consistently.

The remainder of the paper is organized as follows: Section 2 presents the state of the art analyzing existing prognostics methodologies and classifications. Section 3 defines the overall methodology and the activities undertaken within the methodology. Section 4 specifies the design decision framework as a crucial activity within the design methodology. Section 5 presents the applicability of the design decision framework through the analysis of different case studies within the power industry. Finally, Section 6 draws conclusions and presents the future work of this research.

Jose Aizpurua et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
2. State of the Art

Due to the fast growth of prognostics applications, there are divergences in the literature with respect to the definition of prognostics (e.g., see (Sikorska, Hodkiewicz, & Ma, 2011) for different definitions). Literally, the word prognosis is a combination of two Greek words: prog - before; and gnosis - knowledge. Accordingly, a widely accepted prognosis definition is: the ability to acquire knowledge about events before they actually occur (Vachtsevanos et al., 2007). While in the medical field it has been used to predict the probable course of a disease, in the industrial field it is aimed at foretelling the Remaining Useful Life (RUL) of a component after a fault (or a specific failure mode) is diagnosed, i.e., prognosis specifies the fault-to-failure progression of an asset.

Accordingly, in this work we consider prognosis as the process of assessing the RUL of a component based on run-to-failure data, degradation-specific equations, or their combinations. These predictions must include mechanisms to represent the inherent uncertainty of a prognosis, and predict within reasonable bounds (Sankararaman, 2015).

Prognostics techniques focus on predicting fault progression and providing an early indicator of the RUL in order to implement asset-specific maintenance strategies, and thereby reduce costs and increase availability. In recent years a plethora of new techniques have been proposed for prognosis of engineering assets. Our goal is to design a prognostics methodology for the systematic design of prognostic applications including systematic prognostics algorithm selection. Accordingly, we review the scientific literature addressing the proposed prognostics algorithm classifications (cf. Subsection 2.1) and prognostics methodologies (cf. Subsection 2.2).

2.1. Classification of Prognostics Techniques

Two groups, data-driven and model-based prognostics techniques, have been identified by many authors as prognostics approaches derived from historical data and expert knowledge respectively (e.g., see (An, Kim, & Choi, 2015)). However, not all the proposed classifications in the prognostics arena have been limited to these groups. This situation emphasizes the general lack of agreement on fundamental design activities. There is no unique solution for the classification criteria, and depending on the viewpoint, the same approach can be classified in a different way. However, a generally accepted classification framework is needed for the systematic design of prognostic systems.

(Sikorska et al., 2011) define four groups for RUL prediction influenced by the ISO 13881-1 (ISO, 2004): knowledge-based, life expectancy, Artificial Neural Networks (ANNs), and physical models. Prognostics techniques are grouped accordingly, and their advantages and disadvantages are discussed. Despite ANNs having been widely used for many prognostics applications (Haykin, 1998), attributing a group level entity to a specific technique may not be accurate. Besides, due to the ambiguity of some groups, some techniques can fit in more than one group. For instance, Particle Filtering (PF) (Daigle, Saha, & Goebel, 2012) is grouped within life expectancy models. However, according to the engineering requirements, PF needs a degradation equation and observation data for RUL estimation. Therefore, it could fit within physical models as well. They define advantages and disadvantages and tips for using or avoiding each approach. This classification is useful for case-by-case comparison between alternative approaches, but it is necessary to link these approaches through a design process to integrate them seamlessly (e.g., design decision points to choose a model according to design requirements).

The classification proposed in ISO 13881-1 focuses on 12 different groups (ISO, 2004) (see Table 1). This results in a flat classification tree without hierarchies. It is possible to further refine this classification by gathering the proposed groups to create structured and non-overlapping boundaries and choose a model according to design requirements.

(Si, Wang, Hu, & Zhou, 2011) further develop data-driven statistical approaches based on the direct or indirect nature of the condition monitoring data. For direct condition monitoring data the following groups are addressed: regression based, Wiener, Gamma, and Markov processes; while for indirect condition monitoring data: stochastic filtering approaches, covariate hazard approaches, and Hidden Markov Model based approaches are covered.

(Lee et al., 2014) provide an overview of alternative approaches with their respective advantages and disadvantages. Unfortunately, they are considered separately and there is no link between them. The authors suggest a ranking method based on concepts of quality function deployment and house of quality (Govers, 1996) to rank the suitability of prognostics algorithms with respect to the specific problem. A combination of engineering attributes and customer needs is used to rank prognostics algorithms. The idea of ranking prognostics algorithms is interesting, but still the designer needs to assess the adequacy of the algorithm on a case-by-case basis.

(An et al., 2015) group approaches into model-based and data-driven techniques. The authors present practical options to select a prognostic algorithm identifying possible issues for data-driven (Neural Networks, Gaussian Process Regression) and model-based (Particle Filtering) techniques and comparing their results through a case study. Aligned with our design decision framework, the authors present a model selection tree with 3 decision points (1) existence of information: physical model, loading or no information; (2) damage growth: simple or complex; and (3) noise level: small or large. From these decision points four prognostic techniques are suggested. In our design framework we address a com-
plete set of prognostic approaches including combinations of model-based and data-driven approaches. To this end, it is necessary to consider more design decision points, highlighting the cause/consequences of alternative paths in the tree.

(Liao & Kottig, 2014) classify prognostics approaches into Experience-Based (EB), Data-Driven (DD), and Physics-Based (PB) models. From the combination of these approaches, they provide a comprehensive overview of hybrid approaches identifying the following groups: (1) EB with DD; (2) EB with PB; (3) DD with DD; (4) DD with PB; and (5) a combination of EB, DD, and PB. The proposed flowchart for hybrid approaches is influenced by this work (cf. Subsection 4.3). We complement this work by including (1) high-level drivers to select a hybrid prognostic configuration; and (2) different connections between DD and MB approaches.

Table 1 displays the approaches gathered in this subsection considering relevant grouping aspects and analyses if the approach addresses model selection aspects.

Table 1. Summary of prognostics classification approaches

<table>
<thead>
<tr>
<th>Reference</th>
<th>Prognostic groups</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sikorska et al., 2011)</td>
<td>Knowledge-based, life expectancy, ANNs, &amp; physical models</td>
<td>x</td>
</tr>
<tr>
<td>(ISO, 2004)</td>
<td>Behavioral models, statistical, probabilistic, ANNs, life expectancy, reliability based, deterioration based, knowledge based, rule based, causal tree, &amp; case-based reasoning</td>
<td>x</td>
</tr>
<tr>
<td>(Si et al., 2011)</td>
<td>Data-driven</td>
<td>x</td>
</tr>
<tr>
<td>(Lee et al., 2014)</td>
<td>No grouping</td>
<td>✓</td>
</tr>
<tr>
<td>(An et al., 2015)</td>
<td>Model-based &amp; data driven</td>
<td>✓</td>
</tr>
<tr>
<td>(Liao &amp; Kottig, 2014)</td>
<td>Experience-based, data-driven, physics based, &amp; hybrid</td>
<td>x</td>
</tr>
</tbody>
</table>

Legend: MS: Model Selection; ANNs: Artificial Neural Networks

There are some papers in the literature that deal with model selection related issues (Lee et al., 2014; An et al., 2015). However, the addressed techniques are only a subset of the existing approaches for prognostics applications. As for the classification criteria, the common factors for all the reviewed approaches are the data-driven (including Neural Networks, reliability, and life-expectancy groups) and model-based (including behavioral and physical groups) techniques. Besides, it is possible to consider experience (knowledge) based techniques as another group, but there are not many techniques which can be grouped here other than Fuzzy logic. Therefore, for the sake of simplicity, we will not consider it as a separate group (see Section 4 for more details).

The classification of prognostic approaches is not of practical use without a clear connection with the system design process. It helps the designer to choose a group of approaches, but within the same group further design choices need to be adopted to select a suitable prognostics algorithm according to system requirements. This requires introducing engineering criteria into the classification trees in order to adopt prognostics design decisions systematically. To this end, we propose the transformation from classification-like approaches towards design decision-like flowcharts based on trade-off analyses and design decision metrics.

2.2. Prognostics Design Methodologies

The need to develop a generally applicable methodology has been recognized in the literature (Uckun, Goebel, & Lucas, 2008). However, some of the proposed approaches have used a particular solution technique (e.g., see (Peysson et al., 2009)), and others need to be developed further in order to be generally applicable. This subsection analyses some of the prognostics methodologies presented in the literature, to explain the direction of this work.

(Kumar, Torres, Chan, & Pecht, 2008) proposed a methodology for electronic products. To this end, they (1) identify the critical failures, (2) establish a healthy baseline based on monitoring data, (3) incorporate a physics-of-failure model into the prognostics model, and (4) evaluate the RUL based on the Mahalanobis distance from baseline. Although the hybrid approach reduces uncertainty, the method is not generally applicable because it may not be feasible for specific requirements (e.g., lack of run-to-failure data). For the sake of generality, prognostics model selection criteria is necessary instead of focusing on a specific prognostics algorithm.

(Uckun et al., 2008) identified the need of a universal methodology to design prognostics and health management systems and gather some of the key activities of the methodology (see Table 2). Some of these activities have been formalized: transformation from high-level requirements to business case (Saxena et al., 2012); (2) metric selection (Saxena et al., 2008); and (3) validation and verification tests (L. Tang, Orchard, Goebel, & Vachtsevanos, 2011). A key activity that the methodology must integrate is the definition and integration of metrics as a means to introduce consistency for alternative techniques. This standardization provides mechanisms to compare prognostics approaches consistently.

(Peysson et al., 2009) introduced a methodology to perform prognostics of complex systems using damage trajectory models. The methodology introduces a generic modeling formalism for system specification linking environment, mission and process (or resources) variables. The environmental model is specified using Fuzzy logic and the damage models used are abaci. The generalization comes from the formal system specification in order to perform prognostics of complex multi-component systems. However, the methodology lacks a generalized prognostic model selection process.

(Lee, Liao, Lapira, Ni, & Li, 2009) presented a methodology for the design of e-manufacturing systems comprised of the
following steps: (1) streamline: identify critical components and sort/filter/prioritize data to ensure quality; (2) smart processing: evaluate degradation, predict performance, and diagnose the failure; (3) synchronize: use of advanced technologies (e.g., agents) to introduce transparency; (4) standardize: systematic prognostics selection, platform integration, and maintenance information standardization; (5) sustain: closed-loop life cycle design (real-time feedback); embedded self-learning; and user-friendly development. The methodology integrates the Watchdog Agent (Djurdjanovic, Lee, & Ni, 2003) for automated tool selection. It ranks prognostics algorithms based on process properties (stationarity, expert knowledge, cost, computation, data dimension, or prediction span) and implements the highest ranked technique. However, the prognostics techniques considered in this toolbox are a subset of data-driven techniques, and they do not include model-based and hybrid prognostics techniques.

Table 2 shows the approaches gathered in this subsection considering relevant design activities and analyses if the approach addresses model selection aspects.

Table 2. Summary of prognostics methodology approaches

<table>
<thead>
<tr>
<th>Reference</th>
<th>Methodology steps</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Kumar et al., 2008)</td>
<td>FMEA; health monitoring; baseline definition; anomaly detection; param. isolation; &amp; PoF-load matching</td>
<td>x</td>
</tr>
<tr>
<td>(Uckun et al., 2008)</td>
<td>Requirements transformation; metric; fault, sensor and model selection; validation &amp; verification</td>
<td>x</td>
</tr>
<tr>
<td>(Peysson et al., 2009)</td>
<td>System modeling; &amp; prognostics analysis (damage evaluation)</td>
<td>x</td>
</tr>
<tr>
<td>(Lee et al., 2009)</td>
<td>Streamline; &amp; prognostics analysis (damage evaluation)</td>
<td>✓</td>
</tr>
</tbody>
</table>

Legend: MS: Model Selection; PoF: Physics of Failure

In summary, there is no generally applicable methodology which suggests a prognostic technique according to the user requirements. To aid in this process we introduce a formal procedure for the design of prognostic systems in order to approach the task systematically. This should simplify prognostic system design, and avoid repetition of redundant process steps for every application.

3. METHODOLOGY OVERVIEW

The proposed methodology framework assumes four design stages: (1) fault coverage, (2) model selection, (3) requirements transformation, and (4) validation and verification. Prognostic system developers must consider each in turn. Figure 1 depicts the generic prognostics methodology structuring these activities to meet the design requirements.

From the literature analysis some of these steps have been identified (cf. Subsection 2.2). The four stages integrated in the methodology are:

- Fault coverage or Failure Mode (FM) choice through formal criticality assessment techniques (e.g., FMECA (US Department of Defense, 1980), importance measurements (Borgonovo & Apostolakis, 2001)). Applications may prioritize a single fault type, aging behavior, or a number of important failure modes.
- Systematic prognostics model selection: a prognostic system must contain a model of degradation. This model can be simple (e.g. linear decrease of a single parameter) or more complex. It could be derived from data, or based on engineering understanding (e.g. a physics-of-failure model). According to available engineering resources, the failure mode of interest, and application specific requirements, this activity determines which is the best prognostics model.
- Transformation from high-level requirements into application specific metrics (e.g., see (Saxena et al., 2012)). This step introduces consistency by defining a transformation step to evaluate different prognostics models under the same criteria, i.e., prognostics metrics.
- Validation and verification: validate the proposed model according to the prognostics metrics (e.g., see (L. Tang et al., 2011)).

The choices made throughout the methodology impact the immediately connected steps, and may lead to iteration of previous steps. For instance, if system requirements are not met, the designer should reconsider the initial system requirements or the adopted failure mode.

While all the outlined activities are important for the design of prognostic applications, the main focus of this paper is on prognostics model selection. We plan to address the remainder of the design activities in forthcoming publications (see Section 6).
4. Design Decision Framework for Model Selection

To present a comprehensive model selection decision framework, the applicability of different algorithms must be well understood. To synthesize this knowledge, we define a design framework based on strategic decision points, developed by analyzing case study prognostic systems. Enough cases have to be considered that general guidance can be usefully extracted, and also that a broad set of differing requirements are represented. The framework guides the designer through the prognostics algorithm selection process illuminating the trade-offs and cause-effect influences of alternative design decision points.

The approaches presented in the scientific literature focus on comparing alternative algorithms by implementing quantitative metrics (e.g., error, cost) after the development of the algorithm as post-implementation indicators. This approach results in a case-specific analysis that increases the design cost due to the need of implementing alternative algorithms for the same application. Interestingly, there is room to guide the designer in the pre-implementation phase towards an adequate prognostic algorithm by examining relevant design options, e.g., data properties; computational complexity; degradation patterns; failure thresholds; or uncertainty management.

Existing prognostic approaches are classified into three high-level groups: data-driven, model-based, and hybrid prognostic techniques. Data-driven techniques use monitoring data to fit a model of system behavior to the historical run-to-failure data (see Subsection 4.1). Model-based techniques require system knowledge in the form of the system’s degradation equations (see Subsection 4.2). Hybrid approaches emerge in different configurations arising from the (intra or inter) combinations of data-driven and model-based techniques. Input requirements for the hybrid approaches depend on the hybrid configuration itself (see Subsection 4.3).

The selection of the high-level prognostics algorithm group is driven by the available engineering resources. That is, when run-to-failure data or knowledge of system’s degradation equation is available, data-driven or model-based approaches are selected respectively. However, when both engineering resources are available, the selection of the high-level group incurs a trade-off decision between the availability of statistically significant run-to-failure data and complexity of the degradation equation. It may be the case that the degradation equation is too complex to model the system behavior accurately. Accordingly, data-driven techniques can be selected, provided that statistically significant run-to-failure data is available. Otherwise, hybrid prognostics techniques can be selected if the complexity is manageable and there is enough run-to-failure data.

Once the high-level prognostics group is chosen, other design criteria are used to trace a path through the group-specific flowchart, i.e., requirements and failure mode of interest.

4.1. Data-Driven Approaches

Data-driven prognostics algorithms rely on the available data to fit a model of the system behavior. The data must include run-to-failure conditions of the component under study in order to predict the RUL. Generally data-driven approaches are based on statistical pattern recognition and machine learning techniques. The main assumptions of data-driven approaches are that (i) the statistical features remain unchanged until a failure occurs or they change in a predictable way as the fault progresses; and (ii) availability of run-to-failure data. Thereby, the quality of the dataset determines the performance of the data-driven prognostic application. In some fields it is difficult to obtain the run-to-failure data (e.g., safety-critical or new systems).

Assuming that data-driven approaches have been selected as appropriate solutions for the application under study, the design decision process starts by examining uncertainty requirements for RUL estimation. Adequate management and representation of the uncertainty is necessary to predict the RUL with confidence, especially for safety-critical systems (Sankararaman, 2015). The deterministic estimate of the RUL may not be an adequate indicator because of its lack of judgment about the inherent uncertainty of the system. While the confidence intervals over the RUL provide a means to bound the estimation, the Probability Density Function (PDF) of the RUL estimation not only determines RUL bounds, but can also be propagated for system level uncertainty assessment. Consequently, the most accurate and potentially useful prognostics estimation will include the PDF of the RUL estimation.

Therefore, the first decision point evaluates if it is necessary to include the PDF of the RUL or not (see Figure 2). Accordingly, different prognostics algorithms can be selected.

If there is no need to extract the PDF of the RUL according to design requirements, there are alternative solutions depending on the complexity of the data, prediction span (short-term or long-term prediction), system specifications, available dataset, and knowledge of reliability distributions. Monotonicity ($m$) is used as a measure of the data complexity calculated as follows (Coble, 2010):

$$m = \text{mean}(\frac{\#\text{positive} \cdot dt}{n} - \frac{\#\text{negative} \cdot dt}{n})$$  \hspace{1cm} (1)

where $n$ is the number of data windows in the dataset and $t$ is the time scale. Monotonicity is a relevant degradation parameter under the assumption that an asset will not go through repair until reaching the system failure.
If the data reflects a simple linear monotonic degradation ($0.8 \leq m \leq 1$) Linear Regression is an appropriate solution for RUL estimation (e.g., see (Rudd, Catterson, McArthur, & Johnstone, 2011)). However, if the data is not clearly monotonic ($m < 0.8$), more sophisticated techniques are needed. If the goal is to perform a short-term prediction (e.g., 1 step ahead prediction), linear time-series models provide an easy to implement and accurate prognostics implementation (Ling, 2013): ARMA models are better suited for weakly stationary processes, while ARIMA is a generalization of the ARMA model able to deal with non-stationary processes. A weakly stationary process must satisfy two conditions: mean and variance must be constant; and the autocovariance between $X_t$ and $X_{t+\tau}$ must only depend on the lag $\tau$. (Ling, 2013) introduced a Bayesian updating method for ARIMA models for uncertainty management.

For long-term prediction models, the next decision point is if the designer has knowledge of the system’s state-space specification. State-based models define the system behavior through a multi-state specification transitioning from a healthy state towards a failed state through multiple degradation states. In Hidden Markov Models (HMMs) (Tobon-Mejia, Medjaher, Zerhouni, & Tripot, 2011) the state is not directly observable, but it is deduced from observations. HMMs hold the Markovian assumption (future states are independent of all past states but the current one— independent degradation) which may be too restrictive for some systems. To overcome this assumption Hidden semi-Markov Models (HSMM) were proposed assuming a general distribution between states (Tobon-Mejia et al., 2011). With HMM and HSMM it is possible to calculate confidence values with the deterministic RUL estimation.

However, if it is not possible to specify the system behavior through state-based approaches, the system’s behavioral pattern may be inferred from past historical experience. If multiple run-to-failure datasets of the same component are available, case-based reasoning approaches may be implemented. These techniques analyze data, define the health index (or baseline) based on data features, and accordingly evaluate if the new test data is healthy or not and predict the RUL. These approaches assume that components used for testing and training go through the same degradation process and require multiple run-to-failure histories to reuse knowledge and create predictions. If the available dataset has multiple different features Match Matrix is an appropriate solution (J. Liu, Djurdjanovic, Ni, Casoetto, & Lee, 2007). Match matrix improves ARMA models for long-term multivariate predictions, but it suffers from com-

---

**Figure 2. Flowchart for data-driven algorithm selection**

A flowchart for selecting data-driven algorithms based on the characteristics of the data and the prediction requirements. The algorithm selection process is guided by questions regarding the monotonicity of the degradation process, the need for short-term vs. long-term prediction, the availability of expert knowledge or historical data, and the type of model that can handle covariates or multiple features.
putational efficiency. Otherwise, if the dataset has a single feature, distance based approaches are more efficient for online applications. If there is expert knowledge to define the similarity or difference among alternative runs, Fuzzy-Based Similarity (Zio & Maio, 2010) evaluates the distance between alternative run-to-failure data based on Fuzzy membership functions instead of crisp distance evaluations. For online univariate implementations without expert knowledge, Trajectory Based Similarity (TBS) (Tianyi, 2010) approaches could be implemented.

If there is little run-to-failure data and the designer has knowledge of reliability models, the next decision point evaluates if it is necessary to take into account covariate influences (i.e., external factors). If so, the Proportional Hazard Model (PHM) (and its variants) can estimate the RUL considering external environmental influences on the component’s lifetime (Gorjian, Ma, Mittinty, Yarlagadda, & Sun, 2010). For univariate reliability models, Weibull regression approaches (Trappey, Trappey, Ma, & Tsao, 2014) are well suited for non-monotonic data. Weibull based regression approaches require fitting the data according to the Weibull distribution parameters. Note that the Weibull distribution can be adapted to a variety of reliability distributions by fitting the parameters (e.g. exponential, Rayleigh) to provide the corresponding failure time distribution.

If the designer does not have practical knowledge of reliability distributions, it is still feasible to implement a Curve Fitting approach in order to fit the data with, for example, a polynomial function. Otherwise, black-box prognostic approaches estimate the RUL without interpreting the transformation process from the input data towards the output data, i.e., RUL estimation. These approaches may be useful for complex applications in which it is difficult to come up with a relationship between the input and output data.

Both Artificial Neural Networks (ANN) (Haykin, 1998) and Support Vector Regression (SVR) (Smola & Schlkopf, 2004) are semi-parametric black-box approaches suitable for prognostics analyses. ANN is a widely adopted black-box prognostic technique which provides a deterministic estimate of the RUL prediction. SVR estimates the functional relation between input and output random variables under the assumption that the joint distribution is completely unknown. The model created by SVR depends only on a subset of the training data.

For the SVR the kernel function parameters have to be estimated from the data, while for ANNs the architecture needs to be determined. The estimation of these parameters constrain the accuracy of both techniques. Another difference is that ANN suffers from the local minimum problem, while SVR gives globally optimal solutions. Probably, the wider acceptance of ANN is because there are many software implementations for ANNs, while fewer easy-to-use implementations are available for SVR. To provide RUL confidence intervals, ANNs have been extended towards Confidence Prediction Neural Networks (CPNN) (Khawaja, Vachtsevanos, & Wu, 2005).

As for the approaches which estimate the PDF of the RUL, the first decision point analyses if the system’s state-based specification is available or not. If it is available, Dynamic Bayesian Networks (DBN) are a feasible option (Iamsumang, Mosleh, & Modarres, 2014). DBN models can be specified using graphical models making them an appropriate framework for the prognostic assessment of complex systems. If the state-based specification is not available, but there are multiple run-to-failure data histories, an Enhanced TBS approach can be implemented (Lam, Sankaraman, & Stewart, 2014).

Otherwise, if the degradation process can be represented with the Markovian memoryless property, there are different options depending on the monotonicity of the dataset: if the dataset represents a monotonic degradation pattern ($0.8 \leq m \leq 1$) Gamma process based prognostic implementations are feasible (Son, Fouladiarad, & Barros, 2012); otherwise, Wiener process is more appropriate for non-monotonic degradation patterns (S. Tang, Yu, Wang, Guo, & Si, 2014). Both approaches require fitting the data to the process-specific parameters.

Finally, if the degradation process does not adhere to the Markovian process, data/function-dependent techniques are considered. These techniques require choosing correct parameters and functions to fit the actual data. Namely, Relevance Vector Machines (RVM) (Tipping, 2001) and Gaussian Process Regression (GPR) (Rasmussen & Williams, 2006) approaches require choosing an appropriate Kernel and covariance functions respectively. The final performance of RVM and GPR depends on the chosen data and function (Goebel, Saha, & Saxena, 2008). GPR is a Kernel method with Bayesian treatment for regression. It integrates multiple variables by fitting a normal distribution and then applies Bayes’ rule to predict the future based on the past. However, it has relatively expensive memory and CPU requirements, and therefore may not be suitable for online operation. One solution to this problem is to distribute the implementation as in (Saha, Saha, Saxena, & Goebel, 2010). RVM is a Bayesian-inference inspired implementation of Support Vector Machines. See (Yan, Liu, Han, & Qiu, 2013) for a RVM application and see (Goebel et al., 2008) for a comparison between RVM and GPR (and ANN).

The flowchart for data-driven algorithm selection has been designed symmetrically with respect to the uncertainty requirements for the RUL specification. The majority of approaches which estimate the probability density function of the RUL, extend their non-PDF counterpart techniques including mechanisms for uncertainty analysis and representa-
tion, i.e., DBN generalizes HMM; enhanced TBS generalizes TBS; and RVM generalizes SVR.

The order of the model-selection decision points defines priorities for the prognostics algorithm selection process. The ordering is dependent on the preference of the system designer. The flowchart in Figure 2 prioritizes system knowledge (e.g., state-based specification) with the idea that system knowledge provides added value compared with generic prognostics approaches (e.g., curve fitting, black-box techniques). In other words, situation-specific prognostics algorithms are prioritized with respect to generally applicable techniques. Note that other orderings are also possible according to the designer’s preference (e.g., complexity of the prognostic technique implementation). An interesting extension would be to parametrize decision points according to different properties (e.g., system knowledge, complexity) resulting in different algorithm selection flowcharts. This way, the decision points can be rearranged dynamically according to user-defined preferences (see Section 6).

4.2. Model-Based Approaches

For some safety-critical systems, and when the new system has not been produced yet, data-driven approaches are not viable because there will not be enough run-to-failure data to apply data-driven techniques — although there are exceptions such as the use of high fidelity simulators which can produce the necessary run-to-failure data (e.g., see (McGhee, Galloway, Catterson, Brown, & Harrison, 2014)). In these cases model-based prognostic approaches can be considered. The selection of model-based prognostic techniques is motivated by the availability of knowledge of the physical degradation phenomenon, or both knowledge of the degradation equation and actual observations (see Figure 3).

If there are no observations and only engineering knowledge is available, a Physics of Failure (PoF) model should be created defining the system degradation behavior through physics of failure equations. PoF approaches use the system’s degradation properties (e.g., material, loading conditions, geometry) to identify degradation trends (typically due to over-stress or wear-out) and estimate the RUL (Vachtsevanos et al., 2007).

If observations are available in conjunction with the engineering knowledge, the RUL prediction can be solved via Bayesian tracking (or filtering) approaches. These approaches use two dependent equations to predict the future degradation of the system: the measurement equation which estimates the current state of the system (posterior PDF); and the process model which predicts the future state of the system using the current state of the system. The main assumptions to apply Bayesian tracking methods are: (i) the states follow a Markov process (the current state depends only on the previous state and actual conditions); and (ii) the observations are independent of the given states.

Among Bayesian tracking methods, the Kalman Filter focuses on the analysis of linear degradation trends. The following conditions must be satisfied to consider a function \( f(x) \) as linear: (1) \( f(x_1 + x_2) = f(x_1) + f(x_2), \forall x_1, x_2 \); and (2) \( f(\alpha x) = \alpha f(x), \forall \alpha \).

However, if these conditions are not satisfied, there are other alternatives for non-linear degradation trend analysis. Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) both are non-linear filters which assume Gaussian distribution for the states and noise (Daigle et al., 2012). Since the UKF provides better accuracy for highly non-linear degradation trends compared with the EKF (Daigle et al., 2012), in Figure 3 we have not added the EKF approach. For non-linear systems without the Gaussian distribution assumption, Particle Filtering approaches have been widely implemented with accurate results. (Daigle et al., 2012) showed in their case study that the accuracy and computational cost of UKF outperform Particle Filtering.

Generally, model-based prognostics techniques are more specific (and complex) than data-driven techniques (e.g., PoF models). For simplicity, we have not further developed the flowchart in Figure 3 and we have included a discussion for asset-specific model-based approaches in Section 5.

4.3. Hybrid Approaches

Hybrid prognostics approaches combine different techniques to determine the RUL of the system under study. To this end, model-based and data-driven prognostic techniques are integrated through (i) the fusion of their respective results or (ii) using as input the results of complementary prognostic tech-
Hybrid approaches combine Data-Driven (DD) and Model-Based (MB) techniques in series and parallel configurations (Penha & Hines, 2002). Series combinations (denoted with the symbol ‘+’) use the outcome of one approach to feed another approach. Possible series combinations include intra-combinations (DD + DD) and inter-combinations (DD + MB, MB + DD) of prognosis approaches. The first approach on the series operation complements the second approach, which performs the prognostics evaluation. Parallel intra- and inter-combinations (denoted with the symbol ‘∥’|’) fuse the outcomes of DD and MB approaches through fusion techniques such as (Goebel & Eklund, 2007): bagging and boosting, fuzzy fusion, or statistics based fusion. As opposed to the series configuration, the parallel operation is interchangeable without influencing the result, i.e., MB ∥ DD = DD ∥ MB.

These configurations determine the goal of the combination of data-driven and model-based approaches: while series applications focus on parameter estimation (e.g., initial parameter estimation or measurement equation estimation); parallel applications are aimed at improving the accuracy of the prognostics application.

When designing hybrid prognostics applications it is possible to create them by (i) combining previously implemented data-driven or model-based approaches with other approaches; or (ii) implementing hybrid approaches upfront. If the results from the already implemented data-driven (or model-based) algorithm (selected according to the flowchart in Figure 2 or 3) are unsatisfactory, it is possible to combine it with other data-driven or model-based approaches. As Figure 4 depicts, this is the first decision point for hybrid prognostics approaches.

If the designer implements a data-driven approach, gets unsatisfactory results, and if they do not have PoF knowledge, it is possible to create data-driven combinations to improve the accuracy of the results. If the designer has datasets with different features or datasets of different scenarios of the same system, then parallel fusion combinations (i.e., DD_1 (x) ∥ DD_2 (y), where x and y indicate different input datasets) may improve the system’s prediction accuracy (e.g., see (J. Liu, Vitelli, Seraoui, & Zio, 2014) for an ensemble of SVR models).

Otherwise, if there is some form of expert knowledge it is possible to integrate it with the DD approach to improve the accuracy of the results, (e.g., see (Soualhi, Razik, Clerc, & Doan, 2014) for a combination of HMM with Adaptive Neuro Fuzzy Inference System (ANFIS)). Note that the expert knowledge considered for prognostics applications is implemented in the form of fuzzy logic and not as rule-based or case-based systems used for diagnostics.

If there is no expert knowledge, it is feasible to use one DD approach as a parameter estimation technique in order to implement another DD approach with more accuracy, i.e., DD_1 (x) + DD_2 (y) (e.g., see (Z. Liu, Li, & Mu, 2012) for a complementary series combination of SVR and HMM).

Finally, if none of the above conditions are satisfactory, it is possible to fuse alternative DD approaches with the same dataset (i.e., DD_1 (x) ∥ DD_2 (x), where x is the available dataset) to improve the accuracy of the RUL estimation (e.g., see (Hu, Youn, Wang, & Yoon, 2012) for an ensemble of multiple algorithms combined with a weighted-sum formulation).

Data-driven and model-based combinations complement each other providing mechanisms to strengthen possible deficiencies. If there exists expert knowledge, it is possible to combine data-driven, model-based, and expert knowledge in a single prognostic approach (e.g., use fuzzy logic to improve data-driven parameter estimation and accordingly, use the Fuzzy + DD configuration to estimate input parameters of a model-based algorithm). Surprisingly, we have not come up with any example that uses this configuration.

If there is no expert knowledge, the typical goal for hybrid prognostics approaches is the parameter estimation through complementary approaches. That is, a data-driven approach estimates input or initial parameters of a model-based approach (DD + MB) and accordingly, improves the accuracy of the final RUL estimation of the PoF model (e.g., see (Baraldi, Compare, Sauco, & Zio, 2013)).

Parallel combinations of MB and DD techniques (MB ∥ DD) focus on improving the accuracy of the RUL estimation through fusion techniques (e.g., see (Baraldi, Mangili, Gola, Nystad, & Zio, 2014) for an ensemble of Kernel Regression models fused with PoF models). To the best of our knowledge, all the fusion configurations between MB and DD approaches are done with different input information due to the dissimilar nature of model-based and data-driven techniques.

As for the configurations comprised of model-based combinations, datasets with different features or scenarios of the same system could be combined to improve the final estimation (i.e., MB_1 (x) ∥ MB_2 (y), where x and y indicate different input datasets). For instance, (Baraldi, Mangili, & Zio, 2012) implement an ensemble of Kalman Filter models. The other possible configuration for model-based intra-combinations is to add expert knowledge to model-based prognostics predictions in order to manage uncertainties (e.g.,
Series and parallel combinations of model-based approaches with the same input dataset are scarce due to the lack of complementary properties between PoF techniques when combining or fusing two different degradation equations of the same system. An example of series combination with the same dataset configuration (i.e., $MB_1(x) + MB_2(x)$) is presented in (Yoon & He, 2015) using UKF to estimate the state of the degrading system, and Particle Filtering to estimate the RUL.

The flowchart for hybrid approaches provides a high-level prognostic algorithm combination guide (cf. Figure 4). This is done deliberately because low-level decisions should be adopted according to technique-specific details.

5. APPLICATION OF THE METHODOLOGY

In this section we will evaluate the design of three prognostics applications in the field of power systems in order to show the applicability of the design decision framework. Namely, the following assets will be examined: cables, transformers, and circuit breakers. Finally, we will assess the applicability of the proposed model selection framework through the analysis of different design requirements.

5.1. Cable Prognostics

There are different parameters which can indicate the fault-to-failure progression of cables such as impedance changes, physical damage, or partial discharge. Particularly, partial discharge accelerates electrical tree growth in the insulation material, and electrical treeing is one of the main causes of electrical breakdown in high voltage cables.

To the best of our knowledge, few attempts have been made to characterize a prognostics model for cables using physics-of-failure equations. In one example, (Dodd, 2003) defined a deterministic model for the growth of electrical tree structures and (Nyanteh, Graber, Edrington, Srivastava, & Cartes, 2011) classified different simulation models for partial discharge and electrical treeing including physics-based and stochastic models.

(Aziz, Catterson, Judd, Rowland, & Bahadoorsingh, 2014) pursued the modeling of the electrical tree growth using a Curve Fitting approach. Analyzing the design requirements for this application, we end up with the following set of decisions according to the data-driven flowchart in Figure 2:

1. According to the design requirements, there is no need to extract the PDF of the RUL estimation. However, confidence bounds are necessary.
2. The dataset is not monotonic: $m = 0.71$.
3. The aim is to predict the RUL at least 1 hour in advance, i.e., long-term prediction.
4. There is no information about the states of the system transiting through the electrical tree growth process before reaching the electrical breakdown.
5. In total there are 25 run-to-failure data histories including multiple variables.
There is no knowledge of reliability models.

Therefore, the data-driven flowchart suggests to implement a Curve Fitting approach as was taken in (Aziz et al., 2014). However, if the designer decides to implement a black-box approach, the only feasible technique would be CPNN for uncertainty management.

If the run-to-failure histories in the dataset were enough to consider case-based reasoning techniques, the data-driven flowchart suggests Match matrix as an appropriate solution for prognosis of long-term multivariate systems.

### 5.2. Transformer Prognostics

(Abu-Elanien & Salama, 2010) presented a taxonomy for transformer physical aging mechanisms divided into two groups: (i) transitive aging reflects the rapid aging of the transformer due to abnormal conditions. Its possible causes are: highly distorted loads with harmonics, high ambient temperature, and overloading. It can be assessed through the measurement of the hot spot temperature. (ii) Intransitive aging assumes that the insulating material can withstand the designed stress. The only possible failure cause is the insulation deterioration. Assessment techniques include: degree of polymerization, dissolved gas analysis, detection of furanic compounds, recovery voltage measurement, and measurement of retaining tensile strength.

Different data-driven prognostics techniques have been presented to estimate the remaining life of transformers, e.g., (Zarei, Shasadeghi, & Ramezani, 2014) implemented an ANFIS model to estimate the end of life of a transformer based on dissolved gas analysis data samples; (Trappey et al., 2014) used linear regression and Weibull distribution to estimate the remaining life of the transformer based on furfural concentration and combustible gases. To the best of our knowledge, only (Catterson, 2014) implemented a model-based prognostics application for transformers using a Particle Filtering approach based on the transformer’s paper aging model. A model for paper aging is given in IEEE standard C57.91 (IEEE Power and Energy Society, 2011). The standard defines an aging acceleration factor based on the hot spot temperature. Accordingly, the implemented method estimates the RUL of the transformer through the degree of polymerization of the paper at its most aged point. According to the model-based flowchart in Figure 3, the design requirements proceed as follows:

1. The degradation equation is available, and the hot spot temperature can be calculated from available observations. Besides, the process is Markovian and therefore, Bayesian tracking solutions are considered.

There are failure precursor variables which indicate the degradation of circuit breakers such as SF6 density, PT, or arc timing. (Rudd et al., 2011) implemented Linear Regression in order to extract a prognostics model based on SF6 density data samples. These are the considered steps according to the data-driven flowchart in Figure 2:

1. There is no need to extract the PDF of the RUL estimation. However, RUL confidence bounds are necessary.

Therefore, we see that the flowchart effectively indicates the same approach as adopted in (Rudd et al., 2011). However, if we assume a more strict limit for the monotonic data assessment, e.g., \( m > 0.9 \), the designer will evaluate different design decisions (cf. Figure 2):

1. There is no need to extract the PDF of the RUL.

Under the non-monotonic assumption, depending on the final design decisions it would be feasible to implement the following prognostics models: (i) Weibull based prediction (no covariance influence); (ii) Curve Fitting; or (iii) CPNN.

### 5.3. Circuit Breaker Prognostics

Circuit breakers do not have a clearly defined physics-of-failure equation model. As pointed out recently in (Westerlund, Hilber, Lindquist, & Krafft, 2014) the ability to predict the aging of circuit breakers is not fully developed. Accordingly, data-driven techniques have been considered for circuit breaker prognostics.

In (Catterson, 2014) a Gaussian distribution was assumed to deal with the lack knowledge of the real behavior. However, as more information is available, the Gaussian assumption is no longer needed. According to the approach adopted in (Catterson, 2014) the model-based flowchart suggests the implementation of a Particle Filtering model.

### 5.4. Analysis of Changing Requirements

To demonstrate the applicability of the model selection framework, in this subsection we will change the design
requirements of the analyzed applications and examine the techniques suggested by the different flowcharts accordingly.

**Uncertainty:** assume that the PDF of the RUL estimation is needed for all the analyzed assets. The transformer application in (Catterson, 2014) already meets the stated requirement. The cable application in (Aziz et al., 2014) and the circuit breaker application in (Rudd et al., 2011) are redesigned according to the data-driven flowchart in Figure 2.

In both cases, the degradation pattern does not follow the Markovian process and system’s state-based specification is unknown. For the cable application there may be enough run-to-failure data to implement an Enhanced TBS. For the circuit breaker model, there is only one run-to-failure history and therefore, RVM or GPR implementations are more appropriate for online or offline implementations respectively.

**Accuracy:** assume that (i) none of the analyzed applications meet the accuracy criteria; and (ii) there is no other engineering resource after the development of the applications. It is possible to combine the implemented approaches with other techniques according to the hybrid flowchart in Figure 4:

- For the transformer application the only feasible suggestion is to improve the parameter estimation (e.g., initial state, process error) of the model-based Particle Filtering technique through a data-driven approach, i.e., \( \text{DD} + \text{MB} \), or using another model-based technique, i.e., \( \text{MB}_1(x) + \text{MB}_2(x) \).
- For the prognostic models developed for circuit breaker and cable assets, series or parallel implementations of different data-driven techniques can be considered in order to improve the accuracy of these applications, i.e., \( \text{DD}_1(x) \parallel \text{DD}_2(x) \) or \( \text{DD}_1(x) + \text{DD}_2(x) \).

6. **Conclusions & Future Work**

In this paper, a methodology to design prognostics applications is presented focusing on the model-selection problem. The main goal of the presented design-decision framework is to construct prognostic models systematically to reduce the effort required to develop a prognostic system and ensure the consideration of all the possible design options. Through different applications in the power industry the applicability of the proposed framework have been demonstrated.

In order to refine the framework and further demonstrate the applicability of the design framework, different prognostics applications will be implemented to reduce the possible subjective (qualitative) criteria. Besides, we plan to complete the methodology by integrating: (i) (automated) fault coverage analysis; (ii) transformation from requirements into prognostic metrics to compare different prognostic algorithms consistently; and (iii) validation and verification steps.

As a long-term goal we plan to develop a decision support tool, which builds semi-automatically prognostics models according to input requirements, engineering resources, and failure modes of interest. The design decision flowchart will benefit from meta-modeling techniques to reuse complex knowledge through automated design decision tools.

**References**


**Biographies**

**Jose Ignacio Aizpuru** is a Research Assistant within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his Eng., M.Sc., and Ph.D. degrees from Mondragon University (Spain) in 2010, 2012, and 2015 respectively. His research interests include prognostics, RAMS analysis and model-based systems engineering.

**Victoria M. Catterson** is a Lecturer within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. She received her B.Eng. (Hons) and Ph.D. degrees from the University of Strathclyde in 2003 and 2007 respectively. Her research interests include condition monitoring, diagnostics, and prognostics for power engineering applications.
System Interdependency Modeling in the Design of Prognostic and Health Management Systems in Smart Manufacturing

M.L. Malinowski¹, P.A. Beling², Y.Y. Haimes³, A. LaViers⁴, J.A. Marvel⁵, and B.A. Weiss⁶

¹²³ University of Virginia, Charlottesville, Virginia, 22904, USA
mlm7yp@virginia.edu
beling@virginia.edu
haimes@virginia.edu
alaviers@virginia.edu

⁵⁶ National Institute of Standards and Technology, Gaithersburg, Maryland, 20899, USA
jeremy.marvel@nist.gov
brian.weiss@nist.gov

ABSTRACT

The fields of risk analysis and prognostics and health management (PHM) have developed in a largely independent fashion. However, both fields share a common core goal. They aspire to manage future adverse consequences associated with prospective dysfunctions of the systems under consideration due to internal or external forces. This paper describes how two prominent risk analysis theories and methodologies – Hierarchical Holographic Modeling (HHM) and Risk Filtering, Ranking, and Management (RFRM) – can be adapted to support the design of PHM systems in the context of smart manufacturing processes. Specifically, the proposed methodologies will be used to identify targets – components, subsystems, or systems – that would most benefit from a PHM system in regards to achieving the following objectives: minimizing cost, minimizing production/maintenance time, maximizing system remaining usable life (RUL), maximizing product quality, and maximizing product output.

HHM is a comprehensive modeling theory and methodology that is grounded on the premise that no system can be modeled effectively from a single perspective. It can also be used as an inductive method for scenario structuring to identify emergent forced changes (EFCs) in a system. EFCs connote trends in external or interdependent sources of risk to a system that may adversely affect specific states of the system. An important aspect of proactive risk management includes bolstering the resilience of the system for specific EFCs by appropriately controlling the states. Risk scenarios for specific EFCs can be the basis for the design of prognostic and diagnostic systems that provide real-time predictions and recognition of scenario changes. The HHM methodology includes visual modeling techniques that can enhance stakeholders’ understanding of shared states, resources, objectives and constraints among the interdependent and interconnected subsystems of smart manufacturing systems. In risk analysis, HHM is often paired with Risk Filtering, Ranking, and Management (RFRM). The RFRM process provides the users, (e.g., technology developers, original equipment manufacturers (OEMs), technology integrators, manufacturers), with the most critical risks to the objectives, which can be used to identify the most critical components and subsystems that would most benefit from a PHM system.

A case study is presented in which HHM and RFRM are adapted for PHM in the context of an active manufacturing facility located in the United States. The methodologies help to identify the critical risks to the manufacturing process, and the major components and subsystems that would most benefit from a developed PHM system.

1. INTRODUCTION

Smart Manufacturing Systems require advanced technologies that facilitate widespread information flow within the system’s components and subsystems. This information can include the health, performance, and risk of the system in failing to meet an objective (Jung, Morris, Lyons, Leong, & Cho, 2015). The engineering focus of Prognostics and Health Management (PHM) is coupled with smart manufacturing. The term “prognostics” refers to the prediction of the future status, health, or performance of components and systems. A commonly used metric within
engineering prognostics is the remaining usable life (RUL) of a machine or system (National Institute of Standards and Technology, 2014). The term “health management” on the other hand refers to the process of making maintenance and logistics decisions from the prognostics information, available resources, and operational demand (Barajas & Srinivasa, 2008). The focus of health management is to minimize operational loss and to maximize the objectives established by the facility (Lee, Wu, Zhao, Ghaffari, Liao, & Siegel, 2014).

The use of PHM models to improve manufacturing performance has been demonstrated in numerous case studies within automotive (Holland, Barajas, Salman, & Zhang, 2010), aerospace (Batzel & Swanson, 2009), machine tool (Biehl, Staufenbiel, Recknagel, Denkena, & Bertram, 2012), and power generation (Hofmeister, Wagoner, & Goodman, 2013) industries. However, as manufacturing processes increase in size and complexity, it can become exceedingly difficult to determine which components or subsystems can most benefit from a PHM system model. Even data-driven approaches, which rely on historical data and mathematical models, lose accuracy and become less predictive as complexity increases (Bai, Wang, & Hu, 2015).

When available resources for PHM efforts are limited, designers and implementers of PHM systems face a difficult problem in deciding where to deploy these scarce resources to maximize benefit. A smart manufacturing system may involve multiple subsystems or processes that present reasonable targets for the development of PHM systems (Barajas & Srinivasa, 2008). This selection problem is made more difficult because the potential costs and benefits of those potential PHM systems are subject to random and known uncertainty (Feldman, Jazouli, & Sandborn, 2009) (Hou-bo, & Jian-min, 2011).

Numerous systems-based risk analysis methodologies designed to support decision-makers within manufacturing industries have successfully been developed and deployed (Lee, Lv, & Hong, 2013) (Fernández, & Pérez, 2015), including Hierarchical Holographic Modeling (HHM) (Haines, 2009) and Risk Filtering, Ranking, and Management (RFRM) (Haines, Kaplan, & Lambert, 2002) (Haines, 2009). The original purpose of these methods (within the field of risk analysis) was to identify the most critical sources of risks to a system and to provide risk assessment, risk management, and risk communication (Haines, 2012). With a few modifications, the critical risks identified in the HHM and RFRM processes can be used to identify the most critical components and subsystems that would most benefit from a PHM system or model.

The contribution of this paper is to introduce HHM and RFRM as methodologies to provide scope and direction for the PHM system designer. The proposed methodologies will be used to identify targets – components, subsystems, or systems – that would most benefit from a PHM system in regards to achieving the following objectives: minimizing cost, minimizing production/maintenance time, maximizing system remaining usable life (RUL), maximizing product quality, and maximizing product output. There currently exist multiple methods to determine the major failure modes of a system after an accident or catastrophe (Cocheteux, Voisin, Levrat, & Iung, 2009) (Lee et al., 2014) (Vykdyal, Plura, Halfarová, & Klaput, 2015). The proposed methodology allows for a thorough analysis to be conducted even before a failure occurs in a manufacturing environment.

The remainder of the paper is organized as follows. Section 2 summarizes and explores the general HHM methodology. Section 3 explains the additional benefit of applying RFRM to the models developed using HHM. Section 4 discusses PHM-specific modifications to the RFRM method. Section 5 provides a specific case study of the application of HHM and RFRM to a major manufacturing facility. Section 6 concludes the paper.

2. Hierarchical Holographic Modeling, Risk Analysis, and PHM

Risk is a combined measure of the probability and severity of adverse effects (Andretta, 2014), which necessitates knowledge and understanding of future probable adverse events and their likely consequences (Haines, 2009). To answer the basic question in risk analysis: “what can go wrong?” it is imperative that all conceivable and likely risk scenarios be identified. This is a daunting task, but can be accomplished by integrating knowledge and experience from multiple experts across different disciplines. The HHM methodology facilitates this collaboration between experts.

HHM has been successfully utilized in numerous projects and for multiple agencies, including the President's Commission on Critical Infrastructure Protection (PCCIP), the Federal Bureau of Investigation (FBI), the National Aeronautics and Space Administration (NASA), the Virginia Department of Transportation (VDOT), and the U.S. Army National Ground Intelligence Center (NGIC) (Haines, 2009) (Lambert, Haines, Li, Schooff, & Tulsiani, 2001). The PCCIP utilized HHM to determine the major hardware, software, human, and environmental risks to a supervisory control and data acquisition system (Chittester, & Haines, 2004). The FBI developed an HHM model to identify varying perspectives, motives, and weaknesses between homeland defenders and terrorist networks (Haines, & Horowitz, 2004). For VDOT, the HHM method identified major interdependencies within Virginia’s transportation infrastructure and outlined critical sectors that were most sensitive to disruptions (Crowther, Dicdican, Leung, Lian, & Williams, 2004). Finally for the Army NGIC, HHM was used prior to a major deployment to identify the critical state variables of the target host country.
U.S. forces, and U.S. allies (Dombroski, Haimes, Lambert, Schlussel, & Sulcoski, 2002).

Haimes (2009) defines HHM as a holistic philosophy and methodology aimed at capturing and representing the essence of the inherent diverse characteristics and attributes of a system. These system attributes include, but are not limited to, the multiple aspects, perspectives, facets, views, dimensions, and hierarchies. The mathematical and systems approach to holographic modeling reveals the interconnectedness, and the interdependencies among the system’s objective functions, constraints, decision variables, and inputs/outputs (Haimes, 2009). The term holographic refers to the desire to have a multi-view image of a system (Crowther et al., 2004). For example, the risk to a system due to emergent forced changes (EFCs) can be represented from its multiple perspectives, which are related to time and geography, and include, but are not limited to: (1) economic, (2) health, (3) technical, (4) political, and (5) social perspectives. To capture a holographic outcome, the modeling team that performs the analysis must represent a broad array of experience and knowledge (Haimes, 2009).

The HHM process considers risks at both the macroscopic (management) and microscopic (component) levels. Most organizational and technology-based manufacturing systems are hierarchical in nature (Alvandi, Bienert, Li, & Kara, 2015) (He, Zhang, & Li, 2014), and the deployments of HHM have effectively addressed the risks at these multiple levels (Haines, Kaplan, & Lambert, 2002). HHM is especially useful in determining the reliability and maintainability of infrastructures that feature a large number of components and subsystems. From a mathematical standpoint, reliability refers to the probability that a system is operational in a given time period, while maintainability is defined as the probability that a failed system can be restored to an operational state within a specified period of time (Haimes, 2009). Both of these metrics are essential to holistic risk assessment and management.

The HHM methodology produces a multilevel decomposition of a system into its many subsystems and components. This breakdown is essential to revealing the complexity and internal hierarchical nature of large-scale systems (He et al., 2014). Decomposition also allows for trade-off analyses and studies to be performed at the component, subsystem, or total system level. Applying the HHM methodology requires an organized team of experts with varied experience and knowledge bases to develop a holographic view of a system with its multiple levels and hierarchies. Although it is possible for individual experts to create different decompositions, the aggregate will yield the same optimal solution. Each expert will provide their own perspective to enforce the desired multi-view image of the system and reveal unique vulnerabilities (Kaplan, Haimes, & Garrick, 2001). Two major types of risks and uncertainties will ultimately come to light: those resulting from 1) exogenous events such as new legislation or natural disasters, and 2) endogenous events such as hardware, software, organizational, and human failures (Haimes, 2009). While knowledge of both types of events is crucial to understanding the entire system, a PHM system will focus more heavily on potential endogenous events which can take the form of critical EFCs.

At their cores, both PHM and risk analysis share two common goals: (1) to ensure that the systems under consideration perform their intended functions and meet their objectives at acceptable tradeoffs and within an acceptable time frame, and (2) to inform decision-makers so they can better predict and respond to faults and failures (Haimes, 2009) (NIST, 2015). Additionally, both practices utilize systemic risk modeling, assessment, management, and communication to achieve their goals (Ahmad & Kamaruddin, 2012) (Al-Habaibeh & Gindy, 2000). Due to these commonalities, the risk analysis theory and methodology of HHM was utilized in a case study to determine the conceivable sources of risk to a system, and finally to help decide where to apply a PHM model within a smart manufacturing facility.

3. Risk Filtering, Ranking, and Management

In total risk management, it is necessary to identify, prioritize, assess, and manage potential risk scenarios to a large-scale system. Stakeholders and decision-makers must consider the likelihoods and consequences of each risk to produce acceptable mitigation options. The Risk Filtering, Ranking, and Management (RFRM) methodology offers eight major phases to guide total risk management in an HHM system (Haimes, 2002). The eight phases are: 1 – Scenario Identification, 2 – Scenario Filtering, 3 – Bicriteria filtering, 4 – Multi-criteria Evaluation, 5 – Quantitative Ranking, 6 – Risk Management, 7 – Safeguarding Against Missing Critical Items, and 8 – Operational Feedback. Details on these eight phases can be found in (Haimes, 2002).

The guiding force behind RFRM is the identification of head topics, which represent major concepts or perspectives of success, and subtopics, which provide detailed requirements or sources of risk (Haimes, 2009). However, it is often impractical to evaluate hundreds of sources of risk when evaluating a large system. Therefore, the risk scenarios and sources should be filtered based on professional experience, expert knowledge, and statistical data. It is also important to consider a variety of risks such as those related to hardware, software, organizational failure, human error, budget, schedule slip, and performance criteria (Haimes, 2002).

The RFRM methodology has been successfully deployed on numerous systems for multiple agencies, including the NASA, the Federal Aviation Administration (FAA), the VDOT, the National Ground Intelligence Center (NGIC),...
and the Department of Homeland Security (DHS) (Haimes, 2009). NASA used RFRM to identify the most common risk scenarios facing future space missions (e.g., inadequate oversight teams), and to compare management strategies to mitigate those risks (e.g., restructure existing teams or hire external consultants) (Haimes, 2009). For VDOT, the RFRM method ranked and prioritized the potential shutdowns of various transportation infrastructure assets (e.g., roads, highways, or bridges) according to their impacts on state transportation inoperability and economic loss (Crowther et al., 2004). Finally, the Army NGIC used the RFRM method to identify the risk scenarios that allied forces might encounter in a foreign country that occurred with the highest likelihood probability and produced the most severe results (e.g., loss of life or major asset) (Dombroski et al., 2002).

The risk assessment portion of RFRM can be summed up by four major questions (Haimes, 2002):

1) What can go wrong?
2) What is the likelihood of that happening?
3) What are the consequences?
4) What is the time frame?

The risk management portion on the other hand encompasses three complementary questions (Haimes, 2009):

1) What can be done and what are the available options?
2) What are the associated trade-offs in terms of costs, benefits, and risks?
3) What are the impacts of current decisions on future options?

After all relevant and potential risks have been identified as either head topics or subtopics they must be evaluated by three major criteria: resilience, robustness, and redundancy. **Resilience** refers to the ability of a system to recover after an emergency, and can be evaluated by time and resources needed. **Robustness** is the insensitivity of system performance to external stresses, so the ability to resist potential risks. **Redundancy** refers to the ability of extra components or subsystems to take over the functions of failed components or subsystems (Haimes, 2009).

The three categories of resilience, robustness, and redundancy are then further broken down into eleven essential criteria for evaluating risk scenarios (refer to Figure 1).

Figure 1. Risk factors with eleven criteria.

The eleven criteria relating the ability of a risk scenario to defeat the defenses of a system are formally defined as follows (Haimes, 2009):

1. **Undetectability** – the absence of modes by which the initial events of a scenario can be discovered before harm occurs
2. **Uncontrollability** – the absence of control modes that make it possible to take action or make an adjustment to prevent harm
3. **Multiple paths to failure** – multiple and possibly unknown ways for the events of a scenario to harm the system
4. **Irreversibility** – a scenario in which the adverse condition cannot be returned to the initial, operational (pre-event) condition
5. **Duration of effects** – a scenario that would have a long duration of adverse consequences
6. **Cascading effects** – a scenario where the effects of an adverse condition propagate to other systems or subsystems (cannot be contained)
7. **Operating environment** – a scenario that results from external stressors
8. **Wear and tear** – a scenario that results from use, leading to degraded performance
9. **Hardware, software, human, and organizational interfaces** – a scenario in which the adverse outcome is magnified by interfaces among one or more these subsystems
10. **Complexity/emergent behaviors** – a scenario in which there is a potential for system-level behaviors that are not anticipated even with knowledge of components and their interactions
11. **Design immaturity** – a scenario in which the adverse consequences are related to the newness of the system design or other lack of a proven concept

Each identified risk scenario must be rated as “high”, “medium”, “low”, or “not applicable” against each criterion.
Scenarios with more “high” ratings must be considered further in the RFRM process. Risk scenarios that score mostly “low” or “not applicable” in the eleven categories can be filtered out unless an emergent change drives it towards a higher level of risk. Alternative rating scales and filtering criteria could also be used with the same goal: reduction of the number of scenarios under consideration.

4. PHM-Specific Modifications to Risk Filtering, Ranking, and Management

The RFRM process is essential because it limits the number of risk scenarios for a manufacturing facility to a manageable quantity. However, the process must be modified to identify the risks that are applicable to realistic and practical PHM strategies. Risks that cannot be handled through PHM should still be considered at a higher system level, but will not be useful to the process described in this paper. The modifications to the standard RFRM filtering process are as follows:

M1. Risks that are rated “high” for undetectability should be filtered out during RFRM, unless there exists the potential to add a detection method (such as a sensor to a robot).

M2. Risks that are rated “high” for uncontrollability should be filtered out during RFRM, unless there exists potential to insert control modes to the process or subsystem.

M3. Risks that are directly related to only the operating environment and thus cannot be mitigated on a day-to-day basis should be filtered out during the RFRM.

M4. Risks that can be directly classified as either “human” or “organizational” should be filtered out during RFRM.

M5. Risks that are only rated “high” in the category of design immaturity should be filtered out during RFRM.

The purpose of the M1 modification is to ensure that only risks that can be detected, identified, and diagnosed will remain after the filtering process. This is because PHM systems rely on prognostics, and thus require predictive capabilities of future health, performance, or RUL of subsystems. They must have a means to detect or sense in order to provide effective health management. However, it should be noted that if it is possible to add a detection method or even a reliability model to the risk in question, then it should not be filtered out on the basis of the M1 modification.

The M2 modification seeks to eliminate risks that have no existing control channels. The purpose of a PHM system is to modify decision variables or inputs to a system in order to create a desired outcome. However, even if the optimal modifications to the variables can be identified, if there is no way to implement them, then there is no benefit to the system. It was additionally noted that if it is possible to add control modes, then this filtering criterion can be ignored.

The purpose of the M3 modification is to filter out risks that are only related to the operating environment. Specifically, these are the risks pertaining to external factors over which there is no control, such as the weather, plant location, and even legislation or industry standards. These risks should be filtered because they cannot be managed on a day-to-day basis and would require solutions outside the scope of a manufacturing PHM system. It should be noted that this should only serve as a filter if it is the only “high” rated risk category.

Modification M4 removes any risks that are primarily classified as either “human” or “organizational.” The purpose here is to eliminate risks that are primarily related to issues that are difficult to control, such as human error or the organizational structure of a corporation. While managing these risks may prove extremely beneficial to a manufacturing facility, there is little opportunity for a PHM system.

Finally, the M5 modification removes risks that are focused on immature or experimental subsystems, which are usually still undergoing optimization or usability testing. These new systems will naturally inherit additional risk since they have not yet been verified. Therefore, we would not want to allocate resources towards developing a PHM system for a new component until it has become stable within its own design cycle.

5. CASE STUDY IN THE APPLICATION OF HHM AND RFRM TO PHM IN SMART MANUFACTURING

The process for identifying the most important sources of risk involves developing a Hierarchical Holographic Model and performing a PHM-oriented Risk Filtering, Ranking, and Management. As a proof of concept for this methodology, consider the following example featuring the packaging process at a major manufacturer located in the United States. Due to the competitive nature of the industry, specific details about the company have been omitted. For the remainder of this paper, the manufacturing facility shall be referred to as Plant A.

5.1. Plant A Packing and Bagging Overview

One of the major processes at Plant A encompasses the packing, transporting, and bagging of their finished product. Refer to Figure 2 for a detailed system diagram of the entire process with the major components, subsystems, sensors, machines, robots, and humans identified.

Once the product has been processed and fully prepared, it is stored on the floor in a sterilized section of the plant. A small end-loader pushes controlled heaps of the product into the process for identifying the most important sources of risk involves developing a Hierarchical Holographic Model and performing a PHM-oriented Risk Filtering, Ranking, and Management. As a proof of concept for this methodology, consider the following example featuring the packaging process at a major manufacturer located in the United States. Due to the competitive nature of the industry, specific details about the company have been omitted. For the remainder of this paper, the manufacturing facility shall be referred to as Plant A.

5.1. Plant A Packing and Bagging Overview

One of the major processes at Plant A encompasses the packing, transporting, and bagging of their finished product. Refer to Figure 2 for a detailed system diagram of the entire process with the major components, subsystems, sensors, machines, robots, and humans identified.

Once the product has been processed and fully prepared, it is stored on the floor in a sterilized section of the plant. A small end-loader pushes controlled heaps of the product into
a grate in the floor that is outfitted with an automated screw conveyor. This screw moves the product up to a storage tank overhead, which then funnels the product to one of a few bagging stations: two 15.88 kg – 22.68 kg bag stations and one jumbo station for bulk product. After the product enters the funnels, an automated machine fills bags to their correct, preset weight. Bags are administered by human workers, one at each station. The human operators take empty bags, load them onto the filler, and then start the filling process. Finally they remove the full bags and shift the bags over to a conveyor where they are sealed, flattened, and sent down the line.

At this point the bags are in queue for a robotic palletizer. The palletizer receives sealed and inspected bags of product and stacks them onto wooden pallets in regular, repeating patterns that can be selected and adjusted by the operator. A forklift is used to remove the finished pallet where it is wrapped in shrink wrap and placed in a holding area for distribution. A central programmable logic controller with a touch screen interface coordinates the overall unit automation that was supplemented by at least six human workers: one end-loader driver, two baggers, one inspector, and two to shrink wrap finished pallets and insert empty pallets to the palletizer cage. The insertion of empty pallets into the robot workspace is accomplished by a light curtain that would turn off when the pallet was completely loaded (and the robot switched to an empty pallet on its other side) so that the loaded pallet could be removed (via forklift) and a new wooden pallet re-inserted (by a human operator who would return the light curtain to active to let the robot know it could switch back to that side when it finished the pallet on its other side).

Figure 2. System diagram of Plant A.
Plant engineers have noted the following known health management issues:

- The funnel openings can become clogged with finished product if not regularly cleaned out.
- Sensors fail with regularity. Common causes of failure include occlusion of optical components by dirt and misalignment through collision with bags of product.
- The maneuvering of heavy bags by human workers is a potential source of slower productivity for the facility.
- Adjusting and reprogramming the palletizer is difficult and generally outside the scope of the work done in house. The robot engineer must be on call and able to reprogram the machine in-person.

5.2. Application of HHM and RFRM

The main objectives of the manufacturer are to maximize production of their packaged product, and to minimize the risk of a system failure (production shutdown or delay). To help achieve these objectives, Plant A wishes to implement a PHM system into their packing and bagging process. However, they currently have limited monetary resources allocated towards this effort. Thus, Plant A requires a full analysis regarding which of their components/machines/subsystems would most benefit from a PHM system. This necessitates a complete understanding of their current industrial process.

5.2.1. HHM for Plant A

First, multiple Hierarchical Holographic Models (HHMs) are developed covering multiple aspects of the manufacturing plant. The HHM models receive input from many different subject matter experts, stakeholders, and decision makers. For Plant A, an HHM model was originally developed with the perspective of the different physical components within the finished product bagging system. The head topics for the model were (1) Machines and Robots, (2) Components, (3) Humans, and (4) Environment. Underneath these major topics, subtopics and possible risk scenarios can be identified. The HHM model for the physical components has been displayed in bullet form below.

1. Machines and Robots
   a. Front End Loader
   b. Screw Conveyor
      i. Horizontal
      ii. Vertical
   c. Storage Tank Dispenser
   d. Bagging Machine
   e. Bag Sealer
      i. Heat sealer
      ii. Conveyor belt
   f. Automated Conveyor
      i. Sensor
      ii. Belt
   g. Bag Flattener
   h. Palletizer
      i. Sensor
      ii. Arm
      iii. Claw
      iv. Controls
     i. Forklift
   j. Pallet Packager

2. Components
   a. Finished Product
   b. Bags
   c. Pallets
   d. Packaging material

3. Humans
   a. Front end loader driver
   b. Baggers
   c. Inspectors
   d. Forklift driver
   e. Packager

4. Environment
   a. Factory Floor
   b. Storage Tank
   c. Air
   d. Moisture
   e. Contaminants

A similar HHM model was also developed from multiple experts covering a new perspective: the different processes within the finished product bagging system. The practice of creating multiple HHM models helps to provide a holographic view of the entire system and ensure that the major sources of risk are properly captured. It provides a more realistic and complete overall model by recognizing the limitations of modeling a complex system with just a single structure. The head topics for the processes model were (1) Storing Product, (2) Transporting Product, (3) Bagging Product, (4) Sealing Bags, (5) Transporting Bags,
(6) Flattening Bags, (7) Stacking Bags on a Pallet, and (8) Preparing Final Product for Delivery. The complete HHM model can be seen in bullet form below.

1. Storing Product
   a. Environment
      i. Factory floor
      ii. Air
      iii. Moisture
   b. Human interactions
   c. Factory contaminant controls

2. Transporting Product
   a. Front end loader
      i. Scoop product
      ii. Push product into floor grates
   b. Screw conveyors
      i. Move product to vertical conveyor
      ii. Move product to storage tank

3. Bagging Product
   a. Human operator
      i. Obtain empty bag
      ii. Fill bag
   b. Bagging machine
      i. Grip bag
      ii. Lock bag
      iii. Sense weight
      iv. Unlock bag
   c. Storage tank
      i. Open hatch to drop product
      ii. Close hatch to secure product

4. Sealing Bags
   a. Human operator
      i. Place bag
   b. Bag sealer
      i. Sense bag
      ii. Grip bag
      iii. Heat seal bag
      iv. Transport bag
      v. Lay bag flat

5. Transporting Bags
   a. Human supervisor
      i. Controls
      ii. Fix unaligned bags
   b. Automated conveyor
      i. Sense bags
      ii. Move bags
      iii. Delay bags

6. Flattening Bags
   a. Human supervisor
      i. Controls
   b. Bag flattener
      i. Sense bag
      ii. Flatten bag
      iii. Move bag

7. Stacking Bags on a Pallet
   a. Forklift
      i. Move empty pallet to palletizer
   b. Human supervisor
      i. Adjust settings for palletizer
      ii. Start/stop process
      iii. Fix fallen bags
   c. Palletizer robot
      i. Sense bag
      ii. Grip bag
      iii. Lift bag
      iv. Position bag
      v. Drop/place bag on pallet

8. Preparing Final Product for Delivery
   a. Forklift
      i. Lift pallet with stacked bags
      ii. Transport to packager
   b. Pallet packager
      i. Rotate pallet
      ii. Dispense shrink wrap
   c. Human operator
      i. Operate machinery
      ii. Transport completed pallet to storage area

Multiple HHM perspectives can be explored to further improve the overall system model, such as organizational, technological, or even social. For this particular case study, the processes perspective was used to develop risk scenarios for the finished product packing and bagging system.

5.2.2. RFRM for Plant A

Next the Risk Filtering, Ranking, & Management (RFRM) method was applied to the HHM model containing the processes within the product packing and bagging system. Each head topic was re-defined as a risk scenario, where the process in question failed to occur. Head topics 2 through 8 were identified as being the most critical to the success of the manufacturing system. The subtopics directly related to
the operating environment or human interactions were then filtered out, as per the PHM-specific RFRM modifications. The remaining risk scenarios of interest are identified in Table 1 below.

Table 1. Risk scenarios of interest for RFRM

<table>
<thead>
<tr>
<th>Risk ID</th>
<th>Risk Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.b</td>
<td>Screw conveyor failure</td>
</tr>
<tr>
<td>3.b</td>
<td>Bagging machine failure</td>
</tr>
<tr>
<td>3.c</td>
<td>Storage tank failure</td>
</tr>
<tr>
<td>4.b</td>
<td>Bag sealer failure</td>
</tr>
<tr>
<td>5.b</td>
<td>Automated conveyor failure</td>
</tr>
<tr>
<td>6.b</td>
<td>Bag flattener failure</td>
</tr>
<tr>
<td>7.c</td>
<td>Palletizer robot failure</td>
</tr>
<tr>
<td>8.b</td>
<td>Pallet packager failure</td>
</tr>
</tbody>
</table>

Next a qualitative severity-scale matrix was applied to the remaining subtopics to filter out the topics that did not meet a predetermined risk threshold. A combination of expert insight from the manufacturers and historic data provided both the evidence for the evaluation and the severity of the impact levels. The results of the matrix are displayed in Table 2 below. The five likelihood/probability columns refer to the probability that an event would normally occur. For example, events in the first column occur with a probability of less than 1%, while events in the second column occur with a probability between 1% and 5%. The descriptions for the matrix scales are displayed in Table 3 and Table 4 below.

Table 2. Severity-scale matrix for identified risk scenarios.

<table>
<thead>
<tr>
<th>Impact</th>
<th>Likelihood/Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pr&lt;0.01</td>
</tr>
<tr>
<td>4</td>
<td>7.c</td>
</tr>
<tr>
<td>3</td>
<td>3.c</td>
</tr>
<tr>
<td>2</td>
<td>5.b, 8.b</td>
</tr>
<tr>
<td>1</td>
<td>4.b</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. Risk description for severity matrix

<table>
<thead>
<tr>
<th>Low Risk</th>
<th>Moderate Risk</th>
<th>High Risk</th>
<th>Extremely High Risk</th>
</tr>
</thead>
</table>

According to the RFRM methodology, the topics classified as either “High Risk” or “Extremely High Risk” must be further evaluated, while the other scenarios can be filtered out. In this case, the remaining risk scenarios were:

- 3.b – Bagging machine failure
- 6.b – Bag flattener failure
- 7.c – Palletizer robot failure

These scenarios must be analyzed for their ability to defeat the major defensive properties of a system: redundancy, resilience, and robustness. This can be determined by rating their performance along the eleven criteria RFRM attributes of risk scenarios, displayed in Table 5.

Table 5. Eleven RFRM attributes of risk scenarios.

<table>
<thead>
<tr>
<th>#</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Undetectability</td>
</tr>
<tr>
<td>2</td>
<td>Uncontrollability</td>
</tr>
<tr>
<td>3</td>
<td>Multiple paths to failure</td>
</tr>
<tr>
<td>4</td>
<td>Irreversibility</td>
</tr>
<tr>
<td>5</td>
<td>Duration of effects</td>
</tr>
<tr>
<td>6</td>
<td>Cascading effects</td>
</tr>
<tr>
<td>7</td>
<td>Operating environment</td>
</tr>
<tr>
<td>8</td>
<td>Wear and tear</td>
</tr>
<tr>
<td>9</td>
<td>Hardware/software/human/organizational</td>
</tr>
<tr>
<td>10</td>
<td>Complexity and emergent behaviors</td>
</tr>
<tr>
<td>11</td>
<td>Design immaturity</td>
</tr>
</tbody>
</table>

Each category receives a qualitative assessment regarding whether the risk scenario has a low, medium, or high susceptibility to the given criterion. The evaluation for the three remaining risk scenarios within the Plant A example can be seen below (refer to Table 6).
Finally, the PHM-specific RFRM modifications must be checked against the three identified risk scenarios. It can be seen that the palletizer robot failure (7.c) rated high for undetectability (1), so according to the PHM modifications it should be removed. However in this case we opt to keep this risk scenario since there are sensors available which can be added to the palletizer robot as detection methods. Additionally the bagging machine failure (3.b) received a high rating for hardware/software/human/organizational (9), but only because it was classified as a strictly “human” process. For this reason this risk scenario can be filtered out before further analysis.

5.2.3. Results and Findings from HHM and RFRM

After eliminating risks using the PHM-specific RFRM rules, the palletizer robot received the highest risk assessment both in the qualitative severity-scale matrix (refer to Table 2) and within the eleven attributes of risk (refer to Table 6). Therefore, the HHM and RFRM methodologies have successfully identified an essential location within the Plant A bagging and packaging process. We are confident that the application of a PHM effort at the palletizer robot will provide the biggest impact towards achieving the main objectives: maximizing production and minimizing the risk of a system failure (production shutdown or delay).

Given limited resources, it is recommended that the product manufacturer begin by implementing a PHM strategy at the palletizer robot, and then if available resources remain, proceed with the other top identified sources of risk. The components of the palletizer (arms, claws, sensors, controls, etc.) can even be evaluated for their individual levels of risk to determine which ones are most critical to the palletizer subsystem. Then a variety of PHM methodologies can be implemented for the palletizer to develop an optimal risk management solution for the entire smart manufacturing system-of-systems. This analysis may be crucial in the development of low-level process management by creating awareness of the interconnected system of systems that manufacturing plants rely on to operate efficiently, safely, and in a timely manner. This holistic understanding should trickle down to inform the structure and communications of future robotic control architecture.

6. CONCLUSION

As smart manufacturing facilities increase in size and complexity, it becomes exceedingly challenging to apply Prognostics and Health Management (PHM) models and strategies to the entire system without recognizing and addressing this emergent complexity as a system of systems. This paper has described a systems-based risk-analysis methodology capable of identifying all conceivable sources of risk to smart manufacturing process in support of PHM.

The well-developed practice of risk analysis provides two powerful tools for this methodology: HHM and RFRM. The original purpose of these methods within the risk analysis field was to identify the most critical risks to a system and to provide risk assessment, risk management, and risk communication. However as demonstrated in this paper, with a few modifications the critical risks identified in the HHM and RFRM processes can provide scope and direction for the PHM system designer. Specifically, HHM and RFRM can be utilized to identify the major components, subsystems, or systems that would most benefit from a PHM system while prioritizing the following manufacturing objectives: minimizing cost, minimizing production and maintenance time, maximizing system remaining usable life (RUL), maximizing product quality, and maximizing product output.

NIST DISCLAIMER

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

ACKNOWLEDGEMENT

This paper documents results from a collaborative effort between the University of Virginia’s Systems Engineering Department, and the National Institute of Standards and Technology (NIST).

REFERENCES

Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial

<table>
<thead>
<tr>
<th>Criteria #</th>
<th>3.b Bagging</th>
<th>6.b Flatten</th>
<th>7.c Palletize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>3</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>6</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>7</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>8</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>9</td>
<td>High (human)</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>10</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>11</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>


**Biographies**

Michael L. Malinowski earned a B.S. in Aerospace Engineering from the University of Virginia (UVA), Charlottesville, VA, USA, 2012. He is currently pursuing an M.S. degree in Systems Engineering from UVA. He worked as a Systems and Test Engineer at Lockheed Martin Corporation between 2012 and 2014, and graduated from their Engineering Leadership Development Program within the Mission Systems and Training business division. He also worked as an Engineer at Science Applications International Corporation between 2009 and 2012. He is a member of AIAA, Tau Beta Pi, and Sigma Gamma Tau. His research interests are risk assessment and management of engineering systems.

Peter A. Beling received his Ph.D. in Operations Research from the University of California, Berkeley, CA, USA. He is an associate professor in the Department of Systems and Information Engineering at the University of Virginia (UVA). He is active in the UVA site of the Broadband Wireless Applications Center, which is an Industry-University Cooperative Research Center sponsored by the National Science Foundation. His research interests are decision making in complex systems, with an emphasis on adaptive decision support systems and on model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including prognostics and health management, mission-focused cyber security, and financial decision-making.

Yacov Y. Haimes received his Ph.D. in Large-Scale Systems Engineering from the University of California, Los Angeles, CA, USA, 1970. He is the Founding Director (1987) of the Center for Risk Management of Engineering Systems at the University of Virginia. On the faculty of Case Western Reserve University, Cleveland, OH, for 17 years he was the Chair of the Systems Engineering Department, and Director of the Center for Large-Scale Systems and Policy Analysis. Between 1977 and 1978, he was an AAAS/American Geophysical Union Congressional Science Fellow, joining the staff of the Executive Office of President Jimmy Carter, and later the staff of the House Science and Technology Committee. He has published more than 250 articles and technical papers, edited or co-edited 21 volumes, and authored or co-authored 6 books.

Amy LaViers earned a B.S.E. in Mechanical and Aerospace Engineering from Princeton University, Princeton, NJ, USA. She completed an M.S. and Ph.D. in Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, GA, USA. She is an Assistant Professor in Systems and Information Engineering and Director of the Robotics, Automation, and Dance Lab at the University of Virginia. She aims to extract useful features from human movement for robotic applications, such as endowing co-robots the ability to work alongside human workers in manufacturing plants.

Jeremy A. Marvel received his Ph.D. in Computer Engineering from Case Western Reserve University, Cleveland, OH, USA, 2010. He is a project leader and research scientist in the Intelligent Systems division of the National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Since joining the research staff at NIST, he has established the Collaborative Robotics Laboratory, which is engaged in research dedicated to developing test methods and metrics for the performance and safety assessments of collaborative robotic technologies. His research focuses on intelligent and adaptive solutions for robot applications, with particular attention paid to human-robot collaborations, multi-robot coordination, safety, perception, self-guided learning, and automated parameter optimization. He is currently engaged in developing measurement science methods and artifacts for the integration and application of robots in collaborative assembly tasks for manufacturing.

Brian A. Weiss earned a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the Smart Manufacturing Operations Planning and Control program and the Project Leader of the Prognostics and Health Management for Smart Manufacturing Systems project within the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). He spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles, tactical applications operating on Android devices, advanced soldier
sensor technologies, and urban search and rescue robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 & 2008), and the Jacob Rabinow Applied Research Award (2006).
Device Health Estimation by Combining Contextual Control Information with Sensor Data and Device Health Prognostics Utilizing Restricted Boltzmann Machine

Linxia Liao¹, Tomonori Honda², and Radu Pavel³

¹,² Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA, 94304, USA
linxia.liao@parc.com
tomo.honda@parc.com

³ TechSolve, Inc., 6705 Steger Dr, Cincinnati, OH, 45237, USA
pavel@techsolve.org

ABSTRACT

The goal of this work is to bridge the gap between business decision-making and real-time factory data. Beyond real-time data collection, we aim to provide analysis capability to obtain insights from the data and converting the learnings into actionable recommendations.

For device health estimation, we focus on analyzing device health conditions and propose a data fusion method that combines sensor data with limited diagnostic signals with the device’s operating context. We propose a segmentation algorithm that provides a temporal representation of the device’s operation context, which is combined with sensor data to facilitate device health estimation. Sensor data is decomposed into features by time-domain and frequency-domain analysis. Principal component analysis is used to project the high-dimensional feature space into a low-dimensional space followed by a linear discriminant analysis to search the optimal separation among different device health conditions. Our industrial experimental results show that by combining device operating context with sensor data, our proposed segmentation and linear transformation approach can accurately identify various device imbalance conditions even for limited sensor data which could not be used to diagnose imbalance on its own.

For device health prediction, we propose a restricted Boltzmann machine based method to automatically generate features that can be used for remaining useful life prediction, which is performed by a random forest regression algorithm. The proposed method was validated through run-to-failure dataset of a machine tool spindle test-bed.

1. INTRODUCTION

The growing Internet of Things is predicted to connect 30 billion devices by 2020 (MacGillivray, Turner, & Lund, 2013). This will bring in tremendous amounts of data and drive the innovations needed to realize the vision of Industry 4.0—Cyber-Physical systems monitoring physical processes, and communicating and cooperating with each other and with humans in real time. One of the key challenges to be addressed is how to analyze large amounts of data to provide useful and actionable information for businesses intelligence and decision making. In particular, to prevent unexpected downtime and its significant impact on overall equipment effectiveness (OEE) and total cost of ownership (TCO) in many industries. Continuous monitoring of equipment, early detection of incipient faults, and prediction of failure before it happens can support optimal maintenance strategies, prevent downtime, increase productivity, and reduce costs.

A significant number of anomaly detection and diagnosis methods have been proposed for device fault detection and health condition estimation. Chandola et al. (Chandola, Banerjee, & Kumar, 2009) discusses various categories of anomaly detection technologies and their assumptions as well as their computational complexity. Several approaches such as statistical methods (Markou & Singh, 2003), neural network methods (Markos & Singh, 2003) and reliability methods (Guo, Watson, Tavner, & Xiang, 2009), have been applied to detect anomalies for various types of equipment. The philosophies and techniques of monitoring and predicting machine health with the goal of improving reliability and reducing unscheduled downtime of rotary machines are presented by Lee et al. (Lee et al., 2014).

Many of these methods focus on analyzing, combining, and modeling sensor data (e.g. vibration, current, acoustics sig-
nal) to detect machine faults. One issue that remains mostly unaddressed in these methods is that they rarely consider the varying operating context of the machine. In many cases, false alarms are generated due to a change in machine operation (e.g., rotational speed) rather than a change in machine condition. A major challenge in addressing this issue is that most machine controllers are built with proprietary communication protocols, which leads to a barrier in obtaining control parameters to understand the context under which the machine is operating. Recently, the MTConnect open protocol (Standard, 2009) was developed to connect various legacy machines independent of the controller providers. MTConnect provides an unprecedented opportunity to monitor machine operating context in real-time. In this paper, we leverage MTConnect to diagnose machine health condition by combining sensor data with operating context information. Additionally, we investigate whether it is possible to diagnose machine health condition using less sensor data when it is combined with context information.

Many methods have been proposed in the literature for device remaining useful life (RUL) prediction. These methods can be generally classified as data-driven method, physics-based method, and hybrid method (Liao & Kottig, 2014). Since detailed information of the assembled components is not available in our case, physics-based modeling is unfeasible. Hence, data-driven method becomes the primary approach in our work for prediction. To enable an accurate prediction using data-driven method, feature extraction is a critical step. If an extracted feature is well correlated with the fault propagation process (e.g., vibration root mean square increases as the machine degrades), a good prediction can be expected by extrapolating the historically observations to the future. Related work can be seen in (Coble & Hines, 2009), which used genetic algorithm to find the optimal feature subset, and in (Liao, 2014), which used genetic programming to discover novel features for prediction. In most of the cases, engineering expertise is need to a certain extent to guide the feature extraction, which might not be directly available for complex systems. We would like to explore automatic feature generation method for remaining useful life prediction when engineering expertise is unavailable. Deep learning has recently gained popularity in machine learning based on learning layers of network structure based on restricted Boltzmann machines (RBM). RBM has been widely used as a generative model in many applications such as image classification, speech recognition, and word representation. It has recently been applied in prognostics health management area for health state classification (Tamilselvan & Wang, 2013). Instead of using RBM in a classification scenario, we explore RBM as a feature extraction tool in a RUL prediction scenario.

Prior work (Pavel, Snyder, Frankle, Key, & Miller, 2010) has demonstrated that vibration data could be used for diagnosing machine imbalance fault conditions. Our study focuses on extending prior work by exploring various types of sensor and control data for diagnosing the imbalance of the machine tools. Prior work (Pavel & Iverson, 2012) proposed self-organizing maps and polynomial curve fitting for RUL prediction based on domain specific features such bearing signature frequencies. Our study focuses on automatic feature generation assuming domain specific expertise is unavailable.

Our contribution includes the following:

- Combining control and sensor signals for machine health condition estimation, while utilizing a different set of sensor data such as temperature, power, flow, and lubricant/coolant pH instead of vibration.
- A novel method of using Restricted Boltzmann Machine as a feature generation model and coupling with a random forest algorithm in remaining useful life prediction applications.

Our hypothesis is that these advancements to prior work will aid in improving the diagnosis and prognostics capability, as well as reducing the cost of machine diagnostics by utilizing cheaper sensors and saving engineering effort in feature engineering for predictive maintenance tasks.

2. TECHNICAL APPROACH

This section contains two subsections to describe the technical approaches for: (1) device health estimation by combining contextual control information with sensor data; and (2) remaining useful life prediction using Restricted Boltzmann Machine and random forest.

2.1. Device Health Estimation

For each extension to prior work listed in Section 1, we performed two main steps for diagnostics:

- Feature Extraction & Synthesis
- Model Selection

2.1.1. Feature Extraction & Synthesis

There are various approaches for condensing time series information into data mining features. Prior work has utilized transfer functions to map control signals to vibration sensor data (Pavel et al., 2010). The diagnosis step is then reduced to comparing the features of transfer function-predicted vibration data and the sensor-derived vibration data. This approach makes sense when the control signal directly impacts the output variables of the machine. For motion control of machine tools, the estimated transfer function should be similar to the transfer function of the implemented control (like PI or PID). Typical vibration data features would include average, standard deviation, and maximum FFT values (Deng, Runger, Tuv, & Vladimir, 2013).
However, we would like to diagnose the state of machine using not only accelerometers, but also other sensors, such as temperature sensors. Since temperatures at various locations are not part of active control loops, there may not exist well defined transfer functions that can map control signals to temperature sensor data very accurately. In such cases where conventional features extracted from temperature signals are not correlated with the fault (imbalance) to a sufficient degree. Additionally, if the associated sensors are too expensive to install, then data fusion may be applied.

There are three data fusion approaches typically used in machinery diagnostics (Liu & Wang, 2001; Jardine, Lin, & Banjевич, 2006)—data-level fusion, feature-level fusion, and decision-level fusion. Data-level fusion involves combining sensor data before feature extraction, such that features contain information gathered from multiple sensors. Feature-level fusion involves generating features from each sensor separately, then fusing this set of features generated from all of the sensors coherently for diagnostics. Finally, decision-level fusion creates diagnostics from each sensor separately, then aggregates these diagnostics into a single diagnostic output.

The choice of the three types of data fusion methods is often application specific. In our application, we found that temperature sensor data cannot resolve imbalance conditions by itself and control signal data is too coarse-grained to aid in classifying imbalance conditions using the standard data-fusion techniques. Note that we did not focus on spindle acceleration data, which could diagnose imbalance on its own (see Subsection 3.1.1) since that would require retrofitting existing machine tools with new expensive sensors and data acquisition hardware. Ideally we would like to use the readily accessible control signals and data from inexpensive temperature sensors to diagnose imbalance. To achieve this goal, we proposed a different type of data fusion approach. We used the control signal to provide the contextual information for temperature sensor data. The control signal is used for the segmentation of sensor data, but does not directly map into feature vectors (see Subsection 3.1.2).

2.1.2. Model Selection

Since the data sets are statistically small and dimensionality of the data is increased by feature synthesis, the models to be used for imbalance classification need to be carefully chosen to avoid over-fitting. The high-dimensional data needs to be projected to a much smaller sub-space to prevent over-fitting. To accomplish this, the main techniques used in this study are Principal Component Analysis (PCA) (Wold, Esbensen, & Geladi, 1987) and Linear Discriminant Analysis (LDA) (Koehler & Erenguc, 1990). These techniques are based on linear coordinate transformation, which makes them more likely to under-fit and less likely to over-fit (Yang, Chen, & Wu, 2011).

2.2. Device Remaining Useful Life Prediction

The remaining useful life (RUL) prediction algorithm can be summarized in Figure 1. The pre-processed data is input to the Restricted Boltzmann Machine to automatically generate features. The preprocessed data can actually be the raw signals, e.g. vibration signals, or time/frequency domain features of vibration, or features extracted by signal processing techniques e.g. discrete wavelet transform. The generated features are then input to a predictor, which is random forest in this case, to predict RUL.

![Figure 1. RUL prediction method.](image)

2.2.1. Feature Generation

Restricted Boltzmann Machine (RBM) can be considered as a two-layer network which consists a visible layer and a hidden layer. The visible layer corresponds to the observed input units (v), and the hidden layer corresponds to the feature detectors which are hidden units (h). Since we consider Gaussian input for both the input and hidden units, the energy function of the RBM is more complex than the common binary case. We defined the energy function as:

\[
E(v,h) = \sum_{i \in \text{vis}} \frac{(v_i - a_i)^2}{2\delta_i^2} + \sum_{j \in \text{hid}} \frac{(h_j - b_j)^2}{2\delta_j^2} - \sum_{i,j} v_i h_j \omega_{ij},
\]

(1)

where \(v_i, h_j\) are the states of the visible unit \(i\) and hidden unit \(j\), \(a_i, b_j\) are their levels, \(\delta_i, \delta_j\) are the standard deviations, and \(\omega_{ij}\) is the weight between them. The probability that the RBM network assigns to a visible vector is given by summing over all hidden vectors:

\[
P(v) = \frac{1}{Z} \sum_h \exp(-E(v,h)),
\]

(2)

\(Z\) is the partition function.
where \( Z = \sum_{v,h} \exp(-E(v,h)) \). Now we can define:

\[
P(v, h) = \frac{\exp(-E(v,h))}{Z} \tag{3}
\]

\[
P(h|v) = \frac{\exp(-E(v,h))}{\sum_h \exp(-E(v,h))} \tag{4}
\]

Then we can use the negative log likelihood gradient to update the parameters \((a_i, b_j, \delta_i, \delta_j, \omega_{ij} \in \theta)\) using:

\[
\frac{d}{d\theta} (-\log P(v)) = \frac{d}{d\theta} (-\log \sum_h P(v, h))
\]

\[
= \frac{d}{d\theta} (-\log \sum_h \exp(-E(v,h)) \frac{Z}{\sum_h \exp(-E(v,h))})
\]

\[
= \sum_h \left( \exp(-E(v,h)) \frac{dE(v,h)}{d\theta} \right) + \frac{1}{Z} \frac{dZ}{d\theta}
\]

\[
= \sum_h P(h|v) \frac{dE(v,h)}{d\theta} - \frac{1}{Z} \sum_{v,h} \exp(-E(v,h)) \frac{dE(v,h)}{d\theta}
\]

\[
= \sum_h P(h|v) \frac{dE(v,h)}{d\theta} - \sum_{v,h} P(v, h) \frac{dE(v,h)}{d\theta}
\]

(5)

The positive part in the last line of Eq. 5 is the so called positive phase contribution and the negative part is the so called negative phase contribution. The algorithm updates the parameters through iterations coupling with a learning rate and/or a momentum parameter until a stop criterion is met. The hidden unit states are used as the extracted features for RUL prediction.

### 2.2.2. RUL prediction

We treat the RUL prediction as a regression problem, in which we will train a supervised learner to match the extracted features and the expected RUL. In our case, we picked random forest algorithm as our prediction algorithm to demonstrate how to make predictions based on the features extracted from RBM. Random forest (Breiman, 2001) is an ensemble algorithm for classification or regression by aggregating the decision result from multiple decision trees. A simple pseudo algorithm of random forest training is described in Algorithm 1. After training, the algorithm outputs a RUL given a feature vector extracted from the RBM model described in Section 2.2.1.

#### Algorithm 1 Random forest training algorithm

- Draw \( N \) bootstrap data samples from original dataset \( D \);
- For each of the bootstrap data samples, build a decision tree. For each node of the decision tree, randomly sample \( M \) of the predictors (observations in our case), and choose the best split among the selected predictors;
- Make a prediction by aggregating the predictions of the \( N \) trees (e.g. majority votes);
- At each bootstrap iteration, predict data not in the bootstrap sample (called out-of-bag data) using the tree built with the bootstrap sample. Aggregating the out-of-bag error rate, and repeat the process until a preset threshold is met (i.e. error rate or maximum number of iteration).

### 3. Results

This section contains two subsections to demonstrate: (1) device health estimation using data collected from a machine tool including sensor data and MTConnect data; (2) remaining useful life prediction using run-to-failure dataset collected from a machine tool spindle testbed.

#### 3.1. Device imbalance condition estimation

We have explored three imbalance scenarios to investigate our hypothesis of diagnostics using:

- Sensor based diagnostics
- Control based temporal segmentation followed by sensor based diagnostics

##### 3.1.1. Sensor based Diagnostics

In this case, each sensor signal was analyzed separately to determine if any of the sensor signals contains enough diagnostic information to detect imbalance on its own. By plotting the time series data we find that spindle acceleration sensors (which captures vibration) show higher oscillation amplitudes (see Figure 2) with increasing imbalance. Since imbalance actually impacts moment of inertia of the spindle, this change in acceleration is expected.

We also considered measuring imbalance through temperature. From the energy flow perspective, additional acceleration caused by imbalance should result in higher energy consumption from the power source and higher energy dissipation to thermal inertias due to friction, which should result in temperature increase in parts of the machine tool. However, the time series data, from each of the temperature sensors, did not show distinguishing features similar to the acceleration sensors. An example of temperature sensor time series data is shown in Figure 3.

For this sensor data analysis, the features extracted are (i) average, (ii) standard deviation, (iii) maximum amplitude of FFT, and (iv) frequency for maximum amplitude of FFT. These four features are inspected visually to determine if im-
balance could be classified by a simple linear classifier. The spindle acceleration (X, Y, and Z) feature (maximum amplitude of FFT) showed easily visible characteristics that can distinguish between degrees of imbalance. See Figure 4 for an example of visual classification based on X-axis acceleration data. Other sensor signals like power, pH, flow, and temperature did not exhibit such classification capability.

3.1.2. Control-based Segmentation followed by Sensor-based Diagnostics

The second diagnostic approach that we explored combines both sensor and control data in a coherent manner. The first step in this approach is to utilize the control signal to provide temporal segmentation, i.e., assuming quasi-steady state, the goal is to find the time intervals in which the following conditions are satisfied: (i) all experiments display same values for the primary control signal (actual spindle speed), and (ii) all the control signals are constant over the same period. Note that, to investigate the dynamic response, rather than quasi-
steady state response, the control signals should be consistent across the experiments so that responses are compared under the same set of control inputs. Figure 5 (a) shows the result of this temporal segmentation scheme. For each of the control signals, we have computed the standard deviation at the each time step and identified the periods with standard deviation below a set threshold to find the consistent time intervals (shown as colored segments along the time axis in Figure 5 (b)). Then we find the intersection of the sets of consistent time intervals over all the control signals to determine the aggregate time intervals over which the control signals are statistically consistent (shown as black segments along the time axis in Figure 5 (c)).

These temporal segments are then mapped to sensor data to facilitate diagnostics. For each of 16 temporal segments, we computed features including (i) average, (ii) standard deviation, (iii) maximum FFT amplitude value, and (iv) FFT frequency at maximum amplitude. This step produces a 64 dimensional feature space to diagnose machine imbalance. As mentioned before, to avoid the overfitting we focus on linear transformation based approaches. We implemented Principal Component Analysis (PCA) to reduce the dimensionality from 64 to 4 (postulating that there should be 4 unique dimensions given the 4 uncorrelated features that we have selected). The PCA step is followed by Linear Discriminant Analysis to find the optimal coordinate transformation that provides maximum separation between classes. Result of this PCA-LDA analysis is shown in Figure 6 for Fluid Temperature sensor data. Another temperature sensor located at Spindle Motor also exhibits similar diagnostic capability after application of control based temporal segmentation. This demonstrates that control data can be used to provide context to sensor data in a way that helps diagnose machine imbalance. Thus, temperature sensor which had inferior diagnostic performance without context data, could classify imbalance perfectly when it is combined with additional context from control signal.

3.2. Spindle Remaining Useful Life Prediction

3.2.1. Experiment Setup and Data

The spindle test-bed was built at TechSolve using a frequency drive, a motor, a poly-V belt transmission, and a simplified spindle using two bearings identical to the ones used in the horizontal machining center. Figure 7 presents a picture of the spindle test-bed showing the motor, the belt transmission, and the actual spindle. A loading mechanism pulling on the nose of the spindle was added to accelerate the degradation. The force was kept constant as 35 lbs. The spindle was rotating at a constant speed of 9120 resolution per minute. The spindle motor was shut down automatically by the frequency drive when the bearing was locked at the end of life. A current sensor was installed on one phase of the power cable in the control box which controls the speed of the motor. An accelerometer was installed on the housing of the spindle to collect vibration data. Four thermal couples were installed to collect temperature data of the motor, spindle bearing, loading deck and ambient temperature, respectively. The sampling rate was 25600 Hz and 765440 data points were collected every hour.

We first examined the time series statistics of the vibration signal and energy based on the rotational speed which was 152 Hz. The purpose was trying to find the feature(s) that has/have trending though the life time so that the feature(s) can be used for prediction based on the historical trend. The result revealed that there was no obvious trending in none of the features at least from visual inspection. Figure 8 showed some of the features that we examined. The solid red lines were the smoothed features using a moving average window of length 3.

![Figure 4. Visual classification using spindle x-axis acceleration sensor.](image-url)
3.2.2. RUL Prediction Using Features Generated by RBM

Since there was no obvious trending in the features that we examined, the RUL prediction was supposed to be not straightforward. We would like to use the proposed RBM method to generate features automatically. In order to test the generative capability of the RBM and assuming there is no engineering guidance on feature extraction, we arbitrarily selected the frequency amplitude values ranging within 76 Hz and 532 Hz. It covered the frequency range up to 3.5 times of the rotating frequency. The frequency amplitude values were used as the input to the RBM. The RBM learning parameters were chosen by trial and error because there was no general guidance available to a practical application. The number of hidden nodes was set to 1000. The number of maximum of epochs was set to 100. The learning rate was chosen as 0.1 and it was fixed through the iterations. The momentum was set to be 0.1. The number of iteration for Gibbs sample of Contrastive Divergence algorithm was set to be 1.

After training, we input the training data itself to the trained RBM, and selected the hidden units whose standard deviations are greater than zero. The purpose was to avoid those hidden units that did not contain any variance information. As a result, 88 hidden nodes were selected.

The ground truth of RUL was calculated by the hours from the time stamp when the data was collected to the failure time stamp. All RUL hours were normalized to be in the range between 0 and 1. The selected 88-dimensional features and
the corresponding normalized RUL were trained by the ran-
dom forest regression algorithm. A five-fold cross validation
was used to validate the prediction. For comparison purposes,
we also directly applied the random forest regression to the

Figure 8. Vibration features.

(a) Average acceleration.
(b) Log of acceleration standard deviation.
(c) Log of maximum FFT amplitude.
(d) Log of maximum acceleration peak.
Table 1. Comparison of prediction methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. RMSE</th>
<th>Avg. MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.077</td>
<td>0.065</td>
</tr>
<tr>
<td>Random Forest + RBM</td>
<td>0.047</td>
<td>0.029</td>
</tr>
</tbody>
</table>

pre-processed data without using RBM feature generation. A five-fold cross validation was also applied for this case.

The result was shown in Table 1, which showed that the performance of the proposed RBM and random forest RUL prediction method was superior to the random forest method without using RBM to generate features in terms of both RMSE (Root Mean Square Deviation) and MAE (Mean Absolute Error) criterion. To show the RUL prediction result, 90% of the data was randomly selected for training and the rest was used for testing. The testing result was plotted in Figure 9. The solid line was the ground truth of RUL and the circles were the predicted RUL values. The predicted RUL was very close to the ground truth and the prediction RMSE was 0.043.

4. Conclusion

We proposed a method based on principal component analysis and linear discriminant analysis to combine control and sensor signals for machine health condition estimation. This work explored various types of sensory and control data for diagnosing the imbalance conditions of the machine tools. Our finding was that by combining context information gained from the control signal, certain sensors can have better sensitivity in diagnosing faulty conditions. In our case, the temperature sensor was able to classify machine imbalance conditions with much higher sensitivity than using itself alone. For practical implementation, using thermal couples can reduce the cost of both sensors and data storage comparing to using accelerometers.

For future work on device health estimation, we will explore diagnostics based on control signal alone. Given that relying on sensor data typically requires adding sensors to existing machine tools, it would be ideal if we could diagnose imbalance of the machine from control signals that are usually recorded (i.e., no additional hardware required). The expectation is that if a machine tool uses feedback controls, then the control signal should be impacted by any change in the operational characteristics (in this case the imbalance of the machine tools).

We also proposed a novel method of using Restricted Boltzmann Machine as a feature generation model and coupling with a random forest algorithm in remaining useful life prediction applications. RBM has been explored in many classification scenarios, but it hasn’t been explored in the RUL prediction scenario to our best knowledge. The run-to-failure test showed RBM can potentially generate useful features for RUL prediction with high accuracy.

For future work on RUL prediction, considering using a discriminative restricted Boltzmann machine (Larochelle & Bengio, 2008) model to integrate feature extraction and prediction as a unified task is of interest. One of the advantages is that as a unified task, model selection, parameter tuning, and initialization can be done only once comparing to using two learning phases (feature generation followed by prediction). Currently, the features generated from RBM barely have physical meaning or are hard to explain. If an extra objective term can be added in addition to the energy function 1 of RBM, it may generate features that are easier to understand e.g., a feature which has a better trending over the life span.

References


**Biographies**

**Linxia Liao** Linxia Liaos research interests include predictive and prescriptive analytics; system and components fault diagnostics and prognostics; signal processing; and machine learning algorithms as well as their integration on embedded systems. Currently, he is developing solutions for device health prognostics and patient disease progression modeling. Prior to joining PARC, Linxia worked as a research scientist with Siemens Corporate, Corporate Technology (previous Siemens Corporate Research) located in Princeton, NJ. He conducted research and implemented various prognostics and health management-related applications in the fields of manufacturing, energy, and transportation. He also previously worked at Siemens Technology-To-Business (TTB) Center in Berkeley, CA to transfer the patented Methods for prognosing mechanical systems technology from the university to industry applications. Dr. Liao received his Ph.D. degree in Industrial Engineering from the University of Cincinnati, where he conducted research at the NSF I/UCR Center for Intelligent Maintenance Systems (IMS). He earned his M.S. and B.S. degrees in Mechanical Science and Engineering at Huazhong University of Science and Technology (HUST) in China. Linxia has one issued patent and seven pending patents, and he has published one book chapter and 25+ papers in leading journals and conferences.

**Tomonori Honda** Tomonori Honda received his Ph.D. and M.S. in Mechanical Engineering from California Institute of Technology and B.S. in Mechanical Engineering and Nuclear Engineering from University of California, Berkeley. He also continued his Ph.D. research at Massachusetts Institute of Technology as a Postdoctoral Scholar and Research Scientist for several years. He has published over 30 papers in journals and peer-reviewed conferences. He has also won the 2013 JALA Ten Award from Society of Laboratory Automation and Screening (SLAS) and the 2013 ASME Design Theory and Methodology Best Paper Award. Prior to joining PARC, Tomonori served as a principal data scientist at Edmunds.com, where he led the modeling and analytics effort including reinforcement learning framework for realtime ads targeting using big data. He also worked as a junior analytics manager at Opera Solutions, where he explored different techniques to improve model performances and won an in-house modeling competition. He is a member of ASME.
Dr. Radu Pavel has over 20 years of research and teaching experience in mechanical and manufacturing engineering. Dr. Pavel is knowledgeable in the areas of modeling, design of experiments, test-bed instrumentation, data analysis, sensor technology, and training. His expertise include advanced grinding and finishing techniques for new materials such as superabrasive grinding of ceramics and composites, ELID grinding, and hard turning. Dr. Pavel has published multiple papers in refereed conference proceedings and journals, and organized symposia focused on advances in material processing and inspection for the ASME International Manufacturing Science and Engineering Conferences.

Dr. Pavel obtained his bachelor, master and doctoral degrees in mechanical engineering from University ‘Politehnica’ of Bucharest, Romania. Dr. Pavel also obtained a Ph.D. in Manufacturing Engineering from University of Toledo, Ohio, U.S.A.

Dr. Pavel was hired by TechSolve in February 2004 on the position of machining/grinding research engineer. For the first three years he has been a part of the research team of the “Intelligent Optimization and Control of Grinding Processes” project sponsored by the NIST-ATP division. Next, Dr. Pavel was selected to be part of the team working for the Smart Machine Platform Initiative (SMPI) program. His main focus areas were Tool Condition Monitoring and Condition-Based Maintenance for the Smart Machine. Under the SMPI program, Dr. Pavel evaluated innovative technologies, and contributed to the development of new technologies including those oriented towards the diagnostics and prognostics of health and maintenance of the machine tool.

Dr. Pavel was also the technical lead for the Adaptive Machining Program (AMPI). In this program, vision and laser sensing technologies were used to accurately measure the position and shape of a workpiece in the three dimensions (3D), before machining. The goal is to adapt the cutter path to the actual geometry of the workpiece and avoid the inaccuracies resulting from a cutter path derived from the ideal virtual model of the part. The 3D scanning and point-cloud processing techniques were used to determine the actual geometry of the part before and after machining.

Dr. Pavel is currently TechSolve’s Chief Technology Officer, leading a team of engineers from the Advanced Machining R&D department.
Distributed Adaptive Fault-Tolerant Consensus Control of Multi-Agent Systems with Actuator Faults

Mohsen Khalili1, Xiaodong Zhang2, Yongcan Cao3, and Jonathan A. Muse4

1 Wright State University, Dayton, OH 45435, USA
khalili.4@wright.edu
xiaodong.zhang@wright.edu

3,4 Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA
yongcan.cao@gmail.com
jonathan.muse.2@us.af.mil

ABSTRACT

This paper presents an adaptive fault-tolerant control (FTC) scheme for leader-follower consensus control of uncertain mobile agents with actuator faults. A local FTC component is designed for each agent in the distributed system by using local measurements and certain information exchanged between neighboring agents. Each local FTC component consists of a fault detection module and a reconfigurable controller module comprised of a baseline controller and an adaptive fault-tolerant controller activated after fault detection. Under certain assumptions, the closed-loop system stability and leader-follower consensus properties of the distributed system are rigorously established. A simulation example is used to illustrate the effectiveness of the FTC method.

1. INTRODUCTION

The study of distributed multi-agent systems focuses on the development of control algorithms that enable a team of interconnected agents to accomplish desired team missions. One unique feature of these algorithms is their distributed nature, where each agent takes actions based on information obtained from its local neighbors. This distributed nature has numerous benefits, such as scalability and robustness. The research on distributed multi-agent systems has received increasing attention due to its broad application in numerous areas, such as spacecraft formation flying (Ren & Beard, 2004), smart grid (Piipattanosompong, Feroze, & Rahman, 2009), and sensor networks (Cortes, Martinez, Karatas, & Bullo, 2004). One key concept in the study of distributed multi-agent systems is to have the team exchange information in order to achieve the desired goal. One typical scenario that was extensively studied is consensus, whose goal is to develop control algorithms such that a team of agents reach agreement on their final states via local interaction. The associated control algorithms are also called consensus algorithms. Albeit simple, the study of consensus provides foundation for the development of more advanced algorithms for more general team missions. For agents with different dynamics, numerous consensus algorithms were developed for single-integrator kinematics (Ol¨afati-Saber & Murray, 2004), double-integrator dynamics (Ren & Atkins, 2007), and general linear dynamics (Li, Duan, Chen, & Huang, 2010), where no model uncertainty was considered. To deal with model uncertainties, new consensus algorithms were developed for single-integrator kinematics (Yu, Chen, & Cao, 2011), double-integrator dynamics (Yu, Chen, Cao, & Kurths, 2010), and general linear dynamics (Li, Ren, Liu, & Fu, 2012).

Since such distributed multi-agent systems are required to operate reliably at all times, despite the possible occurrence of faulty behaviors in some agents, the development of fault diagnosis and accommodation schemes is a crucial step in achieving reliable and safe operations. In the last two decades, significant research activities have been conducted in the design and analysis of fault diagnosis and accommodation schemes (see, for instance, Blanken, Kinnairt, Lunze, & Staroswiecki, 2006). Most of these methods utilize a centralized architecture, where the diagnostic module is designed based on a global mathematical model of the overall system and is required to have real-time access to all sensor measurements. Because of limitations of computational resource and communication overhead, such centralized methods are not suitable for large-scale distributed interconnected systems. As a result, in recent years, there has been a significantly increasing research interest in the development of distributed fault diagnosis schemes for multi-agent systems (see, for instance, Kelliris, Polycarpou, & Parisini, 2013; Yan & Edwards, 2008; Ferrari, Parisini, & Polycarpou, 2012; Shames, Teixeira, Sandberg, & Johansson, 2011).

This paper presents a method for detecting and accommodating actuator faults in a class of distributed nonlinear uncertain multi-agent systems. A fault-tolerant control component is designed for each agent in the distributed system by utilizing local measurements and certain information exchanged between neighboring agents. Each local FTC component consists of two main modules: 1) an online fault detection scheme; and 2) the controller (fault accommodation) module consists of a baseline controller and an adaptive fault-tolerant controller employed after fault detection. Under certain assumptions, the closed-loop system’s stability and leader-following consensus properties are established for the baseline controller and adaptive fault-tolerant controller. A simulation example is used to illustrate the effectiveness of the FTC method.

The rest of this paper is organized as follows. Section 2 provides the graph theory notations. Problem formulation
for fault-tolerant leader-follower consensus control of multi-agent systems is described in Section 3. The closed-loop system stability and performance before fault occurrence is investigated in Section 4. The distributed fault detection is analyzed in Section 5. The design and analysis of the fault-tolerant control scheme after fault detection is rigorously investigated in Section 6. In Section 7, a simulation example is used to illustrate the effectiveness of the FTC method. Finally, Section 8 provides some concluding remarks.

2. Graph Theory Notations

A directed graph \( G \) is a pair \((V, \mathcal{E})\), where \( V = \{v_1, \ldots, v_P\} \) is a set of nodes, \( \mathcal{E} \subseteq V \times V \) is a set of edges, and \( P \) is the number of nodes. An edge is an ordered pair of distinct nodes \((v_j, v_i)\) meaning that the \( i \)th node can receive information from the \( j \)th node. For an edge \((v_j, v_i)\), node \( v_j \) is called the parent node, node \( v_i \) the child node, and \( v_j \) is a neighbor of \( v_i \). An undirected graph can be considered as a special case of a directed graph where \( (v_i, v_j) \in \mathcal{E} \) implies \( (v_j, v_i) \in \mathcal{E} \) for any \( v_i, v_j \in V \). A directed graph contains a directed spanning tree if there exists a node called the root such that the node has directed paths to all other nodes in the graph.

The set of neighbors of node \( v_i \) is denoted by \( N_i = \{j : (v_j, v_i) \in \mathcal{E}\} \). The weighted adjacency matrix \( \mathcal{A} = [a_{ij}] \in \mathbb{R}^{P \times P} \) associated with the directed graph \( G \) is defined by \( a_{ij} = 0, a_{ij} > 0 \) if \((v_j, v_i) \in \mathcal{E} \), and \( a_{ij} = 0 \) otherwise. The topology of an intercommunication graph \( G \) is said to be fixed, if each node has a fixed neighbor set and \( a_{ij} \) is fixed. It is clear that for undirected graphs \( a_{ij} = a_{ji} \). The Laplacian matrix \( L = [l_{ij}] \in \mathbb{R}^{P \times P} \) is defined as \( l_{ii} = \sum_{j \in N_i} a_{ij} \) and \( l_{ij} = -a_{ij}, i \neq j \). Both \( L \) and \( \mathcal{A} \) are symmetric for undirected graphs and \( L \) is positive semidefinite.

3. Problem Formulation

3.1. Distributed Multi-Agent System Model

Consider a set of \( M \) agents with the dynamics of the \( i \)th agent, \( i = 1, \ldots, M \), being described by the following dynamics

\[
\begin{align*}
\dot{x}_i &= \phi_i(x_i) + u_i(y_i, y_j) + \eta_i(x_i, t) + \beta_i(t - T_i)f_i(u_i(y_i, y_j)) \\
y_i &= x_i + d_i,
\end{align*}
\]

where \( x_i \in \mathbb{R}^{n_i}, u_i \in \mathbb{R}^{m_i}, \) and \( y_i \in \mathbb{R}^{n_i} \) are the state vector, input vector, and output vector of the \( i \)th agent, respectively. Additionally, \( y_j \) contains the output variables of neighboring agents that directly communicate with agent \( i \), including the time-varying leader to be tracked (i.e., \( y^r \)).

The time-varying leader is denoted by \( y^r \). The nonlinear term \( \phi_i(x_i) \) in Eq. (1) satisfies the Lipschitz condition (Rajamani, 1998); \( \forall x_i, y_i \in \mathbb{R}^{n_i}, \end{align*} \]

The difference between the nominal model Eq. (2) and the actual (healthy) system dynamics Eq. (3) is due to vector fields \( \eta_i \) and \( d_i \) representing the modeling uncertainty in the state dynamics and output measurement of the \( i \)th agent, respectively.

The term \( \beta_i(t - T_i)f_i(u_i) \) denotes the changes in the dynamics of \( i \)th agent due to the occurrence of an actuator fault. Specifically, \( \beta_i(t - T_i) \) represents the time profile of a fault which occurs at some unknown time \( T_i \); \( f_i(u_i) = [\theta_{i1}u_{i1}, \ldots, \theta_{in}u_{in}]^T \) is an actuator fault function representing partial loss of effectiveness of the actuators, where the fault parameter \( \theta_{ip} \in (-1,0), p = 1, \ldots, n, \) characterizes the unknown magnitude of the actuator fault. In this paper, the time profile function \( \beta_i(\cdot) \) is assumed to be a step function (i.e., \( \beta_i(t - T_i) = 0 \) if \( t < T_i \) and \( \beta_i(t - T_i) = 1 \) if \( t \geq T_i \)). The system model (1) allows the occurrence of faults in multiple agents but it is assumed there is only a single fault in each agent at any time.

Remark 1: The distributed multi-agent system model given by Eq. (1) is a nonlinear generalization of the single integrator dynamics considered in literature (for instance, (Ren & Beard, 2008)). In this paper, in order to investigate the fault-tolerance and robustness properties, the fault function \( f_i(u_i) \) and modeling uncertainties \( \eta_i \) and \( d_i \) are included in the system model.

The objective of this paper is to develop a robust distributed fault diagnosis and fault-tolerant leader-following consensus control scheme for the class of distributed multi-agent systems described by Eq. (1). The following assumptions are made throughout the paper:

Assumption 1. Each component of the modeling uncertainties, represented by \( \eta_i(x_i, t) \) and \( d_i \) in Eq. (1) and also the rate of change of the measurement uncertainty represented by \( d_i \), has a known upper bound, i.e., \( \forall p = 1, \ldots, n, \forall x_i \in \mathbb{R}^{n_i}, \forall u_i \in \mathbb{R}^{m_i}, \forall y_i \in \mathbb{R}^{n_i}, \end{align*} \]

where the bounding function \( \bar{\eta}_{ip}, \bar{d}_{ip}, \) and \( \Xi_{ip} \) are known and uniformly bounded.

Assumption 2. The nonlinear term \( \phi_i(x_i) \) in Eq. (1) satisfies a Lipschitz condition (Rajamani, 1998); \( \forall x_i, y_i \in \mathbb{R}^{n_i}, \end{align*} \]

where \( \sigma_{ip} \) is a known Lipschitz constant.

Assumption 3. The communication topology among followers is undirected and the leader has directed paths to all followers.

Assumption 1 characterizes the class of modeling uncertainty under consideration. The bound on the modeling uncertainty is needed in order to distinguish between the effects of faults and modeling uncertainty during the fault diagnosis process (Emami-Naeini, Akhter, & Rock, 1988). Assumption 2 provides Lipschitz condition on the nominal nonlinearity \( \phi_i(x_i) \) in Eq. (1), which is needed for FDI and FTC designs. Assumption 3 is needed to ensure that the information exchange among agents is sufficient for the team to achieve the desired team goal.
3.2. Fault-Tolerant Control Structure

In this paper, we investigate the FTC problem of leader-following consensus. Specifically, the objective is to develop distributed robust FTC algorithms to guarantee that each agent’s output converges to the time-varying reference output of the leader even in the presence of modeling uncertainty and actuator fault.

We can represent the collective output dynamics as

\[ \dot{y}_p = -L y_p + \tilde{\phi}_p + \zeta_p - \bar{\zeta}_p + \dot{d}_p, \quad (12) \]

where \( u_{ip} \) and \( y_{ip} \) are the \( p \)th component of the input and output vectors of the \( i \)th agent, respectively, \( p = 1, \ldots, n, \) \( i = 1, \ldots, M, \) \( \dot{y}_{ij} \triangleq y_{ip} - y_{jp}, \) \( \kappa_p \) is a positive bound on \( |\dot{y}_{ip}| \), \( i.e., \kappa_p \geq |\dot{y}_{ip}|, \) \( sgn(\cdot) \) is the sign function, \( N_i \) is the set of neighboring agents that directly communicate with the \( i \)th agent including the leader, and \( k_{ij}, \) for \( j \in N_i, \) are positive constants. Notice that \( k_{im} = 0, \) for \( m \notin N_i. \)

Note that, by adding a leader, the topology graph of the system has a spanning tree with the leader as its root. First, we need the following Lemmas:

Lemma 1. (Ren & Beard, 2008) The Laplacian matrix \( L \in \mathbb{R}^{P \times P} \) of a directed graph \( G \) has at least one 0 eigenvalue with \( I_P \) as its right eigenvector, where \( I_P \) is a \( P \times 1 \) column vector of ones, and all nonzero eigenvalues of \( L \) have positive real parts. 0 is a simple eigenvalue of \( L \) if and only if the directed graph \( G \) has a spanning tree.

Lemma 2. Consider a connected graph \( G \) with the leader as the \((M + 1)\)th node. The matrix

\[ \tilde{L} = \Psi L + \mathcal{L}^T \Psi \]  

is positive semidefinite and has a simple zero eigenvalue with \( I_{M+1} \) as its right eigenvector. where \( \Psi \in \mathbb{R}^{(M+1) \times (M+1)} \) is the Laplacian matrix of the graph as if the communication between leader and followers is undirected, and \( \mathcal{L} \in \mathbb{R}^{(M+1) \times (M+1)} \) is the Laplacian matrix of the graph with a directed leader.

Proof. The proof of the above Lemma can be found in (Khalili, Zhang, Polycarpou, Parisini, & Cao, 2015). \( \square \)

Remark 2: It is worth noting that the Laplacian matrix \( \Psi \) for the undirected graph is only considered for the purpose of controller performance analysis. The actual distributed controller topology is directed, since the leader is only sending the data and does not receive any data from other agents.

The following result characterizes the stability and leader-following performance properties of the controlled system before fault occurrence.

Theorem 1. In the absence of faults, the baseline controller described by Eq. (9) has the following properties:

1. The leader-follower consensus is achieved asymptotically with a time-varying reference state, i.e., \( y_i \rightarrow y^r \rightarrow 0 \) as \( t \rightarrow \infty; \)

2. All states are bounded, \( x_i \rightarrow x_j \rightarrow d_j - d_i \) as \( t \rightarrow \infty. \)

Proof. We know from Eq. (1) that \( \dot{y}_{ip} = \dot{z}_{ip} + \hat{d}_{ip}, \) for \( p = 1, \ldots, n. \) Based on Eqs. (9) and (3), the closed-loop system dynamics are given by

\[ \begin{align*}
\dot{y}_{ip} &= \phi_{ip}(x_{ip}) - \phi_{ip}(y_{ip}) - \sum_{j \in N_i} k_{ij} \dot{y}_{ij} + \eta_{ip}(x_i, t) + \hat{d}_{ip} \\
&= -L y_{ip},
\end{align*} \]

where \( u_{ip} \) and \( y_{ip} \) are the \( p \)th component of the input and output vectors of the \( i \)th agent, respectively, \( p = 1, \ldots, n, \) \( i = 1, \ldots, M, \) \( \dot{y}_{ij} \triangleq y_{ip} - y_{jp}, \) \( \kappa_p \) is a positive bound on \( |\dot{y}_{ip}| \), \( i.e., \kappa_p \geq |\dot{y}_{ip}|, \) \( sgn(\cdot) \) is the sign function, \( N_i \) is the set of neighboring agents that directly communicate with the \( i \)th agent including the leader, and \( k_{ij}, \) for \( j \in N_i, \) are positive constants. Notice that \( k_{im} = 0, \) for \( m \notin N_i. \)

We can represent the collective output dynamics as

\[ \dot{y}^p = -L y^p + \hat{d}^p + \zeta^p - \hat{\zeta}^p + \hat{d}^p, \]

Figure 1. Distributed FTC architecture for the \( i \)th agent

The distributed FTC architecture considered is shown in Figure 1. First of all, we define two important time–instants: \( T_i \) is the fault occurrence time; \( T_d > T_i \) is the time–instant when a fault is detected; The structure of the fault-tolerant controller for the \( i \)th agent takes on the following general form (Zhang, Parisini, & Polycarpou, 2004):

\[
\begin{align*}
\omega_i & = \begin{cases} 
  g_0(\omega_i, y_i, y_d, t), & \text{for } t < T_d \\
  g_D(\omega_i, y_i, y_d, t), & \text{for } t \geq T_d
\end{cases} \\
\end{align*}
\]

\[
\begin{align*}
u_i & = \begin{cases} 
  h_0(\omega_i, y_i, y_d, t), & \text{for } t < T_d \\
  h_D(\omega_i, y_i, y_d, t), & \text{for } t \geq T_d
\end{cases}
\end{align*}
\]

where \( \omega_i \) is the state vector of the distributed controller; \( g_0, g_D \) and \( h_0, h_D \) are nonlinear functions to be designed according to the following qualitative objectives:

1. In a fault free mode of operation, a baseline controller guarantees the output of \( i \)th agent \( y_i(t) \) should track the leader’s time-varying output \( y^r \), even in the presence of plant modeling uncertainty.

2. If an actuator fault is detected, the baseline controller is reconfigured to compensate for the effect of the fault. This new controller should guarantee the boundedness of system signals and leader-following consensus, even in the presence of fault.

4. Baseline Controller Design

In this section, we design the baseline controller and investigate the closed-loop system stability and performance before fault occurrence. The dynamics of the agents before fault occurrence (i.e., for \( 0 \leq t < T_i \)) is given by Eq. (3). Without loss of generality, let the leader be agent number \( M + 1 \) with a time-varying reference output (i.e., \( y_{M+1} = y^r \)). The baseline controller for the \( i \)th agent can be designed as:

\[ u_{ip} = - (\bar{\eta}_{ip} + \sigma_{ip} \hat{d}_{ip} + \Xi_{ip} + \kappa_i) sgn \left( \sum_{j \in N_i} k_{ij} \dot{y}_{ij} \right) \\
- \left( \sum_{j \in N_i} \left( k_{ij} \dot{y}_{ij} \right) - \phi_{ip}(y_{ip}) \right), \quad (9) \]
where $y^p \in \mathbb{R}^{M+1}$ is comprised of the $p$th output component of the $M+1$ agents, including the leader as the $(M+1)$th agent, i.e., $y^p = [y_{1p}, y_{2p}, \ldots, y_{M+1p}, y_{p}^T]^T$, the terms $\zeta^p \in \mathbb{R}^{M+1}$, $\tilde{\zeta}^p \in \mathbb{R}^{M+1}$, and $\phi^p \in \mathbb{R}^{M+1}$ are defined as

$\zeta^p = \begin{bmatrix} \eta_{1p}, \ldots, \eta_{M+1p}, 0 \end{bmatrix}^T$, \hspace{1cm} (13)

$\tilde{\zeta}^p = \begin{bmatrix} \tilde{\eta}_{1p}, \ldots, \tilde{\eta}_{M+1p}, 0 \end{bmatrix}^T$, \hspace{1cm} (14)

$\phi^p = \begin{bmatrix} \phi_{1p}, \ldots, \phi_{M+1p}, 0 \end{bmatrix}^T$, \hspace{1cm} (15)

$\delta^p = \begin{bmatrix} d_{1p}, \ldots, d_{M+1p}, 0 \end{bmatrix}^T$, \hspace{1cm} (16)

where $\tilde{\eta}_{ip} = (\eta_{ip} + \sigma_{ip}d_{ip} + \Xi_{ip} + \kappa_{ip}) \text{sgn}\left(\sum_{j \in N_{M+1}} k_{ij}\tilde{y}_{ij}\right)$, and $\phi_{ip} = \phi_{ip}(x_{ip} - \phi_{ip}(y_{ip}))$, $i = 1, \ldots, M$. We consider the following Lyapunov function candidate:

$$V_p = y^p \Psi y^p = \frac{1}{2} \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} (y_{ip} - y_{jp})^2$$

$$+ \frac{1}{2} \sum_{i=1}^{M} k_{i(M+1)} (y_{ip} - y_{i(M+1)p})^2,$$ \hspace{1cm} (17)

where $\Psi$ is defined in Lemma 2, and $y_{i(M+1)p}$ is the $p$th component of the leader’s constant output $y^p$. Then, the time derivative of the Lyapunov function Eq. (17) along the solution of Eq. (12) is given by

$$\dot{V}_p = -y^p \dot{\mathcal{L}} y^p + 2y^p \sum_{i=1}^{M} k_{i(M+1)} (y_{ip} - y_{i(M+1)p})$$

$$+ 2y^p \Psi (\tilde{\phi}^p + \tilde{\phi}^p - \phi^p + \delta^p),$$ \hspace{1cm} (18)

where $\dot{\mathcal{L}}$ is defined in (10). Based on Eq. (13), and noticing that $\eta_{i(M+1)p}$ is zero, we have

$$y^p \Psi \zeta^p = \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} (y_{ip} - y_{jp}) \eta_{ip}.$$ \hspace{1cm} (19)

By using the same reasoning logic and knowing that $\eta_{i(M+1)p} = \hat{d}_{i(M+1)p} = \hat{\phi}_{i(M+1)p} = 0$, we can obtain the following from Eqs. (14), (15) and (16):

$$y^p \Psi \tilde{\zeta}^p = \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \hat{y}_{ij}\tilde{\zeta}_{ij},$$ \hspace{1cm} (20)

$$y^p \Psi \phi^p = \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \hat{y}_{ij}\hat{\phi}_{ip},$$ \hspace{1cm} (21)

$$y^p \Psi \delta^p = \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \hat{y}_{ij}\hat{d}_{ip}.$$ \hspace{1cm} (22)

Using the property that $k_{ij} = k_{ji}$ for $j \in N_i$, $j \neq M+1$ (based on Assumption 3), we know that

$$\sum_{i=1}^{M} \sum_{j \in N_i, j \neq M+1} k_{ij} (y_{ip} - y_{jp}) = 0.$$ Therefore, we have

$$2\dot{y}_{ip} \sum_{i=1}^{M} k_{i(M+1)} (y_{ip} - y_{i(M+1)p}) = -2\dot{y}_{ip} \sum_{i=1}^{M} k_{ij} \hat{y}_{ij}. \hspace{1cm} (23)$$

By substituting Eqs. (19), (20), (21), (22) and (23) into Eq. (18), we have

$$\dot{V}_p = -y^p \dot{\mathcal{L}} y^p + 2 \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \hat{y}_{ij}\eta_{ip} + \phi_{ip} + \tilde{d}_{ip} - \tilde{y}_{ip}$$

$$-2 \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \hat{y}_{ij}\hat{\zeta}_{ip},$$ \hspace{1cm} (24)

Based on the Assumptions 1 and 2, we have

$$(\eta_{ip} + \phi_{ip} + \tilde{d}_{ip} - \tilde{y}_{ip}) \sum_{j \in N_i} k_{ij} \hat{y}_{ij} - \zeta_{ip} \leq 0.$$ \hspace{1cm} (25)

Therefore, by applying the above inequality to Eq. (24), we obtain

$$\dot{V}_p \leq -y^p \dot{\mathcal{L}} y^p.$$ Therefore, using Lemma 2, we know that $\dot{V}_p$ is negative definite with respect to $y_{ip} - y_{jp}$, because the only $y^p$ that makes $-y^p \dot{\mathcal{L}} y^p$ zero is $y^p = \mathbf{1}_{M+1}c$, where $c$ is a constant. Therefore, consensus with respect to the agents’ outputs is reached asymptotically, i.e., $y_{ip} - y_{jp} \to 0$ as $t \to \infty$. More specifically, $y_{ip} - y_{jp} \to 0$ as $t \to \infty$ and therefore, the leader-follower consensus is reached asymptotically. In the presence of the output measurement uncertainty $d_{ip}$, by using Eq. (3), we have $x_{ip} - x_{jp} \to d_{jp} - d_{ip}$ as $t \to \infty$. \hspace{1cm} (26)

5. DISTRIBUTED FAULT DETECTION

The distributed fault detection architecture is comprised of $M$ local fault detection components designed for each of the $M$ agents. The objective of each local fault detection component is to detect faults in the corresponding agent. Under normal conditions, each local fault detection estimator (FDE) monitors the corresponding local agent to detect the occurrence of any fault.

Based on the agent model described by Eq. (1), the FDE for each agent is chosen as:

$$\dot{\hat{x}}_i = \phi_i(y_i) + u_i + H_i(y_i - \hat{y}_i)$$

$$\dot{\hat{y}}_i = \hat{x}_i,$$ \hspace{1cm} (26)

where $\hat{x}_i \in \mathbb{R}^n$ and $\hat{y}_i \in \mathbb{R}^n$ denote the estimated local state and output, $H_i = \text{diag}(h_{i1}, \ldots, h_{in})$ is a positive definite matrix, where $-h_{ip} < 0$ is the estimator pole, $p = 1, \ldots, n$, $i = 1, \ldots, M$. Without loss of generality, let the observer gain be $H_i = h_{i}I_n$ where $I_n$ is a $n \times n$ identity matrix. It is worth noting that the distributed FDE Eq. (26) for the $i$th agent is constructed based on local input and output variables (i.e. $u_i$ and $y_i$) and certain communicated information $y_j$ from the FDE associated with the $j$th agent.
For each local FDE, let $\tilde{x}_i = x_i - \hat{x}_i$ denote the state estimation error of the $i$th agent. Then, before fault occurrence (i.e., for $0 \leq t < T$, by using Eqs. (1) and (26), the estimation error dynamics are given by

$$
\begin{align*}
\dot{\tilde{x}}_i &= -H_i \tilde{x}_i + \phi_i(x_i) - \phi_i(\hat{x}_i) - H_i d_i + \eta_i(x_i, t) \\
y_i &= \tilde{x}_i + d_i.
\end{align*}
$$

(27)

The presence of uncertainties $\eta_i(x_i, t)$ and $d_i$ cause a nonzero estimation error. A bounding function on the state estimation error $\tilde{x}_i$ before the occurrence of the fault can be derived. Specifically, based on Assumptions 1-2, for $0 \leq t < T$, each component of the state estimation error $\tilde{x}_i$ satisfies

$$
|\tilde{x}_i| \leq \int_0^t e^{-h_i(t-\tau)} (\eta_i + (h_i + \sigma_i)d_i)d\tau + \tilde{x}_i e^{-h_i t},
$$

where $\tilde{x}_i$ is a conservative bound on the initial state estimation error (i.e., $|\tilde{x}_i(0)| \leq \tilde{x}_i$). Therefore, for each component of the output estimation error (i.e., $e_i = y_i - \hat{y}_i$), by using Eq. (27) and applying the triangle equality, we have $|e_i| \leq \epsilon_i$, where

$$
\nu_i(t) \triangleq \int_0^t e^{-h_i(t-\tau)} (\tilde{\eta}_i + (h_i + \sigma_i)d_i)d\tau + \tilde{x}_i e^{-h_i t}.
$$

(28)

Note that the integral term in the above threshold can be easily implemented as the output of a linear filter with the input given by $\tilde{\eta}_i + (h_i + \sigma_i)d_i$.

Thus, we have the following:

**Fault Detection Decision Scheme:** The decision on the occurrence of a fault (detection) in the $i$th agent is made when the modulus of at least one component of the output estimation error (i.e., $e_i(t)$) generated by the local FDE exceeds its corresponding threshold $\nu_i(t)$ given by Eq. (28).

The fault detection time $T_d$ is defined as the first time instant such that $|\epsilon_i| > \nu_i$, for some $T_d \geq T$, and some $p \in \{1, \ldots, n\}$, that is,

$$
T_d \triangleq \inf_{p=1}^n \{ t \geq 0 : |\epsilon_i(t)| > \nu_i(t) \}
$$

6. **FAULT-TOLERANT CONTROLLER MODULE**

In this section, the design and analysis of the fault-tolerant control scheme is rigorously investigated for the closed-loop system after fault detection. After the fault is detected at time $t = T_d$, the nominal controller is reconfigured to ensure the system stability and tracking performance after fault detection. In the following, we describe the design of the fault-tolerant controller using adaptive tracking techniques.

For $t \geq T_d$, in the case of an actuator fault, the dynamics of the system takes on the following form: for $p = 1, \ldots, n$

$$
\begin{align*}
\dot{x}_i &= \phi_i(x_i) + (1 + \theta_i)u_i + \eta_i(x_i, t) \\
y_i &= x_i + d_i.
\end{align*}
$$

(29)

Without loss of generality, let the leader be agent number $M + 1$ with a set of neighborhoods $N_{M+1}$. The control objective is to force the output $y_i, t = 1, \ldots, M$, to track the output of the leader with a known time-varying output $y^*$.

After the detection of the actuator fault, i.e., $t \geq T_d$, the following adaptive fault-tolerant controller is adopted:

$$
\begin{align*}
u_i &= \frac{1}{1 + \theta_i} \bar{u}_i \\
\bar{u}_i &= -\phi_i(y_i) - \sum_{j \in N_i} (k_{ij} \bar{y}_{ij}) + \tilde{c}_i \\
\hat{\theta}_i &= P \theta_i \left\{ \Gamma_i \sum_{j \in N_i} k_{ij} \bar{y}_{ij} u_i \right\}
\end{align*}
$$

(30) (31) (32)

where $\hat{\theta}_i$ is an estimation of the unknown actuator fault magnitude $\theta_i$ with the projection operator $P$ restricting $\hat{\theta}_i$ to the corresponding set (i.e., $\hat{\theta}_i \in [\bar{\theta}_i, 0]$), with $\bar{\theta}_i \in (-1, 0)$ is used to ensure that $\hat{\theta}_i$ remains within a certain region to guarantee that the denominator of the control law does not approach zero (Ioannou & Sun, 1996), and $\Gamma_i$ is a symmetric positive definite learning rate matrix.

The following theorem characterizes the stability and following performance properties of the adaptive fault-tolerant controller for $t \geq T_d$:

**Theorem 2.** Assume that a fault occurs at time $T$, and that it is detected at time $T_d$. Then, the fault-tolerant controller Eq. (30) and fault parameter adaptive law Eq. (32) guarantee that

1. The leader-follower consensus is achieved asymptotically with a time-varying reference state, i.e. $y_i - y_i^* \to 0$ as $t \to \infty$;
2. All states are bounded, and $x_i - x_j \to d_j - d_i$ as $t \to \infty$.

**Proof.** Using some algebraic manipulations, we can rewrite Eq. (30) as $u_i = \bar{u}_i - \theta_i \bar{u}_i$. Therefore, substituting $\bar{u}_i$ in Eq. (30) and using Eq. (31), the closed-loop system dynamics are given by

$$
\begin{align*}
y_i &= -\sum_{j \in N_i} (k_{ij} \bar{y}_{ij}) + \eta_i(x_i, t) + d_i - \tilde{c}_i \\
&\quad+ \phi_i(x_i) - \phi_i(\hat{x}_i) + \hat{\theta}_i u_i.
\end{align*}
$$

We can represent the collective output dynamics as

$$
\dot{y}^P = -L y^P + \hat{\phi}^P + \xi^P - \tilde{c}^P + \bar{d}^P + \xi^P
$$

(33)

where $y^P \in \mathbb{R}^{M+1}$, $p = 1, \ldots, n$, is comprised of the $p$th component of the $M$ agents and the leader as the $(M + 1)$th agent, i.e., $y^P = [y_{1p}, y_{2p}, \ldots, y_{Mp}, y_M^P]^T$, and the terms $\xi^P \in \mathbb{R}^{M+1}$, $\tilde{c}^P \in \mathbb{R}^{M+1}$, $\hat{\phi}^P \in \mathbb{R}^{M+1}$, $\bar{d}^P \in \mathbb{R}^{M+1}$ are defined in Eqs. (13), (14), (15) and (16), and the term $\xi^P$ is defined as

$$
\xi^P \triangleq [\hat{\theta}_1 u_1, \ldots, \hat{\theta}_M u_M, 0]^T.
$$

(34)
where $\tilde{\theta}_{ip} = \theta_{ip} - \hat{\theta}_{ip}$ is the actuator fault magnitude estimation error.

We consider the following Lyapunov function candidate:

$$V_p = y^T \Psi y_p + \tilde{\theta}_{ip}^T (\Gamma_p)^{-1} \tilde{\theta}_{ip},$$  \hspace{1cm} \text{(35)}$$

where $\Psi$ is defined in Lemma 2, $\tilde{\theta}_{ip}^T = [\tilde{\theta}_{ip}, \cdots, \tilde{\theta}_{Mp}]^T$ is the collective actuator fault magnitude parameter estimation errors, and $\Gamma_p = \text{diag}\{\Gamma_{1p}, \cdots, \Gamma_{Mp}\}$ is a positive definite adaptive learning rate matrix. Then, using Eqs. (19), (20), (21) and (22), and the same reasoning logic for Eq. (34), the time derivative of the Lyapunov function Eq. (35) along the solution of Eq. (33) is given by

$$\dot{V}_p = -y^T \tilde{L} y_p + 2 \sum_{i=1}^{M} \tilde{\theta}_{ip} \left( \sum_{j \in N_i} k_{ij} \tilde{y}_{ij} u_{ip} - (\Gamma_{ip})^{-1} \tilde{\theta}_{ip} \right)$$

$$\hspace{1cm} + 2 \sum_{i=1}^{M} \sum_{j \in N_i} k_{ij} \tilde{y}_{ij} \left( (\hat{\theta}_{ip} + \eta_{ip} + \hat{d}_{ip} - \bar{y}_p^T) - \bar{z}_{ip} \right),$$

where $\tilde{L}$ is defined in (10). Therefore, by choosing the adaptive law as Eq. (32), and after some algebraic manipulations, we have

$$\dot{V}_p \leq -y^T \tilde{L} y_p - 2 \sum_{i=1}^{M} \left( \sum_{j \in N_i} k_{ij} (y_{ip} - y_{jp}) \right)^2.$$ 

It is worth noting that since the parameter projection modification can only make the Lyapunov function derivative more negative, the stability properties derived for the standard algorithm still hold (Farrell & Polycarpou, 2006). Because $\tilde{L}$ is positive semidefinite, referring to the proof of Theorem 1, we know that $\dot{V}_p \leq 0$ with respect to $y_{ip} - y_{jp}$ and $\tilde{\theta}_{ip}$. Integrating both sides of $\dot{V}_p$, we know that $y_{ip} - y_{jp} \in L_2$. Since $(y_{ip} - y_{jp}) \in L_\infty \cap L_2$ and $y_{ip} - y_{jp} \in L_\infty$, we can conclude that consensus is reached asymptotically, i.e., $y_{ip} - y_{jp} \to 0$ as $t \to \infty$. More specifically, $y_{ip} - y_{ip}^* \to 0$ as $t \to \infty$ and therefore, the leader-follower consensus is reached asymptotically. In the presence of the output measurement uncertainty $d_{ip}$, by using Eq. (3), we have $x_{ip} - x_{jp} \to d_{ip} - d_{ip}$ as $t \to \infty$. \hfill \Box

Remark 3: Note that the convergence of tracking errors does not require the convergence of fault parameter estimation error, which requires the condition of persistency of excitation (Farrell & Polycarpou, 2006). In this paper, we do not assume persistency of excitation.

7. Simulation Results

In this section, a simulation example of a networked multi-agent system consisting of 5 agents is considered to illustrate the effectiveness of the distributed fault-tolerant control method. The dynamics of each agent is given by

$$\dot{x}_i = u_i + \eta_i + \beta_i (t - T_i) f_i (u_i), \hspace{1cm} \text{(36)}$$

$$y_i = x_i + d_i,$$

where, for $i = 1, \cdots, 5$, the state vector $x_i = [x_{i1}, x_{i2}]^T$ represents the $i$th agent’s position in a two-dimension coordinate, $y_i$ and $u_i = [\bar{v}_i \cos(\psi_i), \bar{v}_i \sin(\psi_i)]^T$ are the output and input vectors, and $\psi_i$ and $\bar{v}_i$ in the input vector $u_i$ are the orientation and the linear velocity of each agent representing a ground vehicle, respectively. The ground vehicle model given in (36) is a standard unicycle-like model that can be controlled with the orientation $\psi_i$ and vehicle linear velocity $\bar{v}_i$. Using the developed algorithms, the desired orientation and linear velocity of the ground vehicle robot can be obtained uniquely. Then, a low level controller can be designed to track the desired orientation and linear velocity for driving the ground vehicles to desired positions.

The unknown modeling uncertainty in the local dynamics of the agents are assumed to be sinusoidal signals $\eta_i = [0.5 \sin(t), 0.5 \sin(t)]^T$ which is assumed to be bounded by $\eta_i = [0.6, 0.6]^T$. There is also an unknown uncertainty in the sensor measurement $d_i = [-0.5 \cos(t), -0.5 \cos(t)]^T$, which is assumed to be bounded by $d_{i1} = d_{i2} = 0.6$. The objective is for each agent to follow the leader described by $y_r = [\bar{y}_r, \bar{y}_r]^T = [4 + \sin(t), 4 + \cos(t)]^T$.

The Laplacian matrix of the intercommunication graph of agents plus leader, shown in Figure 2, is chosen as

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & 0 & 0 & -1 & -1 \\ 0 & 0 & 2 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & -1 & -1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$ 

The fault considered here is an actuator fault function $f_i(t) = \theta_i u_i$, where the magnitude of this fault is considered as $\theta_i \in [-0.8, 0]$. The observer gain for fault detection estimator is chosen as $h_i = 2$. After fault detection, the controller is reconfigured to accommodate the actuator fault occurred. We set the adaptive gain $\Gamma_i = 5$ with a zero initial condition (see Eq. (32)).

Figure 3 shows the fault detection results when actuator faults with a magnitude of 0.4 and 0.35 occur to agents 1 and 2 at $T_1 = 5$ and $T_2 = 8$ second, respectively. As can be seen from Figure 3, the residual corresponding to the output generated by the local FDE designed for agents 1 and 2 exceeds its threshold immediately after fault occurrence. Therefore, the actuator faults in agent 1 and 2 are timely detected. Note that the residual signals are time-varying because the uncertainties $\eta_i$ and $d_i$ in (27) are time-varying.

Regarding the performance of the adaptive fault-tolerant controllers, as can be seen from Figure 4, the tracking errors converge to zero. Thus, the leader-following consensus is achieved using the proposed adaptive FTC. On the other hand, the agents cannot follow the leader without the FTC controller (see Figure 5), since the tracking errors do not converge to zero. Therefore, the benefits of the FTC method can be clearly seen.

8. Conclusion

In this paper, we investigate the problem of a distributed FDI and FTC for a class of multi-agent uncertain systems. Under certain assumptions, adaptive thresholds are derived for distributed fault detection. Also, adaptive FTC controllers are developed to achieve the leader-following consensus in the
REFERENCES


Li, Z., Duan, Z., Chen, G., & Huang, L. (2010). Consensus of multi-agent systems and synchronization of complex


Mohsen Khalili received the B.S. degree from Bahonar University of Kerman, Kerman, Iran, the M.S. degree from Sharif University of Technology, Tehran, Iran, both in electrical engineering, in 2007 and 2009, respectively. He is currently a Ph.D. Candidate in Electrical Engineering department at Wright State University, OH, USA. His research interests are fault diagnosis, prognosis and fault-tolerant control of multi-agent systems.

Xiaodong Zhang received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from University of Cincinnati, Cincinnati, OH, USA, all in electrical engineering, in 1994, 1997 and 2001, respectively. He is currently an Associate Professor of Electrical Engineering Department, Wright State University, Dayton, OH. His research interests include intelligent control systems, fault diagnosis and prognosis, fault-tolerant control, and cooperative and distributed control. He is an Associate Editor of the IEEE Transactions on Control Systems Technology and a member of the IFAC SAFEPROCESS Technical Committee.

Yongcan Cao received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree from Utah State University, Logan, in 2010, all in electrical engineering. He worked as a Postdoc Researcher at Utah State University from September 2010 to September 2011 and a Postdoc Scholar at University of California, Riverside, from October 2011 to February 2012. He worked at the Air Force Research Laboratory, Wright-Patterson AFB, from March 2012 to July 2015. Since August 2015, he is an Assistant Professor in the Department of Electrical and Computer Engineering, University of Texas at San Antonio. His research interest focuses on autonomous vehicles, cyber-physical systems, sensor networks, and human-robot interaction. He is currently an Associate Editor on the IEEE Control Systems Society Conference Editorial Board.

Jonathan Muse received an undergraduate degree from the University of Alabama in 2005, a Masters Degree from the Georgia Institute of Technology in 2008, and a PhD in Aerospace Engineering from the Georgia Institute of Technology in 2010. He is currently a research engineer for the Aerospace Systems Directorate at the Air Force Research Lab (AFRL). At AFRL, he is responsible for executing basic research related to hypersonic vehicle control and advanced methods in nonlinear control. He is the Co-PI for HIFiRE flight 6, an adaptive flight control experiment on a scramjet hypersonic vehicle currently being constructed. He also holds an adjunct professor position at Wright State University in the Electrical Engineering department where he teaches control theory and UAV flight control. In addition, he also serves on various advisory boards for both industry and academia.
A Review of Prognostics and Health Management for Power Semiconductor Modules

Nicolas Degrenne¹, Jeffrey Ewanchuk, Erwan David, Roman Boldyrjew and Stefan Mollov

Mitsubishi Electric R&D Centre Europe, Rennes, 35708, France
¹n.degrenne@fr.merce.mee.com

ABSTRACT

In this paper, a review of current techniques used in the prognostics and state-of-health monitoring for power semiconductor modules is provided. Given the increasing trend in power modules having a larger share of the power electronic market, understanding their lifetime limitations is critical to improving the life-cycle cost of the power electronic product. Hence, this paper reviews common failure mechanisms in power modules and the state-of-art in predicting the lifetime of the module based on a given mission profile. Prognostics are reviewed in terms of stress-based and condition monitoring-based methods, while the potential of prognostics is presented for applications that utilize power modules.

1. INTRODUCTION

Owing to their lowered parasitic construction, high performance and relatively low manufacturing effort, power semiconductor modules are often preferred in applications where the power exceeds a few kilo-watts. This preference in higher power applications can be further appreciated by the fact that lower cost dies can be easily paralleled within a single power module to increase the rating of that module without having to manufacture a single large power die (Wintrich, Nicolai, Tursky & Reimann, 2011). Hence, since 2008, the market size for power modules occupies a larger proportion of the total market for power electronic devices - with a predicted market share of 30% by 2020 (Madjou, 2014).

Power modules are constructed with several mechanical layers, as seen in Figure 1. Commonly, the die is attached to a DBC (direct bonded copper), via solder, that is composed of a bonded copper-ceramic-copper structure, allowing electrical isolation but with high thermal conductivity towards the heatsink. In order to mate the DBC with the heatsink and increase the thermal capacity of the system, a base-plate is also typically used in medium and high power modules (Wintrich et al. 2011).

Furthermore, using a thermally conductive interface material, the module (base-plate-DBC-die) is attached to the heatsink, completing the thermal extraction path. In this paper, the presence of these layers within the power module is a critical aspect for determining its lifetime.

Depending on the application of a power converter, the power devices and the capacitors can exceed half the reported failures - with power devices being the largest single component (Yang, Bryant, Mawby, Xiang, Ran & Tavner, 2011). These failures are typically a result of environmental factors, overload conditions, and system transients (Yang et al. 2011). Hence, in difficult-to-service applications such as off-shore wind converters, there is an increasing amount of pressure to predict maintenance intervals and improve replacement costs by studying the reliability of power modules, (Ma, Liserre, Blaabjerg, & Kerekes, 2014). These same techniques have also found their way into automotive applications, where the application conditions vary widely and the reliability of the converter is less deterministic (Hirschmann, Tissen, Schröder & De Doncker, 2007). In both applications, the total lifecycle cost of the converter can be decreased with proper prediction of the lifetime. Hence, this paper provides a review of prognostic and health management techniques.
for power modules, with the aim of predicting the remaining useful life of the power module, as illustrated in figure 2.

As the chief aim of the paper is to review the methods relevant to lifetime prediction of power modules, section 2 of this paper covers the failure mechanisms present in typical IGBT-based power modules. In section 3, mission profile based lifetime estimation techniques are presented, with stress-based and condition monitoring-based prognostic methods highlighted in sections 4 and 5 respectively. Finally, section 6 details some potential applications for prognostics in power modules.

![Figure 2. Remaining useful lifetime of a power module](image)

### 2. Failure Mechanisms

The aging of devices is caused by different material properties of adjacent layers, as shown in Figure 1, specifically due to the different coefficients of thermal expansion (CTE) leading to a bimetal-like effect. Hence, during heat-up and cool-down phases the dissimilar expansion of adjacent layers causes a shear stress along the contact surface. Two principal failure mechanisms have been identified to be responsible for wear: bond wire lift-off and solder fatigue (Lutz, Schlangenotto, Scheuermann & De Donker, 2011) (Fig. 3). Reconstruction of metallization is mentioned in the literature as a third failure mechanism but it is not completely investigated yet (Lutz et al., 2011), (Wu, Held, Jacob, Scacco & Birolini, 1995), (Winrich et al. 2011).

#### 2.1. Bond wire lift-off

Bond wire is a cost-effective process to make an interconnection with the source, on the top of the die. Usually, bond wires are made of Al and have therefore a higher CTE value than Si dies. Hence bond wires expand more than the connected chip. As a consequence wires themselves are under stress caused by bending and the interface between bond and chip is under shear stress (Winrich et al. 2011).

Degradation starts usually at the edges of the interface and is presented as crack growth. The cracks generally spread around grain boundaries above the interface, in the wire bond. A reasonable explanation for this is that a softening and hardening occurs during the bonding process, leading to a higher reliability of the interconnection and a reduced reliability of the bond wire itself (Onuki, Koizumi & Suwa, 2000). Cracks grow towards the center of the interconnection, decreasing the contact area. Therefore the electrical resistance is increasing, measurable as a growing of forward voltage drop. Consequently, the power losses rise up leading to a higher junction temperature and power-on time. Eventually higher thermo-mechanical stress will finally lead to a bond wire lift-off. A single lift-off will accelerate the degradation process, because the full load current will increase the current per bond for the remaining wires leading to higher temperature at the interconnection (Goehre, Schneider-Ramelow, Geißler & Lang, 2010), (Lutz et al. 2011), (Onuki et al. 2000), (Bayerer, Herrmann, Licht, Lutz & Feller, 2008). The last bond wire carrying the highest current density typically leaves a crater just under its heel after lifting-off. Such a crater is usually a characteristic of an arc flash-over (Lutz et al. 2011). The lifetime of bond wires has increased lately thanks to better bond alloys, improved bonding technology and optimization of the bond wire geometry (Amro, Lutz, Rudzki, Sittin & Thoben, 2006).

#### 2.2. Solder delamination

Solder fatigue is the second most common aging mechanism in power modules. In popular semiconductor power modules there are two solder layers where solder fatigue occurs: within the solder between base-plate and DBC or chip and DBC. Stresses cause fractures formation inside the solder interfaces. With Pb-based solders, delamination starts usually in the edges and corners of the layer and is spreading towards centre. The thermal impedance is increased, thereby raising the junction temperature of the die. In this manner, higher thermo-mechanical stress accelerates the aging in a positive feedback loop (Lutz et al. 2011). During the last couple of years, Pb-free solder layers were investigated (Morozumi, Yamada, Miyasaka, Sumi & Seki, 2003), (Herrmann, Feller, Lutz, Bayerer & Licht, 2007). It was shown that with Sn/Ag solder, cracks originate in the center and spread in vertical and reticulated structures at tin grain boundary. It was also shown that Pb-free solder layers lead to higher reliability even if bond wires haven’t been improved (Hensler, Lutz, Thoben & Guth, 2010), indicating a cross-coupling reliability effect between the top- and bottom-side die terminations.
The aging of solder joints can be detected by scanning acoustic microscopy (SAM) and by calculating thermal resistance during the test. Usually, the end-of-life criterion from semiconductor power modules is reached when the thermal resistance has increased by more than the 20% (Hensler et al. 2010).

3. MISSION PROFILE BASED LIFETIME ESTIMATION

Design for reliability consists in designing a power module not only with respect to functional requirements but also to reliability requirements (Lu, Bailey & Yin, 2009). It requires estimating lifetime of the module under study, usually with minimum reliability figure in mind.

The lifetime estimation is generally obtained with a stepwise approach (Mainka, Thoben & Schilling 2011) from an estimation of the usage (mission profile). It employs an electrical, thermal and damage model of the Device Under Test (DUT) (Fig. 4). This approach assumes that the device will be used according to the pre-defined mission profile, and that the device assembly will endure a known number of stress cycles (Fig. 5), an assumption that is difficult to fulfill in reality.

A mission profile in automotive application can be derived from a speed profile representative of the average driving style. A collection of driving styles corresponding to different types of driving categories is available in the literature (Barlow, Latham, McCrae & Boulter 2009) such as FTP-72 (Hirschmann, Member, Tissen, Schröder & Doncker, 2007) or NEDC (Biela, Waffler, & Kolar, 2009) as shown in Fig. 6.

The loss profile of each die in the power module is translated from the mission profile by considering traction chain characteristics and various parameters related to the motor and the converter (Thoben, 2008). A thermal model is then applied and results in a junction temperature profile $T_J$ (Lu et al., 2009)(Fig. 6). Note that $T_J$ is fed back in the calculation of the power losses.

3.2. FROM MISSION PROFILE TO JUNCTION TEMPERATURE PROFILE

A counting method is used to extract and classify the thermal cycles. The rainflow algorithm is able to capture compound temperature variations, and is considered to be amongst the better performing algorithms (Mainka et al. 2011).

The stresses are converted into damage with a damage model. This model is either derived from physical models (Kovacevic, Drofenik & Kolar, 2010) or with data-driven approach using power cycling tests (Scheuermann & Schmidt, 1997). Different models are commonly used such as Coffin-Manson (Halford & Manson, 1967), LESIT (Held, Jacob, Nicoletti, Scacco & Poech, 1997), CIPS08 (Bayerer, Hermann, Licht, Lutz & Feller, 2008).

The damages are then summed using Palmgren-Miner linear accumulation rule (Miner, 1945), though this is increasingly questioned (Aal, 2014). In (Huang & Mawby, 2013), the damage level is fed-back to the thermal model. The lifetime is calculated based on the duration of the mission profile and the corresponding computed damage. Based on the lifetime estimation, the design can be adjusted to meet the reliability requirements.
4. STRESS-BASED PROGNOSTICS

Analyzing a pre-defined mission profile provides insight into potential reliability problems that may or may not develop during the life of a product. Although such insight is beneficial during the design and the operational phases of the product, the information quickly becomes obsolete once the product usage (e.g. operating and environmental characteristics) deviates significantly from the assumed operating conditions that have been originally analysed. That is why monitoring of real-life operating and environmental conditions is used to derive the reliability problems based on the actual operating and environmental conditions.

Stress-based methods quantify the cause of degradation: stress (Fig. 7). They are most adequate when the manufacturing process deviations (i.e. strength range) are narrow and when the usage cannot be easily pre-defined or has large deviations (Fig. 8).

In order to estimate the stress, the methods relying on power and thermal models (as presented in section 3) can be implemented in real-time. Alternatively, the junction temperature $T_J$ can be measured on-line. Then, the stress is quantified with a real-time algorithm and converted to a State-of-Health (SoH). The SoH is extrapolated to estimate the End-of-Life (EoL) and the Remaining Useful Life (RUL).

4.1. Junction temperature measurement

A first approach is to position a conventional thermal sensor in the vicinity of the power chip. Commercially available negative temperature coefficient (NTC) thermistors operate up to 200°C, and can be attached with standard processes (e.g. sintering, wire-bonding). This approach measures the temperature several millimeters away from the junction, and has a low space and time resolution, the relevance of the provided information is thus questionable.

A second approach is to create a temperature sensor on the silicon power die itself. In a patent, Schuler (2011) develops the idea of a thermocouple circuit formed by two bonding wires and a pad to measure the temperature difference between the chip and the carrier. In (Motto & Donlon 2012), a string of diodes is fabricated on the IGBT chip’s surface (Fig. 9). The linear dependence of the forward voltage drop is used to estimate temperature. This solution has the required performance but is not cost-effective as it occupies a valuable real estate on the power die.

A third approach is to monitor Thermo-Sensitive Electrical Parameters (TSEP), inherent to the semiconductor device (Avenas, Dupont & Khatir, 2012), (Baker, Liserre, Dupont & Avenas, 2014). Some TSEPs such as voltage drop at low current and saturation current have to be done in specific electrical conditions and often necessitate an alteration to the structure or operation of a power electronic converter. Table 1 compares methods that can be performed online, either during the conduction time or the switching time. All these methods require accurate sensors and chip-level calibration. They must be compared in terms of linearity, dependence in operating conditions outside of temperature, sensitivity and implementation cost. Converter-level methods such as low-order harmonics identification in an inverter (Xiang, Ran, Tavner, Yang, Bryant & Mawby, 2012) are not included in the table.
4.2. Real-time stress-counting algorithms

The rainflow algorithm traditionally uses the entire time history of the junction temperature. This approach is inconvenient in real-time applications because the algorithm must be applied periodically to large datasets. A first option is to use the rainflow algorithm on windows of data of e.g. 1 day in order to reduce the size of the dataset. A second option is to use another type of stress-counting algorithm.

Three alternative real-time stress-counting algorithms are evaluated with respect to wire-bond and chip solder failures in (Mainka et al. 2011). The rising edge and half-cycle methods process each minima and maxima value as it occurs and give acceptable results. An on-line stress-counting algorithm is used in (Musallam, Johnson, Yin, Bailey & Mermet-Guyennet, 2010) and (James 2012), but the performances of the algorithms are not evaluated experimentally.

<table>
<thead>
<tr>
<th>TSEP</th>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage at high current of transistors and diodes</td>
<td>(Kim &amp; Sul, 1998) (Perpiñà, Serviere &amp; Saiz, 2006) (Koenig &amp; Plum, 2007) (Lutz, Paul &amp; Zill, 2011) (Dupont, Avenas &amp; Jeannin 2013)</td>
<td>Requires very accurate sensors and high-voltage protection. Sensitivity of approx. 20mV/°C. Measurement influenced greatly by parasitic elements inside power modules (bond wires, etc.), and load current. Is also a DSEP (Table 2).</td>
</tr>
<tr>
<td>Turn-off transition time of IGBT transistors</td>
<td>(Barlini &amp; Ciappa, 2006) (Kuhn &amp; Mertens, 2009) (Bryant, Yang, Mawby, Xiang, Ran, Tavner &amp; Palmer, 2011) (Xu, Jiang, Li &amp; Ning, 2013)</td>
<td>Sensitivity of approx. 2ns/K. Changes have also been viewed in harmonics in the output of an IGBT inverter.</td>
</tr>
<tr>
<td>Turn-on delay and/or di/dt of transistors</td>
<td>(Barlini &amp; Ciappa, 2007) (Kuhn &amp; Mertens, 2009) (Sundaramoorthy, Bianda, Bloch &amp; Zarfluh, 2014)</td>
<td>Increase in gate resistance during the measurement is proposed to slow down the process. Sensitivity of approx. 2ns/K or 40A/(µs.K) for high power IGBTs.</td>
</tr>
<tr>
<td>Threshold voltage of MOS transistors</td>
<td>(Chen, Pickert, Atkinson &amp; Pritchard, 2006) (Bahun, Sunde &amp; Jakopovic, 2013)</td>
<td>The sensing circuit can be implemented in the gate driver. The threshold voltage is measured when the current starts flowing through the transistor. Sensitivity of approx. 10mV/K.</td>
</tr>
<tr>
<td>Gate parasitic (internal resistance and capacitances)</td>
<td>(Sundaramoorthy, Bianda, Bloch, Nistor, Knapp &amp; Heinemann, 2013) (Baker, Munk-Nielsen, Liserre &amp; Iannuzzo, 2014) (Niu &amp; Lorenz, 2014) (Niu &amp; Lorenz, 2015)</td>
<td>The sensing circuit can be implemented in the gate driver. The sensed parameter can either be the duration of the Miller plateau phase (approx. 1.2ns/K), the voltage during turn-on delay (approx. 20mV/K), the gate charge (approx. 250pC/K) or the gate current transient such as the peak current (approx. 2.5mA/K).</td>
</tr>
</tbody>
</table>

4.3. State-of-Health extrapolation

The time history of the state of health is required to determine the RUL of a device. It is linearly extrapolated according to the following equation (Fig. 10):

\[ RUL_n = \frac{n \times SoH_n}{100 - SoH_n} \]

with \( n \) the time variable, \( SoH \) the state of health (in %).

5. CONDITION-MONITORING PROGNOSTICS

Unlike stress-based methods, condition-monitoring (CM) methods examine the consequence of degradation, i.e. the evolution in time of one or several Damage-Sensitive Electrical Parameters (DSEP), also mentioned as failure precursors (Fig. 11). As such, CM methods are useful in applications where devices present variations due to manufacturing process dispersion (Fig. 12).
The selection of a DSEP depends on the preferred failure mode and on the ease of implementation. In order to convert the deviation of the DSEP into a RUL estimate, a threshold value is determined, and the time deviation is extrapolated (Fig. 11). Two approaches are opposed in the literature: (a) past-history approaches that only use past DSEP data on the DUT itself, and (b) model-based approaches that use data from other devices to generate a model.

5.1. Damage Sensitive Electrical Parameters

Table 2 sums up the DSEP parameters that have been treated and reviewed (Yang, Xiang, Bryant, Mawby, Ran & Tavner, 2010) in the literature. In the case of solder degradation, all methods discussed in paragraph 3.1 to estimate the junction temperature are useful to estimate the value of $R_{TH}$. In the case of wire-bond failure (lift-off), the electrical path is suddenly degraded, and the voltage at high current experiences an abrupt, distinguishable increase. $T_I$ increases as well, as a consequence to the loss increase. In order to isolate the impact of bond-wire degradation from solder-crack degradation, the impact of temperature on the voltage can be compensated (Rashed, Forest, Huselstein, Martire & Enrici, 2013), (Ji, Pickert, Cao & Zahawi, 2013).

Note that on-line measurement may not be a requirement anymore for CM since it is possible to monitor health discontinuously, before or after the operation of the converter. In his thesis, Ji (2011) performs wire-bond and solder condition monitoring of an automotive inverter employing a calibrated operating point sequence taking approx. 5s. However, applications permitting such calibration procedures are rare in practice.

5.2. DUT past-history approach

The prognostic algorithm only uses the past history of the device to estimate its health and the RUL. The extrapolation typically uses a regression framework based on either statistical or Markov chains.

In (Celaya, Saxena, Saha, Vashchenko & Goebel, 2011), a Gaussian Process Regression (GPR) is used on discrete power MOSFETs. A distribution is tuned to fit available measurements, used to output a mean and covariance function, and to predict mean value and corresponding variance for a future point of interest (Fig. 13).

Past-history approach offers the advantage of requiring no training data set from other devices, even if domain knowledge is necessary to define the prior distribution and the type of covariance function, and to predict mean value and corresponding variance for a future point of interest (Fig. 13).

Figure 11. RUL estimation based on Damage-Sensitive Electro Parameter (DSEP)

Figure 12. Condition monitoring can be used in applications where usage and/or strength is not well identified.

The selection of a DSEP depends on the preferred failure mode and on the ease of implementation. In order to convert the deviation of the DSEP into a RUL estimate, a threshold value is determined, and the time deviation is extrapolated (Fig. 11). Two approaches are opposed in the literature: (a) past-history approaches that only use past DSEP data on the DUT itself, and (b) model-based approaches that use data from other devices to generate a model.
Table 2. Damage-Sensitive Electrical Parameters (DSEP)

<table>
<thead>
<tr>
<th>DSEP</th>
<th>Reference</th>
<th>Failure mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{TH}$-based methods</td>
<td>(Ji, Song, Cao, Pickert, Hu, Mackersie &amp; Pierce, 2014)</td>
<td>Chip solder</td>
</tr>
<tr>
<td></td>
<td>(Saha, Celaya, Wysocki &amp; Goebel, 2009)</td>
<td>DBC solder</td>
</tr>
<tr>
<td>Voltage at high current</td>
<td>(Celaya, Saxena, Wysocki, Saha &amp; Goebel, 2010)</td>
<td>Wire-bond</td>
</tr>
<tr>
<td></td>
<td>(Ji, Pickert, Cao &amp; Zahawi, 2013)</td>
<td>Reconstruction of</td>
</tr>
<tr>
<td></td>
<td>(Smet, Forest, Rached &amp; Richardau, 2012)</td>
<td>metalization</td>
</tr>
<tr>
<td></td>
<td>(Beczkowski, Ghimre, De Vega, Munk-Nielsen, Rannestad &amp; Thogersen, 2013)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Ghimire, De Vega, Munk-Nielsen, Rannestad &amp; Thogersen 2013)</td>
<td></td>
</tr>
</tbody>
</table>

5.3. Model-based approach

In the model-based approach, a training data set from other devices is used to model the evolution of the DSEP with time. The model can either be empirical or physical and is established after accelerated aging experiments on other devices and/or simulations. The prognostic algorithm uses the model and the measurements on the DUT to estimate the health and the RUL. Prognostic methods used in the literature are mainly Extended Kalman Filter (EKF) and Particle Filter (PF) (Baraldisi, Maio & Zio, 2014).

In (Celaya, Saxena, Saha & Goebel, 2011), (Patil, Das & Pecht, 2012), and (Saha, Celaya, Vashchenko, mahiuddin & Goebel, 2011), several aged discrete devices are used to fit an empirical model of the evolution of $R_{DS,ON}$, $V_{CE,ON}$, and $V_{TH}$ respectively. The observations on the DUT are related to the empirical models with a particle filter to estimate the degradation state. In (Alghassi, Perimpanayagam & Samie, 2015), an alternative to EKF and PF is proposed. The $V_{CE,ON}$ value is decomposed into 9 discrete steps. A training dataset of 3 discrete IGBTs is used to provide a statistical distribution for the duration of each state. It is then used with Monte-Carlo simulations to investigate a statistical RUL prediction. The result is a light-weight simulation-based prognostic approach requiring only 0.3ms computation time for each measurement.

6. POTENTIAL OF PROGNOSTICS FOR POWER MODULES

The cost associated with the implementation of prognostic means in a power module has to be counterbalanced with quantifiable benefits. First, the prognostic information can be used in order to develop cost-optimized minimum-intervention and just-in-time maintenance strategies. Second, active health management can be employed to redistribute the stress to extend the life time of a device based on prognostic information.

6.1. Asset Management

Two asset-management strategies are commonly used in the field of power electronics: replacement after failure, and replacement at fixed intervals (in mission critical applications). Prognostic is a key enabler to asset management that ensures a good compromise between the risk of failure and the return on investment.

The development of a business model is necessary to support asset management. Return on investment calculations were applied to various electronic products (Sandborn & Wilkinson, 2007), (Feldman, 2009) and (Haddad, Sandborn & Pecht, 2014) but no business case was published for power modules until now. The main difficulty of establishing a business model is to estimate all costs additions (non-recurring, recurring, infrastructure) and cost avoidance (failure cost, maintenance cost) that are used by the stochastic simulator.

Manufacturers of power modules are facing problems to provide users with a year-based warranty. This is because of the considerable impact of system integration, operating and environmental conditions. Manufacturers would more easily provide a warranty based on applied stressors, such as number of junction temperature cycles.

6.2. Active stress reduction or redistribution

Stress reduction limits the performances of the system in order to extend the life time. The most straightforward implementation consists in de-rating (load current, operating voltage) of the module. This approach increases the system costs, and other solutions are preferred (Table 3). Several methods are commonly combined to offer various degrees of lifetime control. These methods can be used as a protection for over-temperature $T_{MAX}$ and as a mean to decrease power cycling stress, reducing the $\Delta T_j$ around $T_{J,MEAN}$ value. Another type of active control consists in controlling the minimum case temperature for the power module. Hence, as the fluctuations of junction temperature are reduced (Davidson, Stone & Foster, 2014) the lifetime of the module should increase, but no study was yet performed to evaluate CTE mismatch decrease and lifetime gain. Lastly, while the redistribution of stress between distinct power modules is not covered by this paper, some methods have been demonstrated for actively controlling parallel power modules (Hofer, Karrer and Gerster, 1996). However, it should be noted that for the latter method, the aim is to equalize the current distribution across the parallel set of modules, rather than to directly control the temperature to increase the lifetime of the module.
7. Conclusion

As a result of increasing market penetration of power modules, the motivation to better understand their lifetime and health is paramount to continuing this trend. Hence, this paper presented a review of prognostic methods, starting with failure mechanisms and continuing with methods of counting stress or directly measuring the state of health.

References


Table 3. Thermal and lifetime control techniques

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Downside</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load current</td>
<td>Decrease conduction losses</td>
<td>Load current decrease</td>
<td>(Blasko, Lukaszewski &amp; Sladky, 1999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Murdock, Torres &amp; Connors, 2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Lemmens, Driesen &amp; Vanassche, 2012)</td>
</tr>
<tr>
<td>Switching frequency</td>
<td>Decrease switching losses</td>
<td>Load ripple increase</td>
<td>(Blasko et al. 1999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Murdock et al. 2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Lemmens et al. 2012)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Weckert &amp; Roth-Stielow, 2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Wei, McGuire &amp; Lukaszewski, 2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Lo Calzo, Lidozzi, Solero, Crescimbini &amp;ardi, 2012)</td>
</tr>
<tr>
<td>DC-link voltage regulation</td>
<td>Decrease conduction losses</td>
<td>Oversized actives and passives, risk for random failures</td>
<td>(Andersen &amp; Liserre, 2014)</td>
</tr>
<tr>
<td>Modulation strategy</td>
<td>Modify modulation type</td>
<td>EMI difficult to control</td>
<td>(Lo Calzo et al. 2012)</td>
</tr>
<tr>
<td></td>
<td>(continuous/discontinuous)</td>
<td></td>
<td>(Ma &amp; Blaabjerg, 2014)</td>
</tr>
<tr>
<td>Active gate driving</td>
<td>Increase the gate voltage</td>
<td>Potential gate oxide degradation</td>
<td>(Wu &amp; Castellazzi 2010)</td>
</tr>
<tr>
<td>Heatsink</td>
<td>Increase cooling effort</td>
<td>Oversized cooling system</td>
<td>(Law &amp; Harley 2004)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Efficacy not proven</td>
<td>(James, 2012)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(De Rijck &amp; Huisman, 2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Davidson et al. 2014)</td>
</tr>
</tbody>
</table>
Modules - various factors influencing lifetime. In *CIPS 2008*.


Ji, B., Song, X., Cao, W., Pickert, V., Hu, Y., Mackersie, J., & Pierce, G. (2014). In-Situ Diagnostics and Prognostics of Solder Fatigue in IGBT Modules for...


252
A Self-Organizing Map-Based Monitoring System for Insulated Gate Bipolar Transistors Operating in Fully Electric Vehicle

Marco Rigamonti¹, Piero Baraldi¹, Enrico Zio¹,², Allegra Alessi¹, Daniel Astigarraga³, and Ainhoa Galarza³

¹Energy Department, Politecnico di Milano, Via Ponzio 34/3, Milan, 20133, Italy
marcomichael.rigamonti@polimi.it
piero.baraldi@polimi.it
enrico.zio@polimi.it
allegra.alessi@polimi.it

²Chair on Systems Science and the Energetic challenge, European Foundation for New Energy-Electricite’ de France, Ecole Centrale Paris and Supelec, Paris, France
enrico.zio@ecp.fr

³CEIT, Manuel de Lardizabal 15, San Sebastian, 20018, Spain
dastigarraga@ceit.es
agalarza@ceit.es

ABSTRACT

Insulated Gate Bipolar Transistors (IGBTs) are one of the most used power semiconductor devices for energy conversion applications, due to their high performance. However, recent studies have shown that IGBT malfunctioning are responsible of several industrial failures: 38% of unscheduled maintenance actions in variable speed AC drives (Shaoyong, Bryant, Mawby, P., Dawei, Li, and Tavner, 2011) and 35% of power electronic systems faults are caused by IGBTs (Fuchs, 2003; Hudgins, 2013).

Condition-Based Monitoring (CBM) techniques have been developed over the last decade for IGBT monitoring (Lu & Sharma, 2009; Oh, Han, McCluskey, Han, and Youn, 2015). In Chokhawala and Kiraly (1995), IGBT degradation due to open- and short-circuit faults is considered, whereas in Ji, Pickert, Cao, and Zahawi (2013) and Smet, Forest, Huselstein, Rashed, and Richardneau (2013), the authors focus on IGBT degradation caused by wire bond faults. Although the proposed methods are efficient for monitoring IGBT degradation when it is caused by a single degradation mechanism, they are not apt for an overall monitoring of the component which is typically characterized by multiple and competing degradation mechanisms.

The purpose of this work is to develop a method for the identification of the degradation state of IGBTs operating in Fully Electrical Vehicle (FEV) powertrains. The final objective is to develop an automatic system able to inform the FEV driver of the IGBT degradation state and, eventually, of the necessity of performing maintenance. The proposed monitoring system is expected to increase the safety of the FEV and to reduce the overall maintenance costs.
The main difficulties to be addressed in order to effectively monitor FEV IGBTs are:

i. several signals usually employed to monitor IGBT degradation, such as the current at the collector or the transconductance (Patil, Das, Goebel, & Pecht, 2008) are not measurable during FEV operation, due to the intrusiveness of the measurement device;

ii. laboratories tests performed within the European FP7 project HEMIS (Electrical powertrain Health Monitoring for Increased Safety of FEVs) at CEIT laboratories (Centro de Estudios e Investigaciones Técnicas – San Sebastian, Spain) have shown a great variability in the degradation behavior of different IGBTs, even if they are degrading in the same controlled test conditions. This is due to the fact that IGBTs are subject to different, possibly interacting and competing degradation mechanisms, such as bond wire lift-off, solder joint fatigue, and bond wire heel cracking (Busca, Teodorescu, Blaabjerg, Munk-Nielsen, Helle, Abeyasekera, & Rodriguez, 2011).

iii. FEV IGBTs are operating under continuously varying conditions. In particular, FEV speed and motor load variations cause modifications of the power, temperature and currents experienced by the IGBTs. This complicates the diagnostic task since the variations of the signals due to the degradation process are small if compared to those caused by the variations of the operational conditions.

In order to overtake the difficulty in i), we consider the possibility of monitoring the IGBTs by using only signals which can be measured on FEV IGBTs, such as the Case temperature, T, the collector-emitter voltage, \( V_{CE} \), and the phase current, \( I_p \). In order to measure the \( V_{CE} \) when the inverter is connected to an electric motor, we have used the new IR25750 chip developed by International Rectifier (IR).

The approach developed in this work for dealing with the inhomogeneous situation described in ii) and iii) is based on the use of Self-Organizing Maps (SOMs), which allow representing and clustering multidimensional data into a two-dimensional space (Kohonen, 2005; Gonçalves, Schneider, Henrques, Lubaszewski, Bosa, & Engel, 2010). A SOM is trained using signal measurements collected from healthy IGBTs and a degradation indicator is defined by considering the distance between the measured signal values and the corresponding SOM best matching unit (BMU).

Finally, a method for setting the thresholds to be applied to the degradation indicator in order to classify the component degradation state is proposed. It is based on the identification of an optimal trade-off between degradation state misclassifications resulting in false and missed alarms, through the definition of a proper utility function which quantified the consequences of false and missed alarms in terms of safety and costs.

The proposed approach has been verified considering experimental data collected at the Centro de Estudios e Investigaciones Técnicas (CEIT, San Sebastian, Spain) from an inverter providing the required three phases AC current to an electric power train. The remaining part of the paper is structured as follows: Section 2 identifies the problem statement and the aim of the methodology; Section 3 illustrates the data pre-processing and the method; Section 4 presents the experimental dataset and describes the data collection process; Section 5 discusses the application of the developed monitoring system; finally, Section 6 recalls the concluding remarks and results.

2. Problem Statement

The purpose of this work is to develop a method for the online identification of the degradation state of IGBTs working under continuously varying operating conditions, such as those characteristics of powertrains used in FEVs. The output of the monitoring system is expected to be one of the following three degradation classes: healthy (no need of maintenance), partially degraded (the component can still work, but a warning should be provided), and very degraded (maintenance is necessary in order to avoid the component failure). The information available for the development of the monitoring system is made by the measurements of \( S \) signals performed on \( C \) different IGBTs, characterized by different levels of degradation. The \( S \) signal values measured from the \( c \)-th IGBT, \( c = 1, 2, ..., C \), at the generic time \( \tau \), will be indicated by the vector \( \hat{x}^c(\tau) \) formed by the signal values \( x^c_1(\tau), ..., x^c_S(\tau) \).

The IGBTs considered in this work have undertaken an accelerated degradation process characterized by a series of prolonged on-cycles, which cause thermomechanical stresses and accelerate their degradation. For each one of the \( C \) IGBTs, we know the number of accelerated aging cycles performed at the time in which the signals have been measured. Notice that, given the stochasticity of the degradation process, this information is not directly related to the real degradation state of the component.

3. Method

In this Section the proposed approach is described; the SOM basic concepts are recalled in Section 3.1, the degradation indicator construction is described in Section 3.2, Section 3.3 presents the data preprocessing procedure, and the developed strategy for setting the method parameters is discussed in Section 3.4.

3.1. Self Organizing Maps

A SOM is a neural network concept, used to classify and cluster \( S \)-dimensional vectors in a visually simple 2-dimensional lattice. It is formed by an array of \( L \) neurons, or map units, each one represented by a characteristic \( S\)-
dimensional vector \( \mathbf{w}^\ell = [w_1^\ell, ..., w_L^\ell] \), \( \ell = 1,...L \) known as weight vector. Each neuron is connected to the other neurons of the map by a relationship based on a neighborhood function (Kohonen, 2005).

\[
U_{\text{dist}}^{i_1,i_2} = \sqrt{(w_{i_1}^{t_1} - w_{i_2}^{t_2})^2 + (w_{i_1}^{t_2} - w_{i_2}^{t_1})^2 + ... + (w_{i_L}^{t_1} - w_{i_L}^{t_2})^2}
\]

(2)

The values of the unified distance provide a representation of how similar are the neighboring neurons of a SOM (Figure 2).

Figure 1 shows a representation of a SOM where each neuron is connected to its adjacent neurons and to the input vector. Before the training process, the neurons are properly initialized according to the procedure suggested in (Kohonen, 2005): the weight vectors are selected as a regular array of values between the two largest eigenvectors of the training data. Then, during the training process, the generic \( r \)-th training step is based on:

1. a sample vector, \( \mathbf{y^{Training}} \), is randomly selected from the training dataset and its distance to the weight vectors of all the SOM neurons is computed;
2. the nearest neuron is identified. It will be referred to as Best Matching Unit (BMU);
3. the weight vector of the BMU and its neighbor vectors are updated in order to obtain weights more similar to that of the chosen random sample vector \( \mathbf{y^{Training}} \). Weight vector updating between training step \( r \) and \( r+1 \) is performed by applying:

\[
h_nBMU(n_i,r) = \frac{1}{h(nBMU,n_i,r)}(\mathbf{y^{Training}} - \mathbf{w}_i(r))
\]

where \( h(nBMU,n_i,r) \) is the neighborhood function between the best matching neuron \( nBMU \) and the \( \ell \)-th neighboring neuron, \( n_{i_{\ell}} \), and \( \alpha(r) \) is the learning rate, which decreases at each training step.

Once the training phase is terminated, the SOM structure is caught by the unified distance matrix, \( U_{\text{dist}} \), whose generic element \( U_{\text{dist}}^{i_1,i_2} \) is defined as the Euclidean distance between the \( S \)-dimensional weight vectors \( \mathbf{w}^{i_1} \) and \( \mathbf{w}^{i_2} \), of the corresponding to neurons \( i_1 \) and \( i_2 \):

Notice that clusters formed by neurons characterized by small inter-neuron distances can be easily identified from the observation of the U matrix. In this work, a SOM is trained using data collected from healthy IGBTs. The obtained SOM provides a two-dimensional representation of the training data which minimizes the influence of outliers and noisy data, and captures the characteristic behavior of a healthy component.

3.2. Degradation indicator

In order to identify the degradation state of a monitored IGBT which will be identified by the letter \( \epsilon^{BMU} \), we provide in input to the trained SOM the measured signal values, \( \mathbf{y^{cTest}} \), and we compute the Euclidean distance between the input vector and the corresponding SOM BMU, \( MQE(\mathbf{y^{cTest}}) \):

\[
MQE(\mathbf{y^{cTest}}) = \sqrt{(y_1^{cTest} - w_{BMU}^{BMU})^2 + ... + (y_S^{cTest} - w_S^{BMU})^2}
\]

(3)

This distance, which is referred to as Minimum Quantization Error (MQE), indicates how much the vector is different from the behavior represented by the data used for training the SOM (Qi, & Lee, 2004; Huang, Xi, Li, Richard Liu, Qi, & Lee, 2007), and can be interpreted as an indicator of the component degradation. Greater the MQE, more the component is behaving differently from an healthy one and, therefore, more the component is degraded.

In this work, the indicator of the component degradation, \( QE(\mathbf{y^{cTest}}) \), is defined as the minimum quantization error,
\( MQE(\text{Test}) \), divided by the average quantization error, \( MQE_{\text{healthy}} \), of a validation set made by healthy data:

\[
QE(\text{Test}) = \frac{MQE(\text{Test})}{MQE_{\text{healthy}}}
\]

with:

\[
MQE_{\text{healthy}} = \sqrt{\frac{1}{N_{\text{healthy}}} \sum_{c=1}^{3} \left| MQE(\text{Test}) \right|^2}
\]

This normalization allows obtaining a baseline reference value: healthy component will be characterized by a degradation indicator close to 1.

### 3.3. Data Pre-Processing

The construction of the SOM is preceded by a phase of data preprocessing based on the following two steps:

1. the selection of data in a predefined range of values. In practice, in order to deal with the great variability of the signals in a FEV motor, the model is trained and tested by considering patterns \( \tilde{x}(t) = [I_i(t) \ I_e(t) \ Vce(t)] \) characterized by all signal values within properly selected ranges, i.e., whose phase current, \( I_o \) is in the range \([ I_{\text{LowerLimit}}; I_{\text{UpperLimit}} ]\), and whose collector-emitter voltage is in the range \([Vce_{\text{LowerLimit}}; Vce_{\text{UpperLimit}}]\).
2) the computation of the moving average of the signals. This step is performed in order to reduce the impact of the measurement noise on the signal values. The lengths, \( L_o \), of the moving average window will be selected through the optimization procedure described in Section 3.4.

### 3.4. Degradation State Identification

The robustness of the method is improved by considering as indicator of the IGBT degradation the median \( QE(\text{Test}) \) of a number \( L_o \) of consecutive values obtained from the SOM. The median has been chosen since it is more stable than the mean value in case of outliers.

The classification of the IGBT degradation state is then based on the definition of two thresholds \( T_{h_{1,2}} \) and \( T_{h_{2,3}} \) according to the following rules:

- if \( QE(\text{Test}) < T_{h_{1,2}} \), then the IGBT is healthy (class 1)
- if \( T_{h_{1,2}} \leq QE(\text{Test}) \leq T_{h_{2,3}} \), then the IGBT is partially degraded (class 2)
- if \( QE(\text{Test}) > T_{h_{2,3}} \), then the IGBT is very degraded (class 3)

### 3.5. Setting of the Method Parameters

The proposed method is based on the following four parameters: i) the length, \( L_o \), of the moving-average window used in the pre-processing phase, ii) the number, \( L_o \), of QE consecutive values which are considered for calculating the median of the degradation indicator, iii) the threshold values, \( T_{h_{1,2}} \) and \( T_{h_{2,3}} \), used for the classification of the degradation state.

The setting of these parameters is performed considering a set of data (hereafter indicated by “optimization set”), taken from IGBTs not considered for the SOM training and for which the degradation state is known. The objective of the parameter setting is the minimization of the misclassification rates, i.e., the fraction of patterns assigned to an incorrect classification state. To this purpose, the numbers of patterns, \( n_{i,j} \), of the optimization set whose correct class is \( i \) and are misclassified in class \( j \) are found for any combination of \( i \) and \( j \) with \( i \neq j \). Then, the fraction of misclassifications, \( \alpha_{i,j} = \frac{n_{i,j}}{N_i} \), with \( N_i \) indicating the total number of patterns in the optimization set of class \( i \) is identified and the overall performance of a given parameters quadruplet \((L_o, L_o, T_{h_{1,2}}, T_{h_{2,3}})\) is defined by the utility function:

\[
P = \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_{i,j} \cdot IF_{i,j}
\]

where \( IF_{i,j} \) is a coefficient quantifying the consequences of the misclassification of a pattern whose true class is \( i \) and is assigned to class \( j \) in terms of safety and availability of the component. In this work, we assume that the most undesirable event, which can lead to the failure of the component, is that an IGBT whose true state is very degraded (class 3) is classified as healthy (class 1) or partially degraded (class 2). Thus, the highest impact factor is assigned to the coefficients \( IF_{3,1} \) and \( IF_{3,2} \) (Table 2). Since a preventive maintenance action is suggested only if the degradation state reaches “very degraded” (class 3), the misclassifications with the less remarkable consequences are those between classes 1 and 2. For this reason, the lowest impact factors are assigned to \( IF_{1,2} \) and \( IF_{2,1} \). An intermediate impact factor value is assigned to the misclassifications which causes an unnecessary preventive maintenance action, i.e. when patterns of classes 1 and 2 are assigned to class 3.
Table 1. Impact factors for the calculation of parameter $P$

<table>
<thead>
<tr>
<th>$\alpha_{i,j}$</th>
<th>$IF_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{1,2}$</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_{1,3}$</td>
<td>2</td>
</tr>
<tr>
<td>$\alpha_{2,1}$</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_{2,3}$</td>
<td>2</td>
</tr>
<tr>
<td>$\alpha_{3,1}$</td>
<td>5</td>
</tr>
<tr>
<td>$\alpha_{3,2}$</td>
<td>5</td>
</tr>
</tbody>
</table>

The optimal values of the parameters $L$, $L_m$, $T_{h_{1,2}}$, $T_{h_{2,3}}$ are identified by following a trail-and-error procedure, where different values of the four parameters are tested and the corresponding value of the utility function $P$ is computed. The quadruplet with the associated lowest value of $P$ is selected. Notice that, although the specific values of the coefficients $IF_{i,j}$ depend on the characteristics of the monitored component and the opinion of the expert, the proposed method for the parameter setting is general.

4. CASE STUDY

The method described in Section 3 has been verified with respect to data collected in experimental tests performed on degraded FEV IGBTs at Centro de Estudios e Investigaciones Técnicas (CEIT).

Since the average mean time to failure of an IGBT is around several thousand hours, IGBTs have been degraded by applying an accelerated aging process based on thermal fatigue cycles inducing mechanical deformations on the solder joints, that, in turn, cause an accumulation of microcracking and damage (Thébaud, Woirgard, Zardini, Azzopardi, Briat, & Vinassa, 2000). The IGBTs were connected to a generator, producing a direct current of 5A, and were kept closed (i.e. turned on) as long as the junction temperature reached 270°C. Once this temperature was reached, they were opened (i.e. turned off) until the temperature reached 258°C, and a new degradation cycle begun.

Figure 3 shows the behavior of the collector current ($I_c$), the collector-emitter voltage ($V_{ce}$) and the junction temperature ($T$) during the degradation cycles. Notice that these laboratory data are not used in this work for the development of the monitoring system, since they do not refer to IGBTs working in FEV inverters.

In order to obtain the data necessary for the development of the monitoring system and its verification, 6 IGBTs have been degraded for a different number of cycles:

- 2 IGBTs were aged for 900 cycles (they will be referred to as IGBT A and B);
- 2 IGBTs were aged for 1800 cycles (they will be referred to as IGBT C and D);
- 2 IGBTs were aged for 2700 cycles (they will be referred to as IGBT E and F).

Then, each degraded IGBT has been mounted on an inverter connected to a powertrain and the typical conditions of the IGBT operation in FEVs have been reproduced. In practice, each experiment has been carried out with the powertrain subject to a constant load of 1kW, a 9Nm torque and operating at an average speed of 400rpm. Each experiment lasted on average 3 seconds, producing between 200000 and 380000 measurements. Stator phase current, collector-emitter voltage and the inverter case temperature have been measured at a frequency of 80kHz using low cost sensors which can be easily installed in FEV inverters. With respect to the collector-emitter voltage sensor, notice that it records 0.2V when the IGBT is off.

Figure 4 shows an example of the three signal measurements collected during the test of an IGBT aged for 900 cycles. Due to the high switching frequency of the IGBT in a FEV inverter, only few measurements are collected for each on cycle of the IGBT. Thus, the obtained data are very different from those typically used in literature (Patil et al., 2008) which refers to IGBT in laboratory tests and are similar to those of Figure 3.
Figure 4. Time evolution of $T$, $I_p$ and $V_{ce}$ during a 0.02s otor test.

5. RESULTS

5.1. Data Preprocessing

The data acquired during the motor tests are characterized by a great variability of the measured signals. Since the methodology developed in this paper aims at assessing the IGBT level of degradation relying on the quantification of the deviation of the monitored component from its corresponding normal healthy state, it is necessary to limit the analysis on a proper operation region. In practice, the monitoring system is trained and tested only considering patterns whose phase current, $I_p$, and collector-emitter voltage, $V_{ce}$, are in the range reported in Table 2.

Table 2. Ranges used for data selection

<table>
<thead>
<tr>
<th>Variable</th>
<th>Lower Limit</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_p$ [A]</td>
<td>-7.05</td>
<td>-6.95</td>
</tr>
<tr>
<td>$V_{ce}$ [V]</td>
<td>1.1</td>
<td>1.7</td>
</tr>
</tbody>
</table>

The data corresponding to the two IGBTs A and B aged for 900 cycles, which have suffered the lowest number of degradation cycles, will be considered as a reference for the healthy behavior and used for developing the SOM. Although these data represent a quasi-healthy condition of the IGBT, they are preferred to data referring to new IGBTs, since data collected in experimental tests show that there is a period of IGBT running characterized by a modification of the IGBT behavior.

The data from these two IGBTs are divided into a train set, for the training of the SOM, a validation set, used to calculate the normalization constant $MQE_{healthy}$ in Eq. (5), an optimization set, which will be used to optimize the quadruplet of the model parameters according to the procedure in Section 3.4, and, finally, a test set which will be used to verify the performance of the method. The data relative to the other four IGBTs (namely C, D, E and F) aged by 1800 and 2700 thermal cycles will be divided into an optimization set for the model parameters optimization and a test set. For ease of comprehension, Figure 5 shows how the available dataset has been divided.

Figure 5. Schematic representation of the available dataset and its division into train, validation and test sets

5.2. The SOM Diagnostic Model

Figure 6 shows the Unified distance matrix of the obtained SOM, and Figure 7 the distribution of the values for the weight vectors associated to each neuron. In particular, it is possible to notice that the right upper corner of the map contains neurons characterized by high $T$, high $I_p$ and high $V_{ce}$ whereas the left lower corner neurons are characterized by low $T$, low $I_p$ and low $V_{ce}$.

Figure 6. Representation of the unified distances of the SOM map trained with quasi-healthy data
Once the SOM has been trained, the average MQE of the validation test has been computed in order to allow defining the degradation indicator $QE$ according to Eq. (4).

### 5.3. Model Parameters Setting

The model parameters $L_i$, $L_o$ and the two thresholds $Th_{1,2}$ and $Th_{2,3}$, have been set according to the procedure illustrated in Section 3.4. The first column of Table 3 reports the considered range of variation of the parameters, whereas the identified optimum setting of the parameters, which leads to the best classification results is listed in the second column. Notice that values of $L_i$ and $L_o$ greater than 30 and 60, respectively, are not considered since they would require too long time for the collection of the necessary measurements.

Table 3. Ranges and optimum values for the model parameter setting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Th_{1,2}$</td>
<td>[0;1.4]</td>
<td>1</td>
</tr>
<tr>
<td>$Th_{2,3}$</td>
<td>[1;3]</td>
<td>2.75</td>
</tr>
<tr>
<td>$L_i$</td>
<td>[0;30]</td>
<td>30</td>
</tr>
<tr>
<td>$L_o$</td>
<td>[0;60]</td>
<td>60</td>
</tr>
</tbody>
</table>

### 5.4. Results

The SOM-based methodology has been applied to test patterns extracted from the 6 IGBTs (A-F) and not used during the SOM training and parameters identification phases.

Figure 8 shows the obtained results. The round markers represent healthy IGBTs (900 degradation cycles), the triangular markers represent partially degraded IGBTs (1800 degradation cycles), whereas the squared markers represent severely degraded IGBTs (2700 degradation cycles). Misclassifications provided by the SOM are represented by colored markers.

Notice that:

1- there are only two cases of missed alarms, i.e. patterns of class 3 (severely degraded) that have been erroneously assigned to class 2 (partially degraded). These patterns correspond to IGBT E, i.e. one of the two IGBTs which have undergone 2700 degradation cycles. It is also interesting to observe that the other patterns obtained from the same IGBT have $QE$ values close to the threshold $Th_{2,3}$. This may indicate that IGBT E is less degraded than IGBT F, even if they have been aged by the same number of degradation cycles.

2- 97% of the patterns corresponding to partially degraded IGBT C are assigned to class 1. Also in this case, we can interpret the results assuming that IGBT C has been more resistant to the degradation cycles than IGBT D.

3- There are no misclassifications of patterns whose true class is 1 and no cases of false alarms in class 3.

### 6. CONCLUSION

The objective of this work has been the development of an online method for the classification of the degradation state of IGBTs operating under variable operating conditions. The application of this method is specifically designed for IGBTs used on FEVs, which are characterized by continuously varying temperature and current conditions. The developed method is based on the construction of a SOM, which is trained using only data corresponding to healthy IGBTs. Then, relying on the use of the SOM
quantization error as degradation indicator, a condition-based-maintenance strategy has been proposed. The quantization error identified by the SOM is compared to two thresholds, $T_{h1,2}$ and $T_{h2,3}$, which are the limit values to identify the components as either healthy, partially degraded and severely degraded and thus needing maintenance. A general procedure for the optimum setting of these thresholds and of other parameters has been proposed. The procedure is based on the definition of an utility function which takes into account the consequences in terms of costs and unavailability of the component.

The method has been applied to data representative of IGBTs characterized by different levels of degradation. The data have been collected performing laboratory test at CEIT on IGBTs degraded by means of thermal cycles. The obtained results have confirmed the ability of the proposed method to classify different IGBTs as new, partially degraded or needing maintenance, regardless of the inverter operating conditions. The errors performed by the method are satisfactory from the points of view of reliability and availability: in fact, only the 2% the IGBTs which are very degraded are identified as partially degraded (missed alarms), and no false alarms are provided.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community’s Framework Programme (FP7/2007-2013) under grant agreement n° 314609. The authors are grateful for the support and contributions from other members of the HEMIS project consortium, from CEIT (Spain), IDIADA (Spain), Jema (Spain), MIRA (UK), Politecnico di Milano (Italy), VTT (Finland), and York EMC Services (UK). Further information can be found on the project website (www.hemis-eu.org).

REFERENCES


BIOGRAPHIES

**Marco Rigamonti** (MSC in Nuclear engineering, Politecnico di Milano, December 2012) is pursuing his PhD in Energetic and Nuclear Science and Technology at Politecnico di Milano (Milan, Italy). He is co-author of 2 works accepted for publications on international journals.

**Piero Baraldi** (PhD in nuclear engineering, Politecnico di Milano, 2006) is assistant professor of Nuclear Engineering at the department of Energy at the Politecnico di Milano. He has been Technical Committee Co-chair of the European Safety and Reliability Conference, ESREL2014, and Technical Programme Chair of the 2013 Prognostics and System Health Management Conference (PHM-2013). He is co-author of 2 books and more than 100 papers on international journals and proceedings of international conferences.

**Enrico Zio** (Nuclear Engineer Politecnico di Milano (1991); MSc in mechanical engineering, University of California, Los Angeles, UCLA (1995); PhD in nuclear engineering, Politecnico di Milano (1995); PhD in Probabilistic Risk Assessment, Massachusetts Institute of Technology, MIT (1998); Full professor, Politecnico di Milano (2005–); Director of the Chair on Complex Systems and the Energy Challenge at Ecole Centrale Paris and Supelec. Fondation Europeenne pour l’Energie Nouvelle – EdF (2010-present); Chairman of the European Safety and Reliability Association-ESRA (2010- present. He is co-author of seven books and more than 250 papers on international journals.

**Allegra Alessi** (MSC in Safety and prevention engineering for the process industry, Politecnico di Milano, April 2015) is currently involved in the research on the development of monitoring system for assessing the degradation on IGBTs.

**Ainhoa Galarza** is a researcher of the Electronics and Communications Department at CEIT-IK4 and an associated professor at Tecnun (University of Navarra). She received her MS in Industrial Engineering, majoring in Electricity in 1994 and her PhD in Industrial Engineering in 2000 from the University of Navarra (Spain). Her research fields include hybrid storage and energy management systems for electric vehicles, distributed electronics for industrial systems and signaling systems for railway applications (ETCS). She is the coordinator of the HEMIS (Electrical powertrain Health Monitoring for Increased Safety of FEVs) FP7 project.

**Daniel Astigarraga** is a PhD student of the Electronics and Communications Department at CEIT-IK4. He received his MS in Industrial Engineering, majoring in Electronics in 2012. His research fields include prognosis in power electronics and energy management systems for electric vehicles.
Analysis of Electrolytic Capacitor Degradation under Electrical Overstress for Prognostic Studies

Jason Renwick¹, Chetan S. Kulkarni², and José R. Celaya³

¹ University of West Indies, St. Agustine, Trinidad and Tabago
jasondren@gmail.com

² NASA Ames Research Center (SGT Inc.), Moffett Field, CA, USA
chetan.s.kulkarni@nasa.gov

³ Schlumberger, Software Technology and Innovation Center, Menlo Park, CA, USA
JCelaya@slb.com, jose.r.celaya@nasa.gov

ABSTRACT
The implementation of prognostics methodologies to electrical and electronics components and systems has become essential and critical as these systems find more prominence recently as they replace traditional systems in several critical applications. There are several challenges due to great variety of components used in a system, a continuous development of new electronics technologies, and a general lack of understanding of how electronics fail. Traditional reliability techniques in electronics tend to focus on understanding the time to failure for a batch of components of the same type. Just until recently, there has been a push to understand, in more depth, how a fault progresses as a function of usage, namely, loading and environmental conditions. Electrolytic capacitors have become critical components in electronics systems in several domains. They are known for their low reliability and frequent breakdown in critical systems like power supplies of avionics equipment and electrical drivers of electro-mechanical actuators. Capacitors are used as filtering elements on power electronics systems. Electrical power drivers for motors require capacitors to filter the rail voltage for the H-bridges that provide bidirectional current flow to the windings of electrical motors. These capacitors help to ensure that the heavy dynamic loads generated by the motors do not perturb the upstream power distribution system (Kulkarni, Biswas, Koutsoukos, Goebel, & Celaya, 2010; Orsagh, Brown, Roemer, Dabnev, & Hess, 2005).

In this work we establish the hypothesis that tau (τ), the RC time constant on a RC type filter circuit is a good precursor of failure candidate for prognostics. The degradation process in a capacitor must be reflected in this quantity. Degrada-
tion in the capacitors is observed by accelerated aging the devices under higher electrical stress. A charge/discharge square wave is applied to the devices and health is monitored by logging the charge/discharge cycles at regular intervals. Furthermore, we propose a way to estimate τ through the rise time observation of the charge transient in a capacitor. The rise time τ value of the charge cycle is calculated based on which the degradation is observed. In addition electro impedance spectroscopy (EIS) measurements are also taken.
to compute the values of Capacitance (C) and equivalent series resistance (ESR). The tau value calculated from the rise time and EIS measurements are used for studying the degradation in the test devices.

The papers discuss our research work in the following sections. Section 2 discusses the research methodology which we are following to study degradation in the capacitors and previous work. Section 3 discuss the experimental setup for the accelerated aging experiments. The paper concludes with discussion on results observed and next steps.

2. Research Methodology

Our research goal for this work is focused on studying the degradation process from an accelerated life test on real electrolytic capacitors. In the experiments, commercial-off-the-shelf capacitors are subjected to electrical stress conditions in order to observe and record the degradation process and identify performance conditions in the neighborhood of the failure criteria in a considerably reduced period.

2.1. Failure Precursor

The devices under test are subjected to charge/discharge waveforms at different voltage levels. In earlier work Electro-impedance spectroscopy was used periodically during the experiment to characterize the frequency response of the capacitor. These measurements along a reduced order model based on passive electrical elements are used to identify the capacitance and parasitic resistance element of the device.

As the capacitance and resistance of the device changes, these are reflected in the RC time constant (\( \tau \)) values. The RC time constant, also called tau (\( \tau \)), is the time constant of an RC circuit, is equal to the product of the circuit resistance (\( \Omega \)) and the circuit capacitance (\( \mu \)). We implement a methodology to observe degradation at the component level based on the rise time during charging. The capacitors are cycled from 0 to a threshold stress voltage in a 10 second charge/discharge cycle continuously and these cycles are recorded. This recorded data is used to study and interpret the degradation occurring in the devices.

Though with the EIS measurements we compute C and ESR values based on the system identification methods, there are some disadvantages of EIS measurement methods at the component level which are listed as below.

- The measurements cannot be taken inline of the circuitry and the test device has to be removed for measurement.
- Measurement noise is observed.
- Study degradation based on the output waveforms.
- Equipment availability to take EIS measurements. There is a requirement of specific instrument to take the measurements.

2.2. Previous Work

We studied accelerated degradation under electrical stress (Celaya, Kulkarni, Biswas, & Goebel, 2011; Celaya, Kulkarni, & Goebel, 2012) as well as thermal stress (Kulkarni, 2012) in electrolytic capacitors. A preliminary approach to remaining useful life prediction of electrolytic capacitors was presented in (Celaya et al., 2011). This paper here builds upon the work presented in the preliminary remaining useful life prediction in (Celaya et al., 2012).

In earlier work of a physics based degradation model during the accelerated life test a Bayesian framework was implemented to estimate the state of health of the capacitor based on measurement updates of key capacitor i.e, capacitance and ESR parameters. Unscented Kalman Filter (UKF) algorithm is used to track the state of health and the degradation model is used to make predictions of remaining useful life once no further measurements are available. A discussion and physical interpretation of the degradation model is presented (Celaya et al., 2011, 2012).

Earlier work focused on implementing prognostics methodologies to data collected at component level. Usually in field applications it is difficult to measure component level data. Our approach in this work is to collect system level measurements and compare them with component level measurements so as to study and apply prognostics methodologies at system level.

3. Accelerated Aging Experiments

Accelerated life test methods are often used in prognostics research as a way to assess the effects of the degradation process through time. It also allows for the identification and study of different failure mechanisms and their relationships with different observable signals and parameters. In the following section we present the accelerated aging methodology and an analysis of the degradation pattern induced by the aging. The capacitors are subjected to three voltage levels and their degradation was observed over the period of aging time. In this work we discuss specifically aging of the devices at 10V which is one of the voltage levels to which the devices were subjected for accelerated aging.

Capacitors were subjected to high voltage stress through an external supply source using a specifically developed hardware as described by block diagram in Fig. 1. To monitor the degradation of the devices measurements were taken on each of the devices. The first is the online monitoring which captures the voltage and current signature while the other is based on the EIS measurements. The schematic shown in Fig.1 shows the measurements taken for each capacitor.
3.1. Experiments

For this experiment a batch of 7 capacitors of 2200μF capacitance each are used. A hardware pcb board was developed which would subject the devices under test to different voltage conditions. The voltage conditions were selected such that it would stress the devices above their normal operating range as mentioned by the manufacturer. The capacitor components under test are maximum rated voltage of 10V, maximum current rating of 1A and maximum operating temperature of 85°C were used for the study. Under normal operating conditions the devices usually are used for a DC-DC converter with a output of 5V and ripple of less than 1% over its operating range. The boards subject the capacitors to electrical stress condition under 10V, 12V and 15V which is well above their normal operating range. This will stress the devices and we are able to observe and monitor the degradation in the devices within a short time period. Fig. 2 shows the complete setup of the experiments with the developed hardware and data measurement using NI LabView.

As shown in Fig. 1 each device is subjected to a square waveform of a specific voltage level. This simulates the charge/discharge cycle for the capacitor. The charge/discharge cycle is set such that the device is charged at the particular voltage level and the charge is held steady for a brief period of time before it is discharged through a constant load of 100 Ω. The continuous charge/discharge cycles subjects the devices to a high stress which leads to degradation. Due to the charging/discharging cycle the internal temperature of the capacitor increases which leads to changes in the capacitance and internal series resistance parameters of the device. Details of the theory is discussed in (Kulkarni, Biswas, Koutsoukos, Celaya, & Goebel, 2010).

3.1.1. Online Measurements

For the online measurements, charge/discharge waveforms are captured at every 10 minutes segments in burst of 10 cycles. Since we do not anticipate any change every minute due to the stressed conditions, the measurements are taken every 10 mins. These are done through a developed NI Labview integrated software and hardware system. A NI acquisition hardware system is used for collecting the data from the different boards and a developed LabView GUI is used to control the hardware. In addition a function generator is used to generate the required amplitude and frequency waveform to age the devices. As shown in Fig. 1 both the $V_O$ and $V_L$ measurements are captured to calculate the rise time ($\tau$) during each cycle. As per our hypothesis as the devices degrade we will observe the value of $\tau$ decrease over the period of time. The plots in Fig. 3 show $V_O$ transients for one of the capacitors (Capcitor #5) on the board as it ages over the period of time due to electrical stress. The plot also shows the corresponding change in $V_L$ values as the capacitor degrades. (Note: There are no legends for Fig. 3 since the $V_O$ and $V_L$ are plotted over the entire aging data). As discussed earlier the $\tau$ value is calculated based on the rise time of $V_L$, which changes over the operating age as seen in Fig. 3. The $\tau$ value is calculated offline at regular intervals to monitor how the device has been performing due to the stress conditions. $\tau$ calculations for all the devices on board are done, the results of which are discussed later in the sections.

3.1.2. EIS Measurements

The SP-150 Biologic SAS impedance measuring instrument uses Electrochemical Impedance spectroscopy, and finds applications in corrosion, battery, fuel cell development, sensors, and physical electro-chemistry. Impedance measurements can be made in a potentiostatic mode or in a galvanostatic mode. The PEIS mode is used for characterizing all the capacitors under test. Electrochemical impedance spectroscopy measurements are available to characterize the electrical performance of the capacitor under test.

The ESR and capacitance values were estimated from the capacitor impedance frequency response measured using the EIS instrument as shown in the plots of Fig. 4. During each measurement the voltage source was shut down, capac-
itors were discharged completely and then the characteriza-
tion procedure was carried out. This process is discussed in
(Celaya et al., 2012).

Table 1 shows the values of C and ESR for all the devices
aged on the board. It can be observed that the C values for
all the devices cross a soft threshold point of 20% below the
actual capacitance values. Usually a capacitor is considered
not fit for use if its capacitance reaches 20% below its initial
values (MIL-C-62F, 2008; IEC-60068-1, 1988). Though the
devices crossed the threshold we continued the experiments
to study any further effects on the internal structure of the
device to high electrical stress. It was observed that the ca-
capacitance values increase after around 400 hours of operation
under the continuous charge/discharge cycle at 10V.

The change in the capacitance value beyond 400 hours could
be hypothesized to the oxide layer formation phenomenon
which may lead to formation of a parallel capacitance. We
are currently studding the data and the underlying physics
phenomenon models to know more about this change in the
capacitance value increasing after crossing the failure thresh-
old.

In addition to the EIS measurements we also tracked the
degradation of the devices through online measurements
during the charge/discharge cycles. As the capacitor degrades,
computed RC time constant from the charge cycle during the
charge/discharge cycle changes given by $\hat{\tau}(t)$ is constant. As
the device degrades the C and ESR values change and this
affects the RC time constant of the device. As the C and
ESR values change the RC time constant changes affecting
the change in the $\hat{\tau}(t)$ values.

For this work the online measurements were not started until
upto around 1100 hours into the aging of the devices. As can
be seen from the plots in Fig. 5

The two methods used to calculate $\hat{\tau}(t)$ require i) offline
measurements and ii) online measurements. In our previous
work (Celaya et al., 2011, 2012) EIS measurements (offline)
was seen as the ground truth when estimating capacitor pa-
rameters. The opposing method (online) comes from calcu-
lation done to the transient voltage data collected. Fig. 6 is
a comparison of the online to offline values of tau. Although
these values have a comparable trend, they exhibit a distinct
difference in magnitude.

This observed difference can be attributed to a number of
factors related to the physical experimental set-up, operat-
ing temperatures and circuit modeling. The implication of
the opposing methods (offline vs. online) is that the mea-
surements are taken at different temperature i.e. operating
temperature opposed to resting temperature (room tempera-
ture). As seen in previous work (Kulkarni, Biswas, Celaya,
& Goebel, 2013) one of the most significant factors that im-
pact capacitor degradation is temperature.

4. RESULTS

In this section we discuss degradation data observed in de-
vice under test for the experiments performed. Plots in Fig. 4
show the impedance measurement for one of the devices un-
der test. The figure plots the imaginary and real values for a
spectrum of frequencies. Using a system identification model
C and ESR values are retrieved from the measured raw data.
As can be seen from the plots, as the device ages the slope of
the measurement values changes. This is due to the change
in the C and ESR values which change with the aging of the
device.
Table 1. C and ESR computed values at different aging times for 10V Board

<table>
<thead>
<tr>
<th>Device</th>
<th>Age : 0 hrs</th>
<th>Age : 296 hrs</th>
<th>Age : 3063 hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cap (µF)</td>
<td>ESR (Ω)</td>
<td>Cap (µF)</td>
</tr>
<tr>
<td>1</td>
<td>1944.2</td>
<td>1859.0</td>
<td>1521.8</td>
</tr>
<tr>
<td>2</td>
<td>1943.0</td>
<td>1901.8</td>
<td>1541.8</td>
</tr>
<tr>
<td>3</td>
<td>1930.6</td>
<td>1788.8</td>
<td>1300.0</td>
</tr>
<tr>
<td>4</td>
<td>1857.6</td>
<td>1901.8</td>
<td>1495.6</td>
</tr>
<tr>
<td>5</td>
<td>1931.0</td>
<td>1877.2</td>
<td>1481.8</td>
</tr>
<tr>
<td>6</td>
<td>1926.2</td>
<td>1919.8</td>
<td>1512.6</td>
</tr>
<tr>
<td>7</td>
<td>1899.6</td>
<td>1906.0</td>
<td>1482.0</td>
</tr>
</tbody>
</table>

With regards to the experimental setup, the transient voltage data is collected by NI hardware with supporting LabVIEW software whereas the EIS data is collected by different hardware (SP-150 Biologic). A major effect difference in hardware may have is the calibration standards. By using offline EIS measurements we also change the circuit under which the capacitor is being observed, whereas, if the transient data is measurement online, the capacitor circuit will see the effects of the connected hardware. It would be very difficult to determine the effect that the measurement hardware has on a circuit. When determining the capacitor characteristics from the EIS measurements we apply a circuit model to the data and extract a value for capacitance and ESR. We currently use a simplified lumped parameter model (C + ESR) when analyzing EIS data. One option to explain is that our model of the capacitor is no long accurate to the capacitors under test. As capacitors are left unused or while in use, their physical/chemical composition may change hence a simple C+ESR lumped parameter model may be obsolete in the later stages of aging.

5. CONCLUSION AND DISCUSSION

The results presented show that by measuring the $\tau$ value of a capacitor may be a viable way of observing a capacitor’s degradation. Also, by using $\tau(t)$ as the observed parameter we can develop online methods of characterizing capacitors and a metric to calculate remaining useful life or end of life of capacitors still in use. The practicability of doing offline measurements is very low when trying to integrate into a system hence doing online measurements would allow for more integration into electronic systems.

If a generic model is derived for a capacitor based on its $\tau(t)$ values, it may be possible to define the EOL of a system (based on capacitor failure ) based on its circuit design i.e. the components possibly affecting the $\tau$ measurements.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the contributions of researchers at NASA Ames Research Center: Kai Goebel and Scott Poll. The funding for this research is provided by NASA ARMD System-wide Safety & Assurance Technology (SSAT) project. The work was done by José R. Celaya while working at the Prognostics Center of Excellence, NASA Ames Research Center.

REFERENCES


**Biographies**

**Jason Renwick** is a first-year engineering student pursuing a BSc. Electrical and Computer Engineering at The University of the West Indies, St. Augustine Campus, Trinidad and Tobago. In 2014, he was awarded a Fall internship at Prognostics Center of Excellence, NASA Ames Research Center. Under the mentorship of Dr. José Celaya and Dr. Chetan Kulkarni, Jason engaged in performing aging experiments of electrolytic capacitors. He returned in Spring 2015 for a second internship period during which he performed detailed analysis on the experimental data collected in his previous placement.

**Chetan S. Kulkarni** received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. He was a Senior Project Engineer with Honeywell Automation India Limited (HAIL) from 2003 till April 2006. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he has been a Staff Researcher with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and Senior member IEEE.

**José R. Celaya** is a Senior Data Scientist at the Software Technology and Innovation Center, Schlumberger. Previously, he was the Lead Scientist and Co-lead at the Diagnostics and Prognostics Group and a founding member of the Prognostics Center of Excellence, both at the Intelligent Systems Division of NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Electrical Engineering in 2003, all from Rensselaer Polytechnic Institute, Troy New York; and a B. S. in Cybernetics Engineering in 2001 from CETYS University, México.

**Figure 5.** $\tau$ aging plots for all the devices under similar stress conditions(10V Board)
Figure 6. $\tau$, C and ESR aging plots: Cap C5 (10V Board)
Review of Markov Models for Maintenance Optimization in the Context of Offshore Wind

Rafael Dawid¹, David McMillan² and Matthew Revie³

¹²Strathclyde University, Department of Electronic and Electrical Engineering, Glasgow, United Kingdom
rafael.dawid@strath.ac.uk
d.mcmillan@strath.ac.uk

³Strathclyde University, Department of Management Science, Glasgow, United Kingdom
matthew.j.revie@strath.ac.uk

ABSTRACT
The offshore environment poses a number of challenges to wind farm operators. Harsher climatic conditions typically result in lower reliability while challenges in accessibility make maintenance difficult. One of the ways to improve availability is to optimize the Operation and Maintenance (O&M) actions such as scheduled, corrective and proactive maintenance. Many authors have attempted to model or optimize O&M through the use of Markov models. Two examples of Markov models, Hidden Markov Models (HMMs) and Partially Observable Markov Decision Processes (POMDPs) are investigated in this paper. In general, Markov models are a powerful statistical tool, which has been successfully applied for component diagnostics, prognostics and maintenance optimization across a range of industries. This paper discusses the suitability of these models to the offshore wind industry. Existing models which have been created for the wind industry are critically reviewed and discussed. As there is little evidence of widespread application of these models, this paper aims to highlight the key factors required for successful application of Markov models to practical problems. From this, the paper identifies the necessary theoretical and practical gaps that must be resolved in order to gain broad acceptance of Markov models to support O&M decision making in the offshore wind industry.

1. INTRODUCTION
Offshore wind turbines will play a key part in meeting the UK’s renewable energy targets in the future. The US Department of Energy also anticipates a sharp increase in the number of offshore wind farms (US DoE, 2008). However, the offshore wind turbine energy yield in the UK is still badly affected by low availabilities, which have been shown to be around 80.2%, compared to 97% onshore (Feng, Tavner, & Long, 2004). Enabling operators to effectively plan for repairs and inspections would likely improve the availabilities. There has been research (Pahlke, 2007) showing there is significant demand for decision support systems in the offshore wind industry. It was reported that 99% of mechanical failures are preceded by noticeable indicators (Lee, Ni, Sarangapani, Mathew, 2011). Fully utilizing Condition Monitoring (CM) data can lead to improved diagnosis and prognosis, yet some authors argue that the wind industry is not taking the full advantage of it (Cibulka, Ebbesen, Hovland, Robbersmyr, & Hansen, 2012). Attempts have been made to quantify the benefits of monitoring systems for wind turbines (McMillan & Ault, 2007), concluding that rewards of having a CM system outweigh the costs in most cases.

Offshore wind farm operators’ actions are constrained by logistics and the weather, which makes optimizing O&M difficult (Van Horenbeek, Van Ostayen, Düflou, & Pintelon, 2013). High complexity of many wind turbine components and the environment they operate in often means that a number of failure modes exists, hindering effective failure prediction (Fischer, Besnard, & Bertling, 2012). Although there has been a significant amount of work done on maintenance optimization models, both in the wind sector and other industries, there is little evidence of successful application of these models in the offshore wind sector. Markov models have been successfully applied to diagnosis, prognosis and maintenance optimization in other industries, with significant effort being invested to implement them in the wind sector. The purpose of this study is to provide an overview of the work done on O&M optimization, focusing specifically on the use of Markov models, in the context of the offshore wind industry.

Rafael Dawid et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
2. RELATED REVIEWS

2.1. Wind Industry

There have been a number of review papers related to maintenance optimization models, but only a few of them are focused on the wind industry. Welte and Wang (2013) outlined different types of lifetime estimation models, while providing examples of their application to the wind industry. Hofmann (2011) identified 49 models on different aspects across the whole life-cycle of the wind farm, from planning and construction to O&M. However, little information is provided on the inner workings of those models. Moreover, a number of those models are shrouded by commercial interest of the companies who own them, making it difficult to investigate their effectiveness and the methods used.

El-Thalji (2012) provided a comprehensive review of the O&M practices of wind power assets, highlighting that the main issues in offshore wind maintenance lie with site accessibility and environmental factors. Cibulka et al. (2012) provided an insight into the failure modes of different wind turbine components; methods of monitoring different electrical, mechanical and fluid parameters were also reviewed. The paper concluded that the wind turbine industry lacks pro-active use of CM data for estimating the remaining life of components. Finally, Hameed, Hong, Cho, Ahn and Song (2009) provided a comprehensive review of various CM approaches for wind turbine industry.

The aforementioned review papers provide a solid background of the offshore wind maintenance practices and shed some light on the approaches used so far, but no detailed description of the models was provided. However, there is no lack of review papers on the use of Markov models for maintenance in other industries; these are described in the following section.

2.2. Other Industries

Optimizing maintenance using mathematical models is by no means a novel approach; a significant amount of work was done on maintenance optimization models from 1970’s until early the 90’s, yielding numerous review papers (Pierskalla & Voelker, 1976)(Sherif & Smith, 1981) (Monahan, 1982)(Valdez-Flores & Feldman, 1989)(Cho & Parlar, 1991)(Wijnmalen & Hontelez, 1992). Monahan’s review is particularly noteworthy as his POMDP framework and algorithms for computing optimum policies have been quoted and applied numerous times by many researchers in the field of maintenance optimization.

A more recent study by Dekker (1996) reviewed maintenance optimization models, mostly concerning vehicle replacement, road maintenance and power stations. The paper identified 43 case studies, some utilizing Markov models, in which the models were based on real data and the outcome of the model advised decision making. Despite identifying a large number of related case studies, little attention was given to critical analysis of the different approaches used. A brief review of software used for maintenance optimization was also provided, but given that the paper was written almost 20 years ago, most of them are now obsolete. Amongst the 132 references in this paper, there was not a single one on the subject of wind energy. According to Frangopol, Kallen and Noortwijk (2004), Markov models are the most common method of modelling bridge maintenance. The authors also highlighted the need to incorporate the data from imperfect inspections in deterioration systems.

Deterioration prognostic models are a large part of decision support tools. For a prognostic model to be effective, it needs to be able to predict the component’s condition sufficiently far into the future to facilitate preparation of spares and human resources (Heng, Zhang, Tan, & Mathew, 2009). Heng et al. have also stressed that most studies on rotating machinery are done in labs, neglecting practical considerations such as interactions between components and the impact of weather. Tung and Yang (2009) provided another comprehensive review paper on rotating machinery prognostics, highlighting that physical model-based approaches are not suitable for complex systems. The authors also stated that the lack of industrial application of prognostic models is partly due to the complexity of a real-life machine, hindering accurate modelling and hence reducing the accuracy of predictions. The use of Markov models for remaining useful life estimation was reviewed by Si, Wang, Hu and Zhou (2011). The authors highlighted the challenges of capturing the influence of external variables, as well as modelling multiple failure modes for the same component.

Peng, Dong and Zuo (2010) argue that combining 2 or more different prognostic approaches may increase the precision, reduce the computational time and combine the benefits of the different approaches while nullifying their demerits. The authors highlighted the fact that HMMs are easily realizable in software. They also stated that most research in the area is still in the theoretical phase, few of the models have been applied in practice. Scarf (1997) stated that “too much attention is paid to the invention of new models, with little thought, it seems, as to their applicability”. The author also argues that many models are over-complicated, making them hard to follow by the practitioners. Zio (2009) and Kothamasu, Huang and Verduin (2009) also recommended that the implementation of reliability methods should be supported by reasonably user-friendly software.

Dragomir, Gouriveau, Minca and Zerhouni (2009) provided a brief review gathering papers on various prognostic algorithms, stating that real prognostic models are scarce in the industry while Tung (Tung & Yang, 2009) pointed out that the amount of research done on prognosis is much less
compared to diagnosis. A large proportion of papers reviewed in (Wang, 2002) involve maintenance policies based on age or fixed time intervals. Such approaches lack depth and flexibility and hence are not suitable for the offshore wind industry. Vasili, Hong and Ismail (2011) rightly concluded that there is a need to develop approaches which will optimize maintenance, while also considering different aspects of the maintenance management. This is especially true for the wind industry, whereby the factors such as weather and logistics are paramount to successful planning of maintenance strategy.

Nicolai and Dekker (2008) focused on multi-component approach to modelling, which has the potential to capture dependencies between components but potentially over-complicates the model. Kothamasu et al. (2009) looked into different techniques of system health monitoring and their use for prognosis to improve reliability, including some Markov models. Reinersen (1996) reviewed a number of models with practical applications but found no models that could be applied, generally, to a broader range of problems. The author called for more consistency in the industry when applying methods for assessment of deterioration and residual life of structures.

The majority of the papers discussed in this section reviewed Markov models, but only a few were solely focused on them. The next section presents an overview of theoretical and practical Markov models for maintenance optimization.

3. MARKOV MODELS

Markov Models can be broadly split into Markov chains, Markov Decision Processes (MDPs), Hidden Markov Models (HMMs) and Partially Observable Markov Decision Processes (POMDPs), which are described in the following sections. Table 1 in the Appendix contains a summary of Markov models for maintenance optimization referenced in this paper, grouped by the method used and application.

3.1. Markov Chains and Markov Decision Processes

Markov chain is a random process, wherein a probability of transition between states only depends on the current state, not on the sequence of previous events. Markov chains have been used for modelling offshore wind O&M (Özdirik, Skiba, Würtz, Kaltschmitt, & Williams, 2013), in a paper which discusses a number of limitations associated with maintenance of offshore wind farms. Yang, Kwan and Chang (2008) used Markov chains to simulate deterioration of electrical substation components, while a multiobjective evolutionary algorithm was used to provide the user with a number of Pareto curves, facilitating visualization of the trade-offs between overall costs and expected unserved energy. Markov chains have been used to model wind turbine blade deterioration in a study by Besnard and Bertling (2010) which also compared the costs and benefits of using a condition monitoring techniques versus inspections, favoring the former approach.

Wilson and McMillan (2014) used Markov chains and Monte Carlo method for assessing reliability of potential wind farm sites, providing a forecast of future O&M costs. Lee, Li and Ni (2013) argued that equipment operators can be quite conservative when setting the maintenance intervals. By applying Markov models to a semiconductor manufacturing process data it was shown that significant maintenance cost savings can be obtained by increasing the time between maintenance actions.

Semi-Markov models relax the assumption of constant transition probabilities, which is more representative of most engineering systems. They have been used in modelling deterioration (Kleiner, 2001) (Black, Brint, & Brailsford, 2005); the latter research showing that maintenance cost savings can be obtained using this method, given sufficient amount of past deterioration data. Semi-Markov approach was also applied to 2-unit standby systems by Maksoud and Moustafa (2009) and Zhong and Jin (2014), wherein the optimal policy for the former was obtained using an iterative process, while the latter utilized Laplace Transform to solve Markov renewal equations, yielding optimal maintenance policy. A different study (Kharoufeh, Solo, & Ulukus, 2010) used a semi-Markov model to assess the current and future states of a system, taking into account the environmental factors. A framework for asset management of power distribution networks based on Semi-Markov models was proposed by Johnson, Strachan and Ault (2012). The authors argue that the component’s future deterioration can be predicted without any historical data by using condition health indices.

MDPs are an extension of a Markov chain, with the addition of the possibility of taking actions, each with associated cost or reward. The timing, order and choice of actions can then be optimized for a given parameter, usually to minimize the cost. Shafiee (2015) defined the 3 main echelons of decision making for offshore wind; namely strategic (long term), tactical (medium term) and operational (short term). Maintenance optimization models would aim to aid operational decisions through maintenance scheduling and planning of logistics; they could also have some impact on the tactical decisions – for example spare part management or maintenance support organization.

Chan and Asgarpoor (2006) provided a simple example of how an MDP can be used to find the optimal mean time to preventative maintenance. Nielsen and Sørensen (2014) conducted a comparison of different approaches to decision support, concluding that MDP is the most accurate method of optimizing the decision policy, with great potential for application in offshore wind industry. The same authors (Nielsen & Sørensen, 2012) have also shown that an MDP can be easily adapted to take external factors such as weather and vessel rental costs into account. This opens up
possibilities of further O&M cost savings, for example by using clustering of repairs.

Semi-Markov Decision Processes (SMDPs) have also been applied in optimizing the inspection and maintenance schedule (Chen & Trivedi, 2005) (Amari & McLaughlin, 2006), with the latter model being very simple but effective, meaning it could be easily adapted to a given practical problem. Berenguer, Chu and Grall (1997) used an SMDP for optimization of inspection and maintenance; although their model may be more realistic for some systems compared to other MDP approaches, the authors admit it is difficult to exploit it analytically due to its complexity.

Kahrobaee and Asgarpoor (2013) proposed an approach for maintenance optimization based on SMDPs, which was applied to a case study of wind turbines. Once the optimal repair policy was found, a Monte Carlo simulation was used to investigate how the wind turbine availability varies with different factors such as the number of repair technicians employed.

The majority of the papers referenced in this section provide numerical examples, which are useful in understanding and assessing the model’s capabilities, but are no match for validation through an application to an actual problem. Although computationally effective, simple Markov models assume that the exact state of the component is known, which is rarely the case with offshore wind turbines. This assumption is relaxed through the use of HMMs, as described in the following section.

3.2. Hidden Markov Models

Wind turbine CM data provides valuable information on the state of various components, however, the degree of system’s deterioration is usually difficult to predict with certainty. In HMMs, the current state of the component is represented by a probability distribution, making it a very functional tool for wind turbine deterioration modelling, diagnosis and prognosis.

Detection of machine failure using HMMs was investigated by Tai, Ching and Chan (2009). Qian, Jiao, Hu and Yan (2007) trained HMMs using a Baum-Welch algorithm to diagnose the type of fault in large scale power transformers. A different approach to diagnosis was suggested by Kwan, Zhang, Xu and Haynes, (2003) and Zhang, Xu, Kwan, Liang, Xie, and Haynes (2005). In both papers, multiple HMMs were created for different failure modes. The HMM with the highest probability of being in a failed state was used to identify the failure mode. Such approach is particularly useful for systems which have more than one likely failure mode, as is often the case with wind turbine components. Both studies have been based on experimental data and the algorithms have also shown prognostic capabilities. Ghasemi, Yacout and Ouali (2007) have used HMMs to simulate deterioration in their model which calculates the long-run average operating costs for strategies with different observation intervals.

HMM-based clustering was used by Chinnam and Baruah (2003) for diagnostics and prognostics. Numerous HMMs were constructed and tested with 3 best performers being selected to diagnose the condition of the asset. A multivariate distribution of the state transition points generated by HMMs was then used for prognosis. Zhou, Hu, Xu, Chen and Zhou (2010) proposed a HMM for real time failure prognosis. Expert knowledge was incorporated into the model through belief rules to capture the influence of environmental factors. The model shows great potential for application in offshore wind, as it is capable of considering the environmental and logistical factors, however the authors did admit that further testing is required to validate the model. HMM approach has also been used in decision making support tool for offshore wind called ECUME (Douard, Domezq, & Lair, 2012).

Research by Dong (2008) used an Auto-Regressive Hidden Semi-Markov Model, which has a few advantages over a standard HMM; namely it does not follow the standard Markov memory-less approach, it also relaxes the assumption of independent observations. The algorithm was tested on a case study of hydraulic pumps and shown good health state recognition rates, with the possibility of application for prognostics. The same author has also applied Hidden Semi-Markov Model (HSM) methodology for diagnostics and prognostics in his earlier work on hydraulic pumps (Dong & He, 2007) and helicopter transmission system (Dong, He, Banerjee, & Keller, 2006), showing that HSSMs are more effective at current state recognition than HMMs. Recent work by Cartella, Lemeire, Dimiccoli and Sahli (2015) used the HSMM methodology for remaining useful life estimation of bearings. The model was validated using real life vibration data and produced reasonably accurate results (although bearings under lab conditions do exhibit monotonically increasing degradation pattern making it easier to predict). One of the advantages of the approach presented in this paper is that it allows the use of both discrete and continuous observations.

It is worth noting that a large proportion of models reviewed in this section were based on experimental data, which is an improvement over validation approaches through numerical examples and simulations. However, as stated by Heng et al. (2009), lab tests fail to capture many practical considerations. The models described above are capable of deterioration modelling, diagnosis and often prognosis, however, they lack the decision making capability. POMDPs combine the hidden property of the Markov model with the ability to consider various maintenance actions and implications, resulting in a powerful maintenance optimization tool.
3.3. Partially Observable Markov Decision Processes

Early theoretical research by White (1976), Rosenfield (1976) and Monahan (1982) have laid the groundwork for many more recent POMDP models. Since then, a number of other papers presenting theoretical frameworks for the application of POMDPs to the maintenance problem have been published (Madanat, 1993)(David, Friedman, & Sinuany-Stern, 1999)(Makis & Jiang, 2003)(Maillard, 2006). The contribution of these papers to the field of maintenance optimization is without a doubt substantial, however, many researchers are reluctant to apply those frameworks, instead opting to use their own approaches.

Byon and Ding (2010) created a model for the offshore wind industry, which emphasizes the importance of the seasonal weather variation on wind turbine availability. A POMDP model was solved by backward dynamic programming method. It was shown that by using the proposed approach, the maintenance costs over the lifetime of a wind turbine can be reduced significantly compared to both periodic maintenance and condition-based maintenance strategy which does not take into account the seasonal variation. The model could be improved by considering the logistical issues specific to the offshore wind industry such as long lead times on parts and vessels. These factors were given more thought in Byon’s related article (Byon, Ntaimo, & Ding, 2010), which highlights the importance of condition-based monitoring, yet it fails to incorporate the CM data into the model. Dynamic programming method applied by Byon and Ding (2010) is computationally intensive for large problems. It was suggested that the application of a tractable approximation scheme can shorten the computational effort required, while still considering dynamic weather changes (Byon, 2012).

Papakonstantinou and Shinozuka conducted a two part study focused on the use of POMDPs for planning structural inspection and maintenance. Part 1 (Papakonstantinou & Shinozuka, 2014a) highlights the difficulties of solving POMDPs for complex problems, while providing practical solutions. Part 2 (Papakonstantinou & Shinozuka, 2014b) applies the theory from part 1 to a case study on corroding reinforced concrete structure. The model yields a highly complex optimum policy, which, according to the authors, could not have been reached by any other method. The framework used in the two part study is very promising; however the complexity of the model may hinder its applicability. The authors argue that exact solution for a POMDP would be too computationally intensive to solve for large problems and propose an approximate value iteration method instead, which is much more effective dealing with large problems (Papakonstantinou & Shinozuka, 2014c).

AlDurgam and Duffuaa (2012) used a POMDP model to generate policy graphs, which allow the operator to choose the optimal maintenance action and speed setting given the current belief state of the component. Fan, Xu and Chen (2013) and Chen, Fan, Hu and Zhou (2014) investigated repair optimization for systems with imperfect maintenance; they also argue that limiting the number of times a component can be repaired and imposing quicker deterioration on components which have already been repaired multiple times is more representative of engineering systems. The authors have also stated that their work needs extending to include condition-based maintenance through the use of sensor data, which would be particularly applicable for the wind industry.

Srinivasan and Parlikad (2014) combined the advantages of an SMDP and POMDP by creating a Partially Observable Semi-Markov Decision Process (POSMDP) for optimum maintenance decision making. Through the use of belief state, the POSMDP is converted into a SMDP. This approach allows the use of different failure rate distributions, facilitating the method’s potential application to various wind turbine components. A different approach to POSMDPs was used by Zhou, Ma, Matthew, Sun and Wolff (2010). In their research, degradation is modelled using a Gamma-based state-space approach. The model is based on continuous POSMDP, which is then converted to a fully observable SMDP through an application of Monte Carlo-based density projection method to optimize maintenance decision making.

Standard POMDPs require set values of transition and emission probabilities, but there is often an uncertainty associated with those. Memarzadeh, Pozzi and Kolter (2013) propose a Bayes-adaptive POMDP methodology, which treats conditional probabilities as random variables. This can result in the optimal policy being sub-optimal for any specific value of transition and emission probabilities, instead maximizing the value for the entire state. A wind farm case study based on synthetic data showed that this methodology can be more effective than a standard POMDP approach, especially for problems with high conditional probability uncertainty.

POMDPs have also been used in the context of civil engineering. Jiang, Corotis and Ellis (2000) developed a model which considered fatigue and corrosion as main deterioration processes of a steel girder highway bridge. A detailed example of the model’s application is provided, which, despite considering 5 maintenance action types and 4 inspection strategies, remains computationally effective. Later work by the same authors (Corotis, Ellis, & Jiang, 2005) presented POMDP theory and an algorithm for the optimal management and design of structures. Ivy and Pollock (2005) attempted to optimize maintenance on a system with “silent failures”: i.e. a system in which the component can remain operational despite it being in a failed state, with an increased cost being incurred on its operation.
POMDPs and POSMDPs are modelling tools with a number of advantages: they are capable of dealing with imperfect observations, allow flexibility in terms of the choice of maintenance action and deterioration mechanism, are capable of modelling multiple failure modes and have been shown to be able to consider external factors such as weather and logistics in making the optimal decision, making them a suitable methodology choice for successful application to the maintenance problem in the wind industry.

4. Conclusions

This review paper focused on the use of Markov models for deterioration modelling and maintenance optimization in a wide range of industries. Given a projected threefold increase in UK offshore wind O&M spend in the next 10 years (GL Garrad Hassan, 2013), the application of such models will play an important role in keeping the costs of energy low. The main conclusions of this paper are as follows:

- The majority of the models reviewed in this paper have either been validated by lab tests or, in many cases, their capabilities were shown using simple numerical examples/simulations. Very few models have actually been applied in the industrial environment. Other researchers in the field have suggested more emphasis should be placed on application of the existing models rather than invention of new models.

- A large number of the models discussed in this paper contain a sound framework which could be successfully applied to the wind industry, provided significant adaptation was carried out, which would involve ensuring that factors such as access restrictions due to weather and logistical issues are considered.

- One of the constraints to widespread Markov models application in the offshore wind industry could be their complexity. Other researchers in the maintenance optimization field have stated that practitioners are unlikely to apply over-complicated models.

- Some researchers have touched upon the computational constraints of Markov models (Tai et al., 2009) (Papakonstantinou & Shinozuka, 2014b). Although HMMs and POMDPs are effective for small state spaces, it was indicated that computational factors may become prohibitive for larger and more complex systems, especially when the methodologies are applied to wind farm-scale problems. In an attempt to address this issue, researchers started to explore more computationally effective ways of solving POMDPs (Byon, 2012) (Papakonstantinou & Shinozuka, 2014c).

- Some of the interesting concepts which have not been researched in depth within the Markov model framework, but may be useful for offshore wind are: opportunistic maintenance, the silent failure approach and de-rating to potentially slow down the degradation process if the turbine cannot be accessed.

The majority of the papers reviewed here focus on the maintenance of mechanical, electrical or structural systems, which all can be applied to offshore wind turbines. However, the offshore environment, where the majority of UK wind turbines will be built in near future, poses numerous challenges to wind farm operators. Research conducted for other industries often ignores issues such as access restrictions, offshore logistics and the high costs associated with it and the problem of effective utilization of large volumes of CM data for deterioration modelling and maintenance optimization. Although it has been shown by some authors that these challenges can be tackled through the use of Markov models, no comprehensive framework exists capable of considering all these factors. The focus of researchers working on O&M for offshore wind should be to attempt to create such framework. As shown in this, and other review papers, methodologies exist to fit most offshore maintenance problems; the biggest challenge now is to work with practitioners to apply those models to real engineering problems.

References


## APPENDIX

Table 1. Markov models by industry and method used\(^1,2\)

<table>
<thead>
<tr>
<th>Industry</th>
<th>General applications</th>
<th>Electrical systems</th>
<th>Civil engineering</th>
<th>Mechanical/Manufacturing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov Chain</td>
<td>(Wilson, 2014)(^{MU})</td>
<td>(Lee, 2013)(^{MU})</td>
<td>(Yang, 2008)(^{MU})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Özdirik, 2013)(^{MU})</td>
<td>(Besnard, 2010)(^{MU})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semi-Markov Chain</td>
<td>(Kharoufeh, 2010)(^{MU})</td>
<td>(Zhong, 2014)(^{MU})</td>
<td>(Johnson, 2012)(^{IO})</td>
<td>(Black, 2005)(^{IO})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Maksoud, 2009)(^{MU})</td>
<td></td>
<td>(Kleiner, 2001)(^{IO})</td>
</tr>
<tr>
<td>MDP</td>
<td>(Nielsen, 2014)(^{IO})</td>
<td></td>
<td>(Chan, 2006)(^{IO})</td>
<td></td>
</tr>
<tr>
<td>SMDP</td>
<td>(Kahrobaee, 2013)(^{MU})</td>
<td>(Amari, 2006)(^{MU})</td>
<td>(Chen, 2005)(^{MU})</td>
<td>(Berenguer, 1997)(^{MU})</td>
</tr>
<tr>
<td>HMM</td>
<td>(Douard, 2012)(^{IO})</td>
<td>(Zhou, 2010a)(^{IO})</td>
<td>(Qian, 2007)(^{IO})</td>
<td>(Tai, 2009)(^{MU})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Ghasemi, 2007)(^{IO})</td>
<td></td>
<td>(Zhang et al., 2005)(^{MU})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Kwan, 2003)(^{MU})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Chinnam, 2003)(^{MU})</td>
</tr>
<tr>
<td>HSMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POMDP</td>
<td>(Memarzadeh, 2013)(^{IO})</td>
<td>(Chen, 2014)(^{IO})</td>
<td></td>
<td>(Papakonstantinou, 2014b)(^{IO})</td>
</tr>
<tr>
<td></td>
<td>(Byon, 2010)(^{IO})</td>
<td>(Fan, 2013)(^{IO})</td>
<td></td>
<td>(Corotis, 2005)(^{IO})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(AlDurgam, 2012)(^{IO})</td>
<td></td>
<td>(Jiang, 2000)(^{IO})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Maillart, 2006)(^{IO})</td>
<td></td>
<td>(Madanat, 1993)(^{IO})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Ivy, 2005)(^{IO})</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(David, 1999)(^{IO})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSMDP</td>
<td>(Srinivasan, 2014)(^{IO})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Zhou, 2010b)(^{IO})</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{IO}\) – Imperfect Observations: uncertainty associated with the action of observation/inspection.

\(^{MU}\) – Multi-Unit: models formulated specifically for multi-unit systems (rather than models capable of considering multiple components).

---

\(^1\) For clarity, the references have been shortened to 1st author and year of publication only.

\(^2\) This table contains a quick reference for the reader, consolidating the articles mentioned in the body of the paper. It is not a comprehensive list of all Markov models within these industries.
Detection of pitch failures in wind turbines using environmental noise recognition techniques

Georgios Alexandros Skrimpas¹, Kun S. Marhadi², Robert Gomez³, Christian Walsted Sweeney⁴, Bogi Bech Jensen⁵, Nenad Mijatovic⁶, and Joachim Holboell⁷

¹,²,⁴ Brüel and Kjær Vibro, Nærum, 2850, Denmark
alexandros.skrimpas@bkvibro.com
kun.marhadi@bkvibro.com
christian.sweeney@bkvibro.com

³ Brüel and Kjær Vibro, Sugar Land, 77478 Texas, USA
robert.gomez@bkvibro.com

⁵ University of the Faroe Islands, Tórshavn, 100, Faroe Islands
bogibj@setur.fo

⁶,⁷ Technical University of Denmark, Lyngby, 2800, Denmark
nm@elektro.dtu.dk
jh@elektro.dtu.dk

ABSTRACT

Modern wind turbines employ pitch regulated control strategies in order to optimise the yielded power production. Pitch systems can be subjected to various failure modes related to cylinders, bearings and loose mounting, leading to poor pitching and aerodynamic imbalance. Early stage pitch malfunctions manifest as impacts in vibration signals recorded by accelerometers mounted in the hub vicinity, as for example on the main bearings or nacelle frame, depending on the installed condition monitoring system and turbine topology. Due to the location of the above mentioned vibration sensors, impacts of various origin, such as from loose covers, can be generated, complicating the assessment of the impact nature. In this work, detection of pitch issues is performed by analysing vibration impacts from main bearing accelerometers and applying environmental noise and speech recognition techniques. The proposed method is built upon the following three processes. Firstly, the impacts are identified using envelope analysis, followed by the extraction of 12 features, such as energy, crest factor and peak to peak amplitude and finally the classification of the events based on the above features. Eighty nine impacts are analysed in total, where 60 impacts are categorized as valid and 29 as in-valid. It is shown that the frequency band of maximum crest factor presents the best classification performance employing K-means clustering, which is an unsupervised clustering technique. The highest correct classification rate reaches 90%, providing useful information towards coherent and accurate fault detection.

1. INTRODUCTION

Wind industry has been continuously growing over the past decades reaching new global total of 369.6GW at the end of 2014 (Global Wind Report Annual Market Update, 2014). In order to ensure system safety, profitability and uninterrupted operation, condition based maintenance (CBM) has been deployed by many owners and operators, especially in offshore wind turbines (Yang, Tavner, Crabtree, Feng, & Qiu, 2012). Condition monitoring (CM) is integrated part of CBM and specialized solutions are offered by condition monitoring system (CMS) suppliers and wind turbine (WT) manufacturers, mainly based on vibration analysis. Other non destructive techniques (NDT) which are applicable in monitoring specific subcomponents are oil debris analysis, temperature measurement, optical fiber monitoring and acoustic emission (Tchakoua et al., 2014). Regardless the condition monitoring technique, condition based maintenance is performed in the following steps: 1) data acquisition, 2) feature extraction, 3) diagnostics, 4) prognostics and 5) planning of maintenance activities (Coble & Hines, 2011).
CM systems target mainly drive train components, such as main bearings, gearbox and generator bearings, where accelerometers are mounted in strategic locations in order to ensure optimum vibration path and thus enhance fault detection. Furthermore, accelerometers are usually installed on the nacelle frame in order to record any excessive tower oscillation, due to blade, yaw and pitch related issues (Skrimpas et al., 2015). Although the pitch system is frequently subjected to a large number of failures resulting in increased downtime (Bi, Qian, Hepburn, & Rong, 2014), there is not any CMS solution able to detect these failures on time and prevent secondary damages. The latter could be possibly explained due to location of the pitch system in the nacelle hub and the nature of its function.

Modern wind turbines are equipped with pitch systems offering independent control of the three blades. A pitch cylinder is connected to each blade on one side and to the hub on the other side. The cylinder suspension system allows movement in two axes using slide bearings. Due to heavy operation, the suspension system can become loose, causing the pitch cylinder assembly to start moving irregularly leading to random impacts in early stage and one or two impacts per rotor revolution in late stage. The main issue of the above described jarring movement is damage to cables and hydraulic systems, improper pitching and poor power production. Furthermore, the replacement of the pitch system may cause substantial downtime especially in cases of extensive damage to the electric cables and hydraulic system failures (Skrimpas et al., 2015).

Considering a typical wind turbine drive train, accelerometers having adequate vibration path to the pitch system can be utilized aiming on the detection of pitch failures. Bearing in mind the sensor location described above, accelerometers monitoring the main bearings and tower are the ones closest to the pitch system. In this work, impacts captured by the accelerometer installed on the rotor-end (front) main bearing are analysed and correlated to pitch failures. The extracted features are adopted by speech, audio and environmental noise recognition methods, showing that techniques applied in sound applications can be also employed on the analysis of vibration signals. It has been observed that impacts generated by loose pitch suspension share common characteristics with gun-shots and glass breaks, which do not have any apparent substructures. Finally, the impacts are classified using K-means clustering, which is an unsupervised technique.

The structure of the paper is as follows. Section 2 describes the algorithm of impact identification which is divided into three processes, namely the identification of impacts, feature extraction and classification. Section 3 presents the results from 89 impacts classified as valid or invalid based on the feedback provided by the service crews. Finally, the discussion and conclusions are presented in sections 4 and 5.

2. Method Description

Traditionally, vibration based CMS consists of two main modules. The first step is the extraction of features from vibration signals which describe the condition of the component of interest and indicate a potential fault in case of progression or high levels. Typical features are the amplitude of spectral components associated to the operation of the monitored equipment, such as tooth mesh frequencies in gearboxes, or frequency bands which usually describes its overall status. The second block of a successful CMS is the capability of consistent alarming based on the level or progression of the related condition indicators. This stage can also provide an automated preliminary diagnosis of the fault and estimation of its severity.

Sound recognition systems are also composed of two stages, namely the extraction of features and classification (Dufaux, 2001). In the framework of evaluating solely impacts, a signal pre-processing stage is required in order to identify these events, shown in Figure 1 as impact detection. Although the core of any recognition system is the feature extraction, the effectiveness of the the impact detection is assessed to be critical on the performance of the proposed method.

A intermediate step not displayed in Figure 1 is the dimensionality reduction or feature selection. It refers to the algorithm that select the best subset of the input feature set in regards to class discrimination capabilities (Jain, Duin, & Mao, 2000).

![Figure 1. Method Description.](image)

2.1. Impact Detection

The randomness, complexity and non-stationary nature of vibration impacts generated by wind turbine pitch system failures highlights the necessity of signal pre-processing for efficient feature extraction. Signal segmentation is used extensively as pre-processing stage in applications such as cardiac sound recognition (Choi & Jiang, 2008). Algorithms, such as normalized average Shannon energy and Hilbert transformation, are very effective techniques when detection of impacts is the main objective. In this work, the envelope of the signal high frequency bandwidth is utilized as signal segmentation tool. The envelope calculation process is executed in three discrete and simple steps illustrated in Figure 2. The signal passes through a bandpass filter in order to remove any low frequency noise and restrict the bandwidth to mitigate any aliasing effects. Typically low frequency (below 1kHz) spectral components in wind turbines are generated by multi-stage gearboxes and high speed generators. The vibrations created...
by these components are commonly recorded by accelerometers installed on the main bearings or nacelle frame due to the presence of weak vibration paths to them. In addition, vibrations from loose components of minor importance, such as covers, or structural micromovement could also produce noise and random impacts. The filtered signal is then rectified, shown as a diode in Figure 2, and the outcome is an unipolar signal. Finally, a low pass filter is applied in order to compute the envelope signal. Both filters are Butterworth and their settings are listed below. It is noted that the sampling frequency $F_s$ is 25.6kHz in all presented waveforms.

The envelope signal shows the presence of three large impacts, one minor impact in the beginning of the signal and a moderate modulation matching three times the main shaft speed. In order to extract only the large impacts, a limit is defined above which the signal is segmented and characterized as impactive. Following trial and error method, a global satisfactory limit value was found to be 1.5 times the energy of the envelope signal. Figure 4 shows the envelope signal along with the limit for this case. Finally, Figure 5 shows the extracted impacts and segments. It can be seen that the event in the first segment does not have the required energy to be identified as impactive compared to the rest of the signal characteristics.

2.2. Feature Extraction

The most critical part of any recognition system is the effectiveness of the extracted features in discrimination between different classes. One of the main objectives is to improve the signal to noise ratio in order to maximize the information. There is an arsenal of features which have been studied extensively in audio, such as music and speech, and environmental sound recognition applications. Time domain features, such as zero crossing rate, spectral features, such as spectral centroid and Mel-frequency cepstral coefficients (MFCCs), and joint time-frequency characteristics have been traditionally used as features (Ghoraani & Krishnan, 2011).

The basis of this work is to utilize the features applied to sound signals in vibration time waveforms. The following subsection present the features which are extracted from each impactive segment, as the ones illustrated in Figure 5.

- **Peak** value of a signal $S(n)$ defined as the maximum positive amplitude of a time waveform.

$$A_{pk} = \max(S(n))$$  \hspace{1cm} (1)

- **Peak to peak** (P2P) is the difference between the maximum positive and the maximum negative amplitudes of a signal $S(n)$. In the current application, the main purpose of evaluating both peak and peak-to-peak is the identification of malfunctioning sensors which often present unipolar time waveforms

$$A_{pp} = \max(S(n)) - \min(S(n))$$  \hspace{1cm} (2)
• Energy $E_S$ of a discrete signal $S(n)$ is a measure of signal strength. It is defined as:

$$E_S = \frac{1}{N} |S(m)w(n-m)|^2$$  

where $w(m)$ is a window of size equal to the signal length

• Impact duration $T$ is the time duration of each extracted impactive events using the process described in section 2.1

• Power of a signal is given by the following ratio:

$$P_S = \frac{E_s}{T}$$

(4)

• Zero crossing rate (ZCR) occurs when successive samples have different signs (Chu, Narayanan, & Kuo, 2009). It is given by:

$$Z_n = \frac{1}{2} \sum_m \left| \text{sgn}(S(m)) - \text{sgn}(S(m-1)) \right| w(n-m)$$

where

$$\text{sgn}|x(n)| = \begin{cases} 
1 & x(n) \geq 0 \\
-1 & x(n) < 0
\end{cases}$$

(6)

• Standard deviation (Std) is a measure of how spread is a distribution. It is equal to

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (S(n) - \mu)^2}$$

(7)

where the mean value $\mu$ is equal to $1/N \sum_{n=1}^{N} S(n)$

• Kurtosis is a typical measure of signal peakness. It is equal to:

$$K = \frac{1/N \sum_{n=1}^{N} (S(n) - \mu)^4}{(1/N \sum_{n=1}^{N} (S(n) - \mu)^2)^2}$$

(8)

• Spectral centroid (SC) measures the brightness of a sound. The higher the centroid, the brighter the sound (Chu et al., 2009). It is equal to

$$SC = \frac{\sum_{m=1}^{M} m \cdot F(m)}{\sum_{m=1}^{M} F(m)}$$

(9)

where F stands for the Fourier Transformation of signal S.

• Spectral flatness (SF) quantifies the tonal quality, i.e. how much tone-line the sound is as opposed to being a noise (Chu et al., 2009). It is given by:

$$SF = \exp\left(\frac{\sum_{m=1}^{M} \log F(m)}{1/M \sum_{m=1}^{M} F(m)}\right)$$

(10)

• Crest factor (CF) is the ratio of peak value to the signal RMS (root mean square) value. It is commonly used in vibration analysis for bearing wear.

$$SF = \frac{S_{pk}}{S_{rms}}$$

(11)

• Maximum crest factor frequency band. (Max CF band) It is a joint time-frequency feature, which corresponds to the frequency range of 500Hz bandwidth yielding the maximum crest factor.

2.3. Dimensionality Reduction

Dimensionality reduction is defined as the selection of a feature subset of size $m$, out of a set of $d$ features, which leads to the smallest classification error (Jain et al., 2000). The most straightforward is to examine all $(d^m)$ combinations and se-
lect the subset with the lowest classification error. Although the risk of exhaustive search is apparent, the small number of features permits this method to select the feature subset. Reference (Jain et al., 2000) presents a complete list of dimensionality reduction methods, where the most popular and commonly used are principal component analysis (PCA) and Fisher’s Linear Discriminant. Bishop discusses the use of these techniques showing that Linear Discriminant Analysis shows usually better characteristics in classification problems compared to PCA.

2.4. Classification

A simple unsupervised classification technique used in many application is K-means clustering. K-means clustering groups a data set consisting of N observations \( x_1, \ldots, x_N \) of dimension D into K clusters, where the inter-point distances between them is minimized (Bishop, 2006). A binary index \( r_{nk} \), where \( n = 1, \ldots, N \) and \( k = 1, \ldots, K \) is assigned to each data point when the sum of the squares of the Euclidean distances between the cluster centres and the data is minimized, as shown in Eq. 12. Hence, each data point is assigned to the closest cluster centre.

\[
r_{nk} = \begin{cases} 
1 & \text{if } \text{argmin} \sum_{k=1}^{K} \sum_{n=1}^{N} ||x_n - \mu_k||^2 \\
0 & \text{otherwise} 
\end{cases} \tag{12}
\]

K-means clustering is based on the Expectation-Maximization (EM) algorithm, where the Expectation step corresponds to clustering the data points based on equation 12 and the cluster centres \( \mu_k \) are updated at the Maximization step. The initial clustering and cluster centres can be arbitrarily selected, given a known number of clusters K. The process is terminated after a predefined number of iterations or when a desired convergence is achieved.

3. DETECTION OF PITCH FAILURES

Detection of pitch failures is performed in multi megawatt wind turbines, whose topology is depicted in Figure 6. In addition, the installed CMS sensors are represented by red rings, where the blue ring shows the position of the speed sensor. It is shown that the front and rear main bearing accelerometers (Brüel and Kjær Vibro AS-70) are placed at the bottom and top of the bearings respectively in order to take into account the stress applied on the shaft due to the rotor weight. The recorded vibration signals are processed by the Wind Turbine Analysis System Type 3652 (WTAS Type 3652) which calculates scalar values and streams them to central servers every hour for long term trending and alarming. Furthermore, 10.24 second vibration signals recorded by the accelerometers mounted on the generator bearings, gearbox and main bearings sampled at 25.6kHz are delivered to the central servers every one or two days for detailed spectral analysis.

It is assessed that the vibration path from the pitch cylinders to the front main bearing accelerometer is the clearest, and thus this sensor will be utilized as indicator for pitch related issues.

The test set consists of 89 impacts from 35 wind turbines, where 60 impacts from 20 turbines and 29 impacts from 15 turbines are marked as valid and invalid respectively regarding the presence of a fault in the pitch cylinders or pitch suspension. The verification of an actual fault depends highly on the provided feedback from the service technicians troubleshooting the corresponding alarms. Therefore, there is a error margin on what is classified as loose suspension based on the technicians’ assessment.

Normalization of the extracted features is essential in order to obtain more consistent and effective classification. A random feature \( f \) can normalized using the following equation:

\[
f_{\text{norm}} = \frac{f - \mu_f}{\sigma_f} \tag{13}
\]

where \( \mu_f \) and \( \sigma_f \) are the mean and standard deviation of the feature population respectively.

In many cases, it is useful to obtain two- or three- dimensional projection of the features offering a visual examination of the data. The features described in section 2.2 are clustered in pairs as shown in Table 1. It is shown that the best classification performance is seen when one of the utilized features is the frequency band of maximum crest factor, reaching 90% in almost all cases. It is also important to note that the aforementioned feature provides the same results when it is the only one used. Furthermore, moderate results in the vicinity of 70% are recorded using other features, where spectral flatness and spectral centroid are among them.

Figure 7 illustrates the initial classes using spectral flatness and maximum crest factor band on the left, and the recalculated classes based on K-means clustering on the right. The red squares represent the data corresponding to pitch failures and the blue diamonds to negative feedback from the service technicians. The black stars on the right figures represent the new cluster centres. In this case, approximately 90% of the data are assigned to their initial classes. The class separation is maximized in terms of distance between the cluster centres as compared to Figure 8, where spectral flatness and spectral centroid are used. At this point, it is important to note that the misclassified points are mainly related to actual faults which are assigned to the ”no-fault” cluster. However, the above phenomenon could be linked to the the presence of both valid and invalid impacts in the same analysed time waveform. The effectiveness of maximum crest factor band
Figure 6. Wind turbine topology and CMS sensor location.

Table 1. Correct classification percentage when reclustering the extracted features in pairs.

<table>
<thead>
<tr>
<th></th>
<th>Peak</th>
<th>P2P</th>
<th>Energy</th>
<th>Duration</th>
<th>Power</th>
<th>ZCR</th>
<th>SC</th>
<th>Std</th>
<th>Kurtosis</th>
<th>SF</th>
<th>CF</th>
<th>Max CF band</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak</td>
<td>40.67</td>
<td>40.67</td>
<td>35.28</td>
<td>72.26</td>
<td>35.28</td>
<td>40.67</td>
<td>53.95</td>
<td>40.67</td>
<td>41.84</td>
<td>70.22</td>
<td>62.04</td>
<td>90.32</td>
</tr>
<tr>
<td>P2P</td>
<td>40.67</td>
<td>40.67</td>
<td>35.28</td>
<td>72.26</td>
<td>37.97</td>
<td>40.76</td>
<td>53.95</td>
<td>40.67</td>
<td>41.84</td>
<td>70.22</td>
<td>62.04</td>
<td>90.32</td>
</tr>
<tr>
<td>Energy</td>
<td>35.28</td>
<td>35.28</td>
<td>35.28</td>
<td>63.00</td>
<td>35.28</td>
<td>57.25</td>
<td>37.97</td>
<td>38.63</td>
<td>70.22</td>
<td>62.70</td>
<td>90.32</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>72.26</td>
<td>72.26</td>
<td>63.00</td>
<td>35.28</td>
<td>57.25</td>
<td>59.79</td>
<td>48.25</td>
<td>68.54</td>
<td>71.09</td>
<td>61.92</td>
<td>90.32</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>35.28</td>
<td>35.28</td>
<td>35.28</td>
<td>57.25</td>
<td>57.25</td>
<td>59.79</td>
<td>48.25</td>
<td>58.32</td>
<td>39.14</td>
<td>70.22</td>
<td>62.70</td>
<td>90.32</td>
</tr>
<tr>
<td>ZCR</td>
<td>40.67</td>
<td>40.67</td>
<td>57.25</td>
<td>59.79</td>
<td>57.25</td>
<td>59.28</td>
<td>54.25</td>
<td>54.25</td>
<td>70.43</td>
<td>70.94</td>
<td>73.63</td>
<td>82.74</td>
</tr>
<tr>
<td>SC</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>53.95</td>
<td>70.94</td>
<td>70.94</td>
<td>73.63</td>
<td>82.74</td>
</tr>
<tr>
<td>Std</td>
<td>40.67</td>
<td>46.06</td>
<td>37.97</td>
<td>48.25</td>
<td>37.97</td>
<td>48.25</td>
<td>53.95</td>
<td>48.25</td>
<td>41.84</td>
<td>70.22</td>
<td>59.85</td>
<td>90.32</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>41.84</td>
<td>41.84</td>
<td>38.63</td>
<td>68.54</td>
<td>39.14</td>
<td>67.52</td>
<td>70.43</td>
<td>41.84</td>
<td>35.94</td>
<td>70.22</td>
<td>56.13</td>
<td>90.32</td>
</tr>
<tr>
<td>SF</td>
<td>70.22</td>
<td>70.22</td>
<td>70.22</td>
<td>71.09</td>
<td>70.22</td>
<td>68.39</td>
<td>70.94</td>
<td>70.22</td>
<td>70.22</td>
<td>66.71</td>
<td>72.11</td>
<td>88.14</td>
</tr>
<tr>
<td>CF</td>
<td>62.04</td>
<td>62.04</td>
<td>62.70</td>
<td>61.92</td>
<td>62.70</td>
<td>62.79</td>
<td>73.63</td>
<td>59.85</td>
<td>56.13</td>
<td>66.71</td>
<td>72.11</td>
<td>88.14</td>
</tr>
<tr>
<td>Max CF band</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
<td>90.32</td>
</tr>
</tbody>
</table>

as classifier is also displayed in Figure 9 where it is the only used feature.

In order to investigate the optimum feature subset which yields the lowest classification error, all combinations were examined as discussed in section 2.3. No improvement has been observed, whereas in many combinations moderate to high increase of misclassified data was registered. As for example, Figure 10 depicts the clusters when spectral flatness, spectral centroid and maximum crest factor band are used. The proper classification percentage in this case was approximately 87%.

4. DISCUSSION

The method described in the previous sections is the first approach by the authors to correlate single or repetitive vibration impacts recorded from the front main bearing accelerometer in wind turbines to failures related to the pitch assembly. The validity of the technique and results depends highly on two factors; the provided feedback from the field and the extracted features. The human factor and thoroughness on reporting the presence of a fault is a parameter which cannot be explicitly quantified and it will always add uncertainty on the method. The feature extracted in the time-frequency domain presented the best performance compared to conventional spectral or temporal features, suggesting that the correct classification percentage can be further improved if more features are tested. Furthermore, the impact population un-

Figure 7. Original and new clusters using spectral flatness and maximum crest factor band. Correct classification is at 90.32%.
Figure 8. Original and news clusters using spectral flatness and spectral centroid. Correct classification is at 70.94%.

Figure 9. Original and news clusters using only maximum crest factor band feature. Correct classification is at 90.32%.

Figure 10. Original and news clusters employing three features, namely spectral flatness, spectral centroid and maximum crest factor band.

der investigation, i.e. 89 impacts, is considered relatively limited so as to serve as a platform for establishing a database or training set. The limited number of confirmed impacts has been the main motivation for using an unsupervised clustering technique and not employing advanced classification methods, such as space vector machine (SVM), Gaussian mixture models (GMM) or neural networks (NN).

An important aspect of the impact classification method is the lack of severity estimation. Although the evaluation of the remaining useful lifetime of the pitch assembly is assessed to be challenging, the proper identification of the issue could assist the maintenance organization to plan its troubleshooting more efficiently. Based on the nature of the described failure mode, a single impact over 10.24s, i.e. approximately three rotor revolutions, is usually an early failure indicator, whereas the presence of repetitive impacts matching the rotor running speed suggests severe looseness of the pitch suspension or cylinders. It is the belief of the authors that a more complete impact database could potentially offer the ground for holistic condition monitoring of pitch systems in wind turbines.

5. CONCLUSIONS

The present work shows an effective technique to diagnose pitch assembly malfunctions, an area which has received little attention, by analysing vibration signals recorded in wind turbine main bearing accelerometers. The proposed impact recognition scheme consists of three blocks, namely impact detection, feature extraction and classification, inspired by research areas not related to machinery diagnostics, such as in environmental noise and speech recognition systems. Consistent impact recognition is achieved using envelope analysis, where the limit for assigning an event as impact was found to be equal to 1.5 times the envelope signal energy. The maximum correct classification percentage reaches 90% in a test sample of 89 impacts. Out of the 12 extracted features, the best classification characteristics are seen on a joint time-frequency feature representing the spectral bandwidth of maximum crest factor. Conventional spectral and temporal features present poorer class discrimination tendency ranging from 30% to 70%.

REFERENCES


stitute of Microtechnology, University of Neuchatel, Switzerland.


**Biographies**

Georgios Alexandros Skrimpas received the Diploma in electrical and computer engineering from the Aristotel University of Thessaloniki, Greece, in 2009 and the M. Sc. in wind energy from the Technical University of Denmark (DTU) in 2012. He joined Brüel and Kjær Vibro in 2012 and since 2013 he is pursuing the Industrial Ph.D. degree at the Centre of Electric Power and Energy at DTU in cooperation with Brüel and Kjær Vibro. His research interests are diagnosis and prognosis of electrical and mechanical faults in wind turbines.

Kun S. Marhadi is an engineer in the Remote Monitoring Group at Brüel and Kjaer Vibro. He joined Brüel and Kjaer Vibro in 2012. Previously, he worked as a postdoctoral fellow in the Department of Mathematics at the Technical University of Denmark (DTU). He received his PhD in computational science in 2010 from San Diego State University and Claremont Graduate University. He has M.S. and B.S. in aerospace engineering from Texas A&M University. His expertise is in structural vibration and analyses, probabilistic methods, and design optimization.

Robert Erick Gomez received a Bachelor of Science in Mechanical Engineering from the University of Florida in 2010. He began his career in the power generation field in 2008, with his key focus being to optimize productivity and efficiency in rotating equipment. He joined Bruel and Kjaer Vibro in 2013 as diagnostic engineer. His research interests are diagnosis of faults on wind turbine drive train components.

Christian Walsted Sweeney received his B.Sc. degree from the University of Southern Denmark in 2006 and M.Sc degree from the Technical University of Denmark in 2008 both in mechanical engineering. From 2008 to 2010 he was employed by Bruel and Kjaer Vibro as diagnostic engineer and since 2010 he is the team leader of the diagnostic services group. His research focus is on the development of condition monitoring systems and handling of large data quantities.

Bogi Bech Jensen received the Ph.D. degree from Newcastle University, Newcastle Upon Tyne, U.K., for his work on induction machine design. He was in various engineering and academic positions in the marine sector from 1994 to 2004. He was at Newcastle University from 2004 to 2010 first as a Postgraduate, then Research Associate and finally as a Lecturer. From 2010 to 2014 he was Associate Professor and later Head of Research Group at the Technical University of Denmark (DTU), Lyngby, Denmark. He is currently Professor of Energy Engineering at the University of the Faroe Islands (UFI), where he is responsible for education and research in energy. Prof. Jensen is Associate Editor of IEEE Transactions on Industry Applications and a Senior Member of IEEE.

Nenad Mijatovic received his Ph.D. degree from the Technical University of Denmark for his work in superconducting machine. After obtaining his Dipl.Ing. education at University of Belgrade, Serbia, he enrolled as a doctoral candidate in 2012. Upon completion of the Ph.D, he has continued to work in the same field of machine research - superconducting machines, as an Industrial PostDoc. The 3 year industrial PostDoc grant has been provided by Hojteknologifonden and supported by Envision Energy Aps., Denmark. Dr. N. Mijatovic is a member of IEEE from 2008 and his field of interest and research includes novel electrical machine design, operations and diagnostic.

Joachim Holboell is associate professor and deputy head of center at the Technical University of Denmark, Department of Electrical Engineering, Center for Electric Power and Energy. His main field of research is high voltage components, their properties, condition and broad band performance, including insulation systems performance under AC, DC and transients. Focus is also on wind turbine technolgy and future power grid applications of components. J. Holboell is Senior Member of IEEE.
PHM Based Predictive Maintenance Option Model for Offshore Wind Farm O&M Optimization

Xin Lei1, Peter A. Sandborn1, Navid Goudarzi1 and Maira A. Bruck1

1CALCE, Department of Mechanical Engineering, College Park, MD, 20742
xlei@umd.edu
sandborn@umd.edu
navid@umd.edu
mbruck@termail.umd.edu

ABSTRACT
A simulation-based real options analysis (ROA) approach is used to determine the optimum predictive maintenance opportunity for multiple wind turbines with remaining useful life (RUL) predictions in offshore wind farms managed under outcome-based contracts, i.e., power purchase agreements (PPAs). When an RUL is predicted for a subsystem in a single turbine using PHM, a predictive maintenance option is triggered that the decision-maker has the flexibility to decide if and when to exercise before the subsystem or turbine fails. The predictive maintenance value paths are simulated by considering the uncertainties in the RUL predictions and wind speeds (that govern the turbine’s revenue earning potential). By valuating a series of European options expiring on all possible predictive maintenance opportunities, a series of option values can be obtained, and the optimum predictive maintenance opportunity can be selected. The optimum predictive maintenance opportunity can also be determined using a stochastic discounted cash flow (DCF) approach that assumes the predictive maintenance will always be implemented on the selected opportunity. For a wind farm managed via a PPA with multiple turbines indicating RULs concurrently, the predictive maintenance value for each turbine depends on the operational state of the other turbines, the amount of energy delivered and to be delivered by the whole wind farm. A case study is presented in which the stochastic DCF and European ROA approaches are applied to a single turbine and to a wind farm managed via a PPA. The optimum predictive maintenance opportunities obtained from the two approaches are compared and it is demonstrated that the European ROA approach will suggest a more conservative opportunity for predictive maintenance with a higher expected option value than the expected net present value (NPV) from the stochastic DCF approach.

1. INTRODUCTION
The global cumulative wind power capacity at the end of 2013 was 318,105 megawatts (MW), representing an average annual growth of approximately 25% over the last 10 years (Fried, Sawyer, Shukla, and Qiao, 2014a). For offshore wind, at the end of 2013 the global cumulative capacity was roughly 6.8 gigawatts (GW), of which 6.6 GW was in the European Union (EU), providing 0.7% of the EU’s total energy consumption (Fried, Shukla, Sawyer, and Teske, 2014b).

Operation and maintenance (O&M) cost, as a major contributor to the wind levelized cost of energy (LCOE), accounts for 0.027 to 0.048 US dollars/kilowatt-hour (USD/kWh), (IRENA Secretariat, 2012). Maintenance for wind turbines has been categorized as scheduled preventive maintenance, corrective maintenance and predictive maintenance (Karyotakis, 2011; Kovacs, Erdos, Viharos, and Monostori, 2011; Nilsson & Bertling, 2007). The cost of corrective maintenance (after failure happens) is expensive for offshore wind farms, since it requires expensive resources such as vessels, and maintenance windows are limited due to the harsh marine environment (Kovacs et al., 2011).

Prognostics and Health Management (PHM) technologies have been introduced into wind turbines to assess the reliability and forecast remaining useful life (RUL) of key subsystems (Haddad, Sandborn, and Pecht, 2014). PHM based predictive maintenance is expected to reduce the wind farm O&M cost (Tchakoua, Wamkeue, Ouhrouche, Slaooui-Hasnaoui, Tameghe, and Ekemb, 2014). Once a PHM indication and a RUL prediction is triggered for a subsystem in a turbine, the maintenance decision-maker needs to decide if and when to perform the predictive maintenance. To address this challenge, Haddad et al. (2014) treated the predictive maintenance opportunities as American style real options. An American real option can be exercised on or prior to a predetermined expiration time (Kodukula & Papudesu, 2006). Haddad et al. (2014) determined the latest predictive maintenance opportunity (the optimum American real option

Xin Lei et al. is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
expiration time) by minimizing the risk of expensive corrective maintenance after failures, while reducing the RUL thrown away by predictive maintenance.

Lei, Sandborn, Bakhshi, and Kashani-Pour (2015) developed a European style Real Options Analysis (ROA) approach based on Haddad et al. (2014). Different from the American real option, a European real option can only be exercised on the predetermined expiration time (Kodukula and Papudesu, 2006). Lei et al. (2015) determined the optimum predictive maintenance opportunity (the optimum European real option expiration time), and extended the European ROA approach to multiple turbines with remaining useful life (RUL) predictions in offshore wind farms managed under outcome-based power purchase agreements (PPAs). Lei et al. (2015) considered the operational status of the other turbines, the lower price for over-delivered energy, and the penalties for under-delivered energy defined by PPAs.

During the predictive maintenance value formulation, both Haddad et al. (2014) and Lei et al. (2015) only considered the cumulative revenue earned between the RUL indication and the predictive maintenance opportunities. However, if the predictive maintenance is not implemented and the turbine is run to failure (where corrective maintenance occurs), more revenue can be earned. Therefore, to reflect the true value of predictive maintenance, the difference between the cumulative revenue earned up to either predictive or corrective maintenance should be considered, and the revenue lost due to predictive maintenance should be included in the analysis.

According to Lei et al. (2015), the European ROA approach assumes that the predictive maintenance is an option but not an obligation, and will only be implemented if the predictive maintenance value is higher than the predictive maintenance cost. Alternatively, the optimum predictive maintenance opportunity can also be determined by a stochastic discounted cash flow (DCF) approach that assumes the predictive maintenance will always be implemented at the selected opportunity no matter how much the predictive maintenance value is.

In this paper, for multiple turbines indicating RULs in an offshore wind farm managed via a PPA, the optimum predictive maintenance opportunity is determined. The time-history cost avoidance and cumulative revenue lost paths are simulated and combined to form the predictive maintenance value paths. By applying the simulation-based European ROA approach (Lei et al., 2015), a series of predictive maintenance options are evaluated by considering all possible maintenance opportunities. Assuming that all turbines with RULs are maintained concurrently, the optimum predictive maintenance opportunity can be determined as the one with the maximum option value. The stochastic DCF approach is also applied to the simulated predictive maintenance value paths, and the results from the two approaches are compared.

The remainder of the paper is structured as following: Section 2 explains the European ROA and stochastic DCF approaches. Section 3 presents a case study for the two approaches applied to both a single turbine and multiple turbines indicating RULs. Finally, Section 4 concludes the work and discusses future research opportunities.

2. ANALYSIS METHODOLOGY

We assume an offshore wind farm is operated under a PPA. At time $t_0$, $K$ turbines are indicating RULs (while $J$ turbines operate normally without RUL indications). Each RUL is predicted for some subsystem (e.g., for the gearbox or main shaft in cycles), and that subsystem will fail before the end of the year (called $EOY$) if the predictive maintenance is not implemented. Once the subsystem fails, the turbine will fail. From $t_0$ to $EOY$ there are multiple discrete predictive maintenance opportunities, and the decision-maker wants to decide which predictive maintenance opportunity should be scheduled for all $K$ turbines. If the predictive maintenance is not implemented, there will be a corrective maintenance event at $EOY$ to fix all failed turbines and restore them to operation.

Using the wind speed historical data from the National Data Buoy Center (NDBC) Station 44009 (National Data Buoy Center, 2013), $M$ buoy height wind paths can be simulated according to Lei et al. (2015), each of which represents a possible future wind profile for the whole wind farm.

For each turbine with a RUL indication, a unique triangular distribution is assumed to represent uncertainties in the subsystem RUL prediction as used in Sandborn and Wilkinson (2007). For each simulated wind path, Monte Carlo simulation can be used to obtain an actual RUL sample (called $ARUL_s$, e.g., in cycles) for turbine $k$, then time to failure for turbine $k$ ($TTF_k$) can be obtained as the actual time to failure in calendar time according to Lei et al. (2015). $M$ $TTF_k$ samples can be simulated for turbine $k$, and then this procedure is repeated for all $K$ turbines.

2.1. Power Purchase Agreement (PPA) Modeling

A PPA is an outcome-based contract between a seller who generates electricity and the buyer who wants to purchase electricity. Wind farms are typically under PPAs for several reasons. First, although wind power can be sold directly into the local market, the average local market energy prices that vary daily and hourly tend to be lower than the contract prices defined in PPAs, (Stoel Rives Wind Team, 2014). Second, PPAs guarantee to the buyer and the seller that the energy generated and delivered will be paid for at the agreed price schedule. Third, as shown by Barradale (2008), utilities don’t
want to build and operate their own wind farms; they prefer to simply purchase power.

PPA terms are typically 20 years (Stoel Rives Wind Team, 2014). Barradale (2008) made the observation that PPAs often set annual energy delivery targets. The contract price can be either constant or escalated annually throughout the whole term. For each year, the buyer will generally agree to pay for all power generated and delivered to a specified transmission point. However, a maximum and a minimum annual energy delivery limit can also be set. Once the seller has delivered beyond the specified maximum limit, the buyer may choose to buy at a lower excess price or not to buy the excess energy at all. (Bonneville, 2007; Gloucester, 2011; Sonoma, 2014). The buyer may also have the right to adjust the annual target of the next year downward for the amount of energy over-delivered (Anaheim, 2003; Delmarva, 2008; World Bank, 2002; Xcel, 2013). If the seller is unable to reach the minimum limit, then the seller may have to compensate the buyer for the energy not produced at a predefined price (Delmarva, 2008; PacifiCorp, 2008; World Bank, 2002). The buyer may also adjust the annual target of the next contract year upward to compensate for the under-delivered amount (Anaheim, 2003).

As described in Lei et al. (2015), we assume in the PPA governing the wind farm, there is a constant annual energy delivery target (which is also the maximum/minimum annual energy delivery limit) set at the beginning of each year, reflecting the buyer’s exact annual demand. During each year, if the delivery target is reached, the energy generated by the whole farm will be priced by a constant contract price; if it is not reached, a lower constant excess price applies for all power generated thereafter until the Eoy. On the other hand, at Eoy, if the target is not met, the buyer has to purchase energy from other sources with a price higher than contract price (called replacement price) to fulfill the demand, and the seller must pay the buyer a compensation equals to the shortfall amount of energy priced by the difference between the replacement and contract price.

The next step is to develop a PPA framed revenue and penalty calculation model. We assume that the turbine energy generation capacity will not degrade as damage accumulates in the subsystems, and the downtime for predictive maintenance is negligible.

If the predictive maintenance is going to be implemented on all K turbines at time t, the cumulative energy generated by the whole wind farm from the beginning of the year (BOY) to time t, $EC_{PM}(t)$ can be calculated as

$$EC_{PM}(t) = EC(t_0) + \sum_{t=t_0+1}^t \sum_{j=1}^J \sum_{k=1}^K E_{PM,j,k}(t)$$

where $EC(t_0)$ is the cumulative energy delivered by the whole wind farm from BOY to time $t_0$, $E_j(t)$ and $E_{PM,j,k}(t)$ are the energy generated by turbine $j$ (the $j$th turbine operates normally without RUL indication) and $k$ (the $k$th turbine indicating RUL), respectively, from time $t-1$ to $t$ according to Lei et al. (2015).²

The revenue earned from time $t-1$ to $t$ by all J and K turbines $R_{PM,j}(t)$ and $R_{PM,K}(t)$, respectively, can be calculated as

$$R_{PM,j}(t) = PPM(\tau) \cdot \sum_{j=1}^J E_j(t)$$

(2)

$$R_{PM,K}(t) = PPM(\tau) \cdot \sum_{k=1}^K E_{PM,k}(t)$$

(3)

where $PPM(\tau)$ is the price per unit of energy at time $\tau$ with predictive maintenance implemented at time $t$, defined as

$$PPM(\tau) = \begin{cases} PC & E_{PM}(\tau) \leq ET \\ PE & E_{PM}(\tau) > ET \end{cases}$$

(4)

where $PC$ is the constant contract price, $PE$ is the constant excess price, and $ET$ is the annual energy delivery target for the wind farm.

The cumulative revenue earned from time $t_1$ to $t_2$ by all J turbines and by the whole wind farm $RC_{PM}(t_1,t_2)$ and $RC_{PM}(t_1,t_2)$, respectively, can be calculated as

$$RC_{PM,K}(t_1,t_2) = \sum_{t=t_1+1}^{t_2} R_{PM,K}(t)$$

(5)

$$R_{PM}(t_1,t_2) = \sum_{t=t_1+1}^{t_2} R_{PM,j}(t) + R_{PM,K}(t_1,t_2)$$

(6)

If $ET$ hasn’t been met at Eoy, there will be under-delivery compensation $UP_{PM}$ paid by the seller to the buyer calculated as

$$UP_{PM} = \begin{cases} (ET - E_{PM}(\text{Eoy})) \cdot (PR - PC), & E_{PM}(\text{Eoy}) < ET \\ 0, & E_{PM}(\text{Eoy}) \geq ET \end{cases}$$

(7)

where $PR$ is the constant replacement price.

Similarly, if the predictive maintenance is not going to be implemented on all K turbines before Eoy, the corrective maintenance will fix all failed K turbines at Eoy. The cumulative energy generated by the whole wind farm from BOY to time t, $EC_{CM}(t)$ can be calculated as

$$EC_{CM}(t) = EC(t_0) + \sum_{t=t_0+1}^t \sum_{j=1}^J \sum_{k=1}^K E_{CM,j,k}(t)$$

(8)

where $E_{CM,j}(t)$ is the energy generated by turbine $j$ from time $t-1$ to $t$, calculated as

$$E_{CM,j,k}(t) = \begin{cases} E_{PM,j,k}(t), & t_0 < \tau < TTF_k \\ 0, & TTF_k \leq \tau \leq \text{Eoy} \end{cases}$$

(9)

When turbine $k$ fails at $TTF_k$, it will be down for the corrective maintenance event at Eoy.

---

² For detailed calculation method for $E_j(t)$ and $E_{PM,j,k}(t)$ see Lei et al. (2015).
The revenue earned from time \( t-1 \) to \( t \) by all \( J \) and \( K \) turbines \( R_{CM,J}(t) \) and \( R_{CM,K}(t) \), respectively, can be calculated as

\[
R_{CM,J}(t) = P_{CM}(t) \cdot \sum_{j=1}^{J} E_j(t) \tag{10}
\]
\[
R_{CM,K}(t) = P_{CM}(t) \cdot \sum_{k=1}^{K} E_{CM,K}(t) \tag{11}
\]

where \( P_{CM}(t) \) is the energy price at time \( t \) with predictive maintenance not implemented before \( EOY \), defined as

\[
P_{CM}(t) = \begin{cases} 
PC, & E_{CM}(t) \leq ET \\
P_E, & E_{CM}(t) > ET 
\end{cases} \tag{12}
\]

The cumulative revenue earn from time \( t_1 \) to \( t_2 \) by all \( K \) turbines and by the whole wind farm \( RC_{CM,K}(t_1,t_2) \) and \( RC_{CM}(t_1,t_2) \), respectively, can be calculated as

\[
RC_{CM,K}(t_1,t_2) = \sum_{t=t_1}^{t_2} R_{CM,K}(t) \tag{13}
\]
\[
RC_{CM}(t_1,t_2) = \sum_{t=t_1}^{t_2} R_{CM,J}(t) + RC_{CM,K}(t_1,t_2) \tag{14}
\]

The under-delivery compensation \( UP_{CM} \) paid by the seller to the buyer at \( EOY \) can be calculated as

\[
UP_{CM} = \begin{cases} 
(ET - E_{CM}(EOY)) \cdot (PR - PC), & E_{CM}(EOY) < ET \\
0, & E_{CM}(EOY) \geq ET 
\end{cases} \tag{15}
\]

2.2. Predictive Maintenance Value Simulation

If predictive maintenance is implemented on all \( K \) turbines at time \( t \), the cumulative revenue earned by all \( K \) turbines from \( t_0 \) to \( t \) is \( RC_{PM,K}(t_0,t) \); if corrective maintenance is implemented on all \( K \) turbines at \( EOY \), the cumulative revenue earned by all \( K \) turbines from \( t_0 \) to \( t \) is \( RC_{CM,K}(t_0,t) \).

The predictive maintenance value \( V(t) \) at time \( t \), representing the extra value obtained by carrying out the predictive maintenance on all \( K \) turbines at time \( t \) rather than waiting for the corrective maintenance at \( EOY \), is defined as

\[
V(t) = (RC_{PM,K}(t_0,t) - RC_{CM,K}(t_0,EOY)) + CA(t) \tag{16}
\]

where \( t_0 < t < TTF_{min} \), and \( TTF_{min} \) is the shortest \( TTF \) of all \( K \) turbines. It is assumed that all \( K \) turbines will be maintained predictively together before \( TTF_{min} \). Therefore once the first turbine failure happens, the predictive maintenance option expires, and the value path simulation will be stopped. The first item in parentheses reflects the revenue lost or the value of the RUL thrown away due to predictive maintenance. The earlier the predictive maintenance is scheduled, the more revenue will be lost (more of RUL will be wasted). The second item represents the cost avoidance by replacing corrective maintenance with predictive maintenance, can be calculated as

\[
CA(t) = \sum_{k=1}^{K} C_{CM,k} + (UP_{CM} - UP_{PM}) + RL \tag{17}
\]

Figure 1 shows a graphical representation of Eq. (16).

In Eq. (17), \( C_{CM,k} \) is the corrective maintenance cost for turbine \( k \) at \( EOY \), which includes the cost of parts, equipment and facilities and labor. The second item in parentheses is the under-delivery penalty due to corrective maintenance, and \( RL \) is the revenue lost during downtime for corrective maintenance at \( EOY \), can be calculated as

\[
RL = RC_{PM}(t,EOY) - RC_{CM}(t,EOY) \tag{18}
\]

2.3. Stochastic DCF Approach

The predictive maintenance can be seen as an investment, and the predictive maintenance value can be treated as its gross
profit. If we assume that the predictive maintenance will always be implemented at some selected opportunity, the optimum predictive maintenance opportunity can be determined by optimizing the net profit of the predictive maintenance as

$$NPV(t) = \begin{cases} V(t) - \sum_{k=1}^{K} C_{PM,k}, & t_0 < t < TTF_{min} \\ 0, & TTF_{min} \leq t \leq EOY \end{cases}$$

where $NPV(t)$ is the net present value (called NPV) at $t_0$ of the predictive maintenance implemented on all $K$ turbines at $t$, and this is called the stochastic DCF approach. $C_{PM,k}$ is the predictive maintenance cost for turbine $k$ at time $t$, including cost of parts, equipment and facilities, and labor. The discount rate is ignored assuming the time period from time $t_0$ to $t$ is short. When $t_0 < t < TTF_{min}$, $NPV(t)$ can also be expressed as

$$NPV(t) = \left( R_{PM,k}(t_0, t) - \sum_{k=1}^{K} C_{PM,k} - UP_{PM} \right) - \left( R_{CM,k}(t_0, EOY) - \sum_{k=1}^{K} C_{CM,k} - UP_{CM} - RL \right)$$

where the first item in parentheses is the present net profit of predictive maintenance on all $K$ turbines at time $t$, and the second item in parentheses is the present net profit of corrective maintenance on all $K$ turbines at time $EOY$.

Equations (19) or (20) can be used to value the NPVs of all possible maintenance opportunities after $t_0$, then the optimum predictive maintenance opportunity can be selected that generates the maximum NPV.

### 2.4. European ROA Approach

There is an implicit assumption in Eqs. (19) and (20) that the predictive maintenance will be implemented at the selected optimum maintenance opportunity whether the NPV is positive, zero or negative. According to Eq. (20), if the present net profit of predictive maintenance is lower than corrective maintenance, a negative NPV will be generated. In other words, replacing corrective maintenance with predictive maintenance will not always be beneficial, which is the limitation of the stochastic DCF approach.

It is reasonable to assume that the decision-maker is willing to schedule a predictive maintenance only if it is more beneficial than corrective maintenance (a positive NPV is generated from Eq. (19) or (20)), otherwise it is better to have all $K$ turbines run to failure for corrective maintenance. Therefore, as demonstrated in Lei et al. (2015), the predictive maintenance opportunities that follow PHM prediction for wind turbines can be treated as real options, and on each opportunity, a European ROA can be applied to valuate the predictive maintenance option as a “European” style option

$$OV(t) = \begin{cases} \max \{V(t) - \sum_{k=1}^{K} C_{PM,k}, 0\}, & t_0 < t < TTF_{min} \\ 0, & TTF_{min} \leq t \leq EOY \end{cases}$$

where $OV(t)$ is the present option value at $t_0$ of the predictive maintenance implemented on all $K$ turbines at $t$. The risk free rate is ignored for the short time period from time $t_0$ to $t$.

By applying the European ROA approach, we assume before $TTF_{min}$ on each predictive maintenance opportunity, if the predictive maintenance value is higher than the predictive maintenance cost, it will be implemented on all $K$ turbines; otherwise, all $K$ turbines will be run to failure, and the option value is 0. After $TTF_{min}$, the option expires and the option value is 0.

An ROA process can be implemented to valuate the option values of all possible maintenance opportunities after $t_0$ as a series of European options. The optimum predictive maintenance opportunity can be selected as the opportunity with highest predictive maintenance option value.

It is worth mentioning that the decision-maker may also want to schedule predictive maintenance for each of the $K$ turbines individually, in that case the predictive maintenance value paths can be generated for each of the $K$ turbines till its own $TTF$. Then the European ROA can be applied to each turbine to determine its own optimum predictive maintenance opportunity, which may be different from each other. Due to the harsh environment and limited availability of the maintenance resources, in reality the decision-maker may prefer to maintain multiple turbines during a single visit to the farm, as assumed in the presented model.

During the valuation process for each predictive maintenance opportunity, the stochastic DCF approach has to carry out the predictive maintenance, while the European ROA approach has the flexibility and may choose not to carry out the predictive maintenance if corrective maintenance is more beneficial.

### 3. Case Study

In this section, the European ROA and stochastic DCF approaches are applied to a single turbine and a wind farm managed via a PPA. The optimum predictive maintenance opportunities obtained from the two approaches are compared.

1000 wind paths are simulated from $t_0$ to $EOY$ by using the method described in Section 2. The wind turbines under study are Vestas V-112 3.0 MW offshore turbines, with cut-in, cut-out and rational speeds of 3 m/s, 25 m/s and 12 m/s, respectively, and a nominal rotational speed of 14 RPM (Vestas, 2013).

### 3.1. Predictive Maintenance Optimization for Single Turbine

We assume there is a single offshore wind turbine operated under a PPA. $ET$ is 8000 MWh, $PC$, $PE$ and $PR$ are $20/\text{MWh}$, $10/\text{MWh}$ and $40/\text{MWh}$, respectively. At $t_0 = 8400$ hrs when $EC(t_0)$ is 7800 MWh, a PHM indication is
triggered and a RUL of 100,000 cycles is predicted for a key subsystem (e.g., the main shaft). The width of the RUL triangular distribution is 200,000 cycles. Using Monte Carlo simulation, 1000 TTF samples are obtained.

By applying Eqs. (1) to (5), 1000 cumulative revenue paths are simulated if predictive maintenance is implemented. Similarly, using Eqs. (8) to (13), 1000 cumulative revenue paths with corrective maintenance are obtained.

Predictive and corrective maintenance costs are assumed to be $10,000 and $9000, respectively. Using Eqs. (6), (7), (14), (15), (17) and (18), 1000 cost avoidance paths are simulated. Using Eq. (16), the predictive maintenance value paths are obtained, as illustrated in Figure 2.

![Figure 2. Predictive maintenance value paths for one turbine. 1000 paths are shown.](image)

As shown in Figure 2, while the cost avoidance is staying constant (see Figure 1), since the revenue lost due to predictive maintenance decreases over time, all value paths are ascending. Each path terminates at a different time point when the RUL is used up, which represents the uncertainties in the predicted RUL and wind speeds. The earlier the RUL is used up, the higher the path’s initial value is; it is because the revenue lost during downtime for corrective maintenance is also larger. The change in slopes of some paths indicate that ET is reached and then the PE is applied.

For offshore wind turbines, predictive maintenance opportunities are not continuously available. We assume the predictive maintenance is available every 2 days. For the simulated 1000 predictive maintenance value paths, using Eq. (23), 1000 option value paths are obtained. At each predictive maintenance opportunity, all option values are averaged to get the expected option present value as shown in the left graph of Figure 3. In order to compare it to the expected NPV obtained from DCF method, the stochastic DCF approach is also applied to get the expected NPV as a comparison, and the results are shown in the right graph of Figure 3.

As can be seen in Figure 3, the optimum predictive maintenance opportunity predicted by the European ROA approach is 4 days (96 hours) after \( t_0 \), with a higher expected value of $1,563 when 13.5% turbine samples have failed. Stochastic DCF approach suggests 6 days (144 hours) after \( t_0 \) with a lower expected NPV of $1,363 when 31.6% turbine samples have already failed. Since The European ROA approach is an asymmetric approach that only captures the upside value (when predictive maintenance is more beneficial) while limiting the downside risk (when corrective maintenance is more beneficial), it suggests to implement predictive maintenance earlier. Also, because of this asymmetric characteristic, at each maintenance opportunity, the expected option value from the European ROA approach is always greater than or equal to the expected NPV from the stochastic DCF approach. The difference is the additional value provided by the flexibility that the real option approach correctly models.

![Figure 3. Left – expected predictive maintenance option present value (from European ROA approach) and right - expected predictive maintenance net present value (from stochastic DCF approach) for one turbine.](image)

If the predictive maintenance value is higher than the predictive maintenance cost at all predictive maintenance opportunities due to high revenue lost, under-delivery penalty or expensive corrective maintenance cost, then there will be no differences between the results from the European ROA and stochastic DCF approach. As it is shown in Figure 4, under the assumption of having a high corrective maintenance cost of $50,000 and keeping all other parameters the same, both approaches suggest the same result for the predictive maintenance: 2 days (48 hours) after \( t_0 \).
Therefore, unless the predictive maintenance value is much higher than the predictive maintenance cost, the European ROA approach offers a more conservative opportunity to schedule predictive maintenance. This means that when the maintenance crew arrives on the suggested maintenance date, the probability that the turbine has failed is lower, which also means a higher probability for the predictive maintenance to be implemented successfully. The European ROA approach also leads to an expected option value higher than the expected NPV from stochastic DCF approach.

Figure 4. Left - expected predictive maintenance option present value (from European ROA approach) and right - expected predictive maintenance net present value (from stochastic DCF approach) for one turbine with expensive corrective maintenance.

3.2. Predictive Maintenance Optimization for a Wind Farm

We assume there is an offshore wind farm with 5 turbines managed via a PPA with the ET of 40,000 MWh, PC, PE and PR are assumed to be the same as the one turbine case. At \( t_0 = 7800 \) hrs when \( EC(t_0) = 39,000 \) MWh, RULs are predicted for turbine 1 to be 80,000 cycles (with 160,000 cycles width triangular distribution) and for turbine 2 to be 100,000 cycles (with 200,000 cycles width triangular distribution). Assume at the same time, there are two turbines in the wind farm that are not operating. The predictive maintenance value paths can be generated for turbine 1 and 2 in Figure 5.

Assuming the predictive maintenance opportunity is once every 2 days, the expected predictive maintenance option present value and predictive maintenance net present value can be determined as shown in Figure 6. The optimum predictive maintenance opportunity according to European ROA approach is 2 days (48 hours) after \( t_0 \), and by stochastic DCF approach it becomes to 4 days (96 hours) after \( t_0 \). Again, the European ROA approach provides a more conservative opportunity with the expected option value higher than the expected NPV from stochastic DCF approach. Figure 7 shows the results when the corrective maintenance cost assumed to be $50,000; both approaches suggest optimum maintenance opportunity as 2 days (48 hours) after \( t_0 \).

Figure 5. Predictive maintenance value paths for turbine 1 and 2. 1000 paths are shown.

If there are less turbines not operating at time \( t_0 \), the optimum predictive maintenance opportunity will shift to 4 days (96 hours) after \( t_0 \) by using the ROA approach as shown in Figure 8. When two turbines are down, considering the significant revenue loss and under-delivery penalty due to corrective maintenance, the selection of optimum predictive maintenance opportunity will tend to be conservative.

Figure 6. Left - expected predictive maintenance option present value (from European ROA approach) and right - expected predictive maintenance net present value (from stochastic DCF approach) for turbine 1 and 2.
### Figure 7
Left - expected predictive maintenance option present value (from European ROA approach) and right - expected predictive maintenance net present value (from stochastic DCF approach) for turbine 1 and 2 with expensive corrective maintenance.

### Figure 8
Predictive maintenance option present value for turbine 1 and 2 when the number of turbines down is varying.

## 4. Conclusion

The objective of the work presented in this paper is to determine the optimum predictive maintenance opportunity for wind farms managed under PPAs when multiple turbines are indicating RULs. Uncertainties in the wind speed and the RUL predictions from PHM are considered, and both ROA and stochastic DCF approaches are applied. This work demonstrates that the predictive maintenance option’s flexibility to expire if the predictive maintenance value is not enough to cover the predictive maintenance cost, results in the ROA approach always having an expected option value higher than the expected NPV from stochastic DCF approach. For the same reason, the ROA approach always suggests an optimum maintenance opportunity that is earlier than the stochastic DCF approach. However, the results from two approaches are the same when the predictive maintenance value is higher than the predictive maintenance cost at all predictive maintenance opportunities.

For a wind farm managed via a PPA with multiple turbines indicating RULs concurrently, the predictive maintenance value for each turbine depends on the operational state of the other turbines, the amount of energy delivered and to be delivered by the whole wind farm. When there are many turbines not operating in the wind farm, the revenue lost and under-delivery penalties due to corrective maintenance will be significant; therefore, the selection of the optimum predictive maintenance opportunity by ROA approach tends to be more conservative.

In the future, the effects of collateral damage that causes higher corrective maintenance costs, the degradation in power generation capacity and the escalating predictive maintenance cost due to damage accumulation will be studied. The uncertainties in the predictive maintenance opportunities/windows and the energy demands will also be introduced.

## Acknowledgement

Funding for the work was provided by Exelon for the advancement of Maryland’s offshore wind energy and jointly administered by MEA and MHEC, as part of “Maryland Offshore Wind Farm Integrated Research (MOWFIR): Wind Resources, Turbine Aeromechanics, Prognostics and Sustainability”.

## Nomenclature

- $ARUL_k$: simulated actual RUL for turbine $k$
- $BOY$: beginning of the year
- $C_{CM,k}$: corrective maintenance cost of turbine $k$
- $C_{PM,k}$: predictive maintenance cost of turbine $k$
- $CA(t)$: cost avoidance obtained if predictive maintenance is implemented on $K$ turbines at time $t$
- $E_j(t)$: energy generated by turbine $j$ from $t-1$ to $t$
- $ECM_k(t)$: energy generated by turbine $k$ from $t-1$ to $t$ with turbine $k$ running to failure
- $E_{PM,k}(t)$: energy generated by turbine $k$ from $t-1$ to $t$ if predictive maintenance will be implemented
- $EC(t_0)$: cumulative energy delivered by the whole wind farm from BOY to $t_0$
The document contains mathematical expressions and references to various studies and reports on prognostics and health management in the context of offshore wind farms. The symbols used in the document include:

- $EC_{CM}(t)$: cumulative energy generated by the whole wind farm from $BOY$ to $t$ by running $K$ turbines to failure
- $EC_{PM}(t)$: cumulative energy generated by the whole wind farm from $BOY$ to $t$ if predictive maintenance will be implemented on $K$ turbines
- $EY_{O}$: end of the year
- $ET$: annual energy delivery target of the wind farm
- $I$: number of turbines operating normally at time $t$ in the wind farm
- $K$: number of turbines indicating $RUL$s at time $t$ in the wind farm
- $M$: number of simulation paths
- $NPV(t)$: expected predictive maintenance net present value at $t_{0}$ if predictive maintenance scheduled at time $t$
- $OV(t)$: Expected predictive maintenance option present value at $t_{0}$ if predictive maintenance scheduled at time $t$
- $PC$: contract price in PPA
- $PE$: excess price in PPA
- $PR$: replacement price in PPA
- $R_{CM,K}(t)$: revenue earned by $J$ turbines from $t_{1}$ to $t$ with $K$ turbines running to failure
- $R_{CM}(t)$: cumulative revenue earned by the whole wind farm from time $t_{1}$ to $t$, with $K$ turbines running to failure
- $R_{PM,K}(t)$: revenue earned by $J$ turbines from $t_{1}$ to $t$ if predictive maintenance will be implemented on $K$ turbines
- $R_{PM}(t)$: cumulative revenue earned by the whole wind farm from time $t_{1}$ to $t$, with $K$ turbines running to failure
- $R_{CM,K}(t)$: cumulative revenue earned by $K$ turbines from time $t_{1}$ to $t$ with $K$ turbines running to failure
- $R_{CM}(t)$: cumulative revenue earned by $K$ turbines from time $t_{1}$ to $t$, with $K$ turbines running to failure
- $R_{PM,K}(t)$: cumulative revenue earned by $K$ turbines from time $t_{1}$ to $t$ if predictive maintenance will be implemented on $K$ turbines
- $R_{PM}(t)$: cumulative revenue earned by $K$ turbines from time $t_{1}$ to $t$, with $K$ turbines running to failure
- $R_{PM}(t)$: cumulative revenue earned by $K$ turbines from time $t_{1}$ to $t$ if predictive maintenance will be implemented on $K$ turbines
- $RL$: revenue lost during downtime for corrective maintenance at $EOY$
- $RUL$: nominal remaining useful life in cycles
- $t$, $t'$: time of the year
- $t_{0}$: time of the year when $RUL$s are predicted and predictive maintenance decision needs to be made
- $TTF_k$: simulated time to failure of turbine $k$
- $TTF_{min}$: smallest $TTF$ of $K$ turbines
- $UP_{CM}$: under-delivery compensation if corrective maintenance implemented at $EOY$
- $UP_{PM}$: under-delivery compensation if predictive maintenance scheduled at time $t$
- $V(t)$: predictive maintenance value for $K$ turbines at time $t$

The references section includes a list of scholarly works and reports related to the topic, such as:

Xin Lei received the B.S. degree in Reliability System Engineering and the M.S. degree in Systems Engineering from the Beihang University, Beijing. He is a Ph.D. student in the CALCE Electronic Products and Systems Center (EPSC), in the Department of Mechanical Engineering at the University of Maryland, College Park, where his interests include system life cycle economics, electronic part obsolescence and prognostics and health management. Prior to attending the University of Maryland, he was an Integration & Verification Engineer of Ericsson (China) Communications Co. Ltd., Beijing.

Peter A. Sandborn received the B.S. degree in engineering physics from the University of Colorado, Boulder, and the M.S. degree in electrical science and Ph.D. degree in electrical engineering, both from the University of Michigan, Ann Arbor. He is a Professor in the CALCE Electronic Products and Systems Center (EPSC), in the Department of Mechanical Engineering at the University of Maryland, College Park, where his interests include system life cycle economics, electronic part obsolescence and prognostics and health management. Prior to joining the University of Maryland, he was a founder and Chief Technical Officer of Savantage, Austin, TX, and a Senior Member of Technical Staff at the Microelectronics and Computer Technology Corporation, Austin. He is the author of over 200 technical publications and books on multichip module design and part obsolescence forecasting. Dr. Sandborn is an Associate Editor of the IEEE Transactions on Components Packaging and Manufacturing Technology, a member of the board of directors of the International PHM Society and the International Institute of Obsolescence Management. Dr. Sandborn is a Fellow of the IEEE and ASME.

Navid Goudarzi is a Research Associate in the Department of Mechanical Engineering at the University of Maryland, College Park. Dr. Goudarzi’s research interests include renewable energy projects with a focus on novel onshore/offshore and small/large scale wind turbine designs for expanding their operational range and increasing their efficiency at lower cost, system life-cycle and risk economics. He received his Ph.D. in mechanical engineering from University of Maryland.

Maira A. Bruck is an undergraduate student at the University of Maryland, College Park. She has declared Economics as her primary major, Geographic Information Systems as her secondary major and International Engineering as her minor. Ms. Bruck works as an Undergraduate Research Assistant in the CALCE Electronic Products and Systems Center at the University of Maryland, College Park, where her interests include risk management, economic development and contract development.

References


Biographies

Xin Lei received the B.S. degree in Reliability System Engineering and the M.S. degree in Systems Engineering from the Beihang University, Beijing. He is a Ph.D. student in the CALCE Electronic Products and Systems Center (EPSC), in the Department of Mechanical Engineering at the University of Maryland, College Park, where his interests include system life-cycle economics and prognostics and health management. Prior to attending the University of Maryland, he was an Integration & Verification Engineer of Ericsson (China) Communications Co. Ltd., Beijing.

Peter A. Sandborn received the B.S. degree in engineering physics from the University of Colorado, Boulder, and the M.S. degree in electrical science and Ph.D. degree in
An Influence Gauge to Detect and Explain Relations between Measurements and a Performance Indicator

Jérôme Lacaille

Snecma Villaroche, 77550 Moissy-Cramayel, France
jerome.lacaille@snecma.fr

ABSTRACT
What about a software tool that behaves like a gauge able to estimate the quantity of information contained in a group of measurements? Then if we have a performance indicator or a defect rate, how may we compute the maximum performance explanation contained in our dataset? The first question may be answered by entropy and the second by mutual information. The present paper recalls a simple way to use those mathematical tools in an application one wants to launch each time a new dataset has to be studied. Often the PHM team in Snecma is asked to participate in special workforces to analyze sudden crisis. This methodology helps at the very beginning of the process to identify our mathematical capability to build an explanation model. This was the case during a small engine start crisis when some spark plugs were not working. Another time we used this tool to identify the flying condition when a gearbox was heating. This methodology was first developed for industry purposes like the optimization of machine tools or process recipes. Its success is in the simplifications of the computations that enlighten the interpretability of the results. Each signal is quantified in a way that improves the mutual information with the performance indicator. This is done signal by signal, but also for any small subsets of multivariate measurements until the confidence given by the quantity and quality of the data reaches its maximum. The segmentation of the data helps and boosts the computation of the integrals. Moreover, as this methodology uses quantified data as inputs it works as well with any sort of inputs such as continuous, discrete ordered and even categorized measurements. Once a best subset of measurements is selected a simple non-linear model is built using a relaxation algorithm. This model is a set of hypercubes that classifies the input space in a very simple and interpretable way. The methodology given below is a rough approach and may be replaced by more efficient regression algorithms if one only have continuous measurements but it has some advantages like a way to search a “best rule” according to some constraints and a graphic navigation tool very efficient to correct recipes.

1. APPLICATION EXAMPLES
This section gives some application examples of the influence analysis methodology. The next section (2. Mathematic Methodology) details computations implemented for this relatively simple method. Hence it is possible to read section 1 first, references to section 2 are given anyway when necessary, or read section 1 if more interested by the mathematical aspects.

1.1. No light up during the engine start process
Some turbofan aircraft engines do not light up on one start plug (two start plugs are positioned on opposite sides of each engine), but the event is really rare and never happens when using both plugs simultaneously (look at (Flandrois, Lacaille, Massé, & Ausloos, 2009) for analysis of the start capability of an aircraft engine). The number of such events is so small that a statistic analysis was not possible. But for aircraft engines as well as for any kind of engine, a plausible indication of the future risk of no-light-up is the increasing duration of the ignition time. So I get these durations for a set of flights from a fleet of similar engines and renormalize them according to external conditions among external temperature, pressure, altitude and oil temperature before ignition (Lacaille, 2009, 2010). Then I used the influence analysis process to find sets of parameters that may explain increase of this duration.

The potential factors were identified by engine experts and were related to engine conditions: temperatures, pressures at different measurement points, shafts’ rotating speeds, variable geometries, airport, flight time, plug position, fuel system and ignition process. We collected more than 10000 starts (more than 5000 different flights) which allow computing very small confidence intervals for the influence criterion.

Jerôme Lacaille. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
The first monovariate analysis (Figure 1) shows that the oil temperature was the most important factor but it does not explain more than 33% of the delay. Fuel flow regulation and adapted rotation speed increase explanation to 40%.

Finally expert impact analysis identifies atmospheric conditions and fuel density as main causes of variations of the ignition delay.

1.2. A gearbox is heating abnormally during flights

Abnormal heating in a gearbox was detected during flights tests on rare occasions. Data from 51 maneuvers were collected to isolate the flight conditions when this phenomenon arises.

The quality criterion was a difference between observed temperature and the expected temperature of the gearbox estimated according to a physical model. The factor of the analysis described

- flight conditions: aircraft speed, altitude, different attitude angles;
- engine conditions: rotation speeds, engine temperatures;
- measurement conditions: maneuver type, temperature stabilization time.

A first monovariate analysis (Figure 3) identifies the altitude (Alt) as linked to almost 80% of the temperature increase on our set of observations.

![Figure 1. Monovariate influence analysis. Oil temperature shows the maximum impact on the delay but explains only a little more than 30% of this criterion.](image1)

![Figure 2. Computation of the best sets of 1, 2 and 3 parameters which jointly influence the ignition duration. The bar sizes give the influence criterion and the light part on top is the confidence interval. The small blue horizontal bars (and values) are the importance of each parameter (monovariate influence of each single parameter over the set classification). The 55% limit on the top of the graph is the maximum influence that may be obtained using all the available data.](image2)

![Figure 3. Monovariate analysis on 51 experiments for identification of conditions related to increase of a gearbox temperature. The confidences intervals are the light boxes at the end of each bar.](image3)

The multivariate analysis (Figure 4) shows that on those 51 experiments an explanation may be improved using aircraft attitude (A2, ~5%), engine speed (N1) and temperature stabilization time (ST).

The number of observations was low but the confidence intervals were not so big and the analysis confirmed the intuition of the engine experts which were able to redesign the gearbox and eliminate this event.
Figure 4. Multivariate analysis on 51 experiments for identification of conditions related to increase of a gearbox temperature.

1.3. Adjustment of a carbon deposition process

The quality of an electric anode used in the extraction process of aluminum depends on the anode carbon density. In the anode making operation one wants to find how to adjust the chemical bath in order to optimize the density. In this application one does not try to find any potential causes of degradation but to isolate a good set of parameters to adjust.

The first step was to quantify the anode density, fixing some important levels observed during a batch of experiments. Then classification trees where learnt to build a clusterisation of each parameter set adapted to the quantified density (section 2.4). This transformation of the inputs allows a simple computation of entropies and influence criterion (section 2.5). But the quantification process obtained from trees also define bounds for each parameters and once a parameter detected as influent on the process these bounds are used to build an optimized recipe.

Each tree input is a set of parameters, as this is a combinatorial problem; a relaxation chain was implemented by a genetic algorithm which converges to a good enough set of parameters. The algorithm maintains a population of agents, each one corresponding to the selection of a subset of parameters. A mutation step managed random changes, suppressions or adjunctions of one parameter, and a cross-over step combined pairs of sets from random binary partitions.

In this application we had 1681 experiments with 17 adaptable control parameters and an anode density measurement as performance criterion. A cross validation procedure ensured robustness of the analysis with a k-fold methodology with batches of 10% of random experiments kept for the test phase. Four levels of density were retained by the process experts: high, medium high, medium low and low.

Figure 5 displays results obtained by selection of parameters using progressively increasing set sizes. It is not usual that a same parameter maintains its influence when the set size increases. It was however the case here.

Figure 5. Progressive selection of the best parameters to adjust.

The analysis results show that one cannot expect much improvement of the fabrication process by just changing one of the parameters. As said before the classification trees helps to build recipes that optimize some constraints, one of them is proposed on Figure 6.

Figure 6. A recipe for concurrent adjustment of 9 control parameters that limit the number of low density anode production. Top left of the interface is the constraint selection, top right is a flatten representation of the local dispersion of good and bad production around the recipe.
This application concludes by a prototype helping adjustment of parameters in real time when the process was drifting.

1.4. Optimization of a crankshaft production chain

The crankshaft production chain for automobiles engines is made of a succession of machine tools. Each tool has a small list of adjustable parameters. Almost at the end of the production chain was an expansive finishing equipment. Some weeks before a scheduled maintenance operation of this equipment the production yield decreased dramatically but the production experts were not affirmative about the cause of this degradation. The team extracts some measurements, a set of data per operation for three of them which were identified as probable causes: the draft production (20 variables), an intermediate polishing operation (10 variables) and the finish process (30 variables). They also had some process recipe data (25 variables) and context measurements and origin of the input material (13 variables). Each of these factors was a set of measurements identified by the production experts as influential on the quality of the crankshaft.

The performance indicator was a binary output identifying good shafts. It was the output of an abnormality detection test I built from geometric measurements and that gives 87% of good detection and only 9% of false alarms according to manual verification by the experts.

A quantization step transforms the sets of process measurements vectors into 5 categorical indicators using robust classification trees (see section 2.4 below).

The influence analysis on 377 observations (Table 1) gives an influence value on the performance for each factor and identifies the polish operation as the most plausible cause with 45% of the influence on production.

Table 1. Influence analysis results for shaft production.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Influence</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>3%</td>
<td>4%</td>
</tr>
<tr>
<td>Process recipe</td>
<td>1%</td>
<td>3%</td>
</tr>
<tr>
<td>Draft</td>
<td>0%</td>
<td>33%</td>
</tr>
<tr>
<td>Polish</td>
<td>45%</td>
<td>9%</td>
</tr>
<tr>
<td>Finish</td>
<td>11%</td>
<td>3%</td>
</tr>
</tbody>
</table>

However the draft operation may also be a potential cause because of the poor precision of the computation.

An optimal tuning was found for this polishing tool with a measurable impact on the quality of the fabrication process. This recipe is a direct application of the classification tree bounds found for data quantification.

Figure 7. Identification of a recipe for the polishing tool that maximizes the number of good shafts.

Equivalent recipes may be found for each operation on the process.

1.5. Causes of bubble production in glass fabrication

This analysis was at the origin of the development of the influence analysis methodology. It was about glass-making process for automobile windshield production.

The process of glass production is complex, we schematize: raw material enter the oven, gas burners melt this material in a fluid that undergoes two successive vortexes, finally some of this fluid exits the oven on a mercury mat and is shaped according to specifications during the annealing phase.

Figure 8. A schematic view of a glass fabrication oven.

Lots of factors may be source of bubbles production:

- RESUR: displacement of the resurgence point (a position around the middle of the oven where the two vortexes exchange material).
- PROFIL: change in the energy distribution profile given by the gas flow for each burner.
- PRESSURE: the pressure in the oven.
- TIREE: The speed of the glass exiting the oven.
- COMRED: Reduction combustion.
- And lot more.

This application was clearly a case where a temporal filter should be applied on the data. If you input colored material, then the produced glass leaves are progressively colored.
Figure 9 is a schematic view of a classical transfer function for the color.

A temporal filter was then applied on the measurements corresponding to each of the potential causes (as described section 2.2 below). Then a clusterisation helps computing the different influence values.

Figure 10. Influence analysis on the causes of bubbles production during glass fabrication process. Resurgence point position appears to be an important cause as well as energy profile and pressure.

Figure 11. Comparison with correlation analysis by pairs of measurements for the glass-making process. Each pair is cross-correlated with the quality of the product.

Figure 10 shows the result of the influence analysis of bubble production in a plant north of Paris during a small production crisis in 1993. I was able to identify the main causes of defectivity.

Figure 11 shows a correlation analysis of each couple of measurements with the quality of the product (after individual temporal filtering for each measurement). This graph identifies pressure (FP41) but not clearly the main causes of degradation.

2. MATHEMATIC METHODOLOGY

2.1. Performance indicator and potential causes

Influence analysis is the computation of an absolute value that quantify a relation between a set of observations linked to some sort of physical phenomenon and a performance indicator. The performance indicator may be a fab yield, the quality of production or a defectivity rate. The potential causes of degradation or amelioration are more difficult to master. They should be modeled by sets of physical factors given by experts. Sometimes, one factor is enough to produce a clear understanding of the risk linked to the monitored system. Often such risk measurement is computed as a score of a statistic model (a log-likelihood for example). But most of the time a statistic model is difficult to build and a variation of production or a change in the behavior of a system should be extracted from multiple measurements simultaneously. This is the most frequent case when potential causes are not independent and interact in a complex system. In that case the complex system should be identified by the list of all its potential causes of defectivity.

2.2. Time delay and data synchronization

A cause of degradation may also be detected as a change in the temporal behavior of a set of signals. I usually simulate that behavior with a rough model like an autoregressive linear model in the simplest case or a recurrent network for non linear behaviors. The parameters of the model are then taken as state factors for the system to monitor. It is the case when some delay exists between the immediate effect of degradation (eventually an action) and the corresponding result on the performance indicator.

In (Lacaille, 1998) and (Lacaille, 1997) I roughly model time dependency by a rational filter. The resulting estimation is not important but the state of the system exhibits its internal dimensionality and a set of intermediate (computed) factors.

Let for example $x_t$ be the set of measurements or computations, identified by system experts, collected at time $t$ and relevant to explain parts of the performance. A delay $\delta$ may exist between observation $x_t$ and result performance $y_{t+\delta}$. This delay is unknown and must even be more a combination of past observations than just a time laps. This temporal combination of past data may be approximated by
an autoregressive filter. Equation (1) gives a markovian representation of such a model (Akaike, 1975) where the intermediate stochastic process $z_t$ is the state of this system.

$$\begin{align*}
\{ y_t &= C_{t+1} z_t \\
\{ z_{t+1} &= A_{t+1} z_t + B_{t+1} x_t 
\end{align*}$$

(1)

The dimension of vector $z_t$ gives the rank of the system (and matrix $A$ eigenvalues an idea of the delay). Initial values for rank and matrices $A_0, B_0$ and $C_0$ are computed by least square regression and minimization of the AIC criterion. Even evolution of the matrices may be obtained by a recurrent tracking method (2) or Kalman filter (Kalman, 1960; Welch & Bishop, 2006).

$$\begin{align*}
\Delta A_t &= \eta C_t (\hat{y}_t - y_t) z_{t-1}' \\
\Delta B_t &= \eta C_t (\hat{y}_t - y_t) x_{t-1}' \\
\Delta C_t &= \eta (\hat{y}_t - y_t) z_{t}'
\end{align*}$$

(2)

This is a rough linear approximation limited to rational filters but (Lacaille, 1994) gives clues of how to replace the linear model by a three layer recurrent perceptron.

### 2.3. Influence criterion

Once we have a performance indicator and a set of potential causes factors our goal is to sort these factors in order of influence on the performance. A classic solution in statistic is to use linear models and compute correlations between each factor regression and the performance indicator (adjusted $R^2$ coefficient to take the different indicators dimensions into account). When the relation is not clearly linear, if we do not have just monovariate indicators or even non-numeric values, but a set of measurements representing in their whole the state of a system we want to use something more generic than just a linear correlation.

Mutual information $I(X,Y)$ measures the quantity of information shared between two stochastic variables $X$ and $Y$. It comes from the computation of entropy $H(X)$ which defines the total quantity of information contained by a stochastic variable. (Note that the stochastic variables used here are not necessary of dimension 1 or event numeric.) For example, a qualitative variable with values in a set of $k$ labels $\{x_0 ... x_k\}$ has an entropy value between 0 if the variable is constant and its maximum $\log(k)$ if it has a uniform distribution (when all labels are random).

The following equations give the integral formulation of the entropy and mutual information:

$$H(X) = - \int_x dP(X = x) \log P(X = x)$$

(3)

$$I(X,Y) = \int_{xy} dP(X = x, Y = y) \log \frac{P(X = x, Y = y)}{P(X = x)P(Y = y)}$$

(4)

Equation (4) shows that the mutual information may be interpreted as the Kullback-Leibler divergence between the distribution of the couple of variables $(X, Y)$ and the product of their distributions as if they were independent. Hence it measures some sort of distance from the couple’s law to stochastic independence.

The two formulations are linked by the following set of equalities:

$$I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y)$$

(5)

Hence mutual information is less than each of the entropies and clearly corresponds to the entropy of one of the variable from which one suppresses the information given by the second.

The selected influence criterion is defined by the proportion of information contained by factor $X$ that explains the performance indicator $Y$.

$$\lambda_Y(X) = I(Y|X) = \frac{I(X,Y)}{H(X)} \in [0,1]$$

(6)

We do not use $H(Y)$ for denominator because we do not want to favor factors with too much entropy. Such factors may be too complex and will not have enough robustness in a data analysis process.

### 2.4. Data quantification and measurement selection

A generic computation of entropy is difficult to implement. A well known solution is to use a nearest neighbor based approximation known as the Kraskov method described in (Kraskov, St, & Grassberger, 2004). But this method implies that a distance between measurements exists. This may be tricky when observations come from different sources and are not really comparables. It is always possible to define individual comparisons for each factor, but we still have a problem to build a multivariate solution.

A simplest approach consists in the quantization or categorization of performance indicator and factors. Moreover it is easier to convince experts with logical rules like “when $X_1$ is low and $X_2$ is medium then the performance is low” instead of a complex analytical representation.

The quantization of our data is achieved in two phases. First we define thresholds for the performance indicator. This is an easy task and very clear to experts. Then for each factor,
which can be multivariate, we implement a clusterisation driven by the quantified performance factor. Usually we train a regression tree with cross validation procedure to limit tree depth and increase the robustness. Then we associate a different label to each of the leaves. Other clusterisation methods may be used like SVM (Burges, 1998) or Bayesian networks (Pearl, 1988) and much more. Classification trees (Breiman, Friedman, Olshen, & Stone, 1984) keep an advantage in interpretability of the clusters and helps building good recipes for the analyzed process.

2.5. Integrals computation and confidence intervals

Once the input data quantified, the computation of the entropies are straightforward: it is just a matter of counting. Let define the following notations for the quantified factor \( \tilde{X} \):

\[
\tilde{X} \in \{ x_1 \ldots x_k \} \text{ and } P(\tilde{X} = x_i) = p_i
\]  

(7)

Each \( p_i \) is estimated by the proportion \( \hat{p}_i \) of label \( x_i \). Then the estimated entropy of quantified factor \( \tilde{X} \) is given by

\[
\hat{H} = - \sum_{i=1}^{k} \hat{p}_i \log \hat{p}_i
\]  

(8)

\( \hat{p}_i \) is a statistics, then a stochastic variable of mean \( p_i \) and variance \( \sigma^2_i = \frac{p_i(1-p_i)}{N} \), \( N \) being the number of observations.

To estimate a confidence interval for this approximate entropy we use the Neymann-Pearson approximation:

\[
\chi^2 = 2N \sum_{i=1}^{k} \hat{p}_i \log \frac{\hat{p}_i}{p_i}
\]  

(9)

Where this value follows a \( \chi^2_{k-1} \) distribution (Chi\(^2\) with \( k-1 \) freedom degrees). Hence we obtain

\[
\sum_{i=1}^{k} \hat{p}_i \log \hat{p}_i - \sum_{i=1}^{k} p_i \log p_i - \sum_{i=1}^{k} (\hat{p}_i - p_i) \log p_i = \frac{1}{2N} \chi^2
\]  

(10)

\[
H - \hat{H} = \sum_{i=1}^{k} \frac{\hat{p}_i - p_i}{\sqrt{p_i}} (-2 \sqrt{p_i} \log \sqrt{p_i}) + \frac{1}{2N} \chi^2
\]  

(11)

But in this last equation we note that the \( \chi^2 \) value is an estimation of the sum of squared values of each term \( \frac{\hat{p}_i - p_i}{\sqrt{p_i}} \) each one that may be approximated by independent normal laws of variance \( \frac{1}{N} \). Hence Cochran theorem (Cochran, 1934) states that the two terms in the sum equation (11) are independent. Thus the variance of estimated entropy \( \hat{H} \) may be estimated by the sum of variances of those two terms:

\[
\text{Var}(\hat{H}) = \frac{1}{N} \sum_{i=1}^{k} p_i(1-p_i) \log^2 p_i + k-1 \left(\frac{1}{2N}\right)^2
\]  

(12)

The standard deviation \( \sigma_H \) of the entropy estimated by (8) is approximated by the square root of the variance (2) where each theoretical proportion is approximated by its own estimation. A confidence interval around \( \hat{H} \) is chosen as \( \hat{H} \pm \Delta H \) where \( \Delta H \) is a multiple of the estimated standard deviation \( \sigma_H \) (usually \( 2\sigma_H \)).

An analogous computation may be developed for the mutual information:

\[
I(X,Y) = \sum_{i=1}^{k} \sum_{j=1}^{l} p_{ij} \log \frac{p_{ij}}{p_i p_j}
\]  

(13)

but it is easier to use the last equality in (5) to approximate the interval bounds from twice the sum of each standard deviation. Then we finally use a logarithmic approximation for the influence coefficient.

\[
\Delta I(X,Y) \leq \Delta H(X) + \Delta H(Y) + \Delta H(X,Y)
\]  

(14)

\[
\Delta \lambda_I(X) \leq \lambda_I(X) \left( \frac{\Delta I(X,Y)}{I(X,Y)} + \frac{\Delta H(X)}{H(X)} \right)
\]  

(15)

This gives us a rough set of bounds we are using to draw the light bars on the example graphics section 1.

3. CONCLUSION

Influence analysis is more a methodology than a tool. To be efficient one has to prepare the data with most of process and physic knowledge as possible. But at the end it defines an application skeleton which may probably be adapted to any specific process.

The equations from section 2 applied in the examples section 1 are simple and rough approximations for the computation of mutual information but they were sufficient to solve some really interesting problems. A detailed implementation (in French) and example is given in (Lacaille, 2004) and applications in semi-conductors fabrication may be found here (Lacaille & Dubus, 2005; Lacaille, 2005, 2008). In the case of linear relations and monovariate factors one may prefer a L\(_1\) constraint robust regression like the LASSO method (Tibshirani, 1996) to select the factors. This methodology however is entirely generic and may be applied even to non numeric categorical data and multivariate factors.

The proposed methodology to select and identify subset of influential variables may also be seen as a classification, problem. Indeed our goal is to separate datasets according to
different levels of performance. In that case one may refer to
the ample bibliography on the use of mutual information to
help feature selection in the paradigm of classification
(Doquire & Verleysen, 2011). One may even see some
limitation in using mutual information for classification
purposes (Frénay, Doquire, & Verleysen, 2012).

Figure 12 recall each step in our methodology. The two first
steps are essentially expert driven; the synchronization is
managed by linear or non-linear autoregressive models; and
the data quantification is done either with decision trees, or
any other classifier with an optimization loop on the mutual
information computation. This optimization may be
implemented with a genetic algorithm or any relaxation
scheme. Finally we implement a simple model based on
decision trees for inference purposes but any other
estimation tool such as neural network may be used on the
selected subset of measurements.

The main fact to remember is that such computation may be
used at the beginning of any data mining challenge just to
get some clues about the quantity of explanation one may be
able to extract from a dataset.

NOMENCLATURE

AIC Akaike Information Criterion
LASSO Least Absolute Shrinkage and Selection Operator
PHM Prognostic and Health Management
SVM Support Vector Machines

REFERENCES

Processes by Canonical Variables. *SIAM Journal of
Control, 13*(1), 162–173. doi:10.1137/0313010

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J.
Wadsworth statistics/probability series* (Vol. 19).

Machines for Pattern Recognition. In L. T. Bell
Laboratories (Ed.), *Data Mining and Knowledge

Cochran, W. G. (1934). The distribution of quadratic forms
in a normal system, with applications to the analysis
of covariance. *Mathematical Proceedings of the
doi:http://dx.doi.org/10.1017/S0305004100016595

for feature selection with missing data. In *ESANN* (pp.
27–29).

Flandrois, X., Lacaille, J., Massé, J.-R., & Ausloos, A.
(2009). Expertise Transfer and Automatic Failure
Classification for the Engine Start Capability System.
In *AIAA InfoTech*.

Frénay, B., Doquire, G., & Verleysen, M. (2012). On the
Potential Inadequacy of Mutual Information for Feature
Selection. In *ESANN* (pp. 25–27), Bruges.

and Prediction Problems. *Transactions of the ASME-
Journal of Basic Engineering, 82*(Series D), 35–45.
doi:10.1115/1.3662552


Lacaille, J. (1994). Generalization of Stochastic and
Deterministic neural Network with a Continuous State
Space and a Connectivity Greater than Two. In *IEEE
World Congress on Computational Intelligence* (Vol.
2). Orlando. doi:10.1109/ICNN.1994.374304

Synchronize Multivariate Sensors. In *Les réseaux
neuro-mimétiques et leurs applications (NEURAP)*.
Marseille.

with an autoadaptive neural method. *Intelligent &

mathématiques. Université Paris I*. Paris I, Habilitation
thesis.

Lacaille, J. (2005). Mathematical Solution to Identify the
Causes of Yield Deterioration. In *International
Sematech Manufacturing Initiative (ISMI)*. Austin, TX:

from

D&date=20080403&CC=US&NR=2008082197A1&KC=
A1

Lacaille, J. (2009). Standardized failure signature for a
turbofan engine. In IEEE Aerospace conference. Big
Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J. (2010). Standardization of Data used for
Monitoring an Aircraft Engine. USA; FR. Retrieved
from
D&date=20100708&CC=WO&NR=2010076468A1&
KC=A1

Swarm of Intelligent Distributed Agents. In AEC-APC.
Indian Wells, CA: Sematech. Retrieved from
https://sites.google.com/site/jeromelacaille/semiconduct
eurs/2005-09---aec-apc-indian-wells

Systems: Networks of Plausible Inference. Morgan
Kaufmann.

Tibshirani, R. (1996). Regression shrinkage and selection
267–288).

Welch, G., & Bishop, G. (2006). An Introduction to the

BIography

Jérôme Lacaille is a Safran emeritus expert which mission
for Snecma is to help in the development of mathematic
algorithms used for the engine health monitoring and
statistical analysis of company data. Jérôme has a PhD in
Mathematics on “Neural Computation” and a HDR
(habilitation à diriger des recherches) for “Algorithms
Industrialization” from the Ecole Normale Supérieure
(France). Jérôme has held several positions including
scientific consultant and professor. He has also co-founded
the Miriad Technologies Company, entered the
semiconductor business taking in charge the direction of
the Innovation Department for Si Automation (Montpellier
-France) and PDF Solutions (San Jose - CA). He developed
specific mathematic algorithms that where integrated in
industrial process. Over the course of his work, Jérôme has
published several papers on integrating data analysis into
industry infrastructure, including neural methodologies and
stochastic modeling.
Omni-Directional Regeneration (ODR) of Gap Sensor Signal for Journal Bearing System Diagnosis

Joon Ha Jung1, Byung Chul Jeon2, Byeng D. Youn3, Donghwan Kim4, and Yeonwhan Kim5

1,2,3Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, 151-919, Republic of Korea
reallibero@gmail.com
puarni@empas.com
bdyoun@snu.ac.kr

4,5Power Generation Laboratory, KEPCO Research Institute, 305-760, Republic of Korea
kimdonghwan@kepco.co.kr
90101552@kepco.co.kr

ABSTRACT

We have developed a technique that enhances the detectability of sensors used to acquire data from a journal bearing rotor system. Usually, at an axial position for the rotating shaft on a journal bearing system, two sensors are fixed in radial direction at a right angle. The conventional diagnosis researches use only the acquired signals. However, two fixed sensors may not give sufficient information for diagnosis of the system since anomalies can happen in arbitrary direction. To improve the robustness of the diagnosis, coordinate transformed gap sensor signal is generated in arbitrary direction without installing extra sensors or adjusting sensor positions. With the original signals, the generated signals are used in the process of diagnosis. The powerful but simple method is described in the paper, and is verified by data sets from the experiment.

1. INTRODUCTION

The journal bearing supports rotating parts of the mechanical systems with a fluid. Since the fluid ensures smooth rolling of the rotors, it is frequently used in large systems that require safe operation. For example, turbines and pumps in power plants use journal bearings to maintain the systems safely in heavy load and high speed conditions. Without a direct contact between the rotor and the stator, the vibration can be kept below in a certain magnitude. Apart from the stability of the journal bearings, a large rotor system requires an anomaly diagnosis system. Although the design of the rotor systems satisfy the requirements of the system, uncertainties can arouse from operation as well as manufacturing process.

These uncertainties in the rotor systems cause the system to operate in an unexpected way. Sometimes, a sudden failure or an accident can happen if proper maintenance action is not performed and the consequence can be disastrous. Thus, to prevent such unfortunate events and to take proper measures based on the exact condition of the system, diagnosis systems are commonly installed in large rotor systems.

The diagnosis systems for rotors use data-driven method frequently. The method follows the steps: data acquisition, feature generation, and classification. First, the signals from each health state are obtained. Most of the signals for rotors use vibration signals since they can well represent the state of the operating condition. Then the obtained signals are processed by appropriate techniques. Next, the processed data are used to extract features that represent each health state and the condition of the rotors. With the extracted features, a classifier is trained, and the classifier classifies newly acquired data after following the same steps as the training data.

Among the stated steps, various signal processing techniques are developed to generate features of good separation ability, which eventually leads to accurate diagnosis results. Examples of various signal processing methods for vibration data are angular resampling (Bonnardot, El Badaoui, Randall, Daniere, & Guillet, 2005; Villa, Reñones, Perán, & De Miguel, 2011), statistical approach (Jeon, Jung, Youn, Kim, & Bae, 2014), principal component analysis (Malhi & Gao, 2004; Sun, Chen, & Li, 2007), and wavelet transform (Liu, 2003; Sanz, Perera, & Huerta, 2007). However, the gap sensors are mostly fixed in the rotor systems and the sensor location is hard to adjust. Due to the fixed location of the sensors, the diagnosis accuracy of direction oriented anomalies (e.g. rubbing, misalignment) may be demanding. Researches have been conducted to resolve the limitation of the fixed location of sensors. Orbit identification in time-
domain and full-spectrum in frequency-domain are the two methods.

The orbit information can be obtained since most of the rotor systems have two sensors in a right angle. Yan, Zhang, Li, Li, and Huang (2009) modified the orbit into seven different features to identify the state of the steam turbine generator. H. Wang, Wang, and Ji (2013) quantified the orbit information with isometric feature mapping to identify faults in rotors. Other researches also tried to quantify the orbit shape to make more accurate diagnosis of rotors (Bachschmid, Pennacchi, & Vania, 2004; Bo, Jian-Zhong, Wen-Qing, & Bing-Hui, 2004; C. Wang, Zhou, Kou, Luo, & Zhang, 2012; Yan, Zhang, & Wu, 2010). Full-spectrum analysis uses both x- and y- signals to expresses the forward and backward frequency of the rotors (Chen & Chen, 2011; Goldman & Muszynska, 1999; T. H. Patel & A. K. Darpe, 2009; Patel & Darpe, 2011; Zhao, Patel, & Zu, 2012). Fengqi and Meng (2006), and T. Patel and A. Darpe (2009) tried to detect rubbing state by using full-spectrum.

However, full-spectrum as well as orbit identification are not enough to represent thorough information of each state. Quantification of the orbit information varies from papers to papers, and the quantification process has huge effects on the overall performance of diagnosis. Full-spectrum uses x- and y- signals acquired from the fixed gap sensors, but the two signals may not be enough to represent the direction-oriented anomalies. To overcome these limits and to make robust diagnosis system, we have suggested omni-directional regeneration (ODR) technique to enhance the robustness of the various anomalies.

The paper is organized as follows. Section 2 briefly states overview of journal bearing diagnosis system. Section 3 describes the experiment set-up used in this research. Section 4 states procedures of ODR based diagnosis. Section 5 shows the results of ODR based diagnosis. In section 6, short summary of the research and future works are stated.

2. OVERVIEW OF JOURNAL BEARING DIAGNOSIS SYSTEM

This section describes general procedures of journal bearing diagnosis. Section 2.1 describes the characteristics of gap sensor signals used. Section 2.2 describes diagnosis process based on supervised-learning.


The proximity sensors, also known as gap sensors, are widely used in journal bearing systems because the system operates in a low level of vibration, and thus require high resolution sensors. The high resolution of the gap sensor is possible as it measures the change of the eddy current. The vibration signals are acquired as voltage. Alternating current (AC) component of the voltage represents relative vibration, while direct current (DC) component represents absolute radial position of the rotor. Generally, two gap sensors are placed in the same axial location at a right angle to show orbit of the rotor as presented in Figure 1.

2.2. Diagnosis Process Based on Supervised-Learning

The supervised learning method is commonly used in diagnosis process of journal bearing systems. The acquired vibration signals from the gap sensors are used to generate features. The feature generation includes feature selection as well as extraction. Then an appropriate classifier is trained and is used to classify the system into health states. The following subsections describe the feature generation and the classification.

2.2.1. Feature Generation

Feature generation, which can be divided into feature extraction and selection processes, has significant effect on the performance of diagnosis. Since the research targets on steady-state system, time- and frequency- domain features rather than time-frequency domain features are used. The candidate features listed in Table 1 and 2 are widely used ones in detecting the faults or abnormality of rotor systems (Jeon et al., 2014; Sun et al., 2007). Time-domain features include statistical moments and waveform related features. Frequency-domain features include various frequency spectrum features which are closely related to the state of the rotor system.

Total of sixteen features were extracted from the vibration signals. Before extracting the features, angular resampling was applied to the raw vibration signals to enhance the separation ability of the features and to reduce the noise. From the resampled signals, time-domain features were extracted by one cycle basis, while freq.-domain feature by sixty cycle basis (Jeon et al., 2014).

Among the extracted features, optimal features are selected by using the genetic algorithm. Features have different ability in health state separation, so not every feature listed in Table 1 and 2 guarantee accurate diagnosis result. For example, a specific feature can represent the oil whirl very well, but it may not tell difference between normal and rubbing states. In
addition, highly correlated features are redundant features that can be reduced. The optimal feature subset of k number of features was obtained by finding a subset that maximizes the following fitness function, \( f_n \), presented as:

\[
f_n = (1 - \alpha) \times \text{mean}(MI) - \alpha \times \frac{1}{N^2 \sum_{i,j=1,i
eq j}|\rho_{i,j}|}
\]

(1)

**Table 1. Time-domain Features.**

<table>
<thead>
<tr>
<th>Number</th>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Skewness</td>
<td>( \frac{\sum(X_i - \bar{X})^3}{(N-1)s^3} )</td>
</tr>
<tr>
<td>2</td>
<td>Kurtosis</td>
<td>( \frac{\sum(X_i - \bar{X})^4}{(N-1)s^4} )</td>
</tr>
<tr>
<td>3</td>
<td>Crest Factor</td>
<td>Max(</td>
</tr>
<tr>
<td>4</td>
<td>Shape Factor</td>
<td>( \frac{\sum X_i^2}{N} \times \frac{N}{\sum X_i} )</td>
</tr>
<tr>
<td>5</td>
<td>Impulse Factor</td>
<td>Max(</td>
</tr>
</tbody>
</table>

**Table 2. Frequency-domain Features.**

<table>
<thead>
<tr>
<th>Number</th>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>FC</td>
<td>( \frac{\int f \times s(f) , df}{\int s(f) , df} )</td>
</tr>
<tr>
<td>7</td>
<td>RMSF</td>
<td>( \frac{\int f^2 \times s(f) , df}{\int s(f) , df}^{1/2} )</td>
</tr>
<tr>
<td>8</td>
<td>RVF</td>
<td>( \frac{\int (f - FC)^2 \times s(f) , df}{\int s(f) , df}^{1/2} )</td>
</tr>
<tr>
<td>9</td>
<td>0.5X / 1X</td>
<td>( s(f_{0.5X}) / s(f_{1X}) )</td>
</tr>
<tr>
<td>10</td>
<td>2X / 1X</td>
<td>( s(f_{2X}) / s(f_{1X}) )</td>
</tr>
<tr>
<td>11</td>
<td>(1x~10x)/1x</td>
<td>( \sqrt{s(f_{nx})} / s(f_{1X}) )</td>
</tr>
<tr>
<td>12</td>
<td>(0~0.39x)/1x</td>
<td>( \sqrt{s(f) , df} / s(f_{1X}) )</td>
</tr>
<tr>
<td>13</td>
<td>(0.4x~0.49x)/1x</td>
<td>( \sqrt{s(f) , df} / s(f_{1X}) )</td>
</tr>
<tr>
<td>14</td>
<td>(0.5x~0.99x)/1x</td>
<td>( \sqrt{s(f) , df} / s(f_{1X}) )</td>
</tr>
<tr>
<td>15</td>
<td>(3x~5x)/1x</td>
<td>( \sqrt{s(f) , df} / s(f_{1X}) )</td>
</tr>
<tr>
<td>16</td>
<td>(3x,5x,7.9x)/1x</td>
<td>( \sum_{n=1}^{8} \sqrt{s(f_{2n+1X})} / s(f_{1X}) )</td>
</tr>
</tbody>
</table>

where mean(MI) is the average of mutual information between the feature and the class, \( \rho_{i,j} \) is the correlation coefficient between \( i \)th and \( j \)th features, \( \alpha \) is the penalty coefficient, \( k \) is the user defined number of features for the optimal subset, and \( n \) indicates the \( n \)th feature subset among \( N \) number of subsets in a generation (Guo, Damper, Gunn, & Nelson, 2008). The mutual information represents the separation ability of the features, and the correlation coefficient represents redundant features.

**2.2.2. SVM Classification**

The supervised learning method frequently uses support vector machine (SVM) algorithm for the classification step because of its simple and strong performance. First, the classifier is trained by using the optimal features acquired via feature generation process. Only the feature data with known class is used for training the classifier. The trained classifier denotes a hyper-plane that maximizes the margin between the classes. After the classifier is trained, the feature data with unknown class is tested. The testing predicts the class of feature data. In this research, the known feature data is used for the testing step, and the predicted class was compared to the actual class to evaluate the performance of the classifier.

Although SVM was originally designed for linearly separable two-class problem, it can be used for non-linear two-class problem by introducing the slack variable and the kernel function. Furthermore, by applying one-against-one (OAO) decision method, SVM can be expanded to classify multi-class problem. In this research, the LIBSVM algorithm was used (Chang & Lin, 2011).

**3. Diagnosis Methodology Using Omni-Directional Regeneration**

This section describes the ODR signal based diagnosis procedures. Most of the diagnosis procedures follow as in section 2, but few steps are modified and adjusted as described in this section. Section 3.1 defines ODR signals, and section 3.2 describes how ODR signals are used in the diagnosis procedure.

### 3.1. Omni-Directional Regeneration of Gap Signals

**3.1.1. Definition of ODR**

The principal of ODR signal generation is the transform of the coordinate system as presented in Figure 2. The clockwise transform of a data point \((x_1, y_1)\) in two-dimension Cartesian coordinate system can be expressed as:

\[
\begin{bmatrix}
  x_2 \\
  y_2
\end{bmatrix} = \begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix} \times \begin{bmatrix}
  x_1 \\
  y_1
\end{bmatrix}
\]

(2)

where \( \theta \) denotes degree of rotation, and \( x_2 \) & \( y_2 \) denotes the data point in \( \theta \) rotated coordinates system.
The scalar data values, \( x_i \) and \( y_i \), in equation (2) can be replaced by vectors, \( x_i \) and \( y_i \), as:

\[
\begin{align*}
    x_2 &= \cos \theta x_1 - \sin \theta y_1 \\
    y_2 &= \sin \theta x_1 + \cos \theta y_1
\end{align*}
\]

where \( x_i \) and \( y_i \) are vectors in \( i^{th} \) coordinate system. The vibration signals from the sensors (a) and (b) in Figure 3 can be indicated by \( x_1 \) and \( y_1 \), respectively. Then, the coordinate transformed signals by \( \theta \), \( x_2 \) and \( y_2 \), can be regarded as the signals obtained from sensors (c) and (d), respectively.

Using this principle, the ODR signals, \( x_n \) and \( y_n \), can be defined as:

\[
\begin{align*}
    x_n &= \cos(n\Delta \theta) x_0 - \sin(n\Delta \theta) y_0 \\
    y_n &= \sin(n\Delta \theta) x_0 + \cos(n\Delta \theta) y_0
\end{align*}
\]

where \( x_0 \) and \( y_0 \) are the acquired vibration signals from gap sensors, \( \Delta \theta \) is the increment of the rotation angle, and \( N (= \lceil \pi / \Delta \theta \rceil) \) is the maximum number of ODR that can be generated.

The ODR can generate vibration signals from an arbitrary direction. Multiple ODR signals around the rotor can be obtained by adjusting the increment of the angle, \( \Delta \theta \). To diagnose the state of the system accurately, \( \Delta \theta \) should be fine. However, if \( \Delta \theta \) is too fine, the number of ODR signals \( (N) \) will increase, and the computational load will also increase. Thus the increment of the angle, \( \Delta \theta \), is set as \( \pi/16 \) in this research. In addition, the vibration signals are radial symmetric, so ODR signals within the \( \pi \) rotation angle range will be generated. Likewise, there is no need to use both \( x_n \) and \( y_n \), because \( x_n \) signal is equal to \( y_{n+8} \), which is 90° rotated signal of \( y_n \). The \( x_n \) covers all \( y_n \) if ODR covers more than half rotation.

3.1.2. Validation of ODR

To check that ODR signals truly represent the signal acquired from rotated direction, three evidences are provided by the example vibration signals.

First, \( x_8 \) which is 90° counter-clockwise (ccw) ODR signal of \( x_0 \) exactly matched to \( y_0 \). As shown in Figure 4, the two signals are identical to each other. Second, \( x_0 \) is located in the opposite direction of \( y_8 \), so the signals were in reverse of each other. This is also shown in Figure 4. Last, the orbit shape remained the same for any \( x_n \) and \( y_n \). This is shown in Figure 5 that the orbit shape is consistent over rotation angle. Thus from the three evidences, the ODR signals exactly shows behavior of the rotor in arbitrary angle.

3.2. Diagnosis Procedures using ODR Signals

3.2.1. Feature Extraction and Reduction

Each ODR signal generates sixteen features defined in section 2.2.1, which makes \( 16 \times N \) features in total. Since the number of ODR, \( N \), was set as sixteen, total number of features add up to 256. But not all 256 features are useful in classification of health states, so the number of features are reduced by Principal Component Analysis (PCA) (Malhi & Gao, 2004).

PCA decorrelates the multi-dimension features by finding coordinates of principal components. The principal components, \( v_i \), are derived by solving the following equation:

\[
Av_i = \lambda_i v_i
\]

where \( A \) is the covariance matrix of feature vectors and \( \lambda_i \) is indices, \( i \) are sorted in descending order of eigenvalues. The projection matrix, \( V \), is defined as equation (7) (Sun et al., 2007).

\[
V = [v_1, v_2, ..., v_{256}]
\]

By multiplying \( V \) to the feature vectors, new de-correlated features are obtained.
However, not all the de-correlated features have good separation ability. Among the 256 new features, ones that had variances larger than one were used, which counts for ten features. Thus principal components corresponding to ten largest eigenvalues are used in this research. Consequently, 256 ODR features were reduced to ten features by PCA.

3.2.2. Feature Selection and Classification

The ten reduced features by PCA are used for feature selection process. As stated in Section 2.2.1, the number of features for optimal subset, \( k \), should be defined prior to feature selection. Since the number of reduced features are ten, \( k \) ranges from two to ten. For each \( k \), feature selection was performed. Then, the selected features were used for classification process as stated in Section 2.2.2. To validate the effectiveness of ODR, the classification results of non-ODR signals were also performed. The results are presented in Section 5.

4. DESCRIPTION ON EXPERIMENT

4.1. Test-bed Description

The research is based on the data acquired from the experiment. The experiment was conducted on the RK4 test-bed made by GE Bently-Nevada. Four health states—normal, rubbing, misalignment, oil whirl—were tested on RK4. First, the normal health state was tested with two shafts. A short and a long shaft of 10mm diameter were connected by a flexible coupling, and the short shaft was driven by the motor. An 800 gram disc was attached at the middle of the long shaft supported by two bearings. The amplitude of vibration was set to a certain level by balancing procedure. Second, the rubbing state also had the same set-up as that of normal, but the rubbing screw induced direct point contact on the shaft at steady-state. An accelerometer was attached to the screw jig to maintain the consistent level of the rubbing. For the misalignment state, a special jig was added to the normal set-up to shift the shorter shaft horizontally, which represents an angular misalignment. An exact amount of misalignment was controlled by the jig. For the last health state, oil whirl, one shaft and two discs were used to enforce whirling force at the end of the shaft. The pressure of oil in the bearing was controlled to produce whirling in the bearing.

4.2. Data Acquisition

All four health states were tested for sixty-seconds to obtain enough amount of data. Since the speed of the rotor was 3,600 rpm, sixty-seconds of test can collect 3,600 cycles of vibration. In addition, each health state was repeated twice, which adds up to three sets, to apply cross-validation.

The displacement of the shaft was obtained by Bently Nevada 3300 proximity sensors. The two sensors in a right angle were placed at an axial location adjacent to the anomaly position. Additionally, tacho signal was measured to acquire the phase of the rotating shaft, and was used in the resampling process. All three signals were acquired through NI DAQ 4432.

5. RESULTS AND DISCUSSION

5.1. Qualitative Analysis

The four health states were tested on RK4 test-bed. Graphical representations such as orbits are widely used to distinguish each health state. Two cycles of orbits are represented in Figure 5. Example of ODR signal orbits.

Figure 6. Set-up of RK4 test-bed.
Figure 7. Rotors in a normal state rotates smoothly, so the orbit of normal is close to a clear circle. The orbit of rubbing state shows trace of contact between the rubbing screw and the shaft. Misalignment has preloaded shaft, and this is characterized by two small circles. Oil whirl health state shows orbits of incomplete circles, a typical sign of sub-harmonic dominant signals.

As shown in Figure 8, the rubbing and misalignment show characteristics of direction oriented health states. The ODR signals of direction oriented anomalies change over the rotation angle. However, the conventional method only uses the acquired signals, $x_0$ and $y_0$. Therefore, the performance of the classification may depend on the anomaly directions.

5.2. Quantitative Analysis

The effectiveness of the ODR signals with PCA feature reduction can be evaluated by class prediction results. The results are compared for all the number of possible optimal feature subset, $k$. As described in section 3.2.2, the $k$ ranged from two to ten. In addition, as stated in section 4.2, leave-one-out cross-validation was performed. Since three datasets were acquired through experiment, two sets were used to train classifier and the other one set was used as testing data. The three graphs in Figure 9 correspond to each case of cross-validation.

The two dotted lines in Figure 9 are the results using $x_0$ and $y_0$ signals, while the single line is the results using ODR signals with PCA. The dotted lines represent the conventional data-driven method using only the signals acquired from the sensors. On the other hand, the single lines represent the results using ODR signals. The ODR signal not only uses the acquired signals but also signals in other directions. Thus the sixteen ODR signals can characterize health state of the system more accurately, and eventually gives enhanced results.

The (a), (b), and (c) in Figure 9 represent each case of the classification results of cross-validation. When number of optimal features were larger than three, the ODR signal based method predicted the testing data 100% accurate. In contrast to that, the results by the $x_0$ and $y_0$ signals are not consistent. The (a) case shows that the prediction accuracy increases as the number of features are increased, and 100% accuracy is obtained when five or more features are used. However, in (b), the accuracy of $x_0$ signals oscillates. Moreover, the accuracy of $y_0$ signals are relatively lower in (b) and (c). This results indicate that conventional method of using the acquired signals cannot guarantee good results, whereas ODR signals can give good results consistently regardless of direction of anomalies and sensors.

6. Conclusion

This research was conducted to enhance the diagnose performance on the health state of journal bearing systems. The data used in this study were acquired from RK4 test-bed. For non-direction oriented health states, normal and oil whirl were selected, whereas rubbing and misalignment were selected for direction oriented health states. From the two acquired signals, ODR signals were generated for each health state. The generated ODR signals represented vibration signals around the rotor, and all the signals were turned into time- and frequency- domain features. The sixteen features of sixteen ODR signals piled to 256 features, then the features were reduced by PCA. Finally feature selection and classification were performed. To add reliability to the study, leave-one-out cross-validation was performed on the three data sets.

We have suggested PCA based ODR method to diagnose the journal bearing rotor system. To the best of our knowledge, no research had tried to generate vibration signals based on the two signals in a right angle. This method is useful for characterizing the direction oriented health states as well as non-direction oriented ones. Characterizing the health states successfully lead to accurate diagnosis results, while signals without ODR presented inconsistent diagnosis results.

![Figure 7](image1.png)

Figure 7. Two cycle orbits of (a) normal, (b) rubbing, (c) misalignment, and (d) oil whirl.

![Figure 8](image2.png)

Figure 8. ODR signals of (a) normal, (b) rubbing, (c) misalignment, and (d) oil whirl.
Figure 9. Classification results by \( x_0, y_0, \) and ODR signals

For future works, study regarding the optimal number of ODR signals should be defined considering the computational resources and the accuracy of diagnosis. In other words, optimal \( \Delta \theta \) is to be determined using appropriate method. Furthermore, correlation among the ODR signals can be considered to reduce computational loads of calculating ODR signals.

ACKNOWLEDGEMENT

This work was supported by Mid-career Researcher Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science ICT and Future Planning (MSIP) (2013R1A2A2A01068627). This work was also supported by the Power Generation & Electricity Delivery Core Technology Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 2012101010001C).

REFERENCES


**Biographies**

**Joon Ha Jung** received B.S. degree of mechanical engineering from Seoul National University (SNU) in 2012, and is involved in graduate School of Mechanical and Aerospace Engineering at SNU. He is currently doing research on Prognostics and Health Management for rotating machinery.

**Byung Chul Jeon** received the B.S. degree of aerospace engineering from Republic of Korea Air Force Academy in 2001 and the M.S. degree of mechanical engineering from Seoul National University in 2009. He is now a Ph. D. student at the School of Mechanical and Aerospace Engineering in Seoul National University. His current research is focused on the data driven Prognostics and Health Management for rotating machinery.

**Byeng D. Youn** received his Ph.D. degree from the department of Mechanical Engineering at the University of Iowa, Iowa City, IA, in 2001. He was a research associate at the University of Iowa (until 2004), an assistant professor in Michigan Technical University (until 2007), and an assistant professor in the University of Maryland College Park (until 2010). Currently, he is an associate professor at the School of Mechanical and Aerospace Engineering at Seoul National University, Republic of Korea. His research is dedicated to well-balanced experimental and simulation studies of system analysis and design, and he is currently exploring three research avenues: system risk-based design, prognostics and health management (PHM), and energy harvester design. Dr. Youn’s research and educational dedication has led to: six notable awards, including the ISSMO/Springer Prize for the Best Young Scientist in 2005 from the International Society of Structural and Multidisciplinary Optimization (ISSMO), and more than 100 publications in the area of system-risk-based design, PHM and energy harvester design.

**Donghan Kim** received B.S degree from the department of mechanical engineering at the University of Inha, Incheon and the M.S degree from the department of mechanical engineering from Korea Advanced Institute of Science and Technology, Daejeon, in 2012. Currently, he is an associate researcher at Korea Electric power Research Institute, Daejeon, Korea. His research interests are fault detection, fault diagnosis, and failure prognosis of power plant systems and their applications.

**Yeon-Whan Kim** received his BS and MS degree from the department of Mechanical Engineering from Ajou University in 1987 and 1990, and his Ph.D. degree from the department of Mechanical Design Engineering at Chungnam National University in 2008, Korea. He is a Principal Researcher at Power Generation Laboratory in KEPCO Research Institute. His research is dedicated to machinery vibration analysis, noise analysis, mechanism identification on flow induced vibration, health condition monitoring and diagnostics in power plants. His current research is focused on prognostics and health management of rotating machinery in 1000MW class thermal power plant.
Quadrotor Accelerometer and Gyroscope Sensor Fault Diagnosis with Experimental Results

Remus C Avram1, Xiaodong Zhang2 and Jonathan Muse3

1,2 Wright State University, Dayton, Ohio, 45404, USA
avram.3@wright.edu
xiaodong.zhang@wright.edu

3 Air Force Research Laboratory, Dayton, OH, 45433, USA
jonathan.muse.2@us.af.mil

ABSTRACT
This paper presents the design and real-time experimental results of a fault diagnosis scheme for inertial measurement unit (IMU) measurements of quadrotor unmanned air vehicles (UAVs). The objective is to detect, isolate, and estimate sensor bias fault in accelerometer and gyroscope measurements. Based on the quadrotor dynamics and sensor models, the effects of sensor faults are represented as virtual actuator faults in the quadrotor state equations. Two robust diagnostic estimators are designed to provide structured residuals enabling the simultaneous detection and isolation of the sensor faults. Additionally, based on the detection and isolation scheme, two nonlinear adaptive estimators are employed for the estimation of the fault magnitude. The performance of the diagnosis method is evaluated and demonstrated through real-time flight experiments.

1. INTRODUCTION
Unmanned Aerial Vehicles (UAVs) have attracted significant attentions in recent years due to their potentials in various military and civilian applications, including security patrol, search and rescue in hazardous environment, surveillance and classification, attack and rendezvous (Shima & Rasmussen, 2008). Most quadrotors used in research are often equipped with low-cost and lightweight micro-electro-mechanical systems (MEMS) inertial measurement units (IMU) including 3-axis gyro, accelerometer and magnetometer. These sensors serve an essential role in most quadrotor control schemes. However, due to their intrinsic components and fabrication process, IMUs are vulnerable to exogenous signals and prone to faults. Specifically, accelerometer and gyroscope measurements are susceptible to bias and excessive noise as a result of temperature variation, vibration, etc. The detection and estimation of accelerometer and gyroscope faults plays a crucial role in the safe operations of quadrotors. In this paper we present a nonlinear method for detecting, isolating and estimating sensor bias faults in accelerometer and gyroscope measurements of quadrotor UAVs. Based on the fact that the accelerometer and the gyroscope measure forces/torque acting directly on the UAV body, the quadrotor dynamics are expressed in terms of the IMU sensor measurements. Two robust diagnostic estimators are designed to provide structured fault detection and isolation (FDI) residuals allowing simultaneous detection and isolation of gyroscope and accelerometer sensor bias in the presence of measurement noise. In addition, by utilizing nonlinear adaptive estimation techniques (Zhang, Polycarpou, & Parsini, 2001), adaptive estimators are employed to provide an estimate of the unknown sensor bias. The parameter convergence property of the adaptive estimation scheme is analyzed.

The remainder of the paper is organized as follows. Section 2 formulates the problem of sensor FDI for quadrotor UAVs. The proposed fault detection and isolation method is presented in Section 3. Section 4 describes the adaptive estimator algorithms for estimating the unknown sensor bias magnitude and provides conditions for parameter convergence. Section 5 and 6 present experimental results and direction of future research, respectively.

2. PROBLEM FORMULATION
As in (Leishman, Jr., Beard, & McLain, 2014) and (Martin & Salaün, 2010), the dynamic model used in this paper considers the gravity, thrust generated by the rotors and drag forces acting on the quadrotor body. The quadrotor nominal system dynamics are derived from the Newton-Euler equations.
of motion and are given by:

\[
\dot{p}_E = v_E
\]
\[
\dot{v}_E = \frac{1}{m} R_{EB}(\eta) \begin{bmatrix} 0 \\ 0 \\ -U \end{bmatrix} - c_d v_B + 0 \tag{1}
\]
\[
\dot{\eta} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \sin \phi \tan \theta \\ \cos \phi \tan \theta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2}
\]
\[
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} J_x - J_z \\ J_y - J_z \\ J_x - J_y \end{bmatrix} \begin{bmatrix} q r \\ p r \\ p q \end{bmatrix} + \begin{bmatrix} \frac{1}{2} r \phi \\ \frac{1}{2} q \theta \cos \phi \\ \frac{1}{2} q \phi \cos \theta \end{bmatrix} \tag{3}
\]

where \( p_E \in \mathbb{R}^3 \) is the inertial position, \( v_E \in \mathbb{R}^3 \) is the velocity expressed in the Earth frame, \( \eta = [\phi, \theta, \psi]^T \in \mathbb{R}^3 \) are the roll, pitch and yaw Euler angles, respectively, and \( \omega = [p, q, r]^T \) represents the angular rates, \( m \) is the mass of the quadrupor, and \( g \) is the gravitational acceleration. The terms \( J_x, J_y, J_z \) represent the quadrotor inertias about the body x-, y- and z-axis, respectively. Note that the quadrotor is assumed to be symmetric about the xz and yz planes (i.e. the product of inertias is zero). \( U \) represents the total thrust generated by the rotors, \( \tau_\phi, \tau_\theta, \tau_\psi \) are the torques acting on the quadrupor around the body x-, y- and z-axis, respectively. The term \( c_d v_B \) represents the drag force acting on the vehicle, with \( c_d \) being drag force coefficient and \( v_B \) is the velocity of the UAV relative to the body frame.

The system model described by Eq (1) - (4) is expressed with the velocity relative to the inertial frame. The inertial coordinate system is assumed to have the positive x-axis pointing North, the positive y-axis pointing East and positive z-axis pointing down towards the Earth’s center. The transformation from the body frame to inertial frame is given by the rotation matrix \( R_{EB} \) and is defined based on a 3-2-1 rotation sequence as follows:

\[
R_{EB}(\eta) = \begin{bmatrix} c\theta c\psi & s\theta c\psi & -s\psi & c\theta c\psi & s\theta c\psi & s\theta s\psi & c\phi s\psi & -c\phi c\psi & s\phi \end{bmatrix}
\]

where \( s \) and \( c \) are short hand notations for the \( \sin(\cdot) \) and \( \cos(\cdot) \) functions, respectively. As in (Leishman et al., 2014), by assuming that the nonlinear Coriolis terms are small enough to be negligible, the quadrotor velocity dynamics relative to the body frame are expressed as

\[
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -U \end{bmatrix} - c_d v_B + \begin{bmatrix} -g \sin \theta \\ g \sin \phi \cos \theta \\ g \cos \phi \cos \theta \end{bmatrix} \tag{5}
\]

where \( v_B = [u, v, w]^T \), represents the velocities along the body x-, y- and z-direction. The relation between the inertial velocity and body velocity is given by \( v_E = R_{EB} v_B \).

As in (Ireland & Anderson, 2012) and (Lantos & Marton, 2011), it is assumed that Euler angles measurements are available. For instance, these measurements can be generated by a camera-based motion capture system, a technology commonly employed for in-door UAV flight (Guenard, Hamel, & Mahony, 2008).

MEMS sensors, such as accelerometers and gyroscopes, measure forces and moments acting in the body frame. The quantity expressed inside the parenthesis in the inertial velocity dynamics described by Eq (2), represents all the forces acting on the body. Therefore, the inertial velocity dynamic equation can be adjusted to reflect accelerometer measurements. Similarly, the evolution of Euler angles can be rewritten in terms of gyroscope measurements. By considering IMU measurement’s susceptibility to bias faults, the accelerometer and gyroscope sensor measurements are given by:

\[
y_a = a + b_a + d_a
\]
\[
= \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -U \end{bmatrix} - c_d v_B + \beta_a(t - T_a) b_a + d_a \tag{6}
\]
\[
y_\omega = \omega + b_\omega + d_\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \beta_\omega(t - T_\omega) b_\omega + d_\omega \tag{7}
\]

where \( y_a \in \mathbb{R}^3 \) and \( y_\omega \in \mathbb{R}^3 \) are the accelerometer and gyroscope measurements, respectively, \( b_a \in \mathbb{R}^3 \) and \( b_\omega \in \mathbb{R}^3 \) represent the possible faults in accelerometer and gyroscope measurements, respectively. The terms \( d_a \) and \( d_\omega \) represent the noise in the sensor measurements, and \( a \) represents the nominal acceleration measurement without bias, that is:

\[
a = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -U \end{bmatrix} - c_d v_B \tag{8}
\]

The fault time profile functions \( \beta_a(\cdot) \) and \( \beta_\omega(\cdot) \) are assumed to be step functions with unknown fault occurrence times \( T_a \) and \( T_\omega \), respectively. Specifically,

\[
\beta_a(t - T_a) = \begin{cases} 0, & \text{when } t < T_a \\ 1, & \text{when } t \geq T_a \end{cases}
\]
\[
\beta_\omega(t - T_\omega) = \begin{cases} 0, & \text{when } t < T_\omega \\ 1, & \text{when } t \geq T_\omega \end{cases}
\]

In addition, it is assumed that the position measurements in the Earth frame available. Hence, the system model can be augmented by the following output equation:

\[
y_p = p_E + d_p , \tag{9}
\]

where \( d_p \) represents zero mean position measurement noise.

Assumption 1. The bias in accelerometer and gyroscope measurements are assumed to be constant and bounded.
Remark. It is worth noting that, in practical applications, after the occurrence of an IMU sensor bias, its magnitude may be time-varying and grow slowly over time. However, the change in the bias is often small over a short time duration. Therefore, the bias may be assumed to be constant during the short time duration under consideration.

Assumption 2. The sensor measurement noise signals denoted by \( d_x, d_\omega \) and \( d_p \) are assumed to be bounded zero mean signals. That is:

\[
\mathbb{E}(d_x) = 0, \quad \mathbb{E}(d_\omega) = 0, \quad \mathbb{E}(d_p) = 0,
\]

where \( \mathbb{E} \) represents the expectation operator.

The objective of this research focuses on the development and demonstration of a robust fault detection, isolation and estimation scheme for sensor faults in accelerometer and gyroscope measurements.

### 3. Fault Detection and Isolation

This section presents the proposed method for detecting and isolating sensor faults in accelerometer and gyroscope measurements. Substituting the sensor model given by Eq (6)-(7) into the systems dynamics Eq (1)-(4), we obtain:

\[
\begin{align*}
\dot{\hat{p}}_E &= v_E \quad (10) \\
\dot{\hat{v}}_E &= R_{EB}(\eta) y_a + \begin{bmatrix} 0 \\ 0 \end{bmatrix} - R_{EB}(\eta) \beta \alpha b - R_{EB} d_a \quad (11)
\end{align*}
\]

\[
\begin{align*}
\dot{\eta} &= R_{\beta}(\phi, \theta) y_\omega - R_{\beta}(\phi, \theta) \beta \omega b - R_{\beta}(\phi, \theta) d_\omega \quad (12) \\
\dot{\omega} &= \begin{bmatrix} J_x - J_z \begin{bmatrix} y_q - \beta \omega b_q - d_q \end{bmatrix} (y_r - \beta \omega b_r - d_r) \\ J_z - J_x \begin{bmatrix} y_p - \beta \omega b_p - d_p \end{bmatrix} (y_q - \beta \omega b_q - d_q) \end{bmatrix} \\ + \begin{bmatrix} \frac{J_z}{J_x} \phi \\ \frac{J_x}{J_z} \theta \end{bmatrix} && (13)
\end{align*}
\]

where \( R_{\beta}(\phi, \theta) \) is the rotation matrix relating angular rates to Euler angle rates and is given by:

\[
R_{\beta}(\phi, \theta) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}.
\]

As can be seen from Eq (10)-(13), a bias in accelerometer measurements affects only the position and velocity states. Conversely, gyroscope measurements affect only Euler angles and angular rates states. Based on this observation, it follows naturally to also divide the fault diagnosis of these two sensor faults. The proposed fault detection, isolation and estimation architecture is shown in Figure 1. As can be seen, two FDI estimators monitor the system for fault occurrences in accelerometer and gyroscope measurements. Once a fault is detected and isolated, the corresponding nonlinear adaptive estimator is activated for sensor bias estimation purposes.

![Figure 1. Fault detection, isolation and estimation architecture.](image)

### 3.1. Gyroscope Fault Diagnostic Estimator

As can be seen from the dynamics of the quadrotor, given by equations (10)-(13), the bias in the gyroscope measurements only affects the attitude and rotation dynamics given by Eq (12)-(13). Based on Eq (12) and adaptive estimation schemes (Ioannou & Sun, 1996), the fault diagnostic estimator for the gyroscope bias can be designed as follows:

\[
\dot{\hat{\eta}} = -\Lambda (\hat{\eta} - \eta) + R_{\beta}(\phi, \theta) y_\omega , \quad (14)
\]

where \( \hat{\eta} \in \mathbb{R}^3 \) are the Euler angle estimates, \( \Lambda \in \mathbb{R}^{3 \times 3} \) is a positive-definite diagonal design matrix. Let the Euler angle estimation error be defined as:

\[
\tilde{\eta} = \eta - \hat{\eta} . \quad (15)
\]

Based on Eq (12) and Eq (14), the dynamics of the attitude angle estimation error are given by:

\[
\dot{\tilde{\eta}} = \dot{\eta} - \dot{\hat{\eta}} = -\Lambda \tilde{\eta} - R_{\beta}(\phi, \theta) \beta \omega b - R_{\beta}(\phi, \theta) d_\omega . \quad (16)
\]

By design, the homogeneous part of Eq (16) is exponentially stable. In the absence of a gyroscope fault (i.e. \( t < T_0 \)), the attitude angle estimation error is given by:

\[
\tilde{\eta}(t) = e^{-\Lambda (t-T_0)} \tilde{\eta}(0) - \int_0^t e^{-\Lambda (t-\tau)} R_{\beta}(\phi, \theta) d_\omega d\tau = r_\omega (t) + e_\omega (t) , \quad (17)
\]

where \( r_\omega (t) \triangleq e^{-\Lambda (t-T_0)} \tilde{\eta}(0) \) converges exponentially fast to zero, and \( e_\omega (t) \) represents an additive noise term generated by filtering the measurement noise \( d_\omega \) through the following linear filter:

\[
\dot{e}_\omega = -\Lambda e_\omega - R_{\beta}(\phi, \theta) d_\omega
\]

In addition, in the presence of a non-zero bias \( b_\omega \), based on Eq (16), it can be seen that the residual \( \tilde{\eta} \) will deviate from zero. Therefore, if any component of the state estimation error \( \tilde{\eta} \) is significantly different from zero, we can conclude that a fault in the gyroscope measurements has occurred.
3.2. Accelerometer Fault Diagnostic Estimator

The dynamics of UAV position and velocity relative to the inertial frame given by Eq (10) and Eq (11) can be represented by the following state space model:

\[
\dot{x} = Ax + f(\eta, y_a) + G_a(\eta)\beta_a b_a + D_a(\eta, t) \\
y = Cx + d_p,
\]

where \(x = [p_E^T, v_E^T]^T\), \(y = p_E\), and

\[
A = \begin{bmatrix}
0_{3\times3} & I_3 \\
0_{3\times3} & 0_{3\times3}
\end{bmatrix}, \quad G_a(\eta) = \begin{bmatrix}
0_{3\times3} \\
-R_{EB}
\end{bmatrix}, \\
f(\eta, y_a) = \begin{bmatrix}
0_{3\times1} \\
R_{EB} y_a + \begin{bmatrix}
0 \\
0
\end{bmatrix}
\end{bmatrix}, \quad D_a(\eta, t) = \begin{bmatrix}
0_{3\times1} \\
-R_{EB} d_a
\end{bmatrix},
\]

and \(C = [I_3, 0_{3\times3}]\), where \(I_3\) is a 3 \times 3 identity matrix, \(0_{3\times3}\) is a 3 \times 3 matrix with all entries zero and \(0_{3\times1}\) is a 3 \times 1 zero vector. Based on this configuration, the following fault diagnostic observer is chosen:

\[
\dot{\hat{x}} = A\hat{x} + f(\eta, y_a) + L(y - \hat{y}) \\
\hat{y} = C\hat{x},
\]

where \(\hat{x} \in \mathbb{R}^6\) represents the inertial position and velocity estimation, \(\hat{y} \in \mathbb{R}^3\) are the predicted position outputs, \(L\) is a design matrix chosen such that the matrix \(\hat{A} \triangleq (A - LC)\) is stable. Let us define the position estimation error and the state estimation error as:

\[
\tilde{y} \triangleq y - \hat{y} \\
\tilde{x} \triangleq x - \hat{x}.
\]

By using equations (18) - (19), the estimation error dynamics are given by:

\[
\dot{\tilde{x}} = \tilde{A}\tilde{x} + G_a(\eta)\beta_a b_a + D_a(\eta, t) - Ld_p \\
\dot{\tilde{y}} = C\tilde{x} + d_p
\]

(22)

In the absence of accelerometer bias, the position estimation error is given by:

\[
\tilde{y} = Ce^{\tilde{A}(t-T_0)}\tilde{x}(0) + C \int_0^t e^{\tilde{A}(t-\tau)}(D_a(\eta, t) - Ld_p)d\tau + d_p \\
=r_a(t) + e_a(t) + d_p
\]

(23)

where \(r_a(t) \triangleq Ce^{\tilde{A}(t-T_0)}\tilde{x}(0)\) converges exponentially fast to zero, and \(e_a(t)\) represents an additive noise term generated by filtering \(d_a\) and \(d_p\) through the following linear filter:

\[
\dot{e}_a = \tilde{A}e_a + (D_a(\eta, t) - Ld_p).
\]

Clearly, the output estimation error \(\tilde{y}\) reaches a small value, centered around zero, exponentially fast in the absence of the accelerometer bias \(b_a\). Furthermore it can be seen from Eq (22) the residual \(\tilde{y}\) is only sensitive to the bias \(b_a\). Therefore, if any component of the position estimation error \(\tilde{y}\) deviates significantly from zero, we can conclude that a fault in the accelerometer sensor measurement has occurred.

3.3. Fault Detection and Isolation Decision Scheme

As described in Section 3.1 and 3.2, the two fault diagnostic estimators are designed such that each of them is only sensitive to one type of sensor faults. Based on this observation, the residuals \(\tilde{\eta}\) and \(\tilde{\gamma}\) generated by Eq (17) and Eq (23) can also be used as structured residuals for fault isolation. More specifically, we have the following fault detection and isolation decision scheme:

- In the absence of any faults, all components of the residuals \(\tilde{\eta}\) and \(\tilde{\gamma}\) should be close to zero.
- If all components of the residual \(\tilde{\gamma}\) remain around zero, and at least one component of the residual \(\tilde{\eta}\) is significantly different from zero, then we conclude that an accelerometer fault has occurred.
- If all components of the residual \(\tilde{\gamma}\) remain around zero, and at least one component of the residual \(\tilde{\eta}\) is significantly different from zero, then we conclude that a gyroscope fault has occurred.
- If at least one component of the residual \(\tilde{\eta}\) and at least one component of the residuals \(\tilde{\gamma}\) are simultaneously significantly different from zero, then we conclude that both a gyroscope and accelerometer sensor measurement fault has occurred.

The above FDI decision scheme is summarized in Table 1, where “0” represents residuals with zero mean, and “1” represents significantly non-zero residuals.

| Table 1. Fault Isolation Decision Truth Table. |
|---|---|---|---|---|
| No Fault | Gyro Bias | Accel Bias | Accel & Gyro Bias |
| \(\eta\) | 0 | 1 | 0 | 1 |
| \(\gamma\) | 0 | 0 | 1 | 1 |

4. FAULT ESTIMATION

After a sensor fault is detected and isolated, it is also crucial to provide an estimation of the sensor bias to improve the performance of the closed loop control system. As shown in Figure 1, once a fault has been detected and isolated, the corresponding nonlinear adaptive bias estimator is activated with the purpose of estimating the fault magnitude in the accelerometer and/or gyroscope measurements. In this section, we describe the design of nonlinear adaptive estimators for sensor bias estimation.
4.1. Accelerometer Fault Estimation

Based on Eq (18), the adaptive observer for estimating the accelerometer bias magnitude is chosen as:

\[
\dot{x} = Ax + f(\eta, y_a) + L(y - \hat{y}) + \dot{G}_a(\eta)\hat{b}_a + \Omega \hat{b}_a \tag{24}
\]

\[
\dot{\Omega} = (A - LC)\Omega + \dot{G}_a(\eta) \tag{25}
\]

\[
\dot{\hat{y}} = C\hat{x}, \tag{26}
\]

where \(\hat{x}\) is the estimated position and velocity vector, \(\hat{y}\) is the estimated position output, \(\hat{b}_a\) is the estimated sensor bias, and \(L\) is the observer gain matrix. The adaptation in the above adaptive estimator arises due to the unknown bias \(b_a\). The adaptive law for updating \(\hat{b}_a\) is derived using Lyapunov synthesis approach (Bastin & Gevers, 1988; Zhang, 2011) and is given by:

\[
\dot{\hat{b}}_a = \mathcal{P}_\Theta \{\Gamma \Omega^T C^T \hat{y}\}. \tag{27}
\]

In order to guarantee the stability of the parameter estimation in the presence of unknown modeling errors and measurement noise (Ioannou & Sun, 1996), the projection operator \(\mathcal{P}\) restricts the parameter estimate to a known compact convex region \(\Theta\), defined by \(\hat{b}_a^T \hat{b}_a < M^2\), where \(M\) is a positive constant. Specifically, the adaptive algorithm is given by:

\[
\dot{\hat{b}}_a = \begin{cases} 
\Gamma \Omega^T C^T \hat{y}, & \text{if } ||\hat{b}_a|| = M \text{ and } \hat{b}_a^T \Gamma \Omega^T C^T \hat{y} \leq 0 \\
\Gamma \Omega^T C^T \hat{y} - \Gamma \frac{\hat{b}_a \hat{\eta}}{\hat{b}_a^T \hat{\eta}} \Gamma \Omega^T C^T \hat{y}, & \text{otherwise}
\end{cases} \tag{28}
\]

where \(\Gamma > 0\) is a symmetric and positive-definite learning rate matrix, and \(\hat{y}_a \triangleq y_a - \hat{y}_a\) is the output estimation error. Let us also define the state estimation error as \(\hat{x} \triangleq x - \hat{x}\), and the parameter estimation error as \(\hat{b}_a \triangleq \hat{b}_a - b_a\). The stability and performance properties of the above adaptive scheme are described below.

**Theorem 4.1.** In the presence of an accelerometer measurement bias, if there exists constants \(\alpha_1 \geq \alpha_0 > 0\) and \(T_0 > 0\), such that

\[
\alpha_1 I \geq \frac{1}{T_0} \int_t^{t+T_0} \Omega^T C^T C \Omega d\tau \geq \alpha_0 I, \tag{29}
\]

then, the adaptive scheme described by Eq (24) - (26) and Eq (28) ensures that:

1. all signals in the adaptive scheme remain bounded,
2. \(E(\hat{x})\) and \(E(\hat{b}_a)\) converge exponentially to zero.

The proof of the above theorem is omitted here due to space limitation. Interested readers please contact the corresponding author for details (Avram, 2015).

4.2. Gyroscope Fault Estimation

Once a gyroscope bias fault is detected and isolated, the following adaptive estimator is activated in order to estimate the bias in the gyroscope sensor:

\[
\hat{\eta} = -\Lambda(\hat{\eta} - \eta) + R_\eta(\phi, \theta) y_o - R_\eta(\phi, \theta) \hat{b}_o \tag{30}
\]

\[
\hat{b}_o = \Gamma R_\eta(\phi, \theta)(\hat{\eta} - \eta), \tag{31}
\]

where \(\hat{\eta}\) is the Euler angle estimate, \(\hat{b}_o\) represents the estimation of the sensor bias, \(\Lambda\) and \(\Gamma\) are positive definite design matrices. The adaptive law for estimating the bias in gyroscope measurements in Eq (30) - (31) is derived using Lyapunov synthesis approach (Ioannou & Sun, 1996). In addition, in order to ensure parameter convergence, \(R_\eta(\phi, \theta)\) will also need to satisfy the persistence of excitation condition (Ioannou & Sun, 1996), that is:

\[
\alpha_1 I \geq \frac{1}{T_0} \int_t^{t+T_0} R_\eta(\phi, \theta)^T R_\eta(\phi, \theta) d\tau \geq \alpha_0 I\tag{32}
\]

for some constants \(\alpha_1 \geq \alpha_0 > 0\) and \(T_0 > 0\) and for all \(t \geq 0\). Let us define the attitude angle estimation error as \(\hat{\eta} - \eta\) and the bias estimation error as \(\hat{b}_o - b_o\). The stability and learning performance of the adaptive scheme Eq (30) - (31) is summarized by the following theorem.

**Theorem 4.2.** In the presence of a gyroscope bias, if there exists constants \(\alpha_1 \geq \alpha_0 > 0\) and \(T_0 > 0\) such that Eq (32) is satisfied, then the adaptive scheme in Eq (30) - (31) ensures that:

1. all signals in the adaptive scheme remain bounded
2. \(E(\hat{\eta})\) and \(E(\hat{b}_o)\) converge exponentially to zero.

For the sake of space limitation, the complete proof of the above theorem is purposely omitted. Interested readers please contact the corresponding author for details (Avram, 2015).

5. Experimental Results

In this section, experimental results using a real-time quadrotor test environment of Wright State University are described to illustrate the effectiveness of the sensor fault diagnosis algorithm. A block diagram of the experimental system setup is shown in Figure 2. During flight tests, quadrotor position and attitude information is obtained from a Vicon motion capture camera system. Position and Euler angle measurements are collected every 10ms and relayed from a Vicon dedicated PC via TCP/IP connection to a ground station computer. As in (Macdonald, Leishman, Beard, & McLain, 2014), we corrupted position measurements with normal noise. In this research we chose the noise standard deviation to be 0.25m. Additionally, position measurements are down sampled to 1Hz, in order to further simulate real world applications. The fault diagnosis method is evaluated in real-time during autonomous flight of a quadrotor built in-house with off-the-shelf components. The quadrotor is equipped with the Qbrain embedded control module from Quanser Inc. The control module consists of a HiQ acquisition card providing real-time IMU measurements, and a Gumstix Duo Vero micro-
controller running the real-time control software. An IEEE 802.11 connection between the ground station PC and the Gumstix allows for fast and reliable wireless data transmission and on-line parameter tuning. Position and attitude information obtained from the Vicon system along with trajectory commands generated by the ground station are sent to the quadrotor in order to achieve real-time autonomous flight. The control software executes on-board at 500Hz, and accelerometer and gyroscope measurement are logged at 200Hz. During the experimental stage, the quadrotor is commanded to move in a circular trajectory, while maintaining constant orientation and altitude. As previously shown, the fault diagnosis technique employed in this approach is independent of the structure of the controller. Therefore, for brevity, the discussion on the control design is purposely omitted.

In order to evaluate the proposed diagnosis method, we log approximately 2.5 minutes of autonomous flight data. Sensor bias is artificially injected into the accelerometer and gyroscope measurements, respectively, while the quadrotor is airborne. Figure 3 shows the fault time profile of the two sensor faults. As can be seen, an accelerometer fault is injected approximately at time $t = 35s$ until $t = 60s$. From approximately $t = 62s$ until $t = 95s$, a gyroscope bias is introduced into the sensor measurements. Additionally, in order to evaluate the performance of the proposed FDIE algorithm in the presence of multiple faults, both accelerometer and gyroscope faults are injected at approximately $t = 97s$. Flight data is processed on-line, and real-time sensor fault diagnostic decision is provided by the diagnostic algorithm. In the following sections, we present the evaluation results of the diagnosis method using real-time flight data.

### 5.1. Case of Accelerometer Bias

The performance of the proposed FDIE in the presence of accelerometer measurement bias fault is shown in this section. At approximately time $t = 35s$, a constant bias $b_a = [0.1, 0.15, 0.9]^T m/s^2$, is injected into the accelerometer measurements. Figure 4 shows the residuals generated by the two diagnostic estimators described by Eq (14) and Eq (19), respectively. In order to enhance the diagnostic decision based on the FDI logic given by Table 1, the two-sided cumulative sum (CUSUM) test is applied to process the diagnostic residuals (Gustafsson, 2000). Figure 5 shows the statistic property generated by the CUSUM test. A fixed threshold is chosen for the detection and isolation of sensor faults. As can be seen, shortly after the occurrence of the fault, at least one component of the test statistic corresponding to the residuals generated by the accelerometer diagnostic estimator exceeds the detection threshold. On the other hand, all components of the test statistic corresponding to the gyroscope bias remain well below the detection threshold. Based on the detection and isolation logic given in Table 1, we can conclude that a fault has occurred in the accelerometer measurement. In addition, Figure 6 shows the estimation of the bias in the accelerometer converges closely to the actual value.
5.2. Case of Gyroscope Bias

A gyroscope bias with \( b_\omega = [5, -7, -10]^{T} \text{o/s} \) is injected into the sensor measurements at approximately time \( t = 62s \). Figure 7 shows the statistic property generated by the CUSUM test when applied to the residuals generated by Eq (14) and Eq (19), respectively. As can be seen, at least one component of the test statistic corresponding to the residuals generated by the gyroscope diagnostic estimator exceeds the manually chosen detection threshold shortly after the occurrence of the gyroscope fault. On the other hand, all components of the test statistic corresponding to the accelerometer bias remain well below the detection threshold. Based on the detection and isolation logic given in Table 1, we can conclude that a fault has occurred in the gyroscope measurement. In addition, Figure 8 shows the estimation of the bias in the gyroscope for each axis, respectively. As can be seen, after a short time, the estimate of gyroscope bias is reasonably close to its actual value.

5.3. Case of Simultaneous Faults

In this section, we present the results of the diagnostic method in the presence of both accelerometer and gyroscope faults. Specifically, at time \( t = 97s \), biases \( b_a = [0.1, 0.15, 0.9]^{T} \text{m/s}^2 \) and \( b_\omega = [5, -7, -10]^{T} \text{o/s} \) are injected into accelerometer and gyroscope measurements, respectively. Figure 9 shows the statistic property generated by the CUSUM test. As can be seen, shortly after the occurrence of the faults, the test statistics corresponding to both diagnostic estimators exceed their respective detection threshold. Hence, we can conclude that faults have occurred in both accelerometer and gyroscope measurements. Furthermore, Figure 10 and Figure 11 show the estimation of the accelerometer and gyroscope biases, respectively. As can be seen, estimation results are satisfactory.

6. Conclusion and Future Work

In this paper, we present the design of a nonlinear fault diagnostic method for sensor bias faults in accelerometer and gyroscope measurements of quadrotor UAVs. Based on the idea that accelerometer and gyroscope measurements coincide with translational and rotational forces acting on the body, respectively, two FDI estimators are designed to generate structured residuals for fault detection and isolation. In addition,
nonlinear adaptive estimation estimation schemes are presented to provide an estimate of the sensor bias. The proposed diagnostic method is implemented on a quadrotor UAV test environment and is demonstrated during real-time autonomous flight data. An interesting direction for future research is to develop and demonstrate a systematic diagnostic method for quadrotor actuator faults.

**REFERENCES**


**Biographies**

**Remus C. Avram** is a PhD candidate at Wright State University, Dayton OH. Remus, received his B.S. in Electrical Engineering with concentration in Computer Engineering at University of Texas, San Antonio, Texas in 2009. In 2011, Remus obtained his M.S. in Electrical Engineering with concentration in Control Systems from Wright State University, Dayton, Ohio. He is currently pursuing a doctoral degree in Electrical Engineering. Remus’s main research include nonlinear health diagnostic methods and real-time intelligent control.

**Xiaodong Zhang** received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from University of Cincinnati, Cincinnati, OH, USA, all in electrical engineering, in 1994, 1997 and 2001, respectively. He is currently an Associate Professor of Electrical Engineering Department, Wright State University, Dayton, OH. His research interests include intelligent control systems, fault diagnosis and prognosis, fault-tolerant control, verification and validation of control systems for safety assurance, etc. He is an Associate Editor of the IEEE Transactions on Control Systems Technology and a member of the IFAC SAFEPROCESS Technical Committee.

**Jonathan Muse** received an undergraduate degree from the University of Alabama in 2005, a Masters Degree from the Georgia Institute of Technology in 2008, and a PhD in Aerospace Engineering from the Georgia Institute of Technology in 2010. He is currently a research engineer for the Aerospace Systems Directorate at the Air Force Research Lab (AFRL). At AFRL, he is responsible for executing basic research related to hypersonic vehicle control and advanced methods in nonlinear control. He is the Co-PI for HIFiRE flight 6, an adaptive flight control experiment on a scramjet hypersonic vehicle currently being constructed. He also holds an adjunct professor position at Wright State University in the Electrical Engineering department where he teaches control theory and UAV flight control. In addition, he also serves on various advisory boards for both industry and academia.
Measurement Science for Prognostics and Health Management for Smart Manufacturing Systems: Key Findings from a Roadmapping Workshop

Brian A. Weiss¹, Gregory Vogl², Moneer Helu³, Guixiu Qiao⁴, Joan Pellegrino⁵, Mauricio Justiniano⁶, and Anand Raghunathan⁷

¹,²,³,⁴ National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, 20899, USA
  brian.weiss@nist.gov
gregory.vogl@nist.gov
moneer.helu@nist.gov
guixiu.qiao@nist.gov

⁵,⁶,⁷ Energetics Incorporated, Columbia, Maryland, 21046, USA
  jpellegrino@energetics.com
  mjustiniano@energetics.com
  araghunathan@energetics.com

ABSTRACT

The National Institute of Standards and Technology (NIST) hosted the Roadmapping Workshop – Measurement Science for Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) in Fall 2014 to discuss the needs and priorities of stakeholders in the PHM4SMS technology area. The workshop brought together over 70 members of the PHM community. The attendees included representatives from small, medium, and large manufacturers; technology developers and integrators; academic researchers; government organizations; trade associations; and standards bodies. The attendees discussed the current and anticipated measurement science challenges to advance PHM methods and techniques for smart manufacturing systems; the associated research and development needed to implement condition monitoring, diagnostic, and prognostic technologies within manufacturing environments; and the priorities to meet the needs of PHM in manufacturing.

This paper will summarize the key findings of this workshop, and present some of the critical measurement science challenges and corresponding roadmaps, i.e., suggested courses of action, to advance PHM for manufacturing. Milestones and targeted capabilities will be presented for each roadmap across three areas: PHM Manufacturing Process Techniques; PHM Performance Assessment; and PHM Infrastructure – Hardware, Software, and Integration. An analysis of these roadmaps and crosscutting themes seen across the breakout sessions is also discussed.

1. INTRODUCTION

The National Institute of Standards and Technology (NIST) is a research agency with the United States (U.S.) Department of Commerce that develops measurement science to advance innovative and emerging technologies to increase U.S. industry’s competitiveness on a global scale. One specific area of NIST research is focused on Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS). To that end, it is critical for the NIST-PHM4SMS project team to understand the needs of its stakeholder community to develop and evolve the project’s research plan, accordingly. A Roadmapping Workshop – Measurement Science for Prognostics and Health Management for Smart Manufacturing Systems was hosted by NIST on November 19th and 20th, 2014 to discuss the needs and priorities of stakeholders in the PHM4SMS technology area. The workshop brought together over 70 members of the PHM community including representatives from small, medium, and large manufacturers; technology developers and integrators; academic researchers; government organizations; trade associations; and standards bodies. The attendees discussed the current and anticipated measurement science challenges that they felt the PHM community should address to advance PHM methods and techniques for smart manufacturing systems. Attendees also described the
associated research and development (R&D) needs that are hindering the advancement and implementation of condition monitoring, diagnostic, and prognostic technologies within manufacturing environments. Finally, attendees identified the priorities and next steps to meet the needs of PHM in manufacturing.

This paper begins by offering background on NIST’s efforts in Smart Manufacturing and PHM in Section 2. Section 3 summarizes the key findings of the workshop including highlights from the panel discussions and breakout sessions. The breakouts focused on the participants in three areas: PHM Manufacturing Process Techniques; PHM Performance Assessment; and PHM Infrastructure – Hardware, Software, and Integration. For each area, participants identified the area’s goals, desired capabilities, challenges and barriers to developing these capabilities, and specific roadmaps with milestones and targets to achieve these goals and capabilities. Section 3 also highlights some of the crosscutting themes that emerged throughout the workshop. Section 4 concludes with a discussion of the NIST-PHM4SMS team’s existing research plans.

2. BACKGROUND

2.1. Smart Manufacturing Systems (SMS)

NIST’s mission is to promote U.S. competitiveness across many technological areas including manufacturing. Smart Manufacturing has been identified by numerous U.S. leadership organizations (including the Executive Office of the President) as a necessity for U.S. manufacturers to increase their global competitiveness (Manyika, Sinclair, Dobbs, Strube, Rassey, Mischke, Remes, Roxburgh, George, O’Halloran & Ramaswamy, 2012) (PCAST, 2012) (PCAST, 2014). Smart Manufacturing Systems (SMS) are the synthesis and integration of advanced physical and virtual technologies to enable innovative processes and enhance existing methods. SMS includes the convergence of information and communication technologies with a range of sophisticated and emerging capabilities in a wide range of domains including sensing, automation, machining, robotics, and additive manufacturing. The effective and efficient use, and integration of these technologies is promoting manufacturing growth by enabling manufacturers to increase their productivity, quality, and safety, while reducing their costs and waste (Bernaden, 2012).

NIST has developed a suite of Smart Manufacturing programs (including robotics and additive manufacturing) to address the measurement science challenges faced by manufacturers who are actively looking to grow and/or enhance their operations. One of the programs is the Smart Manufacturing Operations Planning and Control (SMOPAC) program which is designed to tackle technological and integration challenges posed at the factory level.

PHM is a critical part of Smart Manufacturing. PHM may ultimately enable a machine or system to self-diagnose and self-heal with enough intelligence to be both aware of its current health and make an appropriate decision given both its state and goals. Presently, condition-monitoring, diagnostic, and prognostic techniques are not at the level required to enable this ultimate PHM vision; additional research is required.

2.2. Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS)

Within SMOPAC, the PHM4SMS project is aimed at developing the necessary measurement science to enable and enhance condition-monitoring, diagnostics, and prognostics. This measurement science includes the development of performance metrics, test methods, predictive modeling and simulation tools, reference data sets, protocols, and technical data.

The first of three phases of the PHM4SMS project is focused on assessment (the other phases are development and standardization): understanding the existing PHM capabilities, challenges, and needs of the manufacturing community and identifying the gaps that, if addressed, could benefit industry. This assessment phase has been marked by extensive research into PHM, both within literature reviews and direct interactions with PHM stakeholders (e.g., manufacturing process maintenance engineers, process design engineers, equipment operators). The NIST team has gained valuable insight about preventative/time-based maintenance (Ahmad & Kamaruddin, 2012) (Coats, Hassan, Goodman, Blechertas, Shin & Bayoumi, 2011); predictive maintenance/condition-based maintenance (Butcher, 2000) (Byington, Roemer, Kacprzykni & Galie, 2002) (Montgomery, Banjevic & Jardine, 2012) (Tian, Lin & Wu, 2012); and proactive/intelligent maintenance including maintenance at complex system levels (Barajas & Srinivasa, 2008) (Lee, Ghaffari & Elmeligy, 2011) (Lee, Ni, Djurdjanovic, Qiu & Liao, 2006).

Likewise, studies and reviews have been identified that compare existing PHM methods along with highlighting their strengths and limitations (Kothamasu, Huang & VerDuin, 2006) (Mulier, Crespo Marquez & Iung, 2008) (Peng, Dong & Zuo, 2010). More specifically, reviews of PHM-based standards have also been conducted (Vogl, Weiss & Donmez, 2014a) (Vogl, Weiss & Donmez, 2014b) (Zhou, Bo & Wei, 2013).

Besides NIST efforts in reviewing the existing PHM techniques and standards landscapes, NIST has actively engaged numerous manufacturers to directly understand their PHM capabilities, successes, challenges, and needs. This has included site visits with many small, medium, and large manufacturers from a range of industries including automotive, aerospace, defense, earth-moving, and electromagnetic. Stakeholder engagement peaked with the
planning and execution of the Roadmapping Workshop on Measurement Science for Prognostics and Health Management for Smart Manufacturing Systems.

3. WORKSHOP
The NIST PHM4SMS project team contracted with workshop facilitation and documentation experts at Energetics Corporation to host a two-day workshop. This workshop brought together PHM stakeholders including small, medium, and large manufacturers; technology developers and integrators; standards bodies; academic researchers; and U.S. government organizations. This section summarizes the workshop activities and the output information from the participants. The full details can be found in the comprehensive workshop report (National Institute of Standards and Technology, 2015a).

3.1. Goals
The workshop was planned and executed with three specific goals. They were to identify and prioritize the:

- Measurement science needs for improving PHM impacts within manufacturing processes;
- Measurement science barriers, challenges, and gaps that prevent the broad use of PHM technologies for manufacturing processes;
- R&D needed to address the priority measurement and standards challenges.

3.2. Plenary Talks and Panel Discussions
The workshop featured five plenary talks and three panel discussions (National Institute of Standards and Technology, 2015b). The plenary talks, presented by NIST personnel and external PHM experts, talked about the needs to evolve PHM technology within manufacturing along with existing PHM successes that several organizations have recently employed. Likewise, the talks highlighted specific challenges that still remain that, if addressed, can present tremendous benefit to the manufacturing community. These challenges included the development of common standards, interoperability among systems, deriving actionable intelligence from extensive data streams, and enabling machines to self-heal (i.e., impending faults or failures and automatically take corrective actions to remedy the problem).

The three panel sessions are discussed in the following sub-sections. Each panel was moderated, and included numerous speakers from diverse industry backgrounds, each with practical PHM experience. Some of the highlights from the question and answer sessions during each panel will be discussed herein. Full presentations given by both the plenary speakers and panelists can be found on the NIST web space (National Institute of Standards and Technology, 2015b).

3.2.1. Panel 1: PHM Capabilities, Best Practices, Challenges, and Needs
This panel focused on the current state of PHM for manufacturing. Panelists focused on PHM technologies and systems including existing capabilities, best practices, and challenges along with technological gaps and limitations. Some of the key highlights of the panel’s question and answer session include:

- Communication and interoperability at the system level – Diversity, varying ages, and non-standard software of numerous systems add complexity to PHM systems. Enhancing, simplifying, and standardizing communications among multiple systems is warranted to streamline PHM.
- Catalog of data sets for understanding failure – It is challenging to obtain sufficient training data to ascertain when equipment or processes will fail. It is rarely practical to let a machine or process fail solely to obtain a realistic data set. Given that the best data is often from real failures, data must be opportunistically captured when a true fault or failure occurs.
- Real-time aspects of PHM technologies – Manufacturers are seeing an increasing need for real-time PHM technologies. This is especially true for high value equipment or processes where any faults or failures can be detrimental to overall manufacturing operations. Not all organizations are ready for this shift; some are still lacking in basic (not real-time) PHM while others do not see the implementation of real-time PHM as being cost effective for their operations.

3.2.2. Panel 2: Performance Assessment – Monitoring and Measurement
This panel discussed the techniques for monitoring and measuring the performance of the PHM systems, themselves, along with identifying the metrics that evaluate how PHM technologies impact overall manufacturing performance. Highlights from this panel’s question and answer session include:

- Equipment monitoring and data collection by suppliers – It is challenging to implement PHM in one’s own organization and it can also be challenging to request PHM be integrated into an external supplier’s operations. Those suppliers that integrate PHM within their operations will likely gain a competitive advantage in that they will have more forewarning of faults and/or be more capable of handling unforeseen failures.
• Cost justification of PHM systems – When manufacturers buy manufacturing equipment that has a history of reliable operations, it is unlikely that they will also want to invest in PHM for this same equipment. The cost justification can be made in terms of maintaining or increasing quality and/or safety. Manufacturers will gain confidence in their equipment if PHM technology providers support any warranties that are tied to the equipment.

3.2.3. Panel 3: PHM and the Human Element

The third panel focused on the influence and understanding of human decision-making on PHM systems within manufacturing and the difficulties that present themselves when humans work with PHM. A few of the highlights of this panel’s question and answer session include:

• Need for increased knowledge of refurbished equipment – It is difficult to accurately assess a machine’s health after it has been repaired (following a fault or failure), refurbished, or undergone extensive maintenance. This lack of knowledge can also complicate understanding a system’s overall health when a constituent component has been extensively repaired or replaced. This situation presents an opportunity to develop inventory tracking in conjunction with PHM that could document individual health states and expected remaining useful life (RUL) of specific components.

• PHM is easier to implement at the onset of a machine’s/process’ life – It is more cost effective and easier to integrate PHM into equipment or a process during the design stage prior to the equipment or process being put into service. This ease of implementation includes making it easier to integrate sensors, technology, and programming for PHM. One disadvantage of integrating PHM at the onset is that it is likely that all of the faults and failures that could/will occur are not known at this initial timeframe; some faults and failures are still likely to occur that the PHM system would either not detect or inaccurately detect. PHM design and implementation is costly, so the specificity and extent of its capabilities should be measured against the projected savings with its usage.

3.3. Breakout Sessions

The workshop featured three separate breakout topics: PHM Manufacturing Process Techniques and Metrics, PHM Performance Assessment; and PHM Infrastructure – Hardware, Software, and Integration. Each breakout topic met four times (Sessions I, II, III, and IV) across the two-day event and held a specific focus:

• Breakout Session I: Goals and Desired Capabilities – For each topic area, the first session focused on capturing the specific PHM capabilities most wanted and needed. Each group identified goals in the near-term (1 to 2 years), mid-term (3 to 5 years), and long-term (5+ years) time horizons. Additionally, each group then categorized the capabilities in the different topic areas in terms of high, medium, and low priorities.

• Breakout Session II: Challenges and Barriers for Achieving the Capabilities – This breakout meeting for each topic focused on identifying the specific measurement and standards barriers, challenges, and gaps that hinder PHM development, implementation, and integration.

• Breakout Session III: Prioritization of Challenges – This breakout meeting identified R&D and standards priorities for each of the challenges and barriers mentioned in the prior session. This included organizing the challenges in terms of high, medium, and low priorities.

• Breakout Session IV: Pathways for a Measurement Science Roadmap – The final breakout meeting organized each topic’s participants in small groups to develop specific roadmaps with recommended approaches, next steps, and actionable plans. Each action plan was also broken out into near-term (1 to 2 years), mid-term (3 to 5 years), and long-term (5+ years) timeframes.

Each of the three breakout topics will be presented in the following subsections. Although the three breakout groups operated separately, some of their identified goals, capabilities, challenges, and priority roadmaps had similar themes. This was natural in that some of these similarities cut across multiple topic areas. Cross-cutting themes are highlighted in Section 3.4.

In the following sections, highlights will be presented for all three breakout topics with a focus on the output roadmaps. Certain roadmaps were selected from each breakout topic for discussion in this paper. The chosen roadmaps were deemed the most important to address immediately and/or were supported by a majority of the participants while being relevant to NIST’s mission.

3.3.1. Breakout Topic: PHM Manufacturing Process Techniques and Metrics

The successful implementation of PHM can have a significant influence on manufacturing operations by providing timely actionable intelligence. This intelligence can then be used to aid maintenance such that downtime is carefully coordinated with manufacturing operations for zero loss of productivity and quality. This breakout topic focused on addressing the specific PHM manufacturing
process capabilities that can enable this timely actionable intelligence along with the metrics necessary to collect and analyze in support of these capabilities. This group focused on PHM techniques and metrics that can ultimately enhance condition-monitoring, equipment and process reliability, safety, operator situational awareness, and overall equipment effectiveness. After the group identified their desired goals and capabilities, and the corresponding challenges and barriers, three priority roadmap topics were developed. Two of the roadmaps are presented below while the third (Enterprise-Wide PHM for Maintenance Planning) is not discussed due to space restrictions.

**Advanced Sensors for PHM in Smart Manufacturing**

The development of this specific roadmap was spurred by the lack of understanding of the full suite of capabilities of sensors, their interfaces and interoperability needs for PHM. This is critical to address because current PHM systems lack re-configurability, flexibility, scalability, and robustness partly due to the lack of knowledge with respect to sensors.

A sub-group within this breakout session focused on outlining a multi-stage method for sensor development. This approach begins by inventorying existing sensor data acquisition (DAQ) systems that are needed for PHM systems and defining the re-configurability requirements for common manufacturing processes. This effort would ultimately breed data communications and analytics standards to promote greater communication among multiple configurations and technologies. Mid-term activities would include the identification of gaps in sensor and DAQ capability and interoperability, and define scalability requirements for several manufacturing processes. Long-term activities feature the development of multi-purpose sensors/DAQ interfaces for use within manufacturing PHM systems; development of standards for data communication, data analysis, and prognostic algorithms; and the development of a taxonomy of PHM systems and capabilities. This would lead to the generation of a taxonomy library and a PHM-handling catalog of generated tools to promote flexible and reconfigurable PHM systems. The completion of this roadmap action plan is envisioned to have high impact within the manufacturing community since it’s very likely to improve reliability/reduce failures of equipment and processes, improve maintenance scheduling, and speed process re-configurability.

**PHM Data Format and Architecture**

The generation of this roadmap was motivated by the desire to solve the lack of interoperability of sensors/data formats and types of communication while preserving the meaning of the data and the semantics. The overall approach of this roadmap is to create protocols for PHM covering formats, storage, organization, semantics, and other key components. Standards would be created to support the protocols along with data interfaces and integration. These protocols and overall architecture would enable the generation of a database of PHM data and information that the community could draw upon.

The near-term activities of this roadmap include determining protocol data types and structure. Guidelines must also be developed for data format, storage and preservation, organization, and semantic requirements. Moving forward, the mid-term activities would focus on standards development for semantic PHM data and the creation of tools to capture and organize the data; and then, extract and visualize the information in a meaningful way. The long-term tasking would focus on the creation, organization, and management of PHM data repositories. This would yield an expansive database that could be used by manufacturers, technology integrators, and technology developers who work with PHM systems. The advancement of this roadmap would have the highest impacts in speeding process re-configurability and improving maintenance scheduling.

### 3.3.2. Breakout Topic: PHM Performance Assessment

Before any new technology can realize its full potential, it is critical to verify and validate its performance. PHM is no exception, and care must be taken to ensure that any PHM technology’s performance and impact is accurately assessed. This breakout topic focused on assessing the performance of PHM along with the necessary technologies, measurement techniques, data, and performance metrics required for such verification and validation.

Breakout participants identified goals in the areas of identifying specific PHM performance characteristics and metrics, and equipment and technologies necessary to monitor a PHM system (or component). In addition, the participants also noted the long-term goal of incorporating the design and validation of a PHM system into the overall equipment/process life cycle. Once the participants identified the subsequent capabilities and existing challenges, six priority roadmap topics were developed. Three of the roadmaps are presented in detail while the remaining three (Cost Model for PHM Performance, Taxonomy of Applications, and Determination of PHM Data and Information Needs) are not presented due to space restrictions.

**Overarching Architecture Framework for PHM with Standards and Key Performance Indicators (KPIs)**

This roadmap was motivated by the participants’ acknowledgement that a PHM framework within multiple industries is either unclear or lacking in standards. This absence of standards promotes inconsistencies in PHM verification and validation.
The ultimate goal of this roadmap is to define a standard PHM architecture and create methods that will enable asset traceability and historical record keeping on PHM performance. To realize this goal, the participants identified the near-term action of benchmarking the current state of machine monitoring (starting with specific industries) and the mid-term tasks of cataloging the KPIs and mapping—out the typical diagnostic and prognostic trends (from the target industries). The vision is that this effort would produce a published catalog that gains some industry acceptance (100% acceptance is too ambitious at this time, yet an initial target was not determined). Likewise, international standards would be developed that are broad enough to cover a range of PHM implementations across multiple industries. These standards would have to be specific enough to guide manufacturers through the process of developing and implementing a means of verifying and validating their PHM capabilities. If successful, this roadmap is expected to have significant impact in improving equipment/process reliability, reducing costs, increasing industry’s competitiveness, and enhancing maintenance scheduling.

Identification of PHM Performance Metrics
Participants produced this roadmap citing a lack of performance metrics capable of characterizing the value of prognostics to equipment or processes prior to failure. This coincides with limited information on key metrics for manufacturing equipment and/or processes at component and system levels. The overall approach proposed is to evaluate existing metrics to determine what metrics can be captured from equipment/processes prior to a fault or failure that sufficiently evaluate the performance of the PHM system in question. This assessment will aid in developing new performance metrics.

The near-term plans of this roadmap feature three activities: 1) survey current metrics that characterize the performance of a PHM system itself and the PHM’s effectiveness when applied to a machine/process, 2) identify the necessary metrics that can apply diagnostics and prognostics to manufacturing equipment/process and integrate with controls/operations and maintenance planning, and 3) determine the gaps present between existing and desired metrics. Mid-term actions include 1) developing the missing metrics, 2) evaluating the metrics across a range of equipment, processes, systems, and PHM algorithms, and 3) studying how performance metrics can be integrated with controls, operations, and maintenance planning systems. Long-term activities conclude with integrating the identified performance metrics with the PHM architecture (described in the prior section) so the metrics can be implemented and demonstrating the applied metrics (in concert with the architecture) at selected pilot plants. The achievement of implementing the metrics and framework in a plant is envisioned to be a stepping-stone to applying the metrics and framework to additional manufacturing facilities.

The expected impact of completing this roadmap action plan includes better decisions being made based upon available PHM results and performance metrics; improved quality and productivity of equipment and processes; and greater availability of actionable information.

Failure Data for Prognostics and Diagnostics
The final roadmap is motivated by the lack of sufficient, available failure data for diagnostics and prognostics. Currently, measurement and data collection methods and appropriate test beds are limited in their availability and capability. For those methods and test beds that do exist, there is a lack of consistency in the data formats for which data is captured and organized. The participants who developed this roadmap proposed the approach of developing methods and services to generate diagnostic and prognostic data sets for public use including verification and validation. This would be supported by the development of specific test beds that would enable both the production of data and the necessary verification and validation.

The roadmap action plan begins with three near-term activities: 1) development of a common database, 2) creation of test beds to assess feasibility, and 3) establishment of a consortium (including NIST and university partners) to examine PHM for specific systems in the form of test bed(s). Mid-term activities include qualifying the data within the common database and further development of the scaled-down test beds. Long-term activities feature the implementation and testing of the common database, standardizing the scaled-down test beds, and performing simulation modeling of processes. Upon the completion of these tasks, the realized capabilities should be the active use of a common database and the adoption of PHM failure data standards. The realized impact of these capabilities is expected to include a significant reduction in cost (this method promotes cost sharing across the industry) and improved access to failure data to support verification and validation of PHM methods.

3.3.3. Breakout Topic: PHM Infrastructure – Hardware, Software, and Integration
Successful PHM methods and technologies require a robust infrastructure including key building blocks such as hardware, software, models, and simulations along with the integration of these elements. Technology has greatly advanced in the last decade (including enhanced capabilities in wireless connectivity, mobile devices, computing power, sensing capability, and human machine interfaces), and the PHM infrastructure has become increasingly complex. The participants in this breakout topic discussed a variety of infrastructure needs from the perspective of enabling and
augmenting PHM within smart manufacturing environments.

Breakout participants identified near-term, mid-term, and long-term infrastructure goals in the areas of PHM design, hardware, software, security, maintenance, and data management. This prompted the participants to identify the capabilities and their corresponding priorities. Next, the participants identified the challenges and barriers to achieving these capabilities and prioritized them accordingly. These efforts led to the development of four roadmap action plans. Two of the roadmaps are presented in detail while the remaining two (PHM as an Equipment Design Feature and Embedded Sensors for PHM of Emerging Manufacturing Technologies) are not discussed due to space restrictions. Those roadmaps not discussed in this paper can be found in detail in the full workshop report (National Institute of Standards and Technology, 2015a).

**Open-Source Community for PHM**

The first roadmap action plan to be presented from this breakout topic is motivated by the fact that it is often costly and overly complex to implement PHM on new equipment. The proposed approach charts the path of developing an open source architecture that will reduce the cost and complexity of PHM design and implementation. The approach features a collection of data and identification of relevant PHM systems and devices.

The near-term activities of this roadmap include: 1) the development of open drivers and adapters enabling PHM through the integration of sensors, equipment, controllers, interfaces, etc. 2) the expansion of the data collection infrastructure to accommodate an open source format, and 3) the development of security, compression, fault tolerance, and schema for the open architecture. Mid-term tasks include: 1) identification of systems and devices to be compatible with the framework, 2) development of frameworks and toolkits to enable users to interface with equipment, and 3) expansion of drivers and adapters. Finally, the long-term task is a continuation of the prior tasks – promote continuous development and improvement (similar to what is done in the Linux community). The goal is to get a majority (ideally, all) of industry (ranging from small to large enterprises) using and contributing to the open architecture.

If this roadmap action plan is successfully completed, numerous impacts could be realized. The most significant impacts that could be realized include reduced individual cost to develop and implement PHM; accelerated pace of innovation since more time could be devoted to developing PHM algorithms as opposed to developing the architecture (since it would already be in place); and enhanced industrial competitiveness since the increased presence of PHM would reduce maintenance costs and enhance versatility.

**PHM Infrastructure to Deliver Relevant Timely Information**

The final roadmap action plan to be presented is similar to the roadmap highlighted in the last section, yet is still unique in scope and objectives. The participants developed this plan to overcome the current inability to make good decisions based upon the available data where PHM users are currently making decisions either with the wrong information, with insufficient detail, and/or at the wrong levels. The proposed approach focuses on developing a traffic light approach (e.g., green, yellow, red) to classifying the value of the data for decision-making.

This roadmap features an extensive action plan with eight near-term and six mid-term tasks identified. Some of the near-term tasks include the development of tools to construct cyber-models of replacement parts/components to better predict RUL or mean time to failure, determination of required data to model diagnostics and prognostics, and assess requirements to determine the necessary information needed at each operational level within a manufacturing environment. Several of the mid-term tasks include the development of a cloud-based data repository and analytic engine to further enhance decision-making and technology generation to enable adaptable alarms based upon equipment/process condition. The participants identified a single long-term goal – develop advanced usage-based models to augment PHM decision-making. Increased and enhanced decision-making is the ultimate desired capability where the participants envision 80% improvement (over existing baselines) after five years of effort on this roadmap.

The significant impacts that could potentially be realized if this action plan is completed are the generation and availability of better data for fault and failure prevention, appropriate data and better decision-making are fused to make timely decisions regarding maintenance scheduling.

**3.4. Cross-Cutting Themes**

Over the entire course of the workshop, numerous themes emerged, both within the individual breakout topics and across the rest of the workshop program (plenary talks and panel discussions). Six specific themes were identified; three are presented in the following sub-sections while the other three (Workforce and Training, Human Factors, and Business Case for PHM) are not presented.

**3.4.1. Data Collection and Extraction of Information**

The challenges of collecting, extracting, and analyzing appropriate and meaningful data were well documented throughout the workshop. Data is a critical piece of designing, verifying, validating, and implementing effective PHM technologies into a manufacturing process or piece of equipment. These challenges stem from a lack of sensors capable of capturing the right data at the appropriate
frequency, accuracy, and resolution; and a lack of rigorous measurement methods to enable efficient and effective data collection methods suited for PHM. Additionally, inconsistent or insufficient data standards are making it difficult to broadly apply PHM across a range of manufacturing equipment and processes; standardization of data formats and taxonomies would play a significant role in overcoming this challenge. Another data challenge is generating accurate PHM data, for the purposes of PHM design, verification, and validation without damaging equipment or decreasing productivity.

3.4.2. Models, Simulation, and Visualization

Validated models to support PHM are limited in availability and capability. The entire scope of modeling, simulation, and visualization (MSV) is also encumbered by the diversity of manufacturing equipment and processes, lack of integration with legacy systems, and data availability (which is critical for effective MSV). A benefit of having accurate and relevant models is that they can help highlight the value of PHM prior to a system being put into practice. This would help generate further organizational support for PHM, and it sets initial expectations of the predicted performance.

3.4.3. Design Considerations

The last cross-cutting theme to be highlighted is the notion that PHM be considered as a design feature that is factored in to the design process of any new piece of manufacturing equipment or process. Most original equipment manufacturers (OEMs) do not consider PHM in their design process; any PHM that is factored typically include limited forms of condition-monitoring and diagnostics. Likewise, most technology integrators will not add PHM into their process design unless their customer specifically requests PHM and is willing to pay the additional costs for it. It is much more challenging to integrate effective PHM into a system/process after that system/process is in service on a factory floor.

4. NIST'S RESEARCH DIRECTION

The workshop provided valuable insight that is envisioned to bring tremendous benefit to the PHM community. Likewise, NIST is carefully reviewing the workshop findings to update its project’s research direction to further align it with industry’s needs and high priorities. The PHM@SMS project team is currently focused on four specific efforts that are all factoring in the workshop findings.

4.1. Machine Tool Linear Axes Diagnostics

This effort is focused on developing a sensor-based method to quickly estimate the degradation of linear axes, and is supported by the development of a linear axes test bed. This method leverages data collected from a NIST-developed sensor suite to detect translational and angular changes due to axis degradation. Real-time data is collected to enable diagnostics and prognostics of linear axes for optimization of maintenance scheduling and part quality. This method to estimate the degradation of linear axes will also enable verification and validation of other (built-in or otherwise) PHM techniques that aim to characterize translation and angular errors and degradation. Likewise, this method will produce reference data sets that can be used by PHM developers as test data so they do not have to risk damaging their own equipment or impacting their productivity. This method will ultimately lead to standards to measure and predict linear axes degradation. The linear axes test bed will yield its first data sets for analysis in Summer 2015.

4.2. Manufacturing Process and Equipment Monitoring

Driven by the need to identify high-value data sources and the most appropriate times to collect data, this manufacturing process and equipment monitoring effort focuses on enabling the seamless and effective use of data to generate timely and actionable intelligence on equipment/process health. This effort is supported by the development of a systems-level test bed of networked machine tools and sensors in an active manufacturing facility. Accordingly, a significant part of this research is the design of a reference implementation that manufacturers may use to collect data safely and efficiently without disruption to operations. Likewise, this effort will also yield a reference dataset of fabrication and inspection data that may be used to identify useful links for improved process monitoring, diagnostic, and prognostic capabilities. This test bed will produce initial results in Fall 2015.

4.3. Systems-Level Diagnostics and Prognostics

Many complex processes and systems-of-systems are lacking in higher-level capabilities to accurately and efficiently forecast faults and failures. This research effort addresses this challenge by developing protocols to communicate data, information and metrics across the component, sub-system, and system levels for diagnostics and prognostics in manufacturing. These protocols will enable the prediction of system-level impacts of events occurring at a single component or sub-system. Moreover, the protocols will enable and enhance process management and control approaches to effectively respond to these events. A hierarchical methodology is being developed with external partners, and will be applied to the two aforementioned test beds in 2016.

4.4. PHM for Robotics

Robotics are increasing in their implementation and complexity of integration within manufacturing operations. PHM considerations of a robotic system extend beyond just
the physical arm, gantry, mobile base, etc. nearly every robotic system features some type of end-effector, sensors, safety system(s), supporting/surrounding automation, controller, etc. Robotic systems, especially in smart manufacturing environments, are often marked by complex interactions among these elements. For example, a fault or failure that presents itself as unexpected or inappropriate behavior of the robot arm is likely to have resulted not from a mechanical failure of the arm, but rather from a failure elsewhere in the system (e.g., sensor failure, or a controller fault). This research effort is actively developing a PHM-focused robotics test bed that features a scaled-down industrial robotic arm system to develop test methods, metrics, assessment protocols, and reference data sets that can evaluate robot system degradation techniques including how such degradation impacts key elements of the robot system (e.g., safety). This test bed is expected to be operational and produce its first data sets in Summer 2016.

5. CONCLUSION

The two-day workshop brought together many PHM experts who shared their best practices, challenges, and visions with respect to PHM in smart manufacturing (National Institute of Standards and Technology, 2015a). Their extensive feedback is well-documented in the roadmap action plans, and will guide the community in devising and updating their research directions, accordingly. As a member of the community, NIST is examining the workshop findings to best determine where its research efforts can have substantial impact in addressing PHM measurement science challenges.

ACKNOWLEDGEMENT

The authors would like to thank the workshop speakers, panelists, and participants for their focused and diligent efforts during the two-day event. Their willingness to share their PHM experiences and vision made this report possible.

DISCLAIMER

The views and opinions expressed herein do not necessarily state or reflect those of NIST. Certain commercial entities, equipment, or materials may be identified in this document in order to illustrate a point or concept. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

REFERENCES


National Institute of Standards and Technology (2015a). Measurement Science Roadmap for Prognostics and Health Management for Smart Manufacturing

PCAST (2012). Report to the President: Capturing Domestic Competitive Advantage in Advanced Manufacturing. Executive Office of the President - President's Council of Advisors on Science and Technology.


**Biographies**

**Dr. Brian A. Weiss** has a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the Smart Manufacturing Operations Planning and Control (SMOPAC) program and the Project Leader of the Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) project within the Engineering Laboratory (EL) at NIST. Prior to his leadership roles in the SMOPAC program and the PHM4SMS project, he spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles; tactical applications operating on Android™ devices; advanced soldier sensor technologies; urban search and rescue robots; and bomb disposal robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 & 2008), and the Jacob Rabinow Applied Research Award (2006).

**Dr. Gregory W. Vogl** is a Mechanical Engineer at NIST. He received his B.S. (2000), M.S. (2003), and Ph.D. (2006) degrees in Engineering Mechanics from Virginia Tech, Virginia, USA. Currently, Greg is a member of the Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) project, which seeks to develop a methodology, protocols, and reference datasets to enable robust real-time diagnostics and prognostics for smart manufacturing systems. Previously, he designed, fabricated, and experimented on microelectromechanical systems as a National Research Council Postdoctoral Researcher at NIST. He then joined the Production Systems Group, in which he worked on machine tool metrology and standards development. His interests include machine tool spindle health, diagnostic and prognostic methods, nonlinear dynamics, engineering mechanics, and metrology.

**Dr. Moneer Helu** is a mechanical engineer in the Engineering Laboratory at NIST where his research currently focuses on performance assurance and prognostics and diagnostics for smart manufacturing systems. Prior to joining NIST in 2014, he was the Associate Director of the Laboratory for Manufacturing and Sustainability and a Lecturer in the Department of Mechanical Engineering at UC Berkeley. He received his Ph.D. from UC Berkeley and has been recognized by the Society of Manufacturing Engineers in the 2014 list of the "30 Under 30: Future Leaders of Manufacturing."
Dr. Guixiu Qiao is a mechanical engineer at the NIST. She is a member of the PHM4SMS project, focusing on PHM for robotic systems. She has 12+ years’ experience in high precision industry, expertise in advanced sensors, automation, manufacturing, and information integration; with a strong background in algorithms and optimization. She is a 2011 R&D 100 award winner for “3D volumetric accuracy error compensation for large machine tool” recognizing the most technologically-significant products introduced into the marketplace; 2009 recipient of the Defense Manufacturing Excellence award for the contribution of pioneering efforts in volumetric accuracy for large machine tools; and a 2003 recipient of Boeing’s Exceptional Performance award. Dr. Qiao holds her Ph.D. in robotics from the Robot Research Institute, Harbin Institute of Technology, China.

Joan Pellegrino is Vice President of Energetics’ Science and Technology Division. She has more than 27 years of experience analyzing advanced industrial technologies, conducting cost/benefit analysis, assessing broad energy and environmental trends/statistics, and evaluating bioenergy systems. She has also worked with NIST for over 15 years assessing the strategic measurement science and metrology needs of external stakeholders across the Nation. Her broad experience includes strategic planning, technology roadmapping, market intelligence and business planning, technology transfer, and stakeholder engagement/community building. She has worked with a diversity of government agencies, academic institutions, national laboratories, state/local governments, and commercial entities and provides high-level technical and management advising. Ms. Pellegrino is a professional facilitator with experience designing effective stakeholder engagements; she has contributed to over 80 strategic plans and technology roadmaps on a variety of technical topics. She holds a M.S. in Chemical Engineering from Auburn University and a B.S. in Chemical Engineering from the University of Tennessee, Knoxville.

Mauricio Justiniano is a Senior Engineer at Energetics with over 15 years of experience in energy and environmental consulting. He specializes in project management of technology deployment projects, such as smart grid technologies, and benefits analysis of emerging technologies. Mauricio is an experienced strategic planning consultant helping diverse clients develop visionary strategies, prioritize investments, and devise solutions to complex problems. Prior to joining Energetics, he was a consultant for the World Bank Group’s International Finance Corporation where he researched technology investment opportunities and the economic environment of potential loan-recipient countries and industrial sectors. Mauricio is a Certified Project Management Professional (PMP), and has a MS in Finance from Johns Hopkins University and a BS in Mechanical Engineering from the University of Maryland – College Park.

Anand Raghunathan is a senior materials engineer at Energetics Incorporated. As a consultant to NIST, U.S. Department of Energy (DOE), and private-sector clients, he works on technology, strategy, and program management activities with various groups that include creating long-range, strategic roadmaps. While at Booz Allen Hamilton and the Schafer Corporation, he worked with program managers at the Defense Sciences, Microsystems Technology, and Strategic Technology Offices within the U.S. Department of Defense’s Defense Advanced Research Projects Agency to identify and advance relevant high-risk/high-reward technologies to protect national interests. As an engineer at 3M and Chorum Technologies, Anand worked on materials and telecommunications technologies across the phases of the product development cycle—conducting research and development, designing specifications with global customers, optimizing manufacturing processes, and securing sales. Anand has BS and MS degrees in materials science and engineering from the Massachusetts Institute of Technology.
A New Scheme for Monitoring and Diagnosis of Multistage Manufacturing Processes Using Product Quality Measurements

Hossein Davari Ardakani¹, Brian Phillips¹, Behrad Bagheri¹, and Jay Lee¹

¹NSF I/UCRC for Intelligent Maintenance Systems (IMS), University of Cincinnati, Cincinnati, OH 45221, USA

davarihn@mail.uc.edu
phillib7@mail.uc.edu
bagherbd@mail.uc.edu
jay.lee@uc.edu

ABSTRACT

The development of robust monitoring systems for assuring the consistency and stability of multistage manufacturing processes necessitates the use of add-on sensors and advanced data collection, storage, and analysis platforms to deal with the high-dimensional data collected from machines and products in multiple stages. In many cases, such an approach may not be feasible due to high implementation costs and the challenges of obtaining the process parameters and analyzing them effectively. This paper proposes an alternative approach for health monitoring and diagnosis of multistage manufacturing processes based on product quality measurements in a sensor-less environment. In the presented work, the available data consists of product quality parameters measured from multiple product types along with the manufacturing route associated with each product. A Gamma distribution is fit to the data for each parameter within a moving time window. Using the distribution fits, a metric is developed to represent the performance of each machine in a stage compared to its peers producing the same product. This metric is then aggregated across all the products produced by the machine to generate the final metric reflecting the overall performance of the machine. This performance metric is first calculated for the machines in the last stage. After flagging the underperforming machines in the last stage, the samples from those machines are removed from the data set and the remaining samples are used to calculate the similar metric for the prior stage. The results demonstrate the effectiveness of such approach for monitoring and diagnosis of multistage manufacturing processes when the data is not available from within the process.

1. INTRODUCTION

With the rapid advancement of science and technology, the manufacturing processes are becoming more and more complicated and so sustaining their performance and reliability becomes increasingly important and a pivotal factor in global market competition. This trend is driving the manufacturers to deploy more advanced and reliable process monitoring systems to improve the quality, productivity, and reliability of their processes. With the fast development of information and sensing technologies, large amounts of data are being collected from industrial processes. The availability of various resources for collecting data has led to the advancement of data-driven process control and monitoring technologies which has had a growing impact on the final quality of the products (Shu & Tsung, 2000). A large amount of research has so far been dedicated to the development of multistage process monitoring and diagnosis methods based on measurements in different stages of the processes (Asadzadeh & Aghaie, 2012; Shu & Tsung, 2000; Tsung, Li, & Jin, 2006; Wolbrecht, D’ambrosio, Paasch, & Kirby, 2000; Zhou, Huang, & Shi, 2003; Zhou, Ding, Chen, & Shi, 2003). The developed methods in this field can be categorized in two types: 1) methods based on variation propagation modeling techniques (Ceglarek, Shi, & Wu, 1994; Ceglarek & Shi, 1996; Hu & Wu, 1992; Liu & Hu, 1997); and 2) methods based on Statistical Process Control (SPC) techniques (Lucas & Saccucci, 1990; Montgomery, 2007; Neubauer, 1997; Page, 1954; Rato & Reis, ; Reynolds, Amin, Arnold, & Nachlas, 1988; Reynolds, Amin, & Arnold, 1990; Roberts, 1959).

One of the well-established variation propagation-based modeling techniques is Stream of Variation (SoV), introduced by J. Hu in (7) to identify the sources of variation in automobile body assembly. By developing the
inherent relationships between errors from various sources in a sheet metal assembly process, further improvements in SoV-based methods helped to solve two fundamental issues: 1) lack of analytical models for variation propagation, and 2) lack of a systematic methodology for analysis of variation propagation and its minimization (Shi, 2010). Despite the effectiveness of this group of approaches in modeling and diagnosis of multistage manufacturing processes, they require product quality measurements from each stage of the process in order to build the variation propagation model of the process.

The second group of methods mentioned previously in this section is based on SPC techniques. The objective of SPC is to monitor the process over time and determine whether the process is in control or not. SPC charts were first developed to monitor the key product variables in a univariate way (Lucas & Saccucci, 1990). The most commonly used traditional SPC charts include Shewhart, Cumulative Sum (CUSUM), and Exponentially Weighted Moving Average (EWMA) control charts (Reynolds, Amin, Arnold & Nachlas, 1988). In this group of methods, process data samples from either fixed or variable time intervals are taken. Depending on the selected method, a statistic is computed and plotted from the samples in each interval and thresholds are set to define the acceptable range of the variable. Similar to the SoV-based methods, SPC-based methods also require measurements in different stages of the process. If only the final product quality measurements are available, SPC can still be used to detect variations in the process. However, it is unable to provide further information regarding the source of the variation in terms of process stage and the machine or equipment causing the variation.

Despite the availability of process data in many manufacturing sites, there are existing challenges that the manufacturers face in measuring the necessary parameters from the processes. PHM implementation should have minimal adverse impacts on the performance of the monitored systems and at the same time, it should be low cost as well. Due to the complexity of industrial machinery and processes, instrumenting each machine with various sensors requires investment both in sensor and data acquisition hardware, and the infrastructure for data handling, storage and analysis. Besides considering the cost, the factors that should be considered for sensor selection include the parameters that need to be measured, performance needs, electrical and physical attributes of the system, and sensor reliability (Cheng, Azarian, & Pecht, 2010). Due to the existence of such barriers, data-driven manufacturing process monitoring often faces major challenges in dealing with limited resources and in many cases insufficient data to extract information from. As a summary, the common barriers for implementing sensor-rich manufacturing process monitoring systems are: 1) cost considerations including the initial and total lifecycle cost of sensors; 2) lack of access to critical locations; 3) sensor reliability; and 4) non-optimal selection of sensors and their location. Despite the significant impact of sensor technologies in recent advancements in PHM, there are still situations in which the implementation of sensors does not seem applicable due to the barriers mentioned above. Sensor cost is the most common issue that needs to be dealt with. Instrumenting numerous complex machines in a factory can impose a significant cost and in many cases is not deemed applicable. Such cost can both be related to the purchase of sensors and additional data acquisition and storage systems, or sensor and hardware maintenance and replacements. Determining and accessing the critical locations within the machines and processes is also a common challenge. An alternative sensor-less approach for manufacturing process monitoring is proposed in this paper. Although it may not be an option or the best option in many cases, it has proved to be a viable option in which sensory data from each machine is not available but the data regarding product quality and their manufacturing routes are available.

2. TECHNICAL APPROACH

2.1. General Concept

The process that this methodology is designed for may consist of several stages, with each stage having several machines. This process may produce multiple types of products. The data gathered from this process should include the quality measurements for each or sampled products, along with the manufacturing route associated with that product. Moreover, it is assumed that the products distribute randomly from machines in stage N to machines in stage N+1.

The data for this study includes a set of quality parameters measured for each product in a manufacturing plant. The data also contains the type of the product, its unique barcode, and its manufacturing route through the process. A gamma distribution is fit to the data for each parameter within a moving time window. Using the distribution fits, a metric is developed to represent the performance of each machine in a stage compared to its peers producing the same product. This metric is then aggregated across all the products produced by the machine to generate the final metric reflecting its overall performance. This performance metric is first calculated for the machines in the last stage. After flagging the underperforming machines in the last stage, the samples from under-performing machines in the last stage are removed from the data set and the remaining samples are used to calculate the similar metric for the prior stage. A flowchart of the proposed method is provided in Figure 1.
2.2. Machine Performance Metric Development

For developing a comparison-based metric for machines, a window with the length of several hours was applied to the data. The length of the data was selected based on the expert knowledge and maintenance concerns. At each step of the window, the distribution of product quality parameters where considered as the source for calculating the metric. The purpose of developing such metric was to extract a standard health value, ranging from zero to one, which represents the performance of a machine relative to its peers. The first step for building such metric was to fit an appropriate distribution function to the data during each window.

As first step, the distribution of parameters for each product time and their possible shift over time was studied. The original values of the product quality parameters were positive real numbers, but due to proprietary concerns, the parameters were normalized between zero and one. The observed shape of the parameter distribution was close to normal (Gaussian), except when the parameter approached zero. This sharp drop-off in the distribution varied with the parameter, but it was never a hard cutoff at zero—it always sloped down to zero magnitude when the domain is zero. The negative slope (after the peak) had the appearance of an exponential distribution. Similar distribution shapes were observed for all the variables considered in this study. Thus, only one variable was picked and provided as an example. An example of the histogram of one of the parameters is shown in Figure 2. Initially, the option of exponential distribution was eliminated since the observed distributions had a peak at a finite positive value, whereas exponential distributions immediately decay from zero. The option of Gaussian distribution was also eliminated since its domain exists over the entire set of real numbers \((-∞, +∞)\), and there was no negative-side “tail” on the distribution. Chi-squared distribution was considered, but it was ultimately eliminated for the more general gamma distribution, since the gamma distribution includes the scale parameter and thus was better able to fit the distribution of the parameter, whereas the chi-squared distribution “spreads out” as the shape parameter increases.

![Flowchart of the proposed method for multistage manufacturing process monitoring and diagnosis.](image)
For a given set of a fairly large number of machines, where “large” is subject to discretion based on failure rate, it can be assumed that the majority of the machines will be functioning properly. The term “majority” indicates a number of machines among the total machines that is enough for creating a steady baseline performance. Thus, it is unlikely that a significant number of machines from this “large” set of machines will be faulty at any time, assuming repairs are regularly made on the faulty machines. For the purposes of this paper, “faulty” means any abnormal condition, whether it is a loss of calibration, a mechanical failure, a controller failure, or any other event which causes the machine’s output to deviate from the correct output. Accordingly, an average of the parameters of all the machines should approximate a “good” condition. For the purposes of this paper, “good” means the state at or nearly at the point at which the machine yields the correct output. When comparing one machine against the average of all the other machines, a significant difference in distribution should indicate that the machine is in an abnormal condition.

When analyzing the data, it quickly became apparent that a method was needed for determining the difference in the distribution of a parameter for a machine versus the distribution of the same parameter for peers. Initially, the shared area under both probability density function (PDF) curves was considered as a percentage of the total area under the “good” average PDF curve. This value became the Confidence Value (CV) and was used as the health metric. Since the total area under a PDF curve is simply one, the CV could obtain a maximum value of one when the selected machine’s parameter distribution function lay directly atop the averaged parameter distribution function. A minimum value of the infinitesimal approaching zero could be obtained when the means of the averaged and selected distributions are very far apart. Ergo, the CV was part of the set of (0, 1], where higher values indicated a more healthy condition and lower values indicated the reverse. The equation for calculating the CV is provided in Eq. (1):

$$CV = \int_0^{+\infty} \min(PDF_{selected}, PDF_{avg}) dP \quad (1)$$

where $P$ is the product quality parameter. The issue with this method was when the selected machine’s parameter distribution had a tighter spread than the averaged machine. While the selected machine’s data still fell within the accepted, averaged bounds, the area under both curves was minimal since a large portion of the area in the averaged parameter is in the positive and negative “tails” of the distribution.

The majority of the values of the given parameter will fall within one standard deviation bounds of the machine. To overcome the aforementioned issue, comparing the PDF’s of the distributions only within one standard deviation bounds was considered. The equation was chosen to just be the area under the selected machine’s PDF over the one standard deviation limits of the PDF of the averaged machines, as shown in Eq. (2):

$$CV = \int_{\mu_{avg}-\sigma_{avg}}^{\mu_{avg}+\sigma_{avg}} PDF_{selected} dP \quad (2)$$

where $P$ is the product quality parameter. Because either the positive or the negative “tails” of the selected machine’s PDF will always pass through this domain, the lower limit of the function is some positive infinitesimal. The upper limit, where the area under the selected machine’s PDF is almost wholly within the limits of the integral, is some infinitesimal less than one. Ergo, the CV is some positive real number on the set of (0, 1). This solved the problem of when the PDF of the parameter of the selected machine is much taller and narrower (sharper) than the PDF of the averaged function.

However, it added a new potential for error. When the PDF’s were almost the same, the CV would approach 0.7 rather than one. This happened because roughly 70% of the area is within the one-sigma limits. To combat this error, the confidence value equation was changed such that the CV was equal to the percentage of the area under the one-sigma limits of the selected machine’s PDF that intersected the one-sigma limits of the averaged PDF (Eq. (3)):

$$CV = \frac{\int_{\mu_{avg}-\sigma_{avg}}^{\mu_{avg}+\sigma_{avg}} PDF_{selected} dP}{\int_{\mu_{selected}-\sigma_{selected}}^{\mu_{selected}+\sigma_{selected}} PDF_{selected} dP} \quad (3)$$

This confidence value is bounded on the range of (0, 1]. For healthy machines that fit the average healthy value, the CVs would be around 0.95 to 1. Faulty CVs are chosen based on a threshold CV that varies dependent upon the application of
While this technique worked, the performance cost of calculating and fitting the gamma distributions was higher than might be necessary. Additionally, having a software statistics package capable of fitting gamma distributions might not always be the case. Thus, as an alternative to gamma distribution, the authors further used normal distribution as an alternative with less computational requirements and slightly less accuracy. Accordingly, the gamma distributions were approximated as normal distributions. For any cases where the shape parameter of the gamma distribution (commonly represented by k) is large enough (normally 5 or higher), the normal distribution becomes a reasonable approximation of the gamma distribution for PDF values not near zero. Since the given technique uses at the farthest from the mean the one-sigma bounds which are still part of the “hump” of the PDF curve, this error does not normally come into play.

For rough estimations, a different equation for calculating the CV can be used. In this method, the standard deviation is ignored and only the mean values of the selected machine’s parameter and the averaged machines’ parameter are compared, as shown in Eq. (4):

$$CV = 1 - \left| 1 - \frac{\mu_{\text{sel}}}{\mu_{\text{avg}}} \right|$$  \hspace{1cm} (4)

Results seemed promising, as shown in Figure 4. The gamma distribution fit is plotted in blue, the normal distribution fit is plotted in red, and the mean comparison is plotted in green.

As can be seen in the Figure 4, the machine was run for a while, and then the CVs began dropping rapidly. Then production stopped for maintenance. When production was restarted following the maintenance, the CVs were back in the normal range. The normal approximation to the gamma distribution worked well, and the mean comparison worked as a rough estimate of the gamma distribution method.

### 2.3. Process Performance Monitoring

Besides calculating the health metric for each machine, the framework suggested in this paper also includes a methodology through which multiple stages of the process could be monitored and the machines adversely impacting the quality of the final products could be identified. The suggested approach for identifying faulty machines along the manufacturing rout is constructed based on a general assumption. It is assumed that products are being randomly distributed from each stage to the next. Having this assumption in place, the suggested approach analyzes the manufacturing process in reverse order. Therefore, the first step is to calculate health metrics for a specific type of product at the last stage of the production. After identifying the machines and the time frames in which the machine(s) affected the product quality, these samples are filtered out from the data and the remainder of the data is used to assess...
the machines performance in the previous stage. Removing these samples help to filter out any effect by current stage machines on the product quality once moving back to the previous stage. Based on the random distribution assumption, this sample removal process will not mask any faulty situation coming from previous stages as the fault signature would be present in the products distributed amongst all the machines at the current stage. Once the metric based on the remainder of the data is calculated, the same procedure is performed for previous stages until it reaches the first stage. The suggested approach for an example of a three-stage manufacturing process is illustrated in Figure 5.

3. RESULTS AND DISCUSSION

In section 2.2, the general approach for calculating a comparison-based machine performance metric based on product quality parameters was discussed. Subsequently, in section 2.3, an approach for extending the machine performance metrics from covering one stage of the process to covering all the stages of the process was introduced. In this section, an example is provided from a small subset of data to demonstrate the performance of the suggested process monitoring approach. The selected subset includes data acquired from the manufacturing process during three days. Over ten quality parameters are measured from each product manufactured during the selected time period. Although for this particular data set all the products are undergone quality check but the suggested approach is applicable to the situations in which the quality check is being done by random sampling. The data is limited to only one type of product at two stages of production. At stage one, there is only one machine, and in stage two, there are 12 machines. Having only one machine at first stage does not allow the implementation of a comparison-based approach. Hence, in this situation, the historical data that one machine is used to create a baseline. This baseline is later used to compare the current performance of the machine against it.

The data was analyzed by calculating comparison-based metric for each of the product quality parameters. Using all the metrics calculated based on these parameters, one final metric needs to be extracted as the health indicator of the machines. Simply averaging all the calculated metrics may not be effective since there are cases in which among all the parameters, only one or two contain deviation from normal. In order to address this issue, during each time window, the minimum value among the metrics is considered as the final metric. Although more complex statistical features might convey additional information about the machine condition or severity of faults, but the minimum value represent robust performance in identifying faulty machines. This can be the direction of future works on this method to improve the suggested approach.

The purpose of this example is to show how the suggested method can determine the root cause of the problem and isolate it within a stage and a machine by taking into account the manufacturing route of each product. Figure 6 shows the machine health metrics for nine selected machines out of total of 12 machines in stage two. The other three machines are not included in the chart to avoid taking up more space as their data does not provide any additional information and is pretty similar to the presented ones. It can be observed that in machines 2, 6 and 9, there is a noticeable deviation from normal behavior. While the performance of the machines 2 and 9 declines in the short term, machine 6 underperforms consistently throughout the time window shown. These machines were detected as faulty as they were deemed to adversely affect the quality of the products. In the next step, the products gone through these machines were filtered out from the data and the remainder of the data was used for calculating the health metric for the machine in stage one. For filtering out the mentioned samples, a threshold was obtained by applying k-means clustering approach to the data, similar to how it is used in (Siegel & Lee, 2011). The purpose of using k-means was to partition all the calculated CV’s into two clusters of normal and faulty. Figure 7 shows the clusters detected by the algorithm. Based on the results from k-means algorithm, the threshold of 0.65 was selected for determining the underperforming machines. After removing the samples affected by the faulty machines, the metric for the machine in stage one was calculated. For comparison, this metric was also calculated without removing the samples from machines in stage two. Figure 8 shows both metrics for machine in stage one. The blue line shows the metric based on all the samples. After removing the samples affected by machine 9 in stage 2, a portion of the metric inclined significantly (red line in Figure 8). The overall increase in the metric within the shown time period is correlated to the removal of samples produced by machine 6 (Figure 6). Therefore it can be inferred that the remaining low CV values within the red line are due to faulty machine in stage 1. Special cases might happen when even until stage 1, some of low quality products are not being captured by the method. In those special cases it is possible that an environmental factor or raw material, which equally affects all the machines, are the root cause of low quality products.

4. CONCLUSION

This paper provides a machine health monitoring approach for manufacturing processes by relying only on the product quality measurements. Although the advancement and availability of sensing technologies provides a rich information source for PHM scientists and engineers to monitor the machines, the suggested method provides an affordable alternative approach for situations in which the implementation of the sensors are too costly or impractical.
Figure 5. The flowchart of the proposed approach for an example of a three-stage manufacturing process.

Figure 6. Calculated CV for four machines in stage 2 producing the same type of product.
The comparison-based method has been applied to a two staged manufacturing data where quality measurement values used as inputs of the suggested method. The method was able to identify underperforming machines in second and first stage of the manufacturing process by only relying on product quality measurements. There is potential to improve the suggested method by using other statistical features in calculating the final CV value (comparing to use minimum now).

![Figure 7](image1.png)

Figure 7. K-means clustering results using CV’s calculated in stage 2. The two clustered identified by k-means are shown in blue (healthy) and red (faulty).

![Figure 8](image2.png)

Figure 8. Calculated CV for the machine in stage 1 before removing the samples from stage 2 (blue) and after removing the samples from stage 2 (red).

**REFERENCES**


**BIographies:**

**Hossein Davari Ardakani** is currently a PhD student of Mechanical Engineering at the University of Cincinnati. His related research experience in the field of PHM includes manufacturing process, induction motor, gearbox, wind turbine, railway systems and data quality for prognostics applications. Related work experience in the field of prognostics and health management includes five years of research assistantship at the Center for Intelligent Maintenance Systems (IMS) at the University of Cincinnati.

**Brian Phillips** is a graduate student at the department of Mechanical Engineering at the University of Cincinnati, where he received his B.Sc degree in 2014. His research interests include structural dynamics and vibration analysis of rotating machinery.

**Behrad Bagheri** is currently a PhD at the Center for Intelligent Maintenance Systems (IMS) at the University of Cincinnati. His research area is Cyber-physical systems for manufacturing, prognostics and health management of rotating machinery and application of intelligent systems in healthcare.

**Jay Lee** is Ohio Eminent Scholar, L.W. Scott Alter Chair Professor, and Distinguished Univ. Professor at the University of Cincinnati and is founding director of National Science Foundation (NSF) Industry/University Cooperative Research Center (I/UCRC) on Intelligent Maintenance Systems (IMS; [www.imscenter.net](http://www.imscenter.net)) which is a multi-campus NSF Industry/University Cooperative Research Center which consists of the University of Cincinnati (lead institution), the University of Michigan, Missouri University of S&T, and the University of Texas-Austin. Since its inception in 2001, the Center has been supported by over 80 global companies. His current research focuses on intelligent prognostics and predictive analytics. He has authored/co-authored numerous highly influential articles and technical papers in the areas of machinery monitoring and prognostics, E-manufacturing, and intelligent maintenance systems. He has over 20 patents and trademarks. He is a Fellow of ASME, SME, as well as a founding fellow of the International Society of Engineering Asset Management (ISEAM) and has received a number of awards including the most recent Prognostics Innovation Award at NI Week by National Instruments in 2012.
A data pipeline for PHM data-driven analytics in large-scale smart manufacturing facilities

P. O’Donovan\textsuperscript{12}, K. Leahy\textsuperscript{12}, D. Og Cusack\textsuperscript{2}, K. Bruton\textsuperscript{12}, D. T. J. O’Sullivan\textsuperscript{12}

\textsuperscript{1} IERG, University College Cork, Ireland
\textsuperscript{2} Department of Civil and Environmental Engineering, University College

peter_odonovan@umail.ucc.ie

\textbf{ABSTRACT}

The term smart manufacturing refers to a future-state of manufacturing, where the real-time transmission and processing of information across the factory will be used to produce advanced manufacturing intelligence that can optimize every aspect of its operation. In recent years, initiatives and groups such as the Smart Manufacturing Leadership Coalition (SMLC), Industry 4.0, and the Industrial Internet Consortium (IIC), have led the way by bringing together industry, academia and government to establish policies, roadmaps and platforms to support smart manufacturing. Although there are many characteristics that can be associated with smart manufacturing across these initiatives, a common theme is the emphasis on transitioning operations from reactive and responsive, to predictive and preventative. The research presented in this paper focuses on the development of a data pipeline that supports the development of data-driven Prognostics and Health Management (PHM) applications. In the context of smart manufacturing, PHM enables facilities to transition from preventative and reactive maintenance strategies, to predictive, preventative and condition-based strategies. The benefits that can be derived by PHM are aligned with those of smart manufacturing, which include the opportunity to decrease costs, increase machine availability, reduce energy consumption, and improve production yield.

However, the process of ingesting, cleaning and transforming real-time data streams for data-driven PHM is a difficult, complex and time-consuming task, with estimates from business intelligence projects ranging from 80% to 90% of total project effort. This effort may only be exacerbated further in manufacturing environments due to additional technology challenges, such as low levels of standardization, disparate protocols and interfaces, and ad hoc data management. While emerging technologies such as Cyber Physical Systems (CPS) and Internet of Things (IoT) can overcome many of these challenges and provide an open platform for transmitting data, existing large-scale manufacturing facilities that are subject to compliance, regulation, and stringent quality assurance policies may not be able to adopt these technologies in the short-term due to the associated cost, risk and effort. Therefore, PHM applications that need to access data streams in large-scale manufacturing facilities must do so using transparent data integration that does not discriminate between emerging and legacy technologies in the factory. To this end, this research presents a real-time, scalable, robust, and fault tolerant data pipeline for ingesting, cleaning, transforming, processing and contextualizing time-series data from a wide-range of sources in the factory.

\textbf{1. INTRODUCTION}

The term smart manufacturing refers to a paradigm that describes the transmission and sharing of real-time information across pervasive networks with the aim of creating manufacturing intelligence in every aspect of the factory (Davis, Edgar, Porter, Bernaden, & Sarli, 2012a; Lee, Lapira, Bagheri, & Kao, 2013; Lee, 2014; Wright, 2014). Experts predict that smart manufacturing may become a reality in the next 10 to 20 years. The objective of smart manufacturing is similar to manufacturing intelligence insofar as it focuses on the transformation of raw data to knowledge, which can improve decision-making and have a positive impact on operations. However, smart manufacturing supersedes manufacturing intelligence in its emphasis on real-time data collection and aggregation, which facilitates knowledge sharing across physical and computational processes that can result in seamless operating intelligence (Manufacturing et al., 2011). In general terms, smart manufacturing can be considered an intensified application of manufacturing intelligence, where every aspect of the factory is monitored, optimized and visualized (Davis et al., 2012a).

While smart technologies facilitate the creation of knowledge, workers must apply this knowledge in some way before it can have a positive impact on operations. Therefore, while technology transformation is arguably the most publicized aspect of smart manufacturing, the transformation and education of workers should not be ignored. The demands placed on workers in smart manufacturing facilities are not be entirely limited to
vertical operations, and will therefore require a multidisciplinary perspective. Many of the technologies and systems associated with smart manufacturing discuss high data visibility across the factory, where the potential impact of a decision can be evaluated in the context of the entire facility rather than being isolated to a particular department. Without adopting this type of holistic decision-making, it is difficult to envisage how smart manufacturing objectives such as demand-driven and intelligent production, real-time data management, system interoperability, and cyber security, can be realized (Manufacturing et al., 2011). Therefore, decision-makers embedded in smart manufacturing operations will need a basic understanding of multiple disciplines, including engineering, computing, analytics, design, planning, automation, and production (Meziane, Vadera, Kobbacy, & Proudlove, 2000; Sharma & Sharma, 2014).

1.1. Groups and initiatives focused on smart manufacturing

There are a number of government, academic and industry groups promoting an awareness of smart manufacturing. These initiatives include the Smart Leadership Coalition (SMLC) (Manufacturing et al., 2011), Technology Initiative SmartFactory (Zuehlke, 2010), Industry 4.0 (Lee, Kao, & Yang, 2014), and The Industrial Internet Consortium (IIC). These initiatives formed from the realization that challenges facing smart manufacturing adoption are too big for any single organization to address, and while terminology used by initiatives may differ, they share an overarching vision of smart manufacturing where real-time data streams are used to realize operational efficiencies. The two most prominent smart manufacturing initiatives are the SMLC and Industry 4.0, with each loosely related to their geographical origin – the US and EU respectively.

The SMLC working group differs from other initiatives in a couple of ways. The SMLC is comprised of numerous academic institutions, government agencies and industry partners. This blend enables the SMLC to identify real problems by consensus, which may mitigate from bias recommendations that do not serve the wider manufacturing community. Furthermore, the SMLC have not only developed theoretical artifacts relating to smart manufacturing, such as roadmaps, recommendations and guidelines, they have also undertaken the development of a smart manufacturing platform that implements many of these ideas. Industry 4.0 is a high-tech strategy that was created by the German government to promote an awareness of smart manufacturing and its potential economic benefits. The term Industry 4.0 is a simple naming convention that serves to partition each industrial revolution, with 4.0 referring to an anticipated fourth revolution. Expert opinions differ regarding a realistic timeline for Industry 4.0, with general estimates ranging from 10 to 20 years. Exploring the Industry 4.0 naming convention further, previous industrial revolutions are predictably labelled 1.0, 2.0 and 3.0. Industry 1.0 was brought about by the introduction of mechanical production using water and steam power, with the first mechanical loom used in 1784. Industry 2.0 was brought about by the division of labor and the realization of mass production, which were largely facilitated by electrical energy, with the first assembly line introduced in the Cincinnati slaughter house in 1870. Finally, Industry 3.0 was brought about by advances in electronics and IT systems, which enabled automation of production using control networks, with the first programmable logic controller (PLC) Modicon 084 introduced in 1969.

1.2. Benefits of smart manufacturing

Smart manufacturing focuses on pervasive networking and intelligent data-driven analytics that are highly integrated, intelligent, and flexible. The combined application of these technologies can be used to facilitate highly customized and optimized demand-driven supply chains that can dynamically respond to the needs of the customer. Furthermore, smart manufacturing addresses many common business and operating challenges, such as increasing global competition and rising energy costs, while also facilitating shorter production cycles that respond quickly to customer demand (Manufacturing et al., 2011; Sharma & Sharma, 2014). In addition to these high-level efficiencies, more quantifiable benefits have also been cited. For example, the SMLC identified realistic performance targets for different aspects of smart manufacturing, including (1) a 30% reduction in capital intensity, (2) up to a 40% reduction in product cycle times, as well as (3) an overarching positive impact across energy, emissions, throughput, yield, waste, and productivity. Furthermore, smart manufacturing can also provide benefits to the wider economy. A recent report from Fraunhofer Institute and Bitkom highlights the potential economic benefit of Industry 4.0 to the German economy. The report states the transformation of traditional factories to Industry 4.0 could be worth 267 billion euros cumulatively to the German economy by 2025 (Heng, 2014).

1.3. Impediments to smart manufacturing adoption

While the potential benefits of smart manufacturing are apparent, there are numerous challenges and issues that must be overcome before they can be realized. In particular, facilities must develop the infrastructure and network-intensive real-time technologies needed to support smart manufacturing, as well as cultivating multidisciplinary workforces and next-generation IT departments that are capable of working with smart
technologies (Manufacturing et al., 2011). The degree to which these challenges exist in each facility will vary. For example, there are obvious differences between implementation challenges in greenfield and brownfield sites (Davis, Edgar, Porter, Bernaden, & Sarli, 2012b). Excluding fundamental challenges, such as budgetary constraints, technology availability and the presence of a skilled workforce, greenfield sites are better positioned to adopt emerging smart technologies when compared with brownfield sites. Brownfield sites may be restricted by legacy devices, information systems, and protocols, which can also include proprietary and ad hoc technologies. These technologies are from a time when low latency distributed real-time networks and large-scale data storage and processing were simply not a concern. In some instances legacy technologies may be replaced with smarter equivalents, but there are numerous reasons why substitution may not be an option;

- **Historical investment in IT and automation.** Many facilities invested in information systems and automation networks over the last 40 years. Therefore, facilities may be reluctant to replace technologies that received significant investment and continue to operate at an appropriate level.

- **Regulatory and quality constraints.** In certain industries, such as pharmaceuticals and medical devices, internal or external constraints may exist in the form of regulatory and/or quality standards. In these instances, the existence of exhaustive processes and procedures may negate the enthusiasm for legacy technology replacement.

- **Dependency on proprietary systems or protocols.** While numerous open standards exist for manufacturing information systems and automation networks, such as ISA95 for system interoperability and OPC for device-level communication, their adoption is sporadic. Therefore, where proprietary and closed technologies are used in place of open standards, technology adoption (i.e. smart technologies) is limited by the proprietary vendors offerings.

- **Weak vision and insufficient commitment.** The transition to smart manufacturing is a significant undertaking that requires strong leadership and a shared vision of the short and long-term benefits for the facility. Facilities that do not have a clear vision of how smart manufacturing can improve their operations may be less likely to have an appetite for technology replacement.

- **High risk and disruption.** The implementation of new and emerging technologies and systems are considered high-risk projects, which can negatively impact operations while technical competency is being achieved. Therefore, the appetite to undertake large-scale IT projects may be weak until such time lost opportunities effect the facilities competitiveness.

- **Skills and technology awareness.** IT and automation departments are entrenched in mature computing, automation and networking methods that have been in existence for decades. However, technologies synonymous with smart manufacturing (e.g. IoT, CPS, Big Data, Cloud Computing) require a shift from these approaches. Therefore, if the relevant departments do not embrace these technologies and contribute to the organizations smart manufacturing roadmap, their lack of knowledge may impede technology replacement.

There are numerous impediments surrounding the introduction of technologies for smart manufacturing, but the majority of these relate to brownfield sites where technology replacement can be problematic. The main challenge facing brownfield sites is the encapsulation and integration of legacy technologies with emerging smart technologies, methodologies and roadmaps. Facilities that do not address these issues may be restricted in their adoption of smart manufacturing, and the realization of its associated performance enhancements and benefits.

This paper focuses on Prognostics and Health Management (PHM) applications for equipment maintenance in the context of smart manufacturing. PHM comprises methods for detecting and predicting equipment faults to optimize equipment uptime and availability (Bruton et al., 2014; Bruton, Coakley, O’Donovan, Keane, & O’Sullivan, 2013; Lee, Bagheri, & Kao, 2015; Lee et al., 2013; O’Donovan, Leahy, Bruton, & O’Sullivan, 2015; Wright, 2014). The main contributions of this paper are high-level requirements for data-driven smart manufacturing systems in highly regulated and quality controlled brownfield sites, where legacy integration may impede smart manufacturing adoption, and a system architecture that satisfies these requirements and enables real-time data ingestion and big data processing in the cloud.

2. Research Methodology

This research employed an embedded study, which was undertaken in DePuy Ireland - a large-scale manufacturing facility which is part of the Johnson & Johnson family of companies. The aim of this research was to identify the main requirements and associated system architecture, to support the development of PHM applications by reducing expensive and time-consuming activities, such as ad hoc data integration, and improving overall data accessibility and reusability.
2.1. Establishing scope
As there were numerous potential applications of PHM and industrial analytics in the context of smart manufacturing, the first priority was to establish research boundaries. After an initial discussion between research team members, and automation personnel in DePuy Ireland, the focus of the research was narrowed using the following specificities.

2.1.1. Type of PHM applications
It was agreed that research efforts would focus on data-driven applications that deal with predictive and intelligent equipment maintenance. Equipment uptime and availability was considered a critical aspect of operations given the potential impact downtime can have on production. Therefore, the development of a solution that can stream data directly to PHM applications focused on promoting machine uptime and availability was deemed a worthwhile pursuit.

2.1.2. Regulation and compliance
Given this research was undertaken in a highly regulated and quality-focused environment, and an empirical research methodology was employed, there is an implied narrowing of the research scope insofar as observations may only apply to facilities with the same characteristics. As legacy technology replacement is not easily achieved in these environments (e.g. smart technologies), it was agreed an emphasis would be placed on legacy technology integration, with the aim of amalgamating legacy and smart technologies in a single framework. This was considered a significant real-world challenge for brownfield sites, which could only improve the value of this research. Furthermore, it was agreed that the final solution requirements and prerequisites should be minimal (e.g. it should not require a facility to use a particular brand of controller).

2.1.3. Time-series data
This research focuses on data-driven PHM applications for equipment maintenance in the context of smart manufacturing. Therefore, it was agreed data ingestion, processing and accessibility aspects of the research could be limited to time-series data measurements. Based on experiences of research team members and feedback from automation personnel in DePuy Ireland, time-series data was the format most relevant to equipment maintenance monitoring, analysis and decision-making. By limiting the pipeline to a particular class of data the number of permutations for extraction, transformation and loading operations were reduced, given the predictable and low-dimensional structure of the data (e.g. time/value pairs).

2.1.4. Data flow and direction
The overarching theme of this research is the investigation of real-time data integration and transmission from large-scale industrial facilities. Therefore, it was agreed that data flows in the pipeline would only move one-way (i.e. factory to cloud) and this data would be immutable (i.e. read only). While this research may not consider a two-way communication channel for PHM applications to send instructions back to the factory, these applications can extend the framework and implement their own protocol to initiate actions in the factory if required.

2.1.5. Industrial data integration
Legacy integration was deemed an important aspect of this research given the prominence of proprietary systems and diverse communication protocols that can exist in industrial environments. However, given the broad and ill-defined nature of this problem it was agreed initial legacy integration would be limited to log files produced by Programmable Logic Controllers (PLC) and Manufacturing/Building Systems, OLE Process Control (OPC), Modbus, and BACnet.

2.1.6. Industry collaboration
To better understand the manufacturing systems, processes and technologies in DePuy Ireland, and to gain a greater appreciation for manufacturing operations in general, we engaged with internal teams across automation, energy, big data and smart manufacturing. Discussions with these teams assisted in the identification of data sources, processes and industrial protocols that were relevant to PHM applications in the factory.

- **Automation** - the automation team consisted of eight staff with skills covering control and automation, production, energy and information technology. The automation team informed the research teams understanding of infrastructure supporting production in the factory, as well as scheduling and maintenance strategies for machinery.

- **Energy** – the energy team comprised of five staff with skills in engineering and energy. The energy team informed the research teams understanding of energy consumption monitoring for equipment, as well as highlighting how malfunctioning equipment can produce energy fluctuations.

- **Big Data and Smart Manufacturing** - there are no dedicated teams currently responsible for big data and smart manufacturing. Therefore, the research team interacted with multiple teams and personnel to form a better understanding of how emerging technologies, such as Internet of Things (IoT) and big data...
technologies, were being considered for use in the factory.

2.2. Research questions

Two research questions were identified to guide research efforts. The purpose of the first question (RQ1) was to establish the real-world data integration requirements for large-scale industrial environments, with an emphasis on those that are not supported by traditional data integration tools (e.g. ETL tools). The purpose of the second question (RQ2) was to create an open and accessible architecture for data ingestion, processing and management that could satisfy requirements identified by RQ1.

2.2.1. RQ1 – What requirements and characteristics are important to large-scale manufacturing facilities when it comes to data integration methods?

This question focuses on establishing requirements and characteristics that may support the development of data-driven PHM applications in real-world large-scale manufacturing environments, with a particular emphasis on facilitating transparent data flows across the entire factory, which is aligned with the vision of smart manufacturing.

2.2.2. RQ2 – How can a data pipeline serve data-driven PHM applications using legacy and emerging technologies in an indiscriminate manner?

This question considers the design of a data pipeline architecture that can provide a framework for PHM applications focused on equipment maintenance, while satisfying the requirements from RQ1. Recommendations from smart manufacturing are combined with those of RQ1 to further inform the pipelines design, incorporating the need for real-time capabilities, open standards, and seamless data access.

3. RESULTS AND DISCUSSION

3.1. RQ1 – Requirements and characteristics

The following requirements and characteristics were identified in response to RQ1 during the study. Although these findings were derived from discussions relating to PHM applications focused on equipment maintenance, they should also be considered representative of industrial data integration challenges facing facilities transitioning to smart manufacturing.

3.1.1. Legacy integration

Some facilities will not be in a position to adopt emerging and smart technologies to realize intelligent systems associated with smart manufacturing. Based on observations of the research team, many large-scale manufacturing facilities may have invested too much time and resources in control and automation networks to consider replacing legacy devices with smarter equivalents. Similarly, while many facilities may be aware of the potential benefits associated with emerging technologies, such as big data analytics, they may not know how they can integrate with existing operations, or fully appreciate the multi-disciplinary and technical skills needed to implement them in the facility. Therefore, facilities may want to leverage and maximize existing investments, skills, knowledge, vocabulary and systems, while incrementally transitioning to smart manufacturing rather than completely overhauling technologies and operations. To achieve this transparent data integration will be an important requirement, whereby legacy and smart technologies are abstracted to deliver indiscriminant data access.

3.1.2. Cross-network communication

Real-time data transmission across pervasive networks is a fundamental aspect of smart manufacturing. However, networks in modern manufacturing facilities were not designed with these characteristics in mind. The research team encountered several instances during the study where equipment maintenance data was restricted by firewalls and other security measures. Furthermore, limited access to equipment data was also encountered due to external maintenance and support agreements with vendors (e.g. wind turbines). While these measures may make sense in the context of traditional manufacturing operations, they represent a challenge to data-driven smart manufacturing. Therefore, to provide data visibility across facilities (and/or multiple sites) it may be necessary to communicate across secure networks.

3.1.3. Fault tolerance

Information systems and technologies that play a role in production, automation and maintenance may have high demands placed on them given their ability to directly impact facilities production yield and operational efficiency. Based on observations of the research team, information systems deployed in industrial environments must be highly available and fault tolerant. Therefore, these characteristics will be expected of new systems and tools operating in similar environments.

3.1.4. Extensibility

Proprietary and/or ad hoc technologies and systems in large-scale manufacturing facilities are common. Based on observations of the research team, it appears that facilities have become more aware of technology integration and consolidation, but duplication and
disparity across systems is still evident. Some of these inefficiencies may be due to the inextensibility of existing information systems, which can result in ad hoc implementations. Therefore, an important requirement for systems operating in industrials environments is extensibility, where new data types, methods and protocols can be supported as requirements emerge.

3.1.5. Scalability

As the digitization of factories accelerate, the ability to dynamically scale based on demand is becoming a desirable characteristic for industrial information systems. This is especially relevant when considering emerging technologies (e.g. IoT) in smart manufacturing, and the unknown load they will place on these systems. While modern large-scale manufacturing facilities are entrenched in technology, the real-time and data-rich nature of smart manufacturing may expose unforeseen limitations due to their inability to scale. For example, the normal resolution for data measurements observed during the study was 15 minutes. Without considering the addition of new sensors and measurements (i.e. IoT), the data production rate in the facility would increase by 900% if measurement intervals were reduced to 1 second. Therefore, the ability to scale based on demand is an important requirement.

3.1.6. Data accessibility

Modern large-scale manufacturing facilities produce a lot of data. However, based on observations of the research team, access is inhibited by diverse protocols, formats and structures. These issues are currently overcome using expensive data discovery and integration procedures, which focus on proprietary and ad hoc data integration routines that address the peculiarities of a particular project. However, these approaches typically result in poor reuse, which results in tasks being repeated and duplicated across projects. Therefore, it is important to abstract and generalize low-level data integration routines to provide a consistent data interface for data sources and devices in the factory.

3.2. RQ2 – Data pipeline architecture

A high-level data pipeline architecture for data-driven PHM applications focused on equipment maintenance was produced using the requirements from RQ1. Figure 1 illustrates the data pipeline architecture, with each stage of the factory-to-cloud workflow numbered and highlighted. The aim of the data pipeline is to deliver a low cost turnkey solution for industrial data integration, which is built on a real-time, open, scalable and fault tolerant infrastructure. The purpose and function of each component and stage in the data pipeline is described in the proceeding sections.
Figure 1. Data pipeline architecture and workflow
3.2.1. Stage 1 – Site manager  

**Purpose:** The site manager resides on a cloud server and stores meta-data regarding each facility and associated data sources. Its purpose in the architecture is to persist essential site information, such as the location of each data point and the type of protocol that needs to be used for integration.

**Functions:** The site manager has multiple functions that are related to the factory – (1) store details relating to the site, such as the type and location of local data sources that shall be ingested, (2) schedule and assign jobs to ingestion engines in the factory based on the availability and location of each node, and (3) derive a suitable amount of data to ingest for each engine based on its current location, bandwidth, CPU and bandwidth availability.

3.2.2. Stage 2 – Ingestion process  

**Purpose:** Ingestion engines are distributed software agents that are deployed across networks in the factory to collect and integrate time-series data for different data-driven PHM applications (e.g. HVAC, Chillers, Boilers). They execute as background workers on a server and continually send their status to the site manager (Stage 1), and when instructed, collect and transmit data from local data sources to the cloud. As illustrated in Figure 1, the distributed and autonomous nature of an ingestion engine enables them to be deployed across different networks that are separated by firewalls and/or geographical boundaries. Furthermore, these characteristics also allow the ingestion process to scale by deploying more ingestion engines, which increases the throughout capacity of the pipeline from the factory.

**Functions:** The ingestion engine has multiple functions – (1) communicate location, bandwidth, CPU and memory availability to the site manager so an appropriate ingestion task can be assigned, (2) interpret ingestion tasks sent by the site manager and automatically extract time-series data from the relevant sources in accordance with the task parameters (e.g. particular data range), and (3) transmit the collected time-series data to the cloud message queue. A novel aspect of the ingestion process is an expert ruleset that can automatically map and extract time-series data to limit expensive and manual data mapping tasks.

3.2.3. Stage 3 – Message queue  

**Purpose:** The highly available and distributed message queue service in the cloud accepts time-series data from ingestion engines in the factory. Its main purpose is to provide intermediary storage between the factory and processing components in the pipeline. This decouples data ingestion components from data processing components, which instills resilience by facilitating asynchronous communication and parallel operations when the pipeline is at peak demand.

**Functions:** The message queue has two main functions – (1) notify subscription service when new data has been ingested, and (2) add received data to a queue so data processing components can access it further down the pipeline.

3.2.4. Stage 4 – Subscription service  

**Purpose:** The subscription service provides an endpoint for the data ingestion process and functions as a notification mechanism for data processing components when new data is received. The notification of new data results in one or more data preparation and/or analysis tasks being undertaken. The number of data processing actions executed can be increased or decreased by subscribing or unsubscribing from the subscription service.

**Functions:** The functions of the subscription service are limited, but essential in the orchestration of events in the pipeline – (1) listen to the message queue for new data and (2) notify subscribers when new data are available for processing.

3.2.5. Stage 5 – Data processing  

**Purpose:** Data processing components are responsible for transforming raw time-series data to a format suitable for analysis. The aim of data processing is to remove the onus on each PHM application to undertake expensive and time-consuming operations. At the most basic level the pipeline aggregates time-series data at different levels of granularity, such as hourly, daily, monthly and annual averages. Examples of more sophisticated processing may include the execution of expert rules to identify faults, or the semantic encoding of time-series data (e.g. Project Haystack) to promote interoperability with other applications. Each data processing component in the architecture is responsible for executing a single processing operation to promote modularity. This enables new data processing components to be easily added to the pipeline as new requirements emerge.

**Functions:** The functions that may be associated with data processing are truly diverse. Therefore, data processing components in the pipeline cannot be strictly prescribed given processing requirements will vary from application-to-application, and factory-to-factory. However, common use cases could be built over time to form a library of default processing components. The current default scenario illustrated in the data pipeline is time-series aggregation – (1) daily average, (2) monthly average, and (3) annual average.

3.2.6. Stage 6 – Data access  

**Purpose:** The data access stage exposes a consistent and open method for PHM applications to consume data that
originated from equipment in the factory. A naming convention is used to promote consistency in data access and contextualize data requests. The convention uses an encoded URL to request data for an object (e.g. HVAC) and date. Figure 1 illustrates the naming convention between stage 5 and 6 in the pipeline, and Table 1 describes each parameter of the naming convention in more detail;

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Refers to a particular data set (e.g. energy).</td>
</tr>
<tr>
<td>Object</td>
<td>Identifying name or code that exists within the data set (e.g. machine number).</td>
</tr>
<tr>
<td>Year</td>
<td>Year relevant to data request/query.</td>
</tr>
<tr>
<td>Month</td>
<td>Month relevant to data request/query.</td>
</tr>
<tr>
<td>Day</td>
<td>Day relevant to data request/query.</td>
</tr>
</tbody>
</table>

Functions: Functions relating to data access include – (1) ensuring data are stored in the appropriate location as per the naming convention and (2) return the appropriate data for requests that utilize this convention.

3.3. Alignment of architecture with requirements

This section discusses how the data pipeline architecture from RQ2 satisfies requirements from RQ1. Table 2 supports this discussion by describing how different stages of the pipeline address different requirements.

Table 1. URL convention for data requests

Table 2. Relationship between requirements and architecture

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Data pipeline stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy integration</td>
<td>Stages 1 and 2 in the architecture are responsible for legacy integration. The site manager creates meta-data for each data source in the factory, which is then used by the ingestion engine to extract data independent of the underlying source (i.e. either legacy or smart).</td>
</tr>
<tr>
<td>Cross-network communication</td>
<td>Stages 2 and 3 in the architecture facilitate communication across networks. Ingestion engines are remote autonomous agents that are network agnostic. Therefore, given an outbound connection to the queue service in the cloud, ingestion engines can be deployed across multiple networks to unify data in the pipeline.</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Stages 5 and 6 promote extensibility in the data pipeline architecture. First, data ingestion instructions are dynamically disseminated from the site manager, which means these instructions can be extended to support new types of data sources etc. Second, data processing components are modular, which enables processing capabilities of the pipeline to be extended through the addition of new processing components. Finally, the type of data served to PHM applications via the data interface can be extended to include additional formats.</td>
</tr>
</tbody>
</table>

Scalability

Stages 1-6 illustrate how scalability is embedded in the pipeline. First, the distributed design of the ingestion process is realized using autonomous agents, which enables integration routines to run in parallel and scale based on the number of data points being measured. Second, the message queue, notification and storage services are inherently scalable given the selection of a cloud provider that supports auto-scaling. Finally, data processing components (i.e. workers) can benefit from load balancing and auto scaling features of cloud computing, which enables these components to dynamically distribute and scale based on the quantity of data to be processed.

Data accessibility

Stage 6 provides a common interface for PHM applications to consume data from the pipeline. The architecture employs a cloud-based repository to serve low-latency precompiled views of time-series data to geographically distributed end-users and PHM applications. Furthermore, interoperability with 3rd party applications is supported by the use of open standards and protocols (e.g. HTTP, JSON etc.).

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a set of challenges and characteristics associated with collecting and integrating industrial data for data-driven manufacturing, and a system architecture that addresses these challenges. In particular, the system architecture supports real-time data ingestion from a range of legacy and smart devices throughout the factory, while using a mix of novel and conventional technologies to promote fault tolerance, scalability and accessibility. The automated data pipeline architecture can provide facilities with a robust, flexible and adaptable managed framework to enable data-driven manufacturing (i.e. smart manufacturing) while mitigating low-level technical details, such as legacy and smart technology integration. While emerging smart sensors and technologies (e.g. IoT) will eventually eliminate the need for legacy
integration, given the fact 20 year old PLC’s are still in operation, it is advisable that researchers and innovators should be conservative when estimating timelines for when large-scale industrial facilities will be operating using smart technologies exclusively. Therefore, to transition to smart manufacturing and develop the insightful data-driven applications that can deliver predictive and efficient operations, facilities must be capable of addressing the fundamental issue of transparent data integration.

ACKNOWLEDGEMENTS

The authors would like to thank the Irish Research Council, DePuy Ireland and Amazon Web Services for their funding of this research, which is being undertaken as part of the Enterprise Partnership Scheme (EPSPG/2013/578).

REFERENCES


Early Detection of Combustion Instability from Hi-speed Flame Images via Deep Learning and Symbolic Time Series Analysis

Soumalya Sarkar1, Kin G Lore2, and Soumik Sarkar3, Vikram Ramanan4, Satyanarayanan R Chakravarthy5, Shashi Phoha6, Asok Ray7

1,6,7 Pennsylvania State University, University Park, Pennsylvania, USA
svs5464@psu.edu, sxp26@arl.psu.edu, axr2@psu.edu

2,3 Iowa State University, Ames, Iowa, USA
kglore@iastate.edu, soumiks@iastate.edu

4,5 Indian Institute of Technology Madras, Chennai, India
vikrambest@yahoo.co.in, src@ae.iitm.ac.in

ABSTRACT
Combustion instability, characterized by self-sustained, large-amplitude pressure oscillations and periodic shedding of coherent vortex structures at varied spatial scales, has many detrimental effects on flight-propulsion dynamics and structural integrity of gas turbine engines. Hence, its early detection is one of the important tasks in engine health monitoring and prognostics. This paper proposes a dynamic data-driven approach, where a large volume of sequential hi-speed (greyscale) images is used to analyze the temporal evolution of coherent structures in combustion chamber for early detection of combustion instability at various operating conditions. The proposed hierarchical approach involves extracting low-dimensional semantic features from images using Deep Neural Networks followed by capturing the temporal evolution of the extracted features using Symbolic Time Series Analysis (STSA). Extensive experimental data have been collected in a swirl-stabilized dump combustor at various operating conditions for validation of the proposed approach. Intermediate layer visualization of deep learning reveals that meaningful shape-features from the flame images are extracted, which enables the temporal modeling layer to enhance the class separability between stable and unstable regions. At the same time, the semantic nature of intermediate features enables expert-guided data exploration that can lead to better understanding of the underlying physics. To the best of the authors knowledge, this paper presents one of the early applications of the recently reported Deep Learning tools in the area of prognostics and health management (PHM).

1. INTRODUCTION
Combustion instability is a very undesirable phenomenon characterized by high-amplitude flame oscillations at discrete frequencies. These frequencies typically represent the natural duct/resonator acoustic modes. Combustion instability, in its most basic form arises when there is a positive coupling between the heat release rate oscillations and the pressure oscillations, provided this driving force is higher than the damping present in the system. The mechanisms of pressure-heat release rate coupling are system dependent and thus, the problem of combustion instability becomes very system specific.

The underlying principle of heat release rate oscillations, that drives the pressure oscillations—which result in velocity oscillations and in turn modulate heat release rate oscillations—all in a turbulent background in case of actual gas turbine combustors, pose significant complexities in determining the mechanisms of combustion instability. Crocco (Mcmanus, Poinsot, & Candel, 1993) modeled unsteady heat release rate as a function of unsteady velocity to determine stability of a ducted zero-mean flow flame. Subsequently, a whole class of reduced order modeling-Flame Transfer/Describing functions were theoretically and experimentally (Palies, Schuller, Durox, & Candel, 2011; Noiray, Durox, Schuller, & Candel, 2008; Bellows, Bobba, Forte, Seitzman, & Lieuwen, 2007) formulated to understand the stability of the system by means of solving the dispersion relation. In addition, flame oscillation saturation mechanisms were also experimentally diagnosed which in addition to experiments based on turbulent non reacting and reacting flows led to the universal feature of combustion instability—heat release rate oscillations driven by coherent structures.

Coherent structures are fluid mechanical structures associ-
ated with coherent phase of vorticity, high levels of vorticity among other definitions (Hussain, 1983). These structures, whose generation mechanisms vary system wise, cause large scale velocity oscillations and overall flame shape oscillations by curling and stretching. These structures can be caused to shed/generated at the duct acoustic modes when the forcing (pressure) amplitudes are high. The interesting case of the natural shedding frequency of these structures, causing acoustic oscillations, has been observed by Chakravarthy et al. (Chakravarthy, Shreenivasan, Bhm, Dreizler, & Janicka, 2007).

Recently, a swirl combustor has been characterized and a wide range of experiments relating swirl flows and coherent structures associated with swirl flows has been reported (Syred, 2006; Paschereit, Gutmark, , & Weisenstein, 1998). The presence of Precessing vortex core as the dominant coherent structure has been reported and non linear interactions between heat release rate oscillations and PVC as the cause of superposed frequencies in time series data has also been reported (Moeck, Bourgouin, Durox, Schuller, & Candel, 2012). Much of the literature is dedicated to detection and correlation of these coherent structures to heat release rate and unsteady pressure. The popular methods resorted for detection of coherent structures are proper orthogonal decomposition (POD) (Berkooz, Holmes, & Lumley, 1993) and dynamic mode decomposition (DMD) (Schmid, 2010), which use tools from spectral theory to derive spatial coherent structure modes. DMD has been used to estimate the growth rates and frequencies from experimental data and thus offered to perform stability analysis on experimental data.

This paper proposes a data-driven hierarchical framework for early detection of thermo-acoustic instability from hi-speed greyscale images. In the lower layer, large volume of hi-speed sequential images are used to train a deep neural network model that extracts hierarchical features from the training data (G. E. Hinton & Salakhutdinov, 2006) through the use of multiple layers of latent variables. An unsupervised pre-training approach with deep-belief networks (DBN) (G. E. Hinton, 2009) is used in particular to automatically learn the coherent structures while reducing the dimension of the images for temporal modeling at the top layer (Erhan, Bengio, et al., 2010). Symbolic time series analysis (STSA) (Ray, 2004), a fast probabilistic graphical model is placed at the top layer to extract temporal feature from the output of deep learning model. The concept of STSA has been used for anomaly detection in physical systems as reported in (Ray, 2004; Rao, Ray, Sarkar, & Yasar, 2009; Sarkar, Jin, & Ray, August, 2011). Recently, STSA is applied on pressure and chemiluminescence time series for early detection of Lean-blow out (Mukhopadhyay, Chaudhari, Paul, Sen, & Ray, 2013; Sarkar, Ray, Mukhopadhyay, Chaudhari, & Sen, 2014) and thermo-acoustic instability (Ramanan, Chakravarthy, Sarkar, & Ray, 2014).

From the above perspectives major contributions of the paper are delineated below.

- A novel data-driven framework, with DBN at lower layer and STSA at upper layer, is proposed for early detection of thermo-acoustic instability from hi-speed images.
- In the above framework, the DBN layers extract meaningful shape-features to represent the coherent structures in the flame images. This phenomenon enables STSA at the temporal modeling layer to enhance the class separability between stable and unstable modes of combustion, which implies higher precision for early detection of the onset of combustion instability.
- The proposed theory and the associated algorithms have been experimentally validated at multiple operating conditions in a swirl-stabilized combustor by characterizing the stable and unstable states of combustion.
- Training and testing of the proposed framework have been performed on different operating conditions (e.g., Reynolds number ($Re$), fuel flow rate, and air-fuel premixing level) of the combustion process to test the transferability of the approach. Performance of the proposed framework (DBN+STSA) have been evaluated by comparison with that of a framework, where DBN is replaced by another extensively used dimensionality reduction tool, principal component analysis (PCA) (Bishop, 2006).

The paper is organized in five sections, including the present one. Section 2 describes a laboratory-scale swirl-stabilized combustor, which serves as a test apparatus for experimental validation of the proposed architecture for early detection of thermo-acoustic instability. Section 3 describes the proposed framework along with its building blocks via explaining the concepts of DBN and STSA. Section 4 presents the capability and advantages of the proposed approach along with the feature visualization at intermediate layers of DBN. Finally, the paper is summarized and concluded in Section 5 with selected recommendations for future research.

2. Experimental Setup

The swirl combustor test bed used in this study has a swirler of diameter 30 mm with 60 degree vane angles, thus yielding a geometric swirl number of 1.28. Air to the combustor is fed through a settling chamber of diameter 280 mm with a sudden contraction leading to a square cross section of side 60 mm. This provides an area ratio of around 17, which thus acts as an acoustically open condition at the contraction. A mesh and honeycomb are mounted in immediate downstream of the contraction to provide uniform flow to the swirler. The combustor, shown in figure 1(a) consists of an inlet section of length 200 mm, an inlet optical access module(IOAM) of length 100 mm to provide optical access to the fuel tube, a primary combustion chamber of length 370 mm, and secondary
duct of the same length. Extension ducts of the same cross section are added to provide length flexibility. The overall length of the constant area ducts was chosen to be 1340 mm.

The fuel injection is done by injecting it coaxially with the air in a fuel injection tube with slots on the surface as shown in Figure 1(b). The fuel injection tube is coaxial to a mixing tube which has the same diameter as that of the swirler. The bypass air that does not enter the mixing tube passes through slots on the swirl plate. The slots on the fuel injection tube are drilled at designated distance upstream of the swirler. The larger this distance, more fuel mixes with the primary air in the mixing tube thus leading to more premixedness. Two upstream distances of $X_1 = 90$ mm and $X_2 = 120$ mm were chosen for this work. The upstream distance of 120 mm provides for full premixing of the fuel with the air thus henceforth, it will be referred to as the premixed case. The 90 mm upstream injection case causes partial premixing of the fuel with air and thus will be referred to as the partially premixed case. The images were acquired at 3 kHz using Photron High speed star with a spatial resolution of $1024 \times 1024$ pixels. The data acquisition was triggered simultaneously using NI card and taken for a duration of 3 s yielding in a sequence of 9,000 images for every operating condition.

Two inlet Reynolds numbers (Re), based on the swirler diameter were chosen, the lower Re having stable combustion behavior and higher Re having exhibiting unstable behavior. The Re’s were chosen to be 7,971 and the higher Re being 15,942 for a fuel flow rate (FFR) of 0.495 g/s. Another protocol followed was keeping the inlet Re constant at 10,628 and having two different fuel flow rates. The higher FFRs exhibited stable combustion, whereas the leaner configuration was unstable. The two FFRs were chosen to be 0.66 g/s and 0.308 g/s. These corresponded to equivalence ratios of 0.955 and 0.445 respectively. Besides these conditions, 3 seconds of images are also collected for $Re = 1,771$ and $FFR = 0.083$ at relatively stable state of combustion. The details of the operating conditions along with their ground truth (e.g., stable or unstable) are presented in table 1.

<table>
<thead>
<tr>
<th>Premixing</th>
<th>FFR (g/s)</th>
<th>Re</th>
<th>Ground truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial</td>
<td>0.495</td>
<td>7,971</td>
<td>Stable</td>
</tr>
<tr>
<td>$(X_1 = 90$ mm)</td>
<td>0.308</td>
<td>15,942</td>
<td>Unstable</td>
</tr>
<tr>
<td></td>
<td>0.66</td>
<td>10,628</td>
<td>Stable</td>
</tr>
<tr>
<td>Full</td>
<td>0.495</td>
<td>7,971</td>
<td>Stable</td>
</tr>
<tr>
<td>$(X_2 = 120$ mm)</td>
<td>0.308</td>
<td>15,942</td>
<td>Unstable</td>
</tr>
<tr>
<td></td>
<td>0.66</td>
<td>10,628</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>0.083</td>
<td>1,771</td>
<td>Relatively stable</td>
</tr>
</tbody>
</table>

Figure 2 presents sequences of images of dimension 392 $\times$ 1000 pixels for both stable ($Re = 7,971$, $FFR = 0.495 g/s$ and full premixing) and unstable ($Re = 15,942$, $FFR = 0.495 g/s$ and full premixing) states. The flame inlet is on the right side of each image and the flame flows downstream to the left. It can be observed that the flame does not have any prominent coherent structure when the combustion is stable. While the combustion is unstable, vortex shedding along the flow is observed. Bottom segment of the figure 2 shows formation of mushroom-shaped vortex at $t = 0.001 s$ and the shedding of that towards downstream from $t = 0.002 s$ to $t = 0.004 s$.

3. Decision Framework and Tools

This section describes the proposed architecture for early detection of thermo-acoustic instability in a combustor via analyzing a sequence of hi-speed images. Figure 3 presents the schematics of the framework where a deep belief net-
work (DBN) is stacked with symbolic time series analysis (STSA). In the training phase, images (or a segment of the images) from both stable and unstable states for various operating conditions are used as the visible layer $V$ of a DBN. Multiple hidden layers (i.e., $h_1$ to $h_n$) with reducing dimensions (G. E. Hinton & Salakhutdinov, 2006) are stacked after the visible layer. The weights (i.e., $W_1$ to $W_n$), connecting adjacent layers, are learned first via greedy layer-wise pre-training (G. E. Hinton & Salakhutdinov, 2009) and then they are fine-tuned in a supervised manner. In this paper, unsupervised pre-training step is emphasized more for capturing the coherent structures in flame images at unstable state. The vector of activation probabilities of the hidden units at the topmost hidden layer is used as input to the STSA module.

While testing, sequence of images are passed through the learned DBN and a time series of $L_2$ norm (equivalent to signal energy) of the activation probability vectors is obtained. In STSA module, the time-series is symbolized via partitioning the signal space and a symbol sequence is created as shown in the figure 3. A probabilistic finite state automata (PFSA) (Ray, 2004) is constructed from the symbol sequence, which models the transition from one state to another as state transition matrix. State transition matrix is the extracted feature which represents the sequence of images, essentially capturing the temporal evolution of coherent structures in the flame. DBN and STSA are explained in detail later in this section.

### 3.1. Deep Learning techniques

Deep Learning is an emerging branch of machine learning with a strong emphasis on modeling multiple levels of abstraction (from low-level features to higher-order representations, i.e., features of features) from data (Deng & Dong, 2014; Bengio, Courville, & Vincent, 2013). For example, in a typical image processing application while low-level features can be partial edges and corners, high-level features may be combination of edges and corners to form parts of an image.

Among various deep learning techniques, Deep Belief Networks (DBNs) have become an attractive option for data dimensionality reduction (G. E. Hinton & Salakhutdinov, 2006), collaborative filtering (Salakhutdinov, Mnih, & Hinton, 2007), feature learning (Coates, Ng, & Lee, 2011), topic modeling (G. E. Hinton & Salakhutdinov, 2009), and solving classification problems (Larochelle & Bengio, 2008). Several other deep learning architectures such as Convolutional Neural Networks, Stacked Denoising Autoencoders, and Deep Recurrent Neural Networks have also gained immense traction recently as they have been shown to outperform all other state-of-the-art machine learning tools for handling very large dimensional data spaces to learn features in order to perform detection, classification and prediction. The basic building block of DBN is the Restricted Boltzmann Machine (RBM), where multiple RBMs are stacked on top of another to form a deep network. An RBM is essentially a generative probabilistic graphical model that is capable of learning a probability distribution over the inputs to best explain the observed data. Individual RBMs consists of visible units (the inputs) which are connected to latent variables in the hidden units. Note that connections exist only between the visible layer and the hidden layer but not among visible units and hidden units—hence termed Restricted. While a single layer of RBM is already quite powerful to represent complex distributions, increasing the number of hidden layers greatly improves modeling capacity where the output of one hidden layer becomes the input of another placed over it.

Deep Belief Networks can be trained in an unsupervised greedy layer-wise manner. In simpler terms, the first RBM layer is trained with the raw input as the visible layer. During training, the first layer acquires a representation of the input by updating its weights and biases between the visible and hidden layers (usually through computing the mean activations or by sampling) which in turn becomes the input of the second layer (G. Hinton, Osindero, & Teh, 2006). The objective during layer-wise training is to find the weight vector $W$ (and biases for both visible and hidden units) that maximizes the expected log likelihood of the training data $V$ (Fischer & Igel, 2014). More formally, the optimization problem can be represented (ignoring the biases) as:

$$\arg \max_W \mathbb{E} \left[ \sum_{v \in V} \log P(v) \right]$$
Typically, the optimization is solved in a gradient descent manner. Keeping the weights and biases of the first layer constant after it is trained, the transformed input from the layer is utilized to train the next layer. This process is repeated for the desired number of layers in the network with each iteration propagating either the samples or mean activations to higher levels. As training continues, the product of probabilities assigned to the input is maximized. Once all the layers are trained, the pre-trained model is finetuned via supervised backpropagation. It is important to note that layer-wise training helps with initializing weights and biases in the network prior to the actual supervised training. Taking classification as an example, a logistic classifier is used to classify the input based on the output of the final hidden layer of the DBN. A predefined error metric is computed between the class labels and the resultant output of the DBN (after applying the logistic classifier) and then the error is backpropagated down the network to further adjust and optimize the weights and biases.

**Visualization of Learned Features**

One of the main claims of a hierarchical semantic feature extraction tool such as DBN is that it learns meaningful patterns in the data that can signify the underlying characteristics of the process. Therefore, visualizing the learned features is crucial to both understand and verify the performance of the feature extractor. Furthermore, intermediate feature visualization may lead domain experts to scientific discoveries that are not easy to figure out via manual exploration of large volume of data.

For the lowest RBM layer, simply plotting the weight matrix may be sufficient to visualize the features learned by the first hidden layer. Since the dimensionality of the input and the weights are in the same order, the vectors of weights for each input can be reshaped into the dimension equal to the resolution of the input image. Thus, the visualizations are usually intelligible. Complexity arises for visualizing features learnt at deeper layers because they lie in a different space from the visible data space. At the same time, the dimension of weight matrix depends on the number of hidden units between the layer and the layer before. Thus, plotting the weight matrix will result in an incomprehensible visualization which typically resembles the appearance of white noise. To obtain filter-like representations of hidden units in the DBN, a recent technique known as Activation Maximization (AM) is used (Erhan, Courville, & Bengio, 2010). This technique seeks to find inputs that maximize the activation of a specific hidden unit in a particular layer and the technique is treated as an optimization problem. Let \( \theta \) denote the parameters of the network (weights and biases) and \( h_{ij}(\theta, x) \) be the value of the activation function \( h_{ij}(\cdot) \) (usually the logistic sigmoid function) of hidden unit \( i \) in layer \( j \) on input \( x \). Assuming the network has been trained, \( \theta \) remains constant. Therefore, the optimization process aims to find

\[
x^* = \arg\max_{x \in \mathcal{X}, |x| = \rho} h_{ij}(\theta, x)
\]

where \( x^* \) denotes the inputs that maximizes the hidden unit activation. Although the problem is a non-convex optimization problem, it is still useful to find the local optimum by...
performing a simple gradient ascent along the gradient of \( h_{ij}(\theta, x) \) because in many cases, the solutions after convergence are able to visualize the patterns of the inputs that are being learned by the hidden units.

3.2. Symbolic Time Series Analysis (STSA)

STSA is a fast time series feature extraction tool that models the temporal evolution of a quasi-stationary time series via symbolization (Ray, 2004). The algorithms of STSA are formulated via symbolization of the time series generated from dynamical systems along with subsequent state machine construction. First, the time series data are partitioned by maximum-entropy partitioning (MEP) (Rajagopalan & Ray, 2006) to construct the symbol alphabet \( \Sigma \) for generating symbol sequences. MEP maximizes the Shannon entropy (Cover & Thomas, 2006) of the symbol sequence via generating more partitions at the information-dense zones in the range domain than information-sparse zones. Once the partitions are obtained, each data point of the time series is assigned a symbol \( s_i \in \Sigma \) same as the partition it belongs to. Then, a D-Markov machine, based on the algebraic structure of probabilistic finite state automata (PFSA) (Ray, 2004), is constructed from the symbol sequence. D-Markov machine is defined as follows.

**Definition 3.1** (Ray, 2004; Sarkar et al., 2014) (D-Markov)

A D-Markov machine is a 4-tuple PFSA \( (K = (\Sigma, Q, \delta, \pi)) \), in which each state is represented by a finite history of \( D \) symbols as defined by:

- \( \Sigma \) is a non-empty finite set, called the symbol alphabet, with cardinality \( |\Sigma| < \infty \);
- \( Q \) is the finite set of states with cardinality \( |Q| \leq |\Sigma|^D \), i.e., the states are represented by equivalence classes of symbol strings of maximum length \( D \) where each symbol belongs to the alphabet \( \Sigma \); \( D \) is the depth of the Markov machine;
- \( \delta: Q \times \Sigma \rightarrow Q \) is the state transition function that satisfies the following condition if \( |Q| = |\Sigma|^D \), then there exist \( \alpha, \beta \in \Sigma \) and \( x \in \Sigma^* \) such that \( \delta(\alpha x, \beta) = x\beta \) and \( \alpha x, x\beta \in Q \);
- \( \tilde{\pi}: Q \times \Sigma \rightarrow [0, 1] \) is the symbol generation function (also called probability morph matrix) that satisfies the condition \( \sum_{\sigma \in \Sigma} \tilde{\pi}(q, \sigma) = 1 \) \( \forall q \in Q \), and \( \pi_{ij} \) is the probability of occurrence of a symbol \( \sigma_j \in \Sigma \) at the state \( q_i \in Q \).

State transition matrix, denoted by \( \Pi (\Pi \triangleq [\pi_{ij}], i = 1, 2, \cdots, |Q|, j = 1, 2, \cdots, |Q|) \), is obtained via combining \( \tilde{\pi} \) and \( \delta \). Each element of \( \Pi \), \( \pi_{ij} \) is the probability of moving from state \( q_i \) to \( q_j \) upon occurrence of a symbol at the next time step. In this paper, depth of the D-Markov machine is chosen to be one and it results in the equality of state transition matrix (II) and probability morph matrix \( \tilde{\pi} \). Depth greater than one can also be chosen via applying generalized D-Markov machine construction (Sarkar et al., 2014; Mukherjee & Ray, 2014). II is considered as the output feature of the D-Markov machine, which represents the time-series in reduced dimension. More details on STSA can be found in (Ray, 2004; Sarkar et al., 2014).

4. Results and Discussions

The DBN used for the study is comprised of three hidden layers with 1,000, 100, and 10 hidden units for the first, second, and third hidden layer respectively. The input image has a dimension of 56 × 98 pixels flattened to a 1 × 5488 row vector. The input image segments are taken from respective images at the flame entry (right end of the images) zone after scaling the original images down by 4 times.

4.1. DBN feature visualization

For visualization, the training set consists of 54,000 training images containing 6,000 images each from 9 conditions, 9,000 validation images containing 1,000 images each from 9 conditions, and 18,000 test images containing 2,000 images each from 9 conditions. A learning rate of 0.01 is used for the gradient descent algorithm for both pre-training and supervised finetuning. Pre-training is performed in batches of 50 samples and each layer undergoes 30 complete iterations of pre-training before moving onto the next layer. During supervised finetuning, classification errors on the validation images is compared against the error from training set as a measure to prevent overtraining the network and consequently overfitting the data. The optimized model is obtained prior to the point when the validation error becomes consistently higher than the training error in subsequent training iterations.

Figure 4 (d) shows the visualization of weights from the first layer with each tile representing a hidden unit in the layer immediately after pre-training. Values of weights connecting from all visible units to this single hidden unit are represented as pixel intensities. Panels (c), (b), and (a) visualize the input that maximizes the activation of the hidden units in the first, second, and third hidden layers respectively. As expected, the weights and the inputs that maximizes the activation of the first hidden layer are similar except that the pixel intensities are inverted. For higher layers, the network is able to capture the whole mushroom-shaped features from the input images. However, visualization for the third hidden layer (with only 10 hidden units) is not as clear due to the activation maximization algorithm converging to a non-ideal local optimum. A faint mushroom shape is still visible, however. In general, the pretrained model acquires a good representation of the input. Prominent features serving as the key to distinguishing between stable and unstable flames can clearly be seen in the visualized weight matrices.
In Figure 5, visualization of weights from the first layer and inputs that maximizes activations for all hidden layers after supervised finetuning are shown. An immediate difference can be clearly observed: visualized weights are now less noisy, whereas the third hidden layer is able to produce a visualization with more clarity compared to the weights prior to finetuning.

For both cases, the learning rate used in the AM algorithm is 0.01. Results have also indicated that depending on the initial value of the input vector, the resulting visualization from solving the optimization problem will be very different in terms of clarity. Thus, initial values of the input vectors are manually tuned by trial-and-error in order to obtain the best result. However, random initialization of the input vectors over a uniform distribution yielded undesirable results most of the time, showing images that are completely noisy without any perceivable features. Even if the results do converge, there are no significant differences between the solution from random initialization compared to the solution from tuning the initial values manually.

**Remark:** It is observed from the feature visualization that, though the DBN is trained on both stable and unstable flame images, the features gravitate more towards the coherent structure which is a characteristic of thermo-acoustic instability. An expert can use this feature visualization as an important tool to choose templates for unstable combustion, especially from the higher layer features. Those templates can be applied in post-processing of images to calculate the extent of instability via appropriate metrics that can effectively replace the age-old need for hand-crafted visual feature.

### 4.2. Performance of STSA module

In this subsection, DBN is pre-trained with 36,000 training images coming from 4 different operating conditions (see table 1) at partial premixing. Half of the training data is collected during stable combustion and other half during unstable combustion. Two sequences of images, consisting of one at stable ($Re = 7,971$, $FFR = 0.495g/s$ and full premixing) and another at unstable ($Re = 15,942$, $FFR = 0.495g/s$ and full premixing) combustion states, are reduced dimensionally via DBN with the parameters learned at pre-training phase. It is to be noted that, pre-training and testing of DBN are done on data at different levels of premixing to test the transferability of the proposed architecture.

Time series of $l_2$ norm of 10 dimensional activation probability vectors from each image are obtained as shown in figure 6(c) and (d). For comparison, $l_2$ norm of 10 largest variance components of those images, based on principal component analysis (PCA) (Bishop, 2006) coefficients learned on same training images, are constructed as presented in the top half of the figure 6. It is observed that the difference in textures of the $l_2$ time series between stable and unstable combustion is amplified in the case of DBN feature learning.

STSA is performed with increasing alphabet size on the $l_2$ time series that are mentioned above. Time series for stable and unstable combustion are partitioned separately via MEP and respective state transition matrices are calculated by the method explained in subsection 3.2. Euclidean distance between state transition matrices of stable and unstable combustion is a measure of class separability between those. The more the class separability the more would be the precision.
of detecting the intermediate states of the combustion while shifting from stable to unstable state. Therefore, this framework is better suited for early detection of onset of instability. It is presented in figure 7 that the class separability is much higher when STSA is applied on pre-trained DBN features than the PCA features. A probable rationale behind this observation is that, while PCA is averaging the image vector based on just maximum spatial variance, DBN is learning semantic features based on the coherent structures seen during unstable combustion. This rationale is also supported by the DBN feature visualizations that are shown in the subsection 4.1.

In a PHM context, the state transition matrix emerging from STSA module at the top can be used in supervised manner to detect instability from hi-speed image data. As the ‘DBN+STSA’ architecture provides a large class-separability between stable and unstable conditions, the state transition matrix can help in early prediction of thermo-acoustic instability. While the training of the proposed architecture is carried out offline in a GPU, the testing in a PHM application can be performed online with a processing power of a regular CPU. This is possible because the feed-forward computation of DBN along with STSA is feasible in real-time.
5. CONCLUSION AND FUTURE WORK

The paper proposes a framework that synergistically combines the recently introduced concepts of DBN and STSA for early detection of thermo-acoustic instability in gas turbine engines. Extensive set of experiments have been conducted on a swirl-stabilized combustor for validation of the proposed method. Sequences of hi-speed greyscale images are fed into a multi-layered DBN to model the fluctuating coherent structures in the flame, which are dominant during unstable combustion. DBN hidden layers along with bottom layer weight matrix are visualized via activation maximization method and mushroom-shaped vortex are demonstrated by higher layers after, both, pre-training and finetuning stages. Although visualization after fine tuning is less noisy, it may lead to overfitting due to limitation of the data volume. Therefore, an ensemble of time series data is constructed from sequence of images based on the $l_2$ norm of the activation probability vectors of last hidden layer at the DBN. Then, STSA is applied on the time series that is generated from an image sequence and ‘DBN pre-training+STSA’ is found to exhibit more class separability with varying alphabet size than ‘PCA+STSA’. More class separability between stable and unstable combustion implies more precision at detecting early onset of thermo-acoustic instability. In summary, while DBN captures the semantic features (i.e., coherent structures) of the combustion flames, STSA models the temporal fluctuation of those features at a reduced dimension.

One of the primary advantages of the proposed semantic dimensionality reduction (as opposed to abstract dimensionality reduction, e.g., using PCA) would be seamless involvement of domain experts into the data analytics framework for expert-guided data exploration activities. Developing novel use-cases in this context will be a key future work. Some other near-term research tasks are:

- Application of deep convolutional network on entire (large) flame images to model coherent structure at varying scales and orientations.
- Dynamically tracking multiple coherent structures in the flame to characterize the extent of instability.
- Multi-dimensional partitioning for direct usage of the last hidden layers for the sequence of images to the STSA module without converting it to time series of $l_2$ norm.

ACKNOWLEDGMENT

This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-12-1-0270. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agency.

REFERENCES


Railcar Bogie Performance Monitoring using Mutual Information and Support Vector Machines

Parham Shahidi¹, Daniel Maraini¹, Brad Hopkins¹, and Andrew Seidel¹

¹Amsted Rail Company, Inc. Chicago, IL, 60606, USA
pshahidi@amstedrail.com
dmaraini@amstedrail.com
bhopkins@amstedrail.com
aseidel@amstedrail.com

ABSTRACT
Railcar condition monitoring is an area of high importance and global relevance. The economic and safety concerns of equipment maintenance in North America mandate efforts in prognostics and health management. This paper presents the results from the development of a vibration based condition monitoring algorithm for freight rail, utilizing mutual information feature selection and support vector machine classification of bogie component faults. The algorithm is an implementation of a previously proposed railcar condition monitoring solution by the authors. The proposed monitoring solution is a data-driven method which was developed with measurements taken at a railroad test laboratory under controlled conditions. Vibration data was collected from multiple locations on a railcar over several test runs, each utilizing wheelsets with different levels of wear. The input of controlled wheel wear levels was aimed at varying the system outputs to resemble those of cars with different levels of mileage in revenue service. The generated data sets were processed and a feature set was extracted from the acceleration signals. The data was divided into training and validation partitions using a cross validation scheme to preserve the sequence for both sets. A mutual information (MI) estimation algorithm was used to rank the features based on their similarity to the classified fault state. Both the optimized feature set from the MI feature selection algorithm as well as the full, non-discriminate feature set were used as inputs to the support vector machine to assess classification accuracy. The results of this assessment are presented in the paper along with a presentation of the methods. The paper concludes with a proposal for a monitoring strategy aimed at specifically detecting faulty components and practicing predictive maintenance.

1. INTRODUCTION
The present work is motivated by a need in the freight rail industry to decrease asset maintenance related downtimes and to improve the effectiveness of maintenance schedules. The authors had previously investigated the viability of applying on-board condition monitoring and diagnostics methods to freight rail applications (Shahidi, Maraini, Hopkins, & Seidel, 2014) and had arrived at the conclusion that condition monitoring methods can significantly benefit the current state of railroad maintenance practices. The study of the authors was concerned only with the railcar. In particular, it was focused on the performance of the undercarriage, the bogie system, on which the railcar body traverses the rail network. Figure 1 shows a standard North American three-piece bogie.

Figure 1. Standard North American three-piece bogie

The focus of the present study remains on the bogie as this is the component of a freight rail car which experiences the most wear and is most susceptible to fault modes.

The trade association tasked with rule-making for railroad transportation, the Association of American Railroads (AAR), has established a set of performance metrics (AAR, 2007) which all bogies have to meet before they can be deployed in service. After they go into service, maintenance is performed either as fixed schedule preventive
maintenance or as reactive maintenance following alerts from wayside detectors. In the first case, maintenance downtimes are mostly avoided at the cost of unused capacity and premature component replacements. In the second case, wayside detectors, which are typically installed on the track, monitor passing railcars (Zakharov & Zharov, 2005). The two most common types of wayside detectors for rail car bogie performance are Truck Performance Detectors (TPD) and Truck Hunting Detectors (THD) 1. Both of these detectors consist of strain gage based instrumentation which is added to the track to measure the lateral and vertical forces that rail car wheels exert on the track. TPDs achieve this through instrumentation of two reverse curves with strain gages to measure the wheel lateral and vertical forces and wheelset angle of attack during curving. THDs use strain gages that are placed on tangent track to measure lateral wheelset oscillations. As of 2013, approximately 15 TPDs and 172 THDs were in service across the 140,000 miles of North American rail network. Other, even less common types of wayside equipment include Acoustics Bearing Detectors (ABD) and laser/vision-based systems. Although these systems are also installed wayside they are aimed at the detection of particular component malfunction vs. the bogie system’s performance as a whole. Deployment of these systems is in the low double digit numbers across the North American rail network. The small number of detectors relative to the large size of the US rail network makes it clear that wayside detectors do not provide sufficient coverage to comprehensively monitor freight train bogie performance.

2. BOGIE CONDITION MONITORING

On-board freight rail bogie condition monitoring is an area with large potential for research. As the name implies, the combination of multiple disciplines is the reason that few studies have been completed directly targeting the issue at hand.

First, on-board condition monitoring has historically not been applied to freight rail applications and is a new technology in the realm of freight rail maintenance. Typically, condition monitoring in the freight rail industry is achieved through wayside equipment and therefore research in this area has traditionally focused on efficiency improvements. Barke and Chiu (Barke, 2005) published a review of existing freight rail bogie condition monitoring technologies but excluded on-board methodologies and solely focused on wayside technologies. Lagnebäck also limited his study of potential cost savings and efficiency improvements through condition monitoring (Lagnebäck, 2007) to wayside techniques.

Second, most on-board condition monitoring studies have been attempted in the area of passenger rail transport (Ward, Goodall, Dixon, & Charles, 2010; Ward et al., 2011). Passenger rail bogies use complete and rigid frames and therefore do not have the issue of non-linearities from the friction based suspension elements of a three-piece bogie. However, passenger bogies still have to deal with other non-linearities such as those from the wheel-rail interface. The difficulty of modeling a friction wedge freight rail suspension was shown in Xia and True’s study to model nonlinear dry friction damping with hysteresis and stick-slip action in the friction forces on the contact surfaces of friction wedges (Xia & True, 2003).

Third, condition monitoring of freight rail applications is not limited to bogies and bogie suspension components only. Other areas of interest where significant work has been completed include the wheel-rail interface (Hubbard, Ward, Goodall, & Dixon, 2013), rail car speed inaccuracies due to stick-slip action (Mei & Li, 2008), end-of-car devices (Hopkins, Seidel, Maraini, & Shahidi, 2015) and on-board weighing (Maraini, Shahidi, Hopkins, & Seidel, 2014) applications. It is understandable that the emergence of on-board monitoring technologies and continuous improvements in accuracy lead to a vast scope of interest which includes monitoring strategies for components which have traditionally not been able to be monitored effectively.

With the high cost of both preventive and reactive maintenance, condition-based maintenance can be considered the best solution to the problem at hand. Typically, applications follow one of two paths: either that of model-based condition monitoring or that of data driven condition monitoring.

For model-based condition monitoring, a physics-based model, derived from first principles is used to determine required system parameters. In (Li & Goodall, 2004) a two degrees-of-freedom, half-vehicle model is developed, and simulated to determine parameter deviations. For the data driven case, features are extracted from existing data from field measurements and are then processed with machine learning techniques such as neural networks (Haykin & Network, 2004) and support vector machines (Bishop, 2006; Cortes & Vapnik, 1995) to identify fault modes from measurements.

In both cases, data is required to either compare against the model or else to feed into the machine learning algorithm. Typically, this data is taken from inertial sensors such as accelerometers mounted on the system under test. If prognostics is also part of the monitoring strategy, advanced filtering techniques such as particle filters (Arulampalam, Maskell, Gordon, & Clapp, 2002) or Kalman filters (Kalman, 1960) can be combined with the algorithm to estimate future states from the current state accelerometer measurements.

---

1 In the context of railroading and for this paper, the terms bogie and truck can be used interchangeably.
3. Field Test

Data collection was conducted at Transportation Technologies Center, Inc. (TTCI) in Pueblo, CO. TTCI is a transportation research and testing organization which offers a wide range of tests for rail applications. The facility has seven test tracks which are designed to induce a wide variety of fault conditions, including lateral and vertical railcar instability modes.

3.1. Field Test Setup

One of the test tracks at TTCI, the Railroad Test Track (RTT), is a 13.5-mile loop with four 50-minute curves and a single 1-degree, 15-minute reverse curve. The maximum speed on the RTT is 165 mph and all curves have 6-inches of superelevation (difference in rail height on the same section of track). The primary purpose of this track is high speed stability testing which is well suited for exciting lateral vehicle dynamic modes. The selection of lateral instability testing was based on the fact that the main drivers for this instability mode are the suspension parameters and wheel wear levels. Furthermore, increased car loads have resulted in wagon bodies with higher yaw/roll moments of inertia that under faulty suspension conditions can lead to coupled oscillatory resonance modes at speeds as low as 47 mph (Tournay, Wu, & Wilson, 2009). The constant increase in axle loads is certain to affect Mean-Time-To-Failure (MTTF) requirements, and as such poses a particularly well-suited example for an application of condition monitoring strategies.

For this study, one of the 50-minute (0.8 degree) curves with 6-inches of superelevation was used to accelerate the train to a target speed onto a tangent section of track. The target speeds ranged from 40 mph to 80 mph and were broken up into approximately 5 mph increments. Figure 2 shows the profile of the segment of the RTT track that was used.

![Figure 2. Test segment of RTT track](image)

The upper graph shows the superelevation and the bottom graph shows the curvature. Once the target speed was reached, data acquisition systems began to measure accelerations at multiple locations on the car body and suspension until the test ended. Test runs were aborted once either 80 mph or prescribed maximum acceleration limits per AAR rule MSRP C-II Chapter 11 were reached. The instrumentation setup included accelerometers with various dynamic ranges from ± 5 G to ± 200 G and gyroscopic sensors with rates of 250 °/sec. The sensor specifications were chosen to accommodate signal dynamic ranges that occurred in various measurement locations. A HBM Somat eDAQ rugged data acquisition system was used to acquire the data from the sensors with a sampling rate of 1000 Hz and aliasing protection through analog filtering. The setup was a modified version of the recommended setup from the MSRP C-II Chapter 11 rules, with slightly higher accelerometer bandwidths and dynamic ranges.

To test the system with known wear conditions as the input signals into the railcar system, wheels with three different levels of wear (new, intermediate and worn) were used. For each round of testing the wheelsets were swapped out for sets with a higher degree of wear. Figure 3 shows the three different wheel profiles that were used for the three rounds of testing.

![Figure 3. Different wheel wear profiles used as inputs](image)

Every other aspect of the railcar and bogies remained unchanged to ensure that the wheel profiles were the sole factors influencing the stability of the railcar.

3.2. Field Test Results

As mentioned before, each round of testing began with a different level of wheel wear at or below 40 mph and increased gradually until the prescribed maximum acceleration limit per AAR regulations or a test speed of 80 mph was reached. With these limitations, table 1 lists the speeds the rail car was tested with for each wheelset. The green measurements indicate the speeds for the test runs which remained within the AAR limits and the red test speeds indicate where the limits were exceeded.
Table 1. Test speeds [mph] for each wheel wear level

<table>
<thead>
<tr>
<th>No Wear</th>
<th>Medium Wear</th>
<th>Fully Worn</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>60</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>65</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>67</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>70</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>72</td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>75</td>
<td></td>
<td>70</td>
</tr>
</tbody>
</table>

Figure 4 shows the vibration signals for the 64 mph runs for each wheel wear level. Overlaid in red is the averaged signal of each time series signal.

![Vibration Signals](image)

It can be seen that the signals from figure 4 are reflective of the tabulated data. The vibration signal collected for the medium worn wheel (second subplot) has the highest vibration amplitude amongst the three signals. This corresponds with the test speeds from table 1, where the medium worn wheel set was run up to only 64 mph before the instable situation, pictured in figure 1, began.

4. ANALYSIS

The analysis of the acceleration data was broken down into multiple subtasks which will be explained in this section. The first task was the extraction of the feature set from the data for each wheel wear state and test run. Then, the data sets with the different wear states but same speeds were assembled in a random sequence as the test signal. This was followed by partitioning the assembled data sets into training and validation sets. The training set was used to reduce the dimensionality of the feature vector through a mutual information scheme which ranked the features and thereby allowed to exclude features with information gain below a user defined threshold. Then the reduced dimensionality training set was used to train a multiclass support vector machine. After training was complete, the validation set was used to evaluate the classification performance of the multiclass support vector machine in a one-versus-the-rest classification scheme.

4.1. Feature Identification and Extraction

In the first analysis step, a set of features had to be identified for extraction and identification of faulty instability modes. The initial feature set was identified as a combination of 14 features including the standard statistical moments, power content in various frequency bands, and spectral measures. The frequency bands were selected based on a qualitative spectrogram analysis in which the bands with the highest frequency content magnitude for faulty conditions were identified. In alignment with previous findings, the most important frequency band was chosen as the band between 2.5 and 3.5 Hz which is the typical range for the track-damaging rigid body rail car oscillation modes. The rigid body modes include yaw, roll, pitch and bounce oscillations which are mainly driven by wheel wear and bogie suspension wear. Since the rigid body oscillation modes exist with new components as well, only at lower frequencies and magnitudes, the first analysis frequency band was selected to be between DC and 5 Hz. Additional frequency bands included 7 – 12 Hz and 25 – 50 Hz. It shall be noted that these bands required frequent changes depending on which one of the measurement locations was chosen for analysis. Since a sampling rate of 1000Hz had been utilized for the testing, the usable bandwidth was from 0 to 500 Hz. The decision to use this bandwidth was based on knowledge of rigid and flexible modes of rail cars experiencing the mentioned oscillation modes. It was also observed that at elevated measurement locations on the carbody, higher frequency content, identifiable on suspension components, became attenuated. This is explicable through the carbody acting as a mechanical filter which attenuated much of the frequency content above 10 Hz.

Since each test run typically lasted longer than 60 seconds and included non-stationary dynamic behavior of the carbody, a windowing approach was selected to compute the feature sets. Multiple window lengths from 2 seconds for statistical features up to 10 seconds for spectral features were selected and incremented in one second intervals to compute the feature set. The complete list of all features is presented in table 2.
validation partitions was selected as approximately one tenth the length of the original set.

### 4.3. Feature Selection Using Mutual Information

In cases with very large feature sets, a means to find and select only the most relevant features for the classification task is required to improve computational efficiency. Mutual information theory is a frequently used feature selection algorithm to reduce the number of features. The idea is to compute a simple score $S(i)$ which measures how informative each feature $x_i$ is about the predefined class labels $y$. The information provided by the algorithm can be used then to discard the features with the least amount of relevancy. Mutual information uses the entropy as the amount of information gain provided by each feature. Entropy is defined as

$$ H(X) = \sum_x p(x) \cdot \log \frac{1}{p(x)} \quad (1) $$

where $p_i$ is the probability of an event taking place with a certain outcome. An approximation of $p_i$ can be obtained through the probability distribution since the algorithm is dealing with random continuous samples. The joint entropy of two events taking place together is defined as

$$ H(X, Y) = \sum_{x,y} p(x, y) \cdot \log \frac{1}{p(x,y)} \cdot (2) $$

Together these quantities can be combined to calculate the mutual information for each feature and the target class as

$$ I(X, Y) = H(X) + H(Y) - H(X, Y) \quad (3) $$

The result is a ranking of the features in the vector together with an information gain score for each feature. Table 3 shows the results for the test sequence of figure 5 and the mutual information based ranking of each feature at 65 mph.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mutual Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak to Peak</td>
<td>0.6768</td>
</tr>
<tr>
<td>Standard Dev.</td>
<td>0.6175</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>0.5839</td>
</tr>
<tr>
<td>Freq. Magnitude</td>
<td>0.4999</td>
</tr>
<tr>
<td>Hyperflatness</td>
<td>0.4905</td>
</tr>
<tr>
<td>Variance</td>
<td>0.4881</td>
</tr>
<tr>
<td>Band Power 1</td>
<td>0.4607</td>
</tr>
<tr>
<td>Fund. Frequency</td>
<td>0.4553</td>
</tr>
<tr>
<td>Band Power 3</td>
<td>0.4512</td>
</tr>
<tr>
<td>Skewness</td>
<td>0.4056</td>
</tr>
<tr>
<td>Band Power 2</td>
<td>0.3677</td>
</tr>
<tr>
<td>Hyperskewness</td>
<td>0.3622</td>
</tr>
<tr>
<td>Crest Factor</td>
<td>0.3066</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1688</td>
</tr>
</tbody>
</table>

---

**Table 2. List of Features**

<table>
<thead>
<tr>
<th>Feature #</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Band Power (1) 0-5 Hz</td>
</tr>
<tr>
<td>2</td>
<td>Band Power (2) 7-12 Hz</td>
</tr>
<tr>
<td>3</td>
<td>Band Power (3) 25-50 Hz</td>
</tr>
<tr>
<td>4</td>
<td>Magnitude at Fund. Frequency</td>
</tr>
<tr>
<td>5</td>
<td>Fundamental Frequency</td>
</tr>
<tr>
<td>6</td>
<td>Mean</td>
</tr>
<tr>
<td>7</td>
<td>Variance</td>
</tr>
<tr>
<td>8</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>9</td>
<td>Peak to Peak</td>
</tr>
<tr>
<td>10</td>
<td>Skewness</td>
</tr>
<tr>
<td>11</td>
<td>Kurtosis</td>
</tr>
<tr>
<td>12</td>
<td>Hyperskewness</td>
</tr>
<tr>
<td>13</td>
<td>Hyperflatness</td>
</tr>
<tr>
<td>14</td>
<td>Crest Factor</td>
</tr>
</tbody>
</table>

---

**Figure 5. Three-class label classification scheme**

A cross validation scheme was applied to the data to divide it into training and validation datasets. In prediction problems it is important to separate training and validation data to avoid overfitting and test generalization for independent datasets. The partitioning scheme was selected as a stratified hold out cross-validation which retained the proportions of the class labels for the training and validation partitions. Additionally, the scheme was reshuffled 10 times to provide additional validation data sets. The length of the

---

**Table 3. Mutual information ranking for 65 mph**
It should be noted that since a stratified partitioning scheme was used in the algorithm, the results may slightly differ each time the mutual information algorithm is executed. The reason for this is that for stratification, samples are chosen from the population in no specific order as long as the overall sequence of class labels is maintained. Therefore single values can still vary under the same label and the variation this introduces may influence the probability distribution of the entropy calculation. A threshold can be applied after the ranking to exclude features with an information gain below a desired limit.

4.4. Multiclass Support Vector Machine Classification

A Support Vector Machine (SVM) is a maximum margin classifier that can be used for classifying both separable and non-separable data. This is achieved by finding an optimal hyperplane which defines the maximum margin between two target classes. When the target classes are separable, the equation for the hyperplane is straightforward. However, for non-separable data, kernel based methods must be utilized to transform the data into a space whereby it becomes separable. In the case of only two indicators for each class this is a simple linear line which separates two classes of data. However, when data with more than two features is to be separated the simple line becomes a plane or hyperplane above 3 dimensions. At its core, the classification problem is defined as the decision rule

\[ y(u) = w^T u + b \]  

where \( y(u) \) is the decision, \( w \) a weight vector orthogonal to the decision surface, \( b \) a bias and \( u \) an unknown input vector. The optimal hyperplane can be found by solving the constrained optimization problem of the form

\[ \min \frac{1}{2} \|w\|^2 \]  

Limited by the constraint

\[ t_i(w^T x_i + b) \geq 1 \]  

For (6), \( x_i \) represents known positive or negative training samples and \( t_i \in \{-1,1\} \) is a factor that is either positive negative depending on the sign of \( x_i \) so that (6) is always true. To deal with the constraints, we introduce Lagrangian multipliers \( \alpha_i \) to find the extremum of equation (5). The Langrangian which combines (5) with the constraints from (6) can be expressed as

\[ L = \frac{1}{2} \|w\|^2 - \sum \alpha_i [t_i(w^T x_i + b) - 1] \]  

Taking the derivative and setting it to zero gives the conditions for the extremum. Those can be plugged back into the original decision rule for a two-class classification problem of the form

\[ y(u) = \sum \alpha_i t_i x_i^T u + b \]  

The vectors in the dot product in equation (8) can be transformed for cases when the classes are not linearly separable and in turn make them separable again. This is achieved using a kernel function of the form

\[ \phi(x^T) \phi(u) = k(x,u) \]  

For the present study all tests were conducted with a linear kernel, meaning the dot product was used.

The support vector machine is fundamentally a two-class classifier. To deal with the fact that in this case the problem is not only a two-class separation problem but a three-class problem \( y \in \{1,2,3\} \) with one class for each wheel wear state, the above introduced support vector machine was modified to be a multiclass support vector machine. A common approach for this is called the one-versus-the-rest approach which constructs \( K \) separate SVMs in which the \( k^\text{th} \) model \( Y_k(x) \) is trained using the data from class \( y_k \) as the positive examples and the data from the remaining \( K-1 \) classes as the negative examples.

4.5. Analysis Results

The analysis was completed with the above outlined algorithm and data from the field test. The focus of this testing was on identifying the three wheel wear states while testing for robustness of the algorithm against railcar speed and assessing which features contribute most to the accuracy through the mutual information score of each feature.

Figure 6 shows the progression of the features vs the speed for each wheel wear level. The colors in figure 6 were chose in accordance with the colors of the wheel profiles in figure 3: green stands for the no wear wheel profile, blue for the medium wear wheel profile and red for the full wear wheel profile. As presented in table 1, due to the experimental nature of the field data, the data sets for each fault were not always recorded at the exactly same speeds. Hence, the features are also only available at the same speeds (as in table 1). Since for comparison purposes the speed has to be the same for each fault, only three speed levels (65, 60 and 50 Mph), at which data was available for the three faults, were selected for analysis.

For the first case, data from the 65 mph test run for each wheel wear state was used to evaluate classification accuracy of the algorithm. The sequence of the wheel wear levels remained the same as presented in figure 5 in 4.2 and the hold out cross validation scheme was reshuffled 10 times for 10 simulations with the multiclass support vector machine. Features were ranked but none were excluded for the first case. The first simulation of the first case (65 mph) produced a classification accuracy of 93 %.
Figure 6. Progression of features versus speed – green stands for the no wear, blue for the medium wear and red for the full wear wheel profile.

Table 4 shows the results in a confusion matrix. As the table shows, only 7 out of 100 samples were incorrectly classified.

Table 4. Confusion Matrix for 65 mph run

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Wear</td>
<td>29</td>
</tr>
<tr>
<td>Med. Wear</td>
<td>0</td>
</tr>
<tr>
<td>Full Wear</td>
<td>1</td>
</tr>
</tbody>
</table>

The next 10 simulations yielded similar accuracies and the results are presented in table 5. The average classification accuracy for the first case with speeds at 65 mph was 92%.

For the second case, the same classification runs were repeated for a test speed of 60 mph. The first simulation produced a dramatically decreased classification accuracy of 81%. Table 6 shows results in the confusion matrix.

Table 6. Confusion Matrix for 60 mph run

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Wear</td>
<td>30</td>
</tr>
<tr>
<td>Med. Wear</td>
<td>31</td>
</tr>
<tr>
<td>Full Wear</td>
<td>20</td>
</tr>
</tbody>
</table>

It can be observed that the majority of the incorrect classifications happened for the full wear class label being incorrectly classified as medium worn. The next 10 simulations yielded accuracies as presented in table 7. The average classification accuracy for the second case at a test speed of 60 mph was 76%.

Table 7. Classification accuracies for 60 mph

<table>
<thead>
<tr>
<th>Simulation #</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.73</td>
</tr>
<tr>
<td>2</td>
<td>0.73</td>
</tr>
<tr>
<td>3</td>
<td>0.74</td>
</tr>
<tr>
<td>4</td>
<td>0.77</td>
</tr>
<tr>
<td>5</td>
<td>0.79</td>
</tr>
<tr>
<td>6</td>
<td>0.77</td>
</tr>
<tr>
<td>7</td>
<td>0.75</td>
</tr>
<tr>
<td>8</td>
<td>0.77</td>
</tr>
<tr>
<td>9</td>
<td>0.78</td>
</tr>
<tr>
<td>10</td>
<td>0.76</td>
</tr>
</tbody>
</table>
For the last case, the same simulations were run for a test speed of 50 mph. The first simulation produced an accuracy of 79%. Table 8 shows the results in the confusion matrix. It can be observed again that class label “Full Wear” created the most inaccurate classification events out of the three class labels.

Table 8. Confusion Matrix for 50 mph run

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
<th>No Wear</th>
<th>Med. Wear</th>
<th>Fully worn</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Wear</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Med.Wear</td>
<td>0</td>
<td>28</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Full Wear</td>
<td>0</td>
<td>14</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

The accuracies of the following reshuffled classifications are presented in table 9 below. The average classification accuracy at 50 mph was 79% which was just slightly above that of the 60 mph simulations.

Table 9. Classification accuracies for 50 mph

<table>
<thead>
<tr>
<th>Table 9. Classification accuracies for 50 mph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation #</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Next, the feature rankings were added to the analysis and a minimum information gain threshold for the features to be included in the analysis was enforced. The data from the three test speeds was tested for mutual information thresholds that ranged from 0.5 to 0 where 0 meant that all features will be included for classification. Each speed and mutual information threshold was evaluated for 10 simulations and the approximate number of selected features of the 10 simulations as well the average accuracy with the selected feature set was recorded. The results are presented in table 10. A number of relationships can be observed in the results: first, the speed has a large influence on the number of selected features for each MI cutoff. This can be explained by virtue of the fact that the faster the train moves on the track, the more reflective of the fault condition the features become and hence higher mutual information between the features and classes exist. Second, of course a clear trend towards higher accuracy with more features can be observed for each of the test speeds. In the case of this study, this is not surprising since not a large number of features were used and no contradicting feature trends, which would require exclusion of features, existed.

Table 10. Results with applied feature selection

<table>
<thead>
<tr>
<th>Table 10. Results with applied feature selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed [mph]</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>65</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

As an example, the features excluded by the MI feature ranking algorithm for low cut offs typically include the mean, crest factor and hyper skewness. However, it must be noted that this should be only cautiously considered as representative, since the stratified partitioning scheme here too causes a level of variation as explained in section 4.3.

5. DISCUSSION

Analysis for the detection of wheel wear states from the vibration signature of acceleration data taken on the rail car was completed. A success rate of 92% was achieved for the ideal case of high test speeds. Particularly, the “1” class label achieved ideal classification accuracy which can be interpreted that the identification of normal operation would be most reliable in an implementation. Performance of the algorithm at lower speeds was worse but still acceptable and of high value with a success rate of approximately 80% classification accuracy.

A few interesting points emerged from the analysis which require a deeper discussion. The first and most important observation was that speed influences the classification accuracy. In the case of this testing the test runs were completed with incrementally increasing test speeds until failure occurred. Failure was considered a lateral instability mode which was more likely to occur with worn wheels than with new wheels at a certain speed. This instability mode, which is linked to wheel wear and entered by the train above a certain speed, creates high amplitude lateral oscillations which severely change the vibration characteristics of the acceleration signal. Therefore it is not
surprising that the directly linked wheel wear state was clearly discernible at higher speeds in the analysis. The simulations supported this conclusion with an average classification accuracy of 92% at 65 mph.

In the second set of simulations with reduced test speeds of approximately 60 mph, the average accuracy dropped to 76%. The plot in figure 6 shows that majority of the misclassification occurred for the samples labeled “3” in the test set. These data samples were typically incorrectly classified as having the label “2”. Conversely, the opposite misclassification of label “2” values as label “3” values did not occur, which leads to the question of why these unidirectional classification inaccuracies occur. One explanation may be that at speeds below 65 mph which do not excite the lateral instability mode, distinction between wear states of medium to fully worn states can be a challenge. Interestingly, this trend does not prevail when the speed is further reduced to 50 mph. Although speeds lower than 65 mph had significantly lower classification accuracies, the 50 mph had higher classification accuracy than 60 mph. This non-linear dependency on speed remains subject to further research. However, in preliminary experiments it has been discovered that by implementing support vector machines with kernels that are more sophisticated such as polynomial or Gaussian radial basis function kernels, improved accuracies can be achieved. Another aspect which will require future work is the addition of a probability score for the one-versus-the-rest multiclass support vector machine. Without a probability score samples may be assigned to multiple states and training sets will be imbalanced. In the above example, the test set had a length of 100 samples but less than one third belong to each class therefore giving the rest class label an undue overweight and loss of symmetry may occur. Additional techniques for multiclass support vector machines shall be explored as alternatives.

Other improvements for future iterations of the algorithm include the addition and grouping of features, such as ratios and derivatives. Furthermore, the investigation of additional measurement locations and the addition of more components for wear estimation can provide insight into additional failure modes. Finally, an extension of the algorithm to also include non-parametric data into the feature set will further be able to enhance classification accuracy.

6. Conclusion

On-board condition-based maintenance for North American freight rail applications is an underdeveloped yet promising field for the application of condition monitoring and machine learning techniques. Past efforts were mainly focused on passenger rail and wayside detection technologies. In this study an algorithm to estimate wear levels of freight rail bogie components based on mutual information and multiclass support vector machines was developed and tested with field data. Promising results were achieved with classification accuracy above 90% for test speeds which excite relevant failure modes. Lower speeds still yield an accuracy of approximately 80% with the full feature set. Very high potential for improved results in the future exists based on the proposed improvements for the algorithm and the expansion of test locations. For the algorithm, feature set extension and improved kernel function will most likely yield the highest improvements.

References


**Biographies**

**Parham Shahidi** is a Project Engineer in the Bogie Systems Engineering group at Amsted Rail Company, Inc. He holds a BS degree in Mechanical & Process Engineering from TU Darmstadt in Germany and a PhD in Mechanical Engineering from Virginia Tech. Parham has been working in the area of condition monitoring since 2007, starting with the development of a speech based fatigue estimation system for train conductors in graduate school. In 2011 he joined Amsted Rail where his major projects in the area of health monitoring include the development of systems for vehicle instability detection, and bearing condition monitoring. He has published more than 10 technical articles related to his research. He is also an adjunct professor at Virginia Commonwealth University’s Mechanical Engineering department and a board member of the German Engineers Association (VDI) in North America.

**Daniel Maraini** is the Manager of Bogie Systems Engineering for Amsted Rail Company, Inc. He leads a team of Project Engineers focusing on condition monitoring solutions for freight railcars. Daniel has been a part of Amsted Rail since 2008. In various positions, he has worked on wireless sensing, remote asset monitoring, and condition monitoring for freight rail applications. Daniel is currently a PhD student in the department of Mechanical Engineering at Villanova University. His research is in the field of machinery diagnostics, with a focus on nonlinear model based techniques for rolling element bearings. He holds a MS in Mechanical Engineering from Villanova University and a BS in Physics from West Chester University of PA.

**Brad Hopkins** is a Project Engineer in the Bogie Systems Condition Monitoring group at Amsted Rail and a Lecturer in the Mechanical Engineering Department at Southern Illinois University at Edwardsville. He holds a BS, MS, and PhD in Mechanical Engineering from Virginia Tech. He has been working on track and railcar condition monitoring since 2010, with a focus on accelerometer-based monitoring and algorithm development. His current work includes broken rail detection, wheel defect monitoring, and end-of-car system monitoring. His additional research interests are vehicle dynamics, modeling, and simulation, controls, and vibration analysis.

**Andrew Seidel** is a Project Engineer in the Bogie Systems Engineering group at Amsted Rail Company, Inc. He holds a BS degree in Mechanical Engineering from Pennsylvania State University in State College, Pennsylvania. Andrew has been working in the area of railroad engineering since 2012 when he began working with end of car impact detection systems and railcar condition monitoring projects. He is a member of the American Society of Mechanical Engineers (ASME) and sits on the Mechanical Engineering Technical Advisory Board for Penn State University (Harrisburg campus).
Neural Network-Based Gear Failure Prediction in a Brushless DC Actuation System

Yan Chen¹, Christina Lucas², Jay Lee² and Michael Buehner³

¹,²School of Dynamic Systems, University of Cincinnati, USA
chen2ym@mail.uc.edu
lucasci@mail.uc.edu
jay.lee@uc.edu

³ Control System, Woodward Inc. Fort Collins, USA
Michael.Buehner@woodward.com

ABSTRACT

Due to its inherent efficiency and reliability, brushless DC (BLDC) driven actuation systems are widely used in a variety of industries such as aerospace, electric transportation and industrial positioning. However, it is inevitable that various types of faults can develop in the actuator either from the BLDC motor or geared positioning systems. This paper, focusing on actuator load positioning system failures, proposes a data-driven based failure prediction method. Run-to-failure data is first collected from test-beds of specific BLDC actuation systems and then critical features representing system performance are extracted. There are also dynamic behavioral tests used, which are designed to provide discrete measurements reflecting system health conditions. Ultimately, based on optimized mapping between the two groups of information, a general neural network model is developed to establish a nonlinear trajectory model for failure progression. The model also allows for prediction of gear failure without the interruption of performing dynamic behavioral tests during continuous working condition. This approach provides for real time monitoring of system behavior as well as possibility of the predicting the Remaining Useful Life (RUL) of the actuation system. Although many efforts have been done to predict gear wear based on vibration signal, the proposed method is formulated within a "sensor-less" environment and makes full use of existing on-board sensing information, which provides the possibility of a closed-loop control system for life management.

1. INTRODUCTION

Yan Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
sensing and inflexible end life, this study attempts to estimate gear failure degradation process of an actuator using only output position and motor stator current signals and predict its progress ahead.

The study in [2-5] mainly introduced and further reviewed both CBM or model based motor fault detection, identification and diagnosis methods. Many faults from both internal motor and external load system are studied in terms of their complex signatures in different signals and their combination. The problem of distinguishing between faults with the same fault signatures is also addressed. Researchers in [6] proposed a data-driven methodology for the remaining useful life prediction of a jet engine actuator system. The method utilizes the control system including hydraulic pressure, current and position to create a classification model, which can identify actuators state of health. For prognostic analysis, targeted at valve health, Kalman filter is used as a tracking or trending algorithm to model the failure progression. The valve health is estimated as a hidden state. Researchers in [7] propose a method to predict the failure state of starter motor gear engagement using Hidden Markov Models (HMMs). They use time–frequency features extracted from the motor current and methods for computing the parameters from limited data are presented.

Similarly, in this study, the real gear wear is not directly measured. A series of dynamic tests are conducted to evaluate gear condition with standard control signal. The failure progression modeling is formulated as a mapping issue between discrete failure "measurement" and real time observation. The position error waveform signatures quantify the failure and the real time observation are representative features of the system performance, which are extracted from state current and position error frequency signature. These features are processed to be more generic avoiding the influence of individual unit setup position. The neural network method is used to track the failure progression and perform the failure and remaining useful life prediction.

The organization of the paper is as follows. In section II, we first introduced experiment test-bed and then overview the proposed approach step-by-step. All of the extracted features are introduced in section IV as well as feature normalization methods. In section V, the failure quantification method is introduced. Finally, the General Neural Network method based failure progression method is introduced in section VI and RUL prediction is validated by two more testing units in section VII. This paper also introduces how to tackle mapping issues between discrete failure measurement and continuous feature variable.

2. ACTUATOR ACCELERATED TESTING

The test bed set up within Woodward Inc. to run three BLDC electric actuators to failure is seen in Fig. 1. The actuators are used in natural gas and diesel engines and are expected to achieve a life goal of 30,000 hours. For the test, springs were attached on the output shafts to intensify the working condition. This test bed had the first actuator, Unit 1, with on-board electronics, the second actuator, Unit 2, had of the electronics off-board and the third actuator, Unit 3, had similar configuration as the Unit 1 but with oil lubrication. Unit 1 was run to complete failure, while Unit 2 was terminated shortly after Unit 1, for it was needed for another project.

The following subsections describe the two types of tests that were conducted on the actuators: endurance testing and dynamic testing.

2.1. Endurance test

In order to run the actuators to failure in a realistic window of time for data acquisition, endurance testing was performed in which the actuators were running constantly in a small range of full travel and five times faster at 10 Hz than normal conditions at 2 Hz. In normal conditions, the actuator would be shut down once per day, at this faster pace it was necessary to have the actuators automatically shut down every 5 hours. This shutdown allowed the spring to drive the actuator to a mechanical stop and also redistributed the grease on the gears. In order to keep dataset size to a minimum, only ten minutes of data were collected daily from the actuators at a 1 KHz sampling rate. The various types of data signals collected were almost identical on the units with the exception of no temperature reading from the actuator with off-board electronics.

2.2. Dynamic testing

In addition to the endurance test, dynamic tests were also performed on the actuators on a week-to-week basis. During the test, gear positioning system will run in full travel range under steady speed, which will allow inspection of all of gear teeth in motor pinion gear and frequent engaged teeth in drive train gears. The dynamic tests consisted of various signal inputs such as: steps, ramps, sweeps and steady state positions. Fig. 3 shows the raw transmission error, or difference between the shaft and motor positions, with the
ramp input for all seven days the dynamic test was performed on Unit 1.

As seen in Fig. 3. With baseline failure degradation model, system failure condition can be predicted without interrupting continuous work for dynamic tests; furthermore, after translating fault measurement to design actuator life information, the RUL prediction can be realized.

4. FEATURE EXTRACTION

4.1. Transmission error

The transmission error (TE) of a gear, gear set, or of a complete gearbox is defined as the deviation between the theoretical angular position of the driven gear (output) and its actual position, when driving the input at a constant steady rotation. In this study, TE is defined as the deviation between gearbox driven motor position and the output gearbox shaft position. The absolute transmission error is also partially adjusted by the feedback control mechanism, because when the gearbox is degrading, the feedback position signal will try to compensate the loss by demanding more current.

\[ TE = \text{Shaft Position} - \text{Motor Position} \]

As able to clearly observe the position error caused over different rotational angles, the transmission error spectrum is calculated over time against position of motor shaft (as Fig. 4 shows). According to endurance tests, the motor is setup to run within one revolution (44 rad-38 rad), however the travel range has been changed twice at the end of actuator life. Before the changes, the transmission error variance at certain position of motor shows gradually increasing. This becomes a good indicator to actuator gear system performance. Therefore, average TE variance over motor drive range (in Fig. 5) is calculated as one of the critical features.

Instead of focusing on the absolute value of TE and motor or shaft position, the frequency domain features, which are directly related to motor and shaft rotation variations, are more concerned. Fig. 6 shows shaft position feature in frequency domain. Because the actuator is operated at 10hz and every stroke movement includes the forward and backward actions, the 20hz becomes signature frequency to indicate vibration increasing caused by friction and backlash. Apparently, Unit 1 has a more dramatic degradation pattern starting about one-third of the way into the test.

4.2. Frequency Response of gear system

Faults such as gear wear caused friction or nonlinear backlash in actuator positioning system can be considered as system instinct characteristic change. The variation of total backlash in the gear system will affect system frequency response characteristics. Therefore, in the study, the feature of power distributed at the low frequency range (around 20
hz) is extracted from the frequency response spectrum with input motor position and output shaft position to indicate system performance change as shown in Fig. 7.

\[ T_L \approx T_{em} = k_T I \]  

(1)

Therefore, ideally, if the motor itself was considered under good condition, a stator current spectral analysis could indicate localized fault from gears in load system, particularly for the motor pinion and driving wheel gear.

In the dynamic testing, the ramp signal is fed into the system, which leads to steady motor rotation. The Welch Power density spectrum for the phase A stator current and its transformation in d-q coordination are shown in Fig. 8. The dynamic test is run at a 1 Hz frequency and the number of teeth on the motor pinion is 14, therefore the motor pinion gear mesh frequency can be seen around 14 Hz.

4.4. Feature Calibration and Normalization

In real utilization, every unit might be subjected to random emerging conditions. One of frequent condition changes comes from the demanding or input signal, material and supply power. During machining, produced parts might be slightly different because of variation in the input material. Actuators performance might also be changing due to drifting power supply and other thermal effects. As Fig. 8 shows, during endurance the test, Unit 2's current sensor suffers a high temperature meltdown. After replacing, the supply voltage has an increase, which caused an obvious change in the current signals. This variation has affected all of current related features.
As to recover the trendability of the system degradation reserved in the significant current features, a linear feature calibration technique is used to adjust feature values after the voltage increase as shown in Figure 9.

![Figure 9. Current frequency feature and its calibration](image)

Besides input variation, individual units will perform slightly different due to mechanical configuration variation or others. It is frequently observed that units start with different initial values; therefore the normalization step would be necessary to make different units compatible. Hence, every value in the feature matrix is normalized by comparing them with the starting baseline value of each feature as (3) shows.

$$f(x_{ij}) = \frac{f(x_{ij}) - M_i}{M_j}$$  

In the equation, $i$ is the instance index and $j$ is feature index. $M_j$ is the mean value of the first 5 samples in every feature.

### 5. FAILURE MEASUREMENT

During the dynamic test, with the ramp signal input to the system, motor is designed to run around 12 resolutions (from 3rad to 88rad) at steady speed. The output shaft finishes one stroke movement and is driven by geared load system that engaged with motor pinion. Since daily endurance tests always excise the system at a smaller range of one working cycle (one output shaft stroke), therefore, the frequent engaged motor pinion or driven train gear teeth are worn early and the same engagement position will have an apparent increase in transmission error compared with other position in dynamic test. As Fig.10 shows, at around 42 rad of every working cycle, compared with other position, the TE increases. In Fig. 10, the depth and time duration of the TE drop increases for each dynamic data set as depicted in the various colors. In Unit 2, because the calibration for the position was set opposite of Unit 1, the TE has an increased instead of decreased spike around the similar position. In this study, the depth of the drop in Unit 1 and the height of the spike in Unit 2 are used to qualify failure of the gear wear. From the Figure 10, it is obvious that Unit 1 has more severe degradation than Unit 2.

Compared with endurance tests that excise the actuator in a daily base and provide performance features, dynamic test is design to benchmark system failure. However, the dynamic tests are conducted less frequent than endurance tests, the size of samples of fault measurement are only 6, compared with around 60 samples from endurance test totally. The system performance and fault measurement mapping cannot be realized unless both the target and predictor variables have a compatible size of samples. Therefore, a curve fitting method is used to generate reasonable fault measurement for each performance sample. The fault measurement from initial dynamic test for Unit 2, before all of endurance tests, is taken as reference for all of Unit 2 fault measurements. The curve fitting result is shown in Fig. 11.

![Figure 10. TE in dynamic test with ramp input](image)

![Figure 11. Curve fitting for gear wear](image)

### 6. SYSTEM DEGRADATION MODELING AND PREDICTION

After evaluating features based on system knowledge and visualization of the trendability, four features, current frequency, shaft position in 20 hz, positioning frequency response and TE average variation over travel range, are extracted from endurance tests to formulate the data pattern of actuator system degradation. Consequently, the system degradation modeling issue becomes an optimized mapping...
issue between endurance test features \((x_i)\) and failure measurement \((y_i)\). 
\[
\min_y \sum_{t=1}^{T} L(f(x_1, x_2, \ldots, x_n), y_t)
\]  
(4)
In equation (4), \(\{x_1, x_2, \ldots, x_n\}\) is performance features. Before the mapping, the features are first converted into Self-organizing map—minimum quantization error (MQE) values, that represent system performance health index.

6.1. Health index of actuator load system
Minimum Quantization Error (MQE) is a method of applying Self-organizing map (SOM) to measure system performance change by calculating the distance of current data with the normal baseline. The normal data are trained as the SOM map and the testing feature vector is compared with the weight vector of the all map units. The minimum distance between the new feature sample and the BMUs are used to quantify the health levels. Here, the baseline is selected from initial testing of unit 2. The MQE value of three unit test are calculated as shown in Fig. 12

SOM algorithm is a one layer neural network model. It also can be used to map high-dimensional data to a lower dimensional grid and convert the nonlinear relationship of the dataset into simple geometric distribution and then visualize it on a distance map [10]. During the map training iteration, the best matching units (BMU) are the neurons that have the closest distance to the input vectors.

\[
MQE = \min \|D - w_{BMU}\|
\]  
(6)

Figure 12. MQE value
Since only part of fault measurement samples are from the real dynamic test not curve fitting results, the original degradation modeling issue becomes a semi-supervised mapping optimization issue [9]. If it’s assumed that \(Z\) is the subset of predictor variable samples in time series which contains the real measurement values:
\[
\min_{f,Y} \sum_{t=1}^{T} L(f(mqe_t), y_t) + \lambda_z \sum_{t=Z} L(f(x_t), z_t)
\]  
(5)
As mentioned in [9], there are two items added in the previous loss function (4). The first added item allows the points with real measurement value to have strong influence. The second item is to favor the progressing sequence of the \(Y\) measurement, adhering to prior knowledge; however, in this study, since the curve fitting results have shown promising goodness of fit, the influence of last term is ignored as equation (5) shows.

6.2. GRNN based degradation modeling
General regression neural network (GRNN) is used to map relationship between system performance health index and the off-board physical fault measurement. It also predicts system degradation. GRNN [11] is a one-pass memory-based neural network. It does not require an iterative training procedures like back propagation networks. It can be used for any regression problems without constrain on linearity. It approximates any arbitrary function between input and output vectors. As the training set size increases, the estimation error approaches zero [12]. A GRNN consists of an input layer, pattern layer, summation layer and output layer. During prediction, the predicted value \(y\), according to a unknown input vector \(x\), is:
\[
y = \frac{\sum_{i=1}^{N} y_i \exp(-D(x, x_i))}{\sum_{i=1}^{N} \exp(-D(x, x_i))}
\]  
(6)

Based on (6), the optimization (5) is designed to find best spread of the radial basis function \(s\), which is a parameter contained in \(D(x, x_i)\), with least sum square error (SSE). It proved that when the spread equaled to 0.00031, the loss function of SSE value reached a global minimum.
Using this approach, an actuator degradation model can be established. Based on the model, given health value of MQE, the actuator degradation can also be predicted, for example, Fig. 13 shows prediction results for all unit 2 mqe value. In Fig. 14, the actual failure measurement of unit 2 from dynamic test is compared with predicted value. For unit 1 the first actual failure measurement already reaches 0.002. Based on the degradation model, unit 1’s 28th cycle’s mqe value is the one corresponding to the failure level.

Figure 13 GRNN degradation model based on unit 2
7. Remaining Useful Life Prediction

Among all of testing units, the unit 2 and unit 1 are finally disassembled after all of tests. Unit 2 proved to have 0.003 inch gear wear at motor pinion gear after 69 running cycle, which is 25% of remaining useful life left according to the product design. Unit 1 proved to fail much early than unit 2 and run further than accepted life term. At the end one tooth even fell. Unit 3, has special oil filter system and no failure symptom when the test is stopped. Therefore, all fault measurements of unit 2 are linearly translated into remaining useful life as shown in Fig. 15. In the model only half of samples are used to train the model and the prediction MSE for all of samples is 0.009.

If assume the unit 2 as the baseline model as mentioned above, RUL of both unit 1 and unit 2 are predicted as Fig 16 shows. The prediction results are consistent with the real condition. Unit 1 is predicted to fail after 31 cycle. Unit 3 just consumed less than 20% of its life after 83 cycles.

8. Conclusion and Future Work

The research discussed in this paper strives to predict gear failure and RUL in BLDC electric actuators using existing feedback sensors rather than adding cost with the inclusion of accelerometers. Through the implementation of a test bed within Woodward Inc., the proposed method is able to use run-to-failure endurance data to extract useful information from only current and position signals about the gear wear within the actuators.

A neural network based optimized mapping method is develop to model generic gear failure progression in the BLDC actuator system, which also lead to the prediction of gear failure before it reaches severe damage. The proposed method also made efforts to compensate for the semi-supervised mapping issue caused by "incomplete" discrete failure measurement samples compared to "complete" high sampling rate process observation features.

In this research, one actuator (Unit2) was used to develop the failure progression model and another two similar units were used to validate the method. Although each unit has different control mechanism and random operation variation, for similar gear failure issues, the proposed method is able to compensate the difference by extracting critical features to represent the failure progression, by calibrating features based on every unit initial condition, and reducing feature dimension by MQE based method.

Using a general regression neural network, the paper successfully demonstrated the ability to predict the actuator gear failure progression and caused RUL. In the future, other regression modeling method will be benchmarked and this method will be improved for compatibility using run-to-failure endurance data from more actuators of the different type.

Acknowledgments

Authors would like to thank and acknowledge Woodward Inc. for supporting this research. The support from NSF I/UCR Center for Intelligence Maintenance Systems in the University of Cincinnati is gratefully acknowledged.

References


Verification of a Remaining Flying Time Prediction System for Small Electric Aircraft

Edward F. Hogge 1, Brian M. Bole 2, Sixto L. Vazquez 3, José R. Celaya 4, Thomas H. Strom 5, Boyd L. Hill 6, Kyle M. Smalling 7, and Cuong C. Quach 8

1,5,7 Northrop Grumman Technical Services, NASA Langley Research Center, Hampton, Virginia 23681
edward.f.hogge@nasa.gov
thomas.h.strom@nasa.gov
kyle.m.smalling@nasa.gov

2,4 Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, Moffett Field, California 94035
brian.m.bole@nasa.gov
jose.r.celaya@nasa.gov

3,8 NASA Langley Research Center, Hampton, Virginia 23681
sixto.l.vazquez@nasa.gov
cuong.c.quach@nasa.gov

boyd.l.hill@nasa.gov

ABSTRACT

This paper addresses the problem of building trust in online predictions of a battery powered aircraft’s remaining available flying time. A set of ground tests is described that make use of a small unmanned aerial vehicle to verify the performance of remaining flying time predictions. The algorithm verification procedure described here uses a fully functional vehicle that is restrained to a platform for repeated run-to-functional-failure experiments. The vehicle under test is commanded to follow a predefined propeller RPM profile in order to create battery demand profiles similar to those expected in flight. The fully integrated aircraft is repeatedly operated until the charge stored in powertrain batteries falls below a specified lower-limit. The time at which the lower-limit on battery charge is crossed is then used to measure the accuracy of remaining flying time predictions. Accuracy requirements are considered in this paper for an alarm that warns operators when remaining flying time is estimated to fall below a specified threshold.

1. INTRODUCTION

Improvements in battery storage capacity have made it possible for general aviation vehicle manufacturers to consider electrically-powered solutions. The development of trust in battery remaining operating time estimates, however, is currently a significant obstacle to be overcome when considering adoption of electrical propulsion systems in aircraft (Patterson, German & Moore, 2012). There are several ways in which predicting remaining operating time is more complicated for battery-powered vehicles than it is for vehicles with a conventionally-powered liquid-fueled combustion system. Unlike a liquid-fueled system, where the fuel tank’s volume remains unchanged over successive refueling procedures, a battery’s charge storage capacity will diminish over time. Another complicating feature of a battery system is the time-varying relationship between battery output power and battery current draw. Whereas a conventional liquid combustion system uses an approximately constant amount of liquid fuel to produce a given motive power, the power from a battery system is equal to the product of battery voltage and current. Thus, as batteries are discharged, their voltages drop lower, and they will lose charge at a faster rate.

Our previous papers introduced several new tools for battery discharge prediction onboard a small electric aircraft. One paper described a battery equivalent circuit model to simulate battery state (Bole, Teubert, Quach, Hogge, Vazquez & Goebel, 2013). The model’s battery capacity, internal resistance and other parameters were identified through two laboratory experiments that used a programmed load. The batteries were slowly discharged in one experiment. In the other experiment a repeated pulsed loading was done. Current
and voltage profiles logged during flights of a small electric airplane further tuned the battery model (Quach, Bole, Hogge, Vazquez, Daigle, Celaya, Weber & Goebel, 2013). The use of a flight plan with upper and lower uncertainty bounds on the required energy to complete the mission successfully was presented along with an approach to identify additional parasitic battery loads (Bole, Daigle & Gorospe, 2014). This paper introduces a verification testing procedure that is intended to build trust in predictions of remaining flying time prior to actual flight testing. The philosophy behind the testing procedure described here is to translate system performance and safety goals into requirements for an alarm that warns system operators when the estimated remaining flying time falls below a certain threshold. Ground testing of the actual vehicle provides the closest possible testing conditions short of actual flight and captures some of the variation that the powertrain hardware and that the pilot may introduce while avoiding the risks inherent in flight. For instance, the batteries may be drained to a lower capacity during testing of the remaining flying time prediction without danger of vehicle loss.

A small electric unmanned aerial vehicle (e-UAV) was used in this study. The e-UAV is a 33% sub-scale version of the Zivko Aeronautics Inc. Edge 540 T tandem seat aerobatic aircraft. This vehicle has been actively used by researchers at NASA LaRC to facilitate the rapid deployment and evaluation of remaining flying time prediction algorithms for electric aircraft since 2010. Examples of prior works using this platform are found in the following papers: (Saha, Koshimoto, Quach, Hogge, Strom, Hill, Vazquez & Goebel, 2011), (Hogge, Quach, Vazquez & Hill, 2011), (Daigle, Saxena & Goebel, 2012), and (Bole et al., 2013).

Remaining flying time prediction algorithms focus on the prediction of battery charge depletion over an e-UAV flight. A lower-bound on the battery state of charge (SOC) that is considered safe for flight is set at 30% in this work. Flying the vehicle with batteries below 30% SOC is considered to be a high-risk mode of operation that violates the vehicle’s safe operating guidelines. Such violations of operating guidelines are referred to here as a functional failure of the vehicle’s mission.

The accuracy of onboard remaining flying time estimation algorithms is tested in this work, by conducting a series of controlled run-to-functional-failure experiments on the ground. The vehicle under test was strapped down to a platform and commanded to follow an RPM profile that creates battery demand profiles similar to those expected for flight. A picture of the e-UAV strapped down for ground-based testing is shown in Fig. 1.

The time it takes for powertrain batteries to reach 30% SOC establishes a truth value for the functional failure time. Unlike actual flight tests, powertrain batteries can be repeatedly run down to their lower-limits in the ground-based testing described here to verify the accuracy of remaining flying time predictions.

The primary use-case for remaining flying time predictions is to warn system operators when landing procedures must be initiated to avoid aircraft batteries falling below 30% SOC. After consulting with system operators, it was determined that initiating landing procedures at least two minutes before e-UAV batteries would reach 30% SOC under normal operations provided a sufficient energy buffer for landing maneuvers. The predictive element to be tested in this work is an alarm that warns system operators when the powertrain batteries are two minutes from reaching 30% SOC under normal operations.

System operators were also consulted to identify performance requirements on the prognostic alarm. The defined performance requirements were then verified by repeating ground-based run-to-functional-failure tests a specified number of times. The performance requirement testing described here was originally introduced in (Saxena, Roychoudhury, Lin & Goebel, 2013).

Section 2 of this paper provides an overview of the Edge 540T powertrain. Algorithms used for onboard battery state estimation and remaining flying time predictions are summarized in Section 3. The process used to verify onboard remaining flying time predictions through ground testing and experimental results are described in Section 4. Finally, concluding remarks are given in Section 5.

2. OVERVIEW OF EDGE 540T POWERTRAIN

A wiring diagram for the vehicle powertrain is shown in Fig. 2. The aircraft has two 3-phase tandem motors that are mechanically coupled to the aircraft propeller. Powertrain batteries are arranged in two pairs of series connected battery packs. A switchable parasitic load $R_p$ is present to test the robustness of remaining flying time estimation algorithms to changes in battery loading demands. The other symbols in the figure identify the location of current and voltage sensors.

Remaining flying time predictions are generated by propagating present battery charge estimates forward. Forward propagation of present battery state estimates is
performed using estimates of future powertrain demands that will occur over a known flight plan. These future loads include propeller loads and parasitic loads. The prognostic tools used in this work make use of a known flight plan to inform future load predictions, but no prior information is assumed to be available regarding when a parasitic load may be injected.

3. REMAINING FLYING TIME PREDICTION

Battery discharge prediction is described here in terms of the following components; (i) online battery state estimation; (ii) prediction of future battery power demand as a function of an aircraft flight plan; (iii) online estimation of additional parasitic battery loads; and (iv) prediction of battery discharge over the future flight plan. The assumptions and algorithms used for each of these steps are summarized in this section.

3.1. Online Battery State Estimation

Our previous papers (Quach et al., 2013) and (Bole et al., 2014), described the use of an equivalent circuit model and unscented Kalman filtering (UKF) to update battery state estimates based on observations of current and voltage at the battery output terminals. This approach is also summarized here for convenience. Figure 3 shows an equivalent circuit battery model that is used to represent battery output voltage dynamics as a function of battery current. This battery model contains six electrical components that are tuned to recreate the observed current-voltage dynamics of the Edge-540T battery packs. The bulk of battery charge is assumed to be stored in the capacitor, \( C_b \). The \( (R_s, C_s) \) and \( (R_{cp}, C_{cp}) \) circuit element pairs are used to simulate standard battery phenomenon, such as internal resistance drops and hysteresis effects. Additionally, because battery input-output dynamics will change as a function of internal battery charge, it is necessary to parameterize some of the circuit components in terms of the bulk charge stored in \( C_b \) as described in (Zhang and Chow, 2010).

The UKF takes in the measured battery current and voltage, and gives probability distributions for the charge states of each of the three capacitors in the equivalent circuit model as output. Implementation details for the equivalent circuit model and UKF state estimation are omitted here. Readers interested in the application of UKF to the estimation of battery SOC are referred to our previous paper, (Bole et al., 2014). It is sufficient to state here that the equivalent circuit battery model and the UKF state estimation routine are assumed to do an adequate job of tracking the total charge within the battery over an arbitrary usage profile.

The ratio of a battery’s current charge to its maximum charge storage capacity is typically referred to as the state of charge (SOC). Battery SOC is defined here as:

\[
SOC = 1 - \frac{q_b}{q_{max}}
\]  

where \( q_b \) represents the charge stored in capacitor \( C_b \), \( q_{max} \) is the maximum charge that the battery can hold, and \( C_{max} \) is the maximum charge that can be drawn from the battery in practice. Here, \( C_{max} \) will always be less than \( q_{max} \), due to electrochemical side-reactions that make some portion of a battery’s charge carriers unavailable. As the battery ages more of its internal charge will become unavailable because of these side reactions. The \( C_{max} \) parameter must be refitted periodically to capture this effect.

3.2. Prediction of Motor Power Demand as a Function of Aircraft Flight Plan

After estimating battery state, the next step towards predicting remaining flying time is the estimation of motor power demand over the remainder of a given flight plan. The aircraft’s flight plan is assumed here to be specified in advance in terms of a fixed set of segments. Each segment includes a desired vehicle airspeed along with an expected duration or other ending condition. An example flight plan is defined here as:
1. **Takeoff and climb to 200 m**: desired airspeed = 20 m/s, duration = 1.0 min
2. **Maintain altitude, airspeed set point**: desired airspeed = 23 m/s, duration = 3.0 min
3. **Maintain altitude, increase airspeed set point**: desired airspeed = 25 m/s, duration = 2.0 min
4. **Maintain altitude, decrease airspeed set point**: desired airspeed = 18 m/s, duration = 2.0 min
5. **Maintain altitude, increase airspeed set point**: desired airspeed = 23 m/s, duration = fly until landing is called by monitors on the ground.
6. **Remote control landing**: airspeed and duration may vary widely depending on pilot and environmental conditions.

It is important to understand the granularity at which the flight plan is specified. Note that this flight plan specifies desired speed set points, but does not specify a rate at which the vehicle must switch from one speed to another. Also note that while the flight plan specifies a desired speed, it does not specify exactly how close the aircraft must be to the desired speed. These details are left open to the interpretation of the pilot or autopilot.

The energy needed for an aircraft to fly the remainder of a given flight plan will necessarily be uncertain due to random variation in pilot behavior and environmental conditions. A minimum, maximum, and median motor power demand for each remaining segment of the flight plan is used in this work to represent prediction uncertainty. These three power estimates can then be integrated to form predictions of the minimum, maximum, and median motor energy consumption over the remaining flight plan.

Figure 4 shows sample predictions of future motor power and energy demand over segments 1-5 of the given flight plan. Here, segment 5 of the flight plan is shown to extend out indefinitely, representing the intent to continue flying until the ground team calls for a landing. The median motor power demands are estimated for each flight plan segment using a previously developed model, discussed in Bole et al. (2013) and Bole et al. (2014). A plus or minus 30% error margin around the median motor power demand estimate was used to generate the minimum and maximum predictions shown in Figure 4.

A constraint on the minimum battery SOC required for safely landing the aircraft is considered to limit the aircraft’s maximum safe flying time. This minimum SOC threshold is considered here to be 30%. Prediction of available flying time remaining can thus be considered in this example as the time until the battery SOC reaches 30%, assuming that a landing will not be called until the last possible moment. A triplet of minimum, maximum, and median remaining flying time estimates will ultimately be produced by estimating when the battery SOC threshold would be reached for each of the minimum, maximum, and median motor power profiles.

### 3.3. Online Estimation of Additional Parasitic Battery Loads

Parasitic demands on the battery system that cannot be known in advance are simulated with a resistive load that may be injected in parallel with the aircraft batteries at any time during flight. This parasitic load is denoted as $R_p$ in Fig. 2. The magnitude of the parasitic load is assumed to be unknown. An online filtering routine, described in Bole et al. (2014), was shown to rapidly converge on estimates of parasitic load using data from the current and voltage sensors shown in Fig. 2. A battery current profile and parasitic load estimates from a sample aircraft data set is shown in Fig. 5. Here, a 5.5 Ω parasitic load is injected in parallel with the aircraft batteries at 5 minutes into the run. The time at which the parasitic load is injected is shown with a dashed line on the third column of plots in Fig. 6. At the time the load is injected the battery current is seen to become notably higher.
The estimated parasitic load is then seen to rapidly converge to approximately 5.5 Ω. Online parasitic load estimates are directly incorporated into battery discharge predictions. This results in an immediate shift in battery discharge predictions each time the parasitic load estimate is updated. This immediate shift in discharge predictions is demonstrated in the following subsection.

3.4. Predicting of Battery Discharge Over a Flight Plan

Figure 6 shows plots of measured and predicted battery current, voltage, and SOC at three sample times over the battery discharge run. The minimum, median, and maximum predictions are plotted from each sample time until the predicted SOC reaches 30%.

The predictions made at the first two sample times occur prior to parasitic load injection. These predictions are seen to under-estimate the future battery current loads, resulting in over-estimates of future battery voltage and SOC. The parasitic load has been detected by the third sample time, and the predictions at that time are seen to be much more closely centered on the measured evolutions of battery current, voltage, and SOC.

Figure 7 shows predictions of remaining flying time for the example run shown in Fig. 6. The dashed line in Fig. 7 indicates the true flying time remaining. The solid line in Fig. 7 represents the median remaining time prediction. The bars in Fig. 7 represent the interval between the minimum and maximum remaining time prediction. Here, the true flying time remaining is found by subtracting the current time from...
the time at which the lowest battery SOC crossed 30%. The predictions are seen to overestimate remaining flying time until the parasitic load is detected at about 5 minutes into the run. After the parasitic load is detected the remaining flying time predictions are immediately shifted down.

4. GROUND TEST VERIFICATION OF REMAINING FLYING TIME PREDICTION

The ground-based verification testing of Edge 540 T hardware and software was performed by strapping the vehicle down in the LaRC Electromagnetics and Sensors Branch High Intensity Radiated Fields (HIRF) test chamber. More information about the HIRF Chamber can be found in a report of an earlier UAS radio frequency emissions test in (Ely, Koppen, Nguyen, Dudley, Szatkowski, Quach, Vazquez, Mielnik, Hogge, Hill & Strom, 2011). The airplane was placed upon expanded-polystyrene blocks centered within the chamber, as seen in Fig. 8. The aircraft powertrain with propeller was operated with the vehicle anchored using a steel cable to the chamber wall. Its motor and actuators were operated from another room using the same remote control radio that will be used in flight tests.

Measured aircraft states, battery SOC estimates, and remaining flying time estimates were broadcast to a ground station over a wireless downlink. The ground station also had an uplink interface that enables the aircraft’s autopilot to autonomously follow a given flight plan in chamber testing. This autopilot hardware-in-the-loop interfacing capability is discussed in (Bole et. al., 2013).

Only manual control was used for the test results described in this paper, although the autopilot interface is expected to be used in future work. Aircraft propeller RPM, estimated battery SOC, and predicted flying time remaining were displayed to system operators by the ground station in near real-time. The motor throttle was commanded using the control radio by a manual operator, who read the RPM display from the ground station. The operator adjusted the remote control throttle to maintain the target values for the time duration as determined by the flight plan described in Section 3.2. The test proceeded until a 28% SOC condition was indicated on the ground station display for the lowest
Battery current draw was then stopped and powertrain batteries were allowed to rest for approximately one hour. The battery terminal voltages at rest were used to compute an empirical approximation of ending battery SOC.

Onboard data logging during the experiment runs was performed by the data system described in (Hogge, 2011).

4.1. Performance Requirements

The specification of performance requirements for ground verification of remaining flying time predictions is described next. The predictive element to be tested in this work is an alarm that warns system operators when the powertrain batteries are two minutes from reaching 30% SOC under normal operations.

Accuracy requirements for the two minute warning were specified as:

- The prognostic algorithm shall raise an alarm no later than two minutes before the lowest battery SOC estimate falls below 30% for at least 90% of verification trial runs.
- The prognostic algorithm shall raise an alarm no earlier than three minutes before the lowest battery SOC estimate falls below 30% for at least 90% of verification trial runs.
- Verification trial statistics must be computed using at least 20 experimental runs

Here, the two minute alarm is biased to occur early rather than late since the landing becomes unsafe if not enough fuel reserve is present. The early alarm prediction bound limits the “opportunity cost” of unnecessarily denied flying time.

The requirement definitions above use the term “SOC estimate”, because the UKF state estimation algorithm, described earlier, is relied upon to provide online estimates of battery SOC from measured battery current and voltages. A more direct measurement of battery SOC can be obtained after the experimental run is complete by allowing batteries to rest until the terminal voltage settles to a constant value. There is a known relationship between resting battery voltage and SOC that can then be used to compute the ending SOC of all powertrain batteries. The difference between the estimated battery SOCs at the end of each experimental run and the measurement of SOC that is computed from the resting battery voltage is referred to here as the ending SOC estimation error.

An additional requirement for remaining flying time verification testing specifies maximum bounds on the ending SOC estimation error:

The ending SOC estimation error as identified from the resting battery voltage must be less than 5% for at least 90% of verification trial runs.

4.2. Experimental Results

Figure 9 shows the difference between the time at which the two minutes remaining alarm was raised and the time at which the lowest battery SOC estimate crosses 30% for 26 verification runs. Runs that were performed with and without parasitic load injection are identified in the figure. The vertical lines in the figure indicate the bounds on acceptable alarm accuracy. Only one verification run out of the 26 performed is seen to violate the desired accuracy bounds. The requirement that 90% of trials pass this benchmark is thus seen to be satisfied.

Figure 10 shows box plots of the SOC estimation error measured over the 26 verification runs performed. Because each verification run requires 4 powertrain batteries, 104 measurements of SOC estimation error are produced. Only one of these measurements falls outside of the 5% error tolerance allowed. The requirement that 90% of trials pass this benchmark is thus seen to be satisfied.

5. Conclusion

A procedure for verifying the performance of remaining flying time predictions for a small electric aircraft was demonstrated. Aircraft battery packs reaching 30% SOC in flight was defined as high risk operation for our experimental flying vehicle, to be avoided if possible. Ground-based simulated flight testing was shown to enable a safe means of running the aircraft power train to 30% SOC in order to obtain an empirical measurement of the aircraft’s available safe operating time.

Ground-based testing enables repeatable run-to-functional-failure testing of remaining flying time predictions using the integrated flight vehicle. Repeatable testing such as that...
described in this paper is necessary to effectively debug, tune, and build trust in prognostic algorithms prior to deployment in mission critical applications.

ACKNOWLEDGEMENT

This work was funded by the NASA System-wide Safety Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD), Prognostics and Decision Making. E. Hogge author thanks Mr. Patrick Tae for his contribution to the testing of the parasitic load apparatus as part of his intern assignment.

REFERENCES


Edward F. Hogge received a B.S. in Physics from the College of William and Mary in 1977. He has provided engineering services to the government and currently is employed by Northrop Grumman Technical Services. He has recently been supporting aviation safety research through the implementation of electronic systems for subscale remotely piloted aircraft and through commercial aircraft simulation. He is a member of the American Institute of Aeronautics and Astronautics.

Brian M. Bole graduated from the FSU-FAMU School of Engineering in 2008 with a B.S. in Electrical and Computer Engineering and a B.S. in Applied Math. He received M.S. and Ph.D. degrees in Electrical Engineering from the Georgia Institute of Technology. His research interests include:
analysis of stochastic processes, risk analysis, and optimization of stochastic systems. He is currently investigating the use of risk management and stochastic optimization techniques for prognostics and prognostics-informed decision making in robotic and aviation applications. From 2011 to 2013 he performed joint research with the Prognostic Center of Excellence at NASA Ames under the NASA graduate student research fellowship.

Sixto L. Vazquez Mr. Vazquez obtained MSEE from Old Dominion University in 1990 and BSEE from the University of Puerto Rico in 1983. He has developed real-time 3D graphical simulations to aid in the visualization and analysis of complex sensory data. He has developed techniques to interactively process, analyze, and integrate sensory data from multiple complex, state-of-the-art sensing technologies, i.e. FMCW Coherent Laser Radar range measuring system, Bragg grating Fiber Optic Strain Sensing system, etc., into simulation. In recent years, he has developed software for the Ardupilot and associated ground station.

Thomas H. Strom was born in Aberdeen, WA, 1924. He graduated from Hoquiam High School in 1942, and attended Seattle University and the University of Washington 1953-1960. He served in the U.S. Navy during World War II as a Radar Technician. He was an employee of the Boeing Corp. Wind Tunnel, Seattle, WA 1947-1972. He was Senior Engineer with principal expertise in flutter and aeroelastic wind tunnel modeling. He was the founder and president of Dynamic Engineering, Inc., Newport News, VA, until 1997. He has served as a consultant to the Aeroelasticity Group of NASA Langley Research Center (LaRC) and continues to provide engineering and technical services to LaRC through Northrop Grumman Technical Services.

Kyle M. Smalling Kyle Smalling obtained his B.S. in Aerospace Engineering from Cal Poly Pomona in 2013. He is an avid remote control vehicle enthusiast both personally and professionally. His areas of research include Health Prognostics and developing Safety Critical hardware and software. He is employed by Northrop Grumman Technical Services.

Cuong C. Quach received his M.S. from the School of Physics and Computer Sciences at Christopher Newport University in 1997. He is a staff researcher in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. His research areas include development and testing of software for airframe diagnosis and strategic flight path conflict detection.
A V-Diagram for the Design of Integrated Health Management for Unmanned Aerial Systems

Andrew E Heaton1, Ip-Shing Fan2, Ian K Jennions3, Craig Lawson4, and Jim McFeat5

1Civic Drone Centre, University of Central Lancashire, Preston, Lancashire, PR1 2HE, UK
aheaton3@uclan.ac.uk

2IVHM Centre, Cranfield University, Cranfield, Bedfordshire, MK43 0AL, UK
I.S.Fan@cranfield.ac.uk
i.jennions@cranfield.ac.uk

3Aircraft Design Centre, Cranfield University, Bedfordshire, MK43 0AL, UK
c.p.lawson@cranfield.ac.uk

5BAE Systems, Warton Aerodrome, Preston, Lancashire, PR4 1AX, UK
Jim.McFeat@baesystems.com

ABSTRACT
Designing Integrated Vehicle Health Management (IVHM) for Unmanned Aerial Systems (UAS) is inherently complex. UAS are a system of systems (SoS) and IVHM is a product-service, thus the designer has to take into account many factors, such as: the design of the other systems of the UAS (e.g. engines, structure, communications), the split of functions between elements of the UAS, the intended operation/mission of the UAS, the cost verses benefit of monitoring a system/component/part, different techniques for monitoring the health of the UAS, optimizing the health of the fleet and not just the individual UAS, amongst others. The design of IVHM cannot sit alongside, or after, the design of UAS, but itself be integrated into the overall design to maximize IVHM’s potential.

Many different methods exist to help design complex products and manage the process. One method used is the V-diagram which is based on three concepts: decomposition & definition; integration & testing; and verification & validation. This paper adapts the V-diagram so that it can be used for designing IVHM for UAS. The adapted v-diagram splits into different tracks for the different system elements of the UAS and responses to health states (decomposition and definition). These tracks are then combined into an overall IVHM provision for the UAS (integration and testing), which can be verified and validated. The stages of the adapted V-diagram can easily be aligned with the stages of the V-diagram being used to design the UAS bringing the design of the IVHM in step with the overall design process. The adapted V-diagram also allows the design IVHM for a UAS to be broken down in to smaller tasks which can be assigned to people/teams with the relevant competencies. The adapted V-diagram could also be used to design IVHM for other SoS and other vehicles or products

1. INTRODUCTION
Unmanned Aerial Systems (UAS) are Systems of Systems (SoS) where the pilot has been removed from the aircraft and been located in a Control Station (CS), usually on the ground. The CS and the Unmanned Aircraft (UA) are only two possible elements which makeup a UAS, others include: communication links, launch & recovery equipment, transportation – essentially anything which is needed for the UAS to fly and complete a mission.

The removal of the pilot from the aircraft allows greater variety of sizes and configurations for the UA, as they are not constrained by need to have space for humans or the equipment needed for their survival. However, removing the pilot from aircraft does come with its own set of problems. Perhaps one the key problems is that there can be a lack of situational awareness by the pilot, who gets the majority of their information about the aircraft from sensors onboard the UA via the communications link. To overcome this reduction in situational awareness and to aid the pilot various automated and autonomous responses are programmed into the UA. The responses could be could be small or large in scope – such as the UA conducting automated flight is the...
communication link to the CS is lost until it is reestablished. There is no typical UAS. Just as there is no typical UAS, there is no ideal IVHM solution for UAS as a whole (MacConnell, 2007).

1.1. IVHM Design

The design of any IVHM does not sit alone; it is part of the overall blend of systems and functions which will be part of the UAS. Because of this there must be an effort made to bring the design of the IVHM into the overall design of the UAS to allow health monitoring to move from a point solution to a problem which arises, to being part of the UAS from its conception. The design process of any one organization will vary form the next. Therefor it will be impractical to consider all the possibilities of how the design of IVHM for UAS will interact with the overall design.

IVHM can be considered a Product-Service System: a product (the physical items needed e.g. sensors, databases, networks) and service (the management of the health of the UAS and the fleet of UASs) integrated as one to deliver value (Baines et al, 2007; Grubic, Jennions, and Baines, 2009) to an asset through its life – shifting the responsibilities from the user to the supplier of the IVHM (which may or may not be the original manufacturer). Both the service and product for the IVHM need to be fully considered during the design.

The designer of IVHM for a UAS is still bound by the fundamental challenges of modern product development: they must attract and retain customers, be competitive in the market place, and satisfy the requirements of diverse global communities and governments (Liu and Boyle, 2009). Maintenance can often be overlooked during the design of a UAS (Drew et al, 2005), and this not helped by there being a lack of tools to address the design of maintenance (Price, Raghunathan, and Curran, 2007). There will need to be trade-offs in the design of the IVHM due to there being limited resources (e.g. power, cost) and constraints imposed on it by other aspects of the UAS design (e.g. weight, size).

1.2. The V-Diagram

The v-diagram (also known as the v-model) is one such way of designing complex systems. The general concept of the v-diagram can be seen in Figure 1, it consists of three parts: decomposition & definition; integration & testing; and verification & validation (Haskins 2007). The general concept has been adapted many times for use in both systems engineering and software engineering to design products, with each side of the v broken down into stages describing what needs to be done on. There is no standard v-diagram.

1.2.1. Decomposition & Definition

In this side of the v-diagram the design task is defined, by assessing customer needs and setting the high level requirements and goals. Once defined, the product can be decomposed into systems, subsystems, components, etc. Through the decomposition a link should be maintained to the higher level requirements.

1.2.2. Integration & Testing

In this side of the v-diagram the individual designs resulting from the decomposition and definition are combined and tested to see if they function as they should.

1.2.3. Verification & Validation

Although not strictly a side in the v-diagram, verification (whether the design meets the requirements established) and validation (whether the design meets the customers’ needs) is an integral part to ensure that the product is of value.

2. ADAPTATION OF THE V-DIAGRAM

2.1. Integrated Vehicle Health Management for Unmanned Aerial Systems V-Diagram

The v-diagram adapted for designing IVHM for UAS is presented in Figure 2 and brief explanations of each stage is provided after. Two factors went into this adaptation. First, the fact that UAS are SoS and IVHM is both product and service have been taken into account. With the v-diagram splitting into different tracks at various points to represent this. However, the v-diagram presented here is generalized version which does not provide full detail for all the tracks. It focusses on the UA, which is common to all UAS. Other elements in the UAS will vary from one to the next as will the elements of IVHM external to the UAS. These additional tracks will be discussed at the point when they split from the UA-track.

Second, is to bring the IVHM design into line with the rest for the design for a particular UAS. As the v-diagram is a
common tool used within systems engineering the adapted version should make the task easier for organizations to integrate it into their design process. Even if an organization does not use the v-diagram to govern its overall design process it may still be of use, but may need further adaptation to work within their design framework or philosophy.

1. **Customer Proposition to IVHM**
   This stage looks at how the IVHM will add value to the UAS and benefit the stakeholders; the stakeholders will be dependent on the business model. It looks to establish the amount and nature of IVHM to be included in the design.

2. **UAS IVHM Concept of operations and Fleet IVHM**
   This stage looks to establish how IVHM will be used support the concept of operations of the UAS. It also will look at IVHM can be used to support the management of fleet of UAS.

3. **Internal-External Split**
   This stage looks to split IVHM functions between being internal to the UAS and external to it.
   a. **IVHM Internal to the UAS**
      The products (e.g. sensors, code) and services (e.g. automated response to a leak) that are to be implemented internally to the UAS.
   b. **IVHM External to the UAS**
      The products and services that are to be implanted externally to the UAS and is expected to follow a different set of stages beyond this point (and they are out of the scope of the project). This will include (but not limited to) the infrastructure needed for IVHM – e.g. data warehouses.

4. **UAS Element Split**
   This stage looks to split the IVHM functions between the elements of the UAS. The IVHM functions for an element may not necessary contained within it – e.g. prognostics for the UA could be implemented in the CS.
   a. **IVHM On-Board the UA**
      As the UA is focus of the project this element being looked into, and the flowing stages relate to it.
   b. **IVHM on Other UAS Element**
      Other elements are assumed to follow a similar process to the UA.

5. **UA Functional Systems Decomposition**
   In this stage the functions and boundaries of the systems on-board the UA will be defined.

6. **UA Health States & Responses**
   a. **UA Key Function & Health Evaluation**
      In this stage the effects of the functions in the UAS will be assessed and how they relate to the health of the UA. Then the key functions which affect the health of the UA will be identified.

   b. **On-Board UA Responses to IVHM**
      This stage is in parallel with the Key Function & Health Evaluation stage and looks what the responses (service) implemented on-board the UA will be, based off the IVHM data and information produced (both on-board and off-board). Responses could be automatic/autonomous or require human action; this choice is dependent on many factors including autonomy of the UA, time criticality of a fault. It is assumed a different set of stages will follow on from this stage.

   c. **Off-Board Responses to IVHM**
      Off-board responses covers any responses to IVHM data and information produced off-board the UA. This will fall under two board categories. First, is responses relating to other elements of the UAS (e.g. a component/system on a specific UA is going to fail in x amount of time). Second, is responses relating to the fleet of UAS (e.g. a component/system is failing to perform as designed across the fleet).

7. **Function KPI Identification**
   In this stage the key performance indicators (KPIs) for the key functions identified in the previous stage will be identified as well as the possible way for them to be to be monitored.

8. **Implementation**
   This stage is the detailed design and testing of the hardware and software needed to fulfill the functionality identified in the previous step.

9. **Function KPI Monitoring Verification**
   This stage verifies that they KPIs identified in stage seven are being monitored by the IVHM built in the implementation stage (stage eight).

10. **Health Evaluation & Responses Verification**
    This stage verifies that the KPIs being monitored can be used to establish a health state for the key UA functions and that each health state will have an appropriate response to it, either on-board the UA, elsewhere in the UAS, or external to the UAS (fleet IVHM, ordering a part, etc.).

11. **UA Functional Systems Health Verification**
    This stage verifies that all conceived health states for the functional systems are covered.

12. **UAS SoS IVHM Validation**
    This stage validates that the IVHM as designed for the UAS meets its needs. This is to ensure that that the IVHM as designed supports the operation of the UAS.
Figure 2. UAS-IVHM V-Diagram
13. Internal-External Split Validation
This stage validates the division of the IVHM functions into those internal and external to the UAS. It should validate the allocation of IVHM functions to the internal and external categories.

14. IVHM Validation
This stage the whole IVHM as designed is validated against both the IVHM needs from UAS concept of operations and the needs of the fleet.

15. Commission IVHM
After all the verification and validation stages have been passed then the IVHM have proved that it meets the needs of the customer and can be commissioned into service.

16. In-Use Changes & Upgrades
This stage deals with changes that need to be made to the IVHM due to a better understanding of how both the UAS and the IVHM for it operate in service. This stage also has connections the in-use changes and upgrades of other systems and elements within the UAS. The first way it connects is that IVHM data and information could instigate a change elsewhere – e.g. a problem component is identified and redesign or replaced with an alternative. The second is that a change in a different system or element will instigate a change in the IVHM – e.g. change in the engine used on the UA.

3. DISCUSSION
The v-diagram is only one possible tool which can be used to design products and services and the adapted version is one possible version that could be used for designing IVHM for a fleet of UAS.

The UAS-IVHM v-diagram presented in this paper is a generic representation of the process, with no particular UAS (or UAS fleet) in mind, and follows only the UA full to avoid unnecessary complication. Due vast diversity that there is in the size, shapes, and configurations of UA, rest of the UAS enabling their flight (e.g. CS), the nature of the relationship between the operate of the UAS (or fleet), the provider of the IVHM service/maintainer, and the original equipment manufacturer, and the design process of used means that it can only be used as a guide.

The v-diagram is an established tool which is used in many companies and already has many adaptations for use with overall product design. The UAS-IVHM v-diagram is been created with the intention that it can be integrated with v-diagrams.

Table 1. Comparison of UAS IVHM V-Diagram Stage to Other V-Diagrams

<table>
<thead>
<tr>
<th>UAS IVHM V-Diagram Stage</th>
<th>Other V-Diagrams for the Overall Design Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
<td>Sydenham, 2004</td>
</tr>
<tr>
<td>1 Customer Proposition to IVHM</td>
<td>Define Requirements</td>
</tr>
<tr>
<td>2 UAS IVHM Concept of operations and Fleet IVHM</td>
<td>End Item Requirements</td>
</tr>
<tr>
<td>3 Internal-External Split Allocation Subsystem Functions</td>
<td>Subsystem Requirements</td>
</tr>
<tr>
<td>4 UAS Element Split</td>
<td></td>
</tr>
<tr>
<td>5 UA Fictional systems Decomposition</td>
<td></td>
</tr>
<tr>
<td>6 UA Health States &amp; Responses</td>
<td></td>
</tr>
<tr>
<td>7 Function KPI Identification</td>
<td></td>
</tr>
<tr>
<td>8 Implementation</td>
<td>(No Direct Comparison)</td>
</tr>
<tr>
<td>9 Function KPI Monitoring Verification</td>
<td>Prove Component Parts Work</td>
</tr>
<tr>
<td>10 Health Evaluation &amp; Responses Verification</td>
<td>Prove Subsystems Work</td>
</tr>
<tr>
<td>11 UA Functional Systems Health Verification</td>
<td></td>
</tr>
<tr>
<td>12 UAS SoS IVHM Validation</td>
<td></td>
</tr>
<tr>
<td>13 Internal-External Split Validation</td>
<td></td>
</tr>
<tr>
<td>14 IVHM Validation</td>
<td>Complete System Operation Proven</td>
</tr>
<tr>
<td>15 Commission IVHM</td>
<td></td>
</tr>
<tr>
<td>16 In-Use Changes &amp; Upgrades</td>
<td>(No Direct Comparison)</td>
</tr>
</tbody>
</table>

394
Table 1 compared the stages of the UAS-IVHM v-diagram with three difference adaptations of the v-diagram depicting the overall design process: two generic v-diagrams from textbooks (Sydenham, 2004; Grady 2007) and the third the US Department of Defense’s 2014 adaptation (Defense Acquisition University 2015). There are differences between these three different adaptations, both in terminology and stages, but the UAS-IVHM adaptation can be mapped roughly to all of them. However, this is just a cursory comparison, consideration will need to be taken when mapping the stages of the UAS-IVHM v-diagram of their own v-diagram in use in their organization. It is also shown that there is no direct comparison for stage sixteen (In-Use Changes & Upgrades) in all three cases. This stage is key for efficient and effective IVHM – lessons will be learnt during the operation and maintenance of the UAS fleet which can be used to improve the IVHM. However, the concept of in use changes and upgrades can be seen in other v-diagram adaptations, such as one used by the US Department of Transpiration (United States Department of Transportation 2013).

3.1. Splits in the V-Diagram

Most notable feature of the UAS-IVHM v-diagram is the splits – at stages three, four, and six. These splits in the design task allow the right people, with the right skills and experience, to be assigned to the different parts of the design problem – i.e. the division of labor. Further division of some stages could also be included in the UAS-IVHM v-diagram. Such as stage eight (Implementation) which could be divided, but into how many different tasks (and the nature of them) would be dependent on the UAS it is being designed for the outcomes of the previous stages.

Stage three (the Internal-External Split) recognizes the differences between product-service of the IVHM and the support structure need to fulfil the service part of the IVHM external to the UAS. This ensure during the design the systems (e.g. fault forwarding, automated part/work orders are produced) and infrastructure (e.g. data warehouses, communications links) needed external to the UAS is considered and designed.

Stage four (UAS Element Split) again reflects the guiding nature of the UAS-IVHM v-diagram and the variety of UASs. Each UAS is difference, and thus will need different aspects of IVHM on difference elements – there is no ideal IVHM for UAS (MacConnell, 2007). The designer should still be considering the UAS a SoS at this point splitting into the different IVHM functions between the difference elements. This allows them to consider placing some IVHM features for one element on another. Such as prognostics for the UA on the CS, which will have less power and weigh constraints opposed to it being implemented on the UA – again this will be subject to the benefits of any trade-offs in the design of a particular UAS. The stages in the different tracts crated for each element are to be similar to for the UA, as described above.

Stage six (UA Health States & Responses) is different from the splits in the previous stages as it does not create new tracts but has two interlinked parts and links to the IVHM external to the UAS. There will need to be a like between the different health states identified and the appropriate responses and their location within the overall IVHM, whether it is on- or off-board the UA, or internal or external to the UAS.

However, these splits in the design process could compromise the ‘integrated’ nature of the IVHM. Breaking down the design task into smaller sections in this way could cause the design teams tasked with each part to lose sight of the whole picture. It is therefore important that the whole design is properly manage, with someone overseeing the whole process, during all stages of the UAS-IVHM v-diagram. This should help ensure that there are less (ideally no) major issues when conducting the integration and testing, and the validation and verification. There should be keeping of why decisions have been made (e.g. cost, test/simulation results) and how they support the IVHM design overall (e.g. ensuring lower level requirements are linked to a higher level one).

3.2. Tasks of Each Stage

Being a generic representation of the design process the v-diagram does not prescribe what the exact tasks of each stage should be, nor how they should be done. Each design will be different, as will the organization designing the IVHM or UAS.

4. CONCLUSION

The UAS-IVHM v-diagram presented in this paper provides a useful framework for designing IVHM for UAS. The UAS-IVHM v-diagram breaks down the design of the IVHM into different tracks. As the UAS-IVHM v-diagram is adapted from an established tool can be integrated in the overall design of the UAS with greater ease that a totally new framework.

However, the diagram as presented in this paper is a generic version and will need further development if it were to be used to design IVHM for a specific UAS. Also, in order to determine whether the UAS-IVHM v-diagram is useful to the designer of an IVHM it will need to be compared with the current method(s) use within various organizations and also used during the design of IVHM for a UAS.

ACKNOWLEDGEMENT

The authors would like to thank the technological and financial support from BAE Systems and the EPSRC (Industrial Case Studentship Voucher No: 10002036) and the knowledge and feedback from the other Cranfield University IVHM Centre partners.
NOMENCLATURE

UAS  Unmanned Aerial System
IVHM  Integrated Vehicle Health Management
SoS   System of Systems
CS    Control Station
UA    Unmanned Aircraft
KPI   Key Performance Indicators

REFERENCES


Diagnostic Reasoning using Prognostic Information for Unmanned Aerial Systems

Johann Schumann1, Indranil Roychoudhury2, and Chetan Kulkarni3

1,2,3 Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, Moffett Field, CA, 94040, USA
{johann.m.schumann, indranil.roychoudhury, chetan.s.kulkarni}@nasa.gov

ABSTRACT

With increasing popularity of unmanned aircraft, continuous monitoring of their systems, software, and health status is becoming more and more important to ensure safe, correct, and efficient operation and fulfillment of missions. The paper presents integration of prognosis models and prognostic information with the R2U2 (REALIZABLE, RESPONSIVE, and UNOBTRUSIVE Unit) monitoring and diagnosis framework. This integration makes available statistically reliable health information predictions of the future at a much earlier time to enable autonomous decision making. The prognostic information can be used in the R2U2 model to improve diagnostic accuracy and enable decisions to be made at the present time to deal with events in the future. This will be an advancement over the current state of the art, where temporal logic observers can only do such valuation at the end of the time interval. Usefulness and effectiveness of this integrated diagnostics and prognostics framework was demonstrated using simulation experiments with the NASA Dragon Eye electric unmanned aircraft.

1. INTRODUCTION

For safe and efficient operation and successful fulfillment of missions, it is imperative for modern autonomous systems, such as unmanned aerial systems (UAS), to detect, in flight, if all their components are working in a nominal mode, or if there are any faults that might hamper their performance, endanger their missions, or even lead to crashes. Hence, continuous monitoring of the UAS systems, their software, and health status is becoming—even for small UAS—more and more important.

Traditional Fault Detection and Diagnosis (FDD) systems use the current system status as provided by on-board sensors to detect faults and perform root cause analysis. Many different FDD systems and approaches exist and are being used for commercial and military aircraft, automobiles, or complex (chemical) plants, e.g., (Frank, 1996). Many diagnostic reasoners that are used for on-board diagnostics are oblivious of any temporal relationships and strictly focus on the current state of the system, or need signal-preprocessing libraries (e.g., TEAMS RT (Mathur, Deb, & Pattipati, 1998)). Other systems like FACT (Karsai et al., 2006) allow the modeller to specify properties of temporal propagation that can be used for diagnosis. For example, if component A fails some time after component B and there is structural relationship, then the reasoner can deduce from the order of the events what actually happened. Again, all information is deduced from the current and past states of the UAS. The R2U2 framework (Schumann, Rozier, et al., 2013; Schumann et al., 2015; Reinbacher, Rozier, & Schumann, 2014) is a diagnostic framework that combines observers for Metric Temporal Logic with Bayesian networks for probabilistic reasoning and root cause analysis. R2U2 is implemented in FPGA hardware (Geist, Rozier, & Schumann, 2014).

The performance and safety of an electrically powered UAS strongly depends on its battery. In most cases, high-powered rechargeable cells (e.g., Li-poly type cells) are used to power engines, electronics, and payload. When the battery gets depleted, its voltage starts decreasing and available power to the engine might be reduced. This can lead to lower speeds and climb rates. If the voltage drops below a certain threshold, the flight computer will stop and the UAS will crash. Prognostic information that gives a prediction of how much time remains before the battery voltage will drop below a certain threshold can be very valuable in such scenarios, ensuring even safer operation by enabling possible mitigation actions.

In this paper, we extend R2U2 to incorporate the use of prognostic models and reasoning information to take future predictions of safety and performance into account. This enables decision making at the present point in time. With this extension, we are able to substantially improve model expressiveness with respect to future events. For example, a
safety rule might say that “the UAS state is only healthy if an upcoming climb is performed with enough battery reserves when the climb starts”. Assuming that the flight plan entails a climb in 10 minutes, a temporal encoding of that flight rule could be: “within the next 10 minutes always have a good battery.” However, the validity of this formula can only be established after 10 minutes, unless the battery becomes depleted before that. That time, however, is in general is too late to re-plan the mission. The synchronous R2U2 observers provide additional instantaneous information if it is still possible (“maybe”) to observe the flight rule, but again, its ultimate validity can only be decided after 10 minutes. Our proposed extension allows for such a flight rule to be evaluated at the current time indicating if the battery will be good in 10 minutes.

With our novel incorporation of prognostics reasoning into the R2U2 framework, we will have information about the expected future battery performance at the present time, and the system can decide immediately if the flight rule will be violated in the future based upon these predictions. Obviously, assumptions about the battery performance and the estimated load profile are necessary. Using simulation experiments with the NASA Dragon Eye unmanned aircraft we will demonstrate how the use of an advanced prognostics model for the main UAS battery can augment and improve the UAS health models.

The rest of this paper is structured as follows: Section 2 presents an overview of the R2U2 framework with temporal observers and Bayesian reasoning and its realization in FPGA hardware. This section also gives an introduction into our model-based prognostics architecture. Section 3 focuses on the integration of prognostic reasoning into R2U2. In Section 4 we present results of simulation experiments with a NASA UAS and a prognostics model for battery performance. Section 5 discusses future work and concludes this paper.

2. Background

In this section we present an overview of the R2U2 framework with its FPGA implementation and provide an introduction into prognostics.

2.1. The R2U2 Framework

Developed to continuously monitor system and safety properties of an UAS in flight, our real-time R2U2 framework has been implemented on FPGA (Field Programmable Gate Array) hardware (Reinbacher et al., 2014; Geist et al., 2014). Health models within this framework (Schumann, Rozier, et al., 2013; Schumann et al., 2015) are defined using Metric Temporal Logic (MTL) and Mission-time Linear Temporal Logic (LTL) (Reinbacher et al., 2014) for expressing temporal properties as well as Bayesian Networks (BN) for probabilistic and diagnostic reasoning.

R2U2 models are constructed in a modular way where outputs of sensor discretization and reasoning components can be connected to other monitoring and reasoning components (Schumann, Rozier, et al., 2013). An R2U2 model is usually specified using a graphical block representation and consists of data processing blocks, temporal logic observer blocks, and Bayesian diagnostic reasoning blocks. Additional block types provide utility functions.

2.1.1. Temporal Logic Monitors

LTL and MTL formulas consist of propositional variables, the Boolean operators ∧, ∨, ¬, or →, and MTL operators. For formulas p, q, we have □p (ALWAYS p), ◦p (EVENTUALLY p), Xp (NEXTTIME p), pUq (p UNTIL q), and pRq (p RELEASES q). For MTL, each of the temporal operators is accompanied by an upper and lower time bound that express the time period during which the operator must hold. Figure 1 illustrates the MTL operators: □[2,6]p means that p must be true at all times between time step 2 and 6 along the time line (red dots). ◦[0,7]p means that p must be true at least once in the interval [0, 7]. In Figure 1, p is true at t = 7 making that formula true. pU[1,5]q signifies that either q is true at the beginning of the interval, or else p is true at the beginning of the interval and will remain true until a future time (here: t = 3) within the interval, when q must become true (blue dot). Finally, pR[3,8]q means that either p ∧ q is true at the beginning of the interval or q is true at the beginning until a future time within the interval when both are true (purple dot at t = 6 in Figure 1). This operator is works like a push-button: pressing p triggers event ¬q that “releases q” in the future. A detailed definition and semantics can be found in (Reinbacher et al., 2014).

2.1.2. Bayesian Networks for Health Models

In many situations, temporal logic monitoring might come up with several violations of properties. In order to be able to disambiguate the root causes, the R2U2 framework uses static Bayesian Networks (BN) for diagnostic reasoning. BNs are directed acyclic graphs, where each node represents a sta-
Figure 2. BN for Health management. Observable nodes are shaded.

The statistical variable (Figure 2). BNs are well-established in the area of diagnostic and health management (e.g., (Pearl, 1985; Mengshoel et al., 2010)). Conditional dependencies between statistical variables are represented by directed edges. Local conditional probabilities are stored in the Conditional Probability Table (CPT) of each node. For example, the CPT of node $S$ in Figure 2 defines $P(S|U, H, S)$.

For our health models, we are using BNs of a general structure as shown in Figure 2. Discrete sensor signals or outputs of the synchronous temporal observers (true, false, maybe) are clamped to the $S$ (sensor) and $C$ (command) nodes. Since a sensor can fail, it has an unobservable health node attached. As priors, these health nodes can contain information on how reliable the component is, e.g., by using a Mean Time To Failure (MTTF) metric. Unobservable nodes, $U$, may describe the behavior of the system or component as it is defined and influenced by the sensor or software information. For details of modeling see (Schumann, Mbaya, et al., 2013).

During flight, the posterior probabilities of the BN’s health nodes, given the sensor and command values as evidence, are calculated at each time step. The probability $Pr(H_S = good|e)$ gives an indication of the status of the sensor or component. For on-board real-time reasoning, we use a representation of BNs that is based upon arithmetic circuits (AC), as these data structures and algorithms provide predictable real-time performance (Chavira & Darwiche, 2005; Mengshoel et al., 2010).

2.2. Hardware Implementation

R2U2 is implemented as a separate hardware component. Figure 3A shows the high-level architecture of R2U2 integrated in a UAS. Controlled by an on-board flight computer and the flight software (FSW), the UAS receives measurements from various sensors (e.g., inertial sensors, GPS, or barometric altitude) and commands from the ground control station (GCS) and calculates the necessary adjustments of the actuators: elevator, rudder, ailerons, and throttle. Our R2U2 monitor obtains information from sensors and the FSW. Instrumentation of the FSW with small footprint (Geist et al., 2014; Schumann, Rozier, et al., 2013) guarantees minimal obtrusiveness of our framework. All monitoring data are transferred via a read-only interface into our R2U2 implementation that is hosted on an Adapteva Parallella board mounted on the UAS.

Figure 3B shows the major components of the R2U2 monitoring unit as implemented in the FPGA: the control subsystem, the signal processing and filtering system (SP), the runtime verification (RV) unit, and the runtime reasoning (RR) unit. Continuous signals as obtained via the read-only interface are filtered and discretized in the SP unit to obtain streams of propositional variables. The RV and RR units comprise the proper health management hardware: RV monitors MTL properties using pairwise observers defined and proved correct in (Reinbacher et al., 2014). After the temporal logic formulas have been evaluated, the results are transferred to the runtime reasoning (RR) subsystem, where the compiled Bayesian network is evaluated to yield the posterior marginals of the health nodes (Geist et al., 2014).

2.3. Prognostics

In this section we discuss our developed electro-chemical model and battery prognosis framework following the general estimation-prediction framework of model-based prognostics (Luo, Pattipati, Qiao, & Chigusa, 2008; Orchard & Vachtsevanos, 2009; Daigle & Goebel, 2013). Details of the specific algorithms are described in (Daigle & Kulkarni, 2013). Similar approaches have been used for prognosis of pneumatic valves (Daigle, Kulkarni, & Gorospe, 2014; Kulkarni, Daigle, Gorospe, & Goebel, 2014) and for Current/Pressure (I/P) Transducers (IPT) (Teubert & Daigle, 2013, 2014) Here, we only summarize the formulation of the prognostics problem, followed by a brief description of the estimation and prediction approach.
2.3.1. Problem Formulation

We assume the system model may be generally defined as

\[ x(k + 1) = f(k, x(k), \theta(k), u(k), v(k)), \]
\[ y(k) = h(k, x(k), \theta(k), u(k), n(k)), \]

where \( k \) is the discrete time variable, \( x(k) \in \mathbb{R}^{n_x} \) is the state vector, \( \theta(k) \in \mathbb{R}^{n_{\theta}} \) is the unknown parameter vector, \( u(k) \in \mathbb{R}^{n_u} \) is the input vector, \( v(k) \in \mathbb{R}^{n_v} \) is the process noise vector, \( f \) is the state equation, \( y(k) \in \mathbb{R}^{n_y} \) is the output vector, \( n(k) \in \mathbb{R}^{n_n} \) is the measurement noise vector, and \( h \) is the output equation.\(^3\)

In prognostics, we predict the occurrence of an event \( E \) that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold \( T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_{\theta}} \times \mathbb{R}^{n_u} \rightarrow \mathbb{B} \), where \( \mathbb{B} \triangleq \{0, 1\} \) changes from the value 0 to 1. That is, the time of the event \( k_E \) at some time of prediction \( k_P \) is defined as

\[ k_E(k_P) \triangleq \inf\{k \in \mathbb{N} : k \geq k_P \land T_E(x(k), \theta(k), u(k)) = 1\}. \]

The time remaining until that event, \( \Delta k_E \), is defined as

\[ \Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \]

For system health management, \( T_E \) is defined via a set of performance constraints that define what the acceptable states of the system are, based on \( x(k), \theta(k), \) and \( u(k) \) (Daigle & Goebel, 2013). For batteries, we are interested in end of discharge (EOD) time, i.e., the time at which the battery voltage will deplete below the voltage threshold \( V_{EOD} \).

Models of the system components are constructed in this paradigm that capture both nominal behavior, as well as faulty behavior and damage progression. Using these models, observations can be mapped back to the health state of the system as represented in \( x \) and \( \theta \). An estimation algorithm, such as the Kalman filter (KF), unscented Kalman filter (UKF), or particle filter (PF), is used to solve this problem (Daigle, Saha, & Goebel, 2012). In this paper we use the UKF. This state-parameter estimate, along with a prediction of the future usage of the component, is used as input to a prediction algorithm that computes the time to EOD. This time is known as end of life (EOL), the difference between EOL and current time is called the remaining useful life (RUL) (Daigle & Goebel, 2013; Daigle, Saxena, & Goebel, 2012).

2.3.2. Prognostics Architecture

In our model-based prognostics architecture (Daigle & Goebel, 2013), there are two sequential problems, (i) the estimation problem, which requires determining a joint state-parameter estimate \( p(x(k), \theta(k)|y(k_0:k)) \) based on the history of observations up to time \( k \), \( y(k_0:k) \), and (ii) the prediction problem, which determines at \( k_P \), using \( p(x(k), \theta(k)|y(k_0:k)) \), a probability distribution \( p(k_E(k_P)|y(k_0:k_P)) \). The distribution for \( \Delta k_E \) can be trivially computed from \( p(k_E(k_P)|y(k_0:k_P)) \) by subtracting \( k_P \).

The prognostics architecture is shown in Figure 4. In discrete time \( k \), the system is provided with inputs \( u_k \) and provides measured outputs \( y_k \). The estimation module uses this information, along with the system model, to compute an estimate \( p(x(k), \theta(k)|y(k_0:k)) \). The prediction module uses the joint state-parameter distribution and the system model, along with hypothesized future inputs, to compute the probability distribution \( p(k_E(k_P)|y(k_0:k_P)) \) at given prediction times \( k_P \).

2.3.3. Estimation

For batteries a detailed physics model of component behavior using nominal data from the testbed has been developed, which is discussed in (Daigle & Kulkarni, 2013). We use an unscented Kalman filter (UKF) to obtain the state estimate from the sensor measurements, as described in (Daigle & Kulkarni, 2013).

Battery Modeling: In order to predict end-of-discharge (EOD) as defined by a voltage cutoff, the battery model must compute the voltage as a function of time given the current drawn from the battery. There are several electrochemical processes that contribute to the cell’s potential that make this a difficult problem. For the purposes of on-line prognostics, we focus here on a lumped-parameter ordinary differential equations form that still considers the main electrochemical processes. We focus on Li-ion 18650 batteries with an average nominal voltage of 4.2V and nominal capacity of 2200mAh.

The voltages of a battery are summarized in Figure 5 (adapted from Rahn & Wang, 2013)). The overall battery voltage \( V(t) \) is the difference between the potential at the positive current collector, \( \phi_s(0, t) \), and the negative current collector, \( \phi_b(L, t) \), minus resistance losses at the current collectors (not shown in the diagram). As shown in the figure, the potentials vary with the distance \( d \in [0, L] \), because the loss varies with distance from the current collectors. The details of the battery model are discussed in (Daigle & Kulkarni, 2013).

State of Charge: State of Charge (SOC) of a battery is conventionally defined to be 1 when the battery is fully charged and 0 when the battery is fully discharged. In this model, it is analogous to the mole fraction \( x_n \), but scaled from 0 to 1. There is a difference here between nominal SOC and apparent SOC. Nominal SOC would be computed based on the combination of the bulk and surface layer control volumes in the negative electrode, whereas apparent SOC would be computed based only on the surface layer. That is, a battery can be discharged at a given rate, and reach the voltage cutoff, i.e.,

\(^3\)Bold typeface denotes vectors, and \( n_a \) denotes the length of a vector \( a \).
apparent SOC is then 0. But, once the concentration gradient settles out, the surface layer will be partially replenished and the battery can be discharged further, i.e., apparent SOC increases whereas nominal SOC remains the same.

Nominal (\(n\)) and apparent (\(a\)) SOC can then be defined using

\[
SOC_n = \frac{q_n}{0.6q_{\text{max}}} \\
SOC_a = \frac{q_{6-n}}{0.6q_{\text{max}},n},
\]

where \(q_{\text{max}},n = q_{\text{max}}, v_n / v_o\). The factor 1/0.6 comes from the fact that the mole fraction at the positive electrode cannot go below 0.4 (Daigle & Kulkarni, 2013), therefore SOC of 1 corresponds to the point where \(q_a = 0.6q_{\text{max}},n\).

**Battery Voltage**: Now that each of the voltage drops in Figure 5 have been defined, battery voltage can be expressed as follows.

\[
V = V_{U,p} - V_{U,n} - V_o - V_{b,n} - V_{b,n}.
\]

Voltages in the battery are not observed to change instantaneously, i.e., the voltage change occurs smoothly. When discharge completes, for example, the voltage rises slowly as the surface layers move to the concentrations of the bulk volumes, as caused by diffusion. In addition to this, there are transients associated with \(V_o\) and the \(V_{n,i}\) terms. To take this into account in a simple way, we compute voltage using

\[
V = V_{U,p} - V_{U,n} - V_o' - V_{n,p}' - V_{n,n}'.
\]

where

\[
V_o' = (V_o - V_o') / \tau_o \\
V_{n,p}' = (V_{n,p} - V_{n,p}') / \tau_{n,p} \\
V_{n,n}' = (V_{n,n} - V_{n,n}') / \tau_{n,n},
\]

where the \(\tau\) parameters are empirical time constants.

The model contains as states \(x, q_{s,p}, q_{b,p}, q_{b,n}, q_{s,n}, V_o', V_{n,p}',\) and \(V_{n,n}'.\) The single model output is \(V'.\)

### 2.3.4. Prediction

For the batteries, we simulate for various SOC values and load values the corresponding remaining time until discharge, and compute a lookup table. Given the SOC, as computed by the UKF, and expected future load, we can then quickly compute the corresponding time of EOD.

#### 2.3.5. Application to Prognostics

With an accurate model and known future inputs to a system, prognostics should in turn be accurate. We use the architecture described in Section 2.3. As an estimation algorithm, we use the UKF with the battery model; see (Julier & Uhlmann, 1997, 2004) for details on the filter and (Daigle, Saha, & Goebel, 2012; Daigle, Saxena, & Goebel, 2012) for its application to prognostics. The UKF operates on a set of deterministically selected samples, called *sigma points*, that are used to represent the joint state-parameter distribution \(p(x(k), \theta(k) | y(k_0:k))\).

For the prediction algorithm, we perform a simple simulation as described in (Daigle & Goebel, 2013). Each sigma point is simulated forward using the model until EOD is reached; from the corresponding EODs for each sigma point we can construct the EOD distribution. In this work, we assume that the future inputs (\(i_{\text{app}}\)) are known, so the only uncertainty present in the prediction is that related to the model. A defined cutoff voltage is assumed to define EOD.

### 3. APPROACH

We integrate prognostics capabilities into our R2U2 framework by defining a new type of model block (Figure 6). It is provided with the analog input signals for battery voltage \(U_{\text{batt}}\) and current \(I_{\text{batt}}\) and produces values for RUL and
SOC. Prognostics blocks for other system components look similar. The prognostics engine can also be provided with prognosis time \( t \), a planned load profile \( \mathcal{P} \), and prognostics model \( \mathcal{M} \). We can use the outputs of the prognostics module in different ways: a model-based accurate estimate of the battery state at the current time. Because SOC is a statistical variable with a probability density, we use \( \mu_b^{SOC} \) for the expected value and \( \sigma_b^{SOC} \) for the standard deviation.

The RUL prediction itself is calculated via \( \Delta k_E(k_p) \). We denote \( \mu_b^{RUL}[t, \mathcal{P}, \mathcal{M}] \) for the expected value for the battery’s RUL in \( t \) time-stamps from now. The load-profile \( \mathcal{P} \) can be obtained from the flight computer by reading the list of upcoming way-points. There can be substantial uncertainty in the prediction because of unmodeled and unpredictable external effects like wind. Finally, \( \mathcal{M} \) is the battery’s prognostic model. For convenience, we introduce \( \mu_b^{RUL}[t] \) for a standard expected load profile and \( \mu_b^{RUL} \) for an RUL prediction at the current time. Those expressions use a nominal battery model.

In the following, we discuss a number of safety and performance properties that actively use the prognosed values of the main battery of the UAS. Most R2U2 health models consist of Boolean and temporal observers to monitor value ranges, relationships, and flight rules. Value checks test whether data values are plausible and relationships encode dependencies among sensor or software data that may originate from different subsystems. Flight rules are defined by institutions (e.g., the Federal Aviation Administration) or are rules that must be obeyed for mission- or system-specific reasons.

### 3.1. Value Checks

**V1:** Always have enough battery.

\[
\square(\mu_b^{SOC} > 50)
\]

requires that the battery always has a charge of at least 50%\(^4\). This formula continuously monitors the actual battery state at each point in time during the flight. This threshold will certainly vary depending on the mission profile. Such constraints can be easily formulated in logic. For example, a loaded UAS should have its batteries charged to a minimum of 80%: \( \text{is} \text{loaded} \rightarrow \square(\mu_b^{SOC} > 80) \). We might require that the battery will be charged at the end of the flight \( t_{end} \),

**V2:** The above formula can be refined to monitor for safe operation in specific scenarios: for a safe takeoff and subsequent mission, the battery must be charged to at least 80%:

\[
\square((\text{cmd} = \text{takeoff}) \land (\mu_b^{SOC} < 80))
\]

**V3:** The maximum current that can be drawn from the battery is, of course, always limited: \( \square(I_{batt} < 100A) \). However, an already substantially discharged battery should not be overburdened. For example, for the remaining 10 minutes before the end of RUL, only 30A should be drawn. Several levels of maximum allowable current, depending on the battery’s RUL can be specified by:

\[
\begin{align*}
\square(\mu_b^{RUL} > 100min \lor I_{batt} < 100A) \quad &\land \\
\square(\mu_b^{RUL} > 50min \lor I_{batt} < 50A) \quad &\land \\
\square(\mu_b^{RUL} > 10min \lor I_{batt} < 30A) \quad &\land
\end{align*}
\]

Here again, multiple versions with different battery models might be of interest. A safety rule might restrict the amount of current to be drawn even further based on a battery model that performs predictions of an overheated battery.

**V4:** don’t draw too much current if you are unsure about the charge status of the battery

\[
\square((\sigma_b^{SOC} > 10) \land (I_{eng} > 50A))
\]

**V5:** Do not heat up the battery as a result of extended use if it is close to empty. If the battery is charged to only 50% or less, a current of more than 30A should not be drawn for more than 30 consecutive seconds:

\[
\square((\mu_b^{SOC} < 50) \rightarrow (I_{batt} > 30A) \mathcal{U}_{[0,29s]}(I_{batt} \leq 30A))
\]

**V6:** This property monitors that the landing command is issued before the battery reaches a dangerous low level and its RUL is less than 10 minutes. We discretize this property into several ranges and require that the landing command is issued at least 10 minutes prior to RUL reaching a certain value:

\[
\begin{align*}
\square(\mu_b^{RUL} < 110min \rightarrow \mathcal{Q}_{[0,100min]}(\text{cmd} = \text{landing})) \quad &\land \\
\square(\mu_b^{RUL} < 60min \rightarrow \mathcal{Q}_{[0,50min]}(\text{cmd} = \text{landing})) \quad &\land \\
\square(\mu_b^{RUL} < 20min \rightarrow \mathcal{Q}_{[0,10min]}(\text{cmd} = \text{landing}))
\end{align*}
\]

\(^4\)Note that this number has been arbitrarily chosen for illustration purposes.
3.2. Relationships

These properties relate signals from different sensors. Here again, rules can be different depending on the state of that battery. For these examples and our simulation experiments, we assume that the throttle value just regulates the motor current \( I_{eng} \), but not engine power or its speed (RPM). With such a simple controller (mostly found in small hobby-style UASs), the motor RPM will decrease if the battery becomes discharged and \( U_{batt} \) drops.

**R1:** Pitching up should result in a strong climb

\[ \square([\alpha > \alpha_0] \land (\mu_b^{SOC} > 80)) \Rightarrow (V_z \, t > 10) \]

For weaker battery the available engine power is lower and thus we cannot expect the same climb speed. We obtain

\[ \square([\alpha > \alpha_0] \land (\mu_b^{SOC} > 30)) \Rightarrow (V_z \, t > 2) \]

3.3. Flight Rules

Flight rules can be complex formulas that must be valid for the UAS to be in a safe and healthy state and to accomplish the mission successfully. They might concern certain minimal performance rules or behavior in specific situations, e.g., a certain loitering altitude should be reached in case the communication link is lost and wait for the communication to be re-established. Some of the flight rules depend on state of the battery. For example, loitering should be attempted only if the battery will have power for at least 5 more minutes of flight. If the battery is too weak, the only safe alternative is to immediately attempt an emergency landing.

**F1:** This flight rule checks on availability of enough battery power for an upcoming climb in 10 min. As discussed in Section 1, we want to be able to decide immediately if the flight rule is violated or not. Figure 7 shows this nominal situation at time \( t \). The blue line corresponds to the future development of the altitude with a climb in 10 minutes. A naive formulation in MTL as \( \square([\alpha > \alpha_0] \land (\mu_b^{SOC} > 80)) \Rightarrow (V_z \, t > 10) \) is not suitable here (green line), because the formula is false until \( t + 10 \text{ min} \). Only then, the formula can be decided, in our nominal case, to be true. A black triangle depicts the point when the formula is decided. Even the R2U2 synchronous observer only can obtain a “maybe” for the next 10 minutes (red line). By using the prognostics engine, this flight rule can be encoded as

\[ (\mu_b^{RUL} > 30 \text{ min}) \land (t_{\text{climb}} - t > 10 \text{ min}) \]

where \( t_{\text{climb}} \) is the planned time of the next climb as obtained from the list of waypoints in the FSW. This formula can be decided right away (magenta in Figure 7). Alternatively, this formula can be written as \( \mu_b^{RUL}[10 \text{ min}] > 30 \text{ min} \).

4. Experiments and Results

4.1. The UAS Testbed

For this paper, we consider a simple and small UAS platform, the NASA Dragon Eye (Figure 8A). With a wingspan of 1.1 m it is small, but shares many commonalities with larger and more complex UASs. Our R2U2 framework has been implemented on an Adapteva Parallella board. This credit card sized board features a Zync 7010 FPGA and is running Linux, which facilitates preprocessing of signals, running the prognostics engine, and perform data logging. For our experiments, we also performed the BN reasoning under Linux. Monitoring of the FSW is performed according to the schematics shown in Figure 3. Figure 8B shows the integration of the Parallella board, located in one of the wings.

For the experiments presented in this paper, we used a simulation environment that is based on the APM:Plane simulator\(^5\). As shown in Figure 9 the UAS is controlled by the operator using a ground control station (GCS). The environment, the aircraft dynamics\(^6\), motors, sensors, and the flight computer is simulated in software running on a host PC. Monitored FSW data are collected from the simulator and sent to the Parallella board via a serial interface (FDTI cable), where monitoring and diagnosis is performed. These data are also recorded for replay purposes. Since this simulator does not contain any suitable battery model, we are using a MACCOR battery test stand which is able to simulate the flight path for the DragonEye. In the MACCOR we simulate the batteries based on the path scenario in Figure 10. The data collected from the MACCOR testbed is fed to the simulator which in turn runs the prognostics algorithm. In our simplified setting, battery model takes throttle position and returns current values of \( U_{batt} \) and \( I_{batt} \).

---

\(^5\)http://plane.ardupilot.com

\(^6\)JSBSim Flight Dynamics Model jsbsim.sourceforge.net
4.2. Simulation experiments

In this section, we illustrate our approach with some simulation experiments. Figure 10 shows a typical example. The UAS is taking off from altitude 0 and reaching, after a short climb, an altitude of 200ft (Figure 10, top panel). It then flies a level course before descending to 100ft at time stamp 600. A final climb, starting at time stamp 850 brings the UAS back to 200ft altitude. The throttle settings, necessary to execute this profile is proportional to the battery current $I_{batt}$ and shown in Figure 10, 2nd panel from top. The battery voltage drops when a large current is drawn, e.g., during the climb. During the descend, when only very little engine current is drawn, the battery can recover and the voltage slightly increases. Figure 10, 3rd panel from top shows $U_{batt}$ for a fully charged battery (green) and a battery that only had been charged to 50% (red line). With both values, the current and the voltage, the prognostics engine calculates an RUL for the battery at each point in time. As a safety threshold, we use $\theta = 1000$ (dashed line). Finally, the bottom panel shows the output of the R2U2 unit for different formulas related to flight rule $F1$: according to the flight plan, a climb will be necessary at $t = 850$ (dashed magenta line). At time stamp 250, we want to ask if the battery is still good to do that climb. The top row shows the valuation of an asynchronous observer for $\square_{[250,850]} RUL > \theta$. This formula can only become true after the interval has been expired. The green line shows this situation for the good battery. The RUL of weak battery, crosses the threshold at around 500 time stamps. After that time, the formula remains false. Thus, the result of this formula can only be used after time stamp 850, usually too late to start any mitigation action. The second line shows the result of the synchronous observer for the same formula. At the beginning its value is “maybe” until the final value can be determined: at $t = 850$ for the good battery and at $t = 500$ for the weak one. Here, this three-valued observer allows us to obtain information earlier. The bottom lines show the valuation of the prognostics-based formula $\mu_{b}^{RUL}(850) > \theta$ for both scenarios. This formula is evaluated immediately and the result can be used to start a necessary mitigation action earlier than with temporal logic alone.

4.3. Prognostics for Root Cause Analysis

For this experiment we assume that the UAS is equipped with a barometric altimeter (BA) and a laser altimeter (LA). Both sensors measure the altitude of the aircraft but can fail in different ways. We want to construct a probabilistic health model to reason about the health of these sensors.

Our BN model in Figure 11 does not reason about actual altitude measurements but rather uses an abstracted indication of increasing or decreasing altitude, noted as UP and DOWN. Observable sensor nodes for BA and LA (Figure 11, bottom left) are clamped to these values based upon the information extracted from the flight software. The health of each of the sensors is reflected in the nodes $H_{BA}$ and $H_{LA}$ with states GOOD and BAD (Figure 11, top).

The priors of these nodes indicate the reliability of each of the sensors (see below). Each of the sensors should measure the same entity and thus should show the same behavior with respect to UAS climbs and descents. This is modeled by the unobservable behavior node $U_{climb}$, which influences both sensors. If both sensors show the same behavior (e.g., UP, UP), the sensors are most likely healthy. Divergent behavior might indicate a problem with either sensor. Table 1A shows the CPT for BA sensor node $S_{BA}$ (the CPT table for the LA sensor has the same structure). The CPT table de-
notes that a healthy sensor follows the climb behavior (center columns). If the sensor is broken, however, probabilities are 0.5, so nothing can be deduced (right columns).

We now need to address the question on how to find the root cause with divergent sensor readings. In our model, we pull in additional information related to climb or descend behavior of the UAS. This behavior is, in particular, related to the question whether an engine-induced climb is happening. This behavior \( U_{\text{Climb}} \) engine is unobservable as well. An engine-induced climb is set to \( U_{\text{Climb}} \). SoC is \( H \) with \( H \) 0.5, so nothing can be deduced (right columns).

Here, we need to use probabilistic reasoning, because other effects like down-drafts can play a major role. The CPT for this node is shown in Table 1B. A good battery will result in a climb in 85% of the cases: \( p(U_{\text{climb}} = \text{UP}|S_{\text{SoC}} = \text{HGH}) = 0.85 \). A weak battery is modeled as \( p(U_{\text{climb}} = \text{UP}|S_{\text{SoC}} = \text{LOW}) = 0.5 \). Note that there is also an edge connecting \( S_{\text{SoC}} \) with \( H_{\text{Laseralt}} \). This edge models the effect that the LA might be less reliable and accurate when the battery is weak. In our model, the BA is reliable with 95%, whereas the LA is only 90% reliable if the battery is good. Otherwise, the probability for a healthy LA drops to 0.6.\(^7\)

Figure 12 shows the BN in a number of different situations. Observable nodes are colored red for \( \text{DOWN} \) and \( \text{LOW} \), and green for \( \text{UP} \) and \( \text{HGH} \), respectively. The marginal probabilities of the behavior and health nodes are shaded in different levels of grey, where white indicates a probability of one. In a nominal scenario (Figure 11) with sensor input consistent to climbing, both sensors appear to be healthy. Figure 12A shows a nominal situation when the UAS is descending. Note that both unobservable behavior nodes now have status \( \text{DOWN} \) with high probability. A failing barometric altimeter can be detected easily as such during descending (Figure 12B) and climbing (Figure 12C) when the battery is fresh. If we get the same altitude sensor readings, but the battery is weak (Figure 12D), a different picture emerges: the model is aware that commanded climbs with a weak battery often do not result in an increasing altitude, because the engine doesn’t receive enough power. In addition, the LA is less reliable with a weak battery. Therefore, the BN reasons that, despite two other disagreeing sources, the BA is most likely in a better shape than the LA.

5. CONCLUSIONS

In this paper we presented the integration of prognostics reasoning into the R2U2 temporal and Bayesian health management framework. We used a UKF-based prognostics engine to provide SOC and RUL prognoses for the battery of a UAS. This information can be used in the R2U2 model to improve diagnostic accuracy. Most notably, prognostic information allows the monitor to instantaneously evaluate at the present time step the properties dealing with events in the future (e.g., a climb in 10 minutes). Temporal logic observers can only do that valuation at the end of the time interval, or only provide minimal information (maybe). Thus, statistically reliable health information is available at a much earlier time, a

\( ^7 \)The numbers and probabilities used in this example have been selected for illustration purposes only.
The presented work is only a first step toward full integration. Future work will investigate how the probability densities of the prognostics outputs can be directly used by a Bayesian network with sensor nodes capable of handling discretized probability density functions. Also the use of multiple models for nominal and off-nominal battery conditions and fault progression might improve fault detection and diagnostic reasoning in advanced R2U2 health models. Furthermore, we will aim to implement the prognostics engine in FPGA hardware (see e.g., (Soh & Wu, 2014) for a possible approach). For monitoring safety and performance of a UAS, we would also like to evaluate the usefulness of information from other components of the UAS (e.g. engine, or other structural components) as well as environmental effects (e.g., icing) or operational performance (e.g., reliability of a radio link in different weather conditions) for prognosis. Such additional information might be amenable to prognostic modeling and subsequent integration into R2U2 health models.

**ACKNOWLEDGEMENTS**

This work was supported in part by NASA ARMD 2014 I3AMT Seedling Phase I, NNX12AK33A.

**NOMENCLATURE**

- $\mu_b^{RUL}$ mean of RUL for battery
- $\sigma_b^{RUL}$ std of RUL for battery
- $\mu_b^{SoC}$ mean of SoC for battery
- $\sigma_b^{SoC}$ std of SoC for battery
- $V_z$ vertical velocity
- SOC state of charge of battery [%]
- RUL remaining useful life
- EOD End of Discharge
- EOL End of Life
- H,X BN health node for component X
- U BN behavior node

**REFERENCES**


Reinbacher, T., Rozier, K. Y., & Schumann, J. (2014). Temporal-logic based runtime observer pairs for syst-


**Biographies**

**Dr. Johann Schumann** is Chief Scientist for Computational Sciences with SGT, Inc. and working at the NASA Ames Research Center. He received his PhD (1991) and German habilitation degree (2000) in Computer Science from the Technische University Munich in Germany. Dr. Johann Schumann is engaged in research on software health management, verification and validation of IVHM algorithms, analysis and V&V of advanced air traffic control algorithms, and the automatic generation of reliable code. He is author of a book on theorem proving in software engineering and has published numerous articles on automated deduction and its applications, automatic program generation, V&V of safety-critical systems, and neural network oriented topics.

**Indranil Roychoudhury** received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems. Dr. Roychoudhury is a member of the Prognostics and Health Management Society and a Senior Member of the IEEE.

**Chetan S. Kulkarni** received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. He was a Senior Project Engineer with Honeywell Automation India Limited (HAIL) from 2003 till April 2006. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. Since Jan 2013 he has been a Staff Researcher with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and Senior member IEEE.
A Sensor-Based Method for Diagnostics of Machine Tool Linear Axes

Gregory W. Vogl¹, Brian A. Weiss¹, and M. Alkan Donmez¹

¹National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, 20899, USA
gregory.vogl@nist.gov
brian.weiss@nist.gov
alkan.donmez@nist.gov

ABSTRACT
A linear axis is a vital subsystem of machine tools, which are vital systems within many manufacturing operations. When installed and operating within a manufacturing facility, a machine tool needs to stay in good condition for parts production. All machine tools degrade during operations, yet knowledge of that degradation is illusive; specifically, accurately detecting degradation of linear axes is a manual and time-consuming process. Thus, manufacturers need automated and efficient methods to diagnose the condition of their machine tool linear axes without disruptions to production. The Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) project at the National Institute of Standards and Technology (NIST) developed a sensor-based method to quickly estimate the performance degradation of linear axes. The multi-sensor-based method uses data collected from a ‘sensor box’ to identify changes in linear and angular errors due to axis degradation; the sensor box contains inclinometers, accelerometers, and rate gyroscopes to capture this data. The sensors are expected to be cost effective with respect to savings in production losses and scrapped parts for a machine tool. Numerical simulations, based on sensor bandwidth and noise specifications, show that changes in straightness and angular errors could be known with acceptable test uncertainty ratios. If a sensor box resides on a machine tool and data is collected periodically, then the degradation of the linear axes can be determined and used for diagnostics and prognostics to help optimize maintenance, production schedules, and ultimately part quality.

1. INTRODUCTION
Linear axes are used to move components of machine tools that carry the cutting tool and workpiece to their desired positions for parts production (Altintas, Verl, Brecher, Uriarte & Pritschow, 2011). Essentially, a linear axis moves along a nominally linear path and is a vital subsystem of computer numerical control (CNC) machine tools. Because a typical 3-axis machine tool has three linear axes, their positional accuracies directly impact load capacity, quality, and efficiency of manufacturing processes.

As a machine tool is utilized for parts production, emerging faults lead to performance degradation, which lowers control precision and accuracy (Li, Wang, Lin & Shi, 2014). Typical faults within feed systems are due to pitting, wear, corrosion, cracks, and backlash (Zhou, Mei, Zhang, Jiang & Sun, 2009). As degradation increases, tool-to-workpiece errors become more likely, and eventually, linear axes of CNC machines may undergo significant wear that results in a failure and/or a loss of production quality (Uhlmann, Geisert & Hohwieler, 2008). Occurrences of faults and failures are becoming more common as higher levels of automation and productivity within manufacturing result in greater wear on machine components. Machine tool faults account for yearly economic losses of tens of billions of US dollars (Shi, Guo, Song & Yan, 2012). Thus, machine tools must be maintained and available for cost-effective production (Verl, Heisel, Walther & Maier, 2009).

Yet knowledge of degradation is illusive; accurately detecting degradation of linear axes is a manual, time-consuming, and potentially cost-prohibitive process. While direct methods for machine tool calibration are well-established (International Organization for Standardization, 2012) and reliable for position-dependent error quantification, measurements for these methods typically halt production and take “a long time” (Khan & Chen, 2009). The “extensive experimental and analytical efforts” for conventional sequential error measurement methods is usually time-consuming and requires expensive equipment, hindering widespread commercial adoption (Ouafi & Barka, 2013). Because degradation differs along a linear axis and the wear changes with production time (Uhlmann et al., 2008), the particular condition of an axis is usually unknown. The
varying loads, hardness, and surface friction of guides affect their performance, so prediction of remaining useful life (RUL) of linear axis guideways may be difficult (Huang, Gao, Xu, Wu, Zhao & Guo, 2010).

Manufacturers need automated and efficient methods for continual diagnosis of the condition of machine tool linear axes without disruptions to production. This need is consistent with a European roadmap that identified three main key enabling technologies (KETs) for the future of sensor technology in manufacturing: new sensors and sensor systems, advanced sensor signal data processing, and intelligent sensor monitoring (Teti, Jemieliñiak, O’Donnell & Dornfeld, 2010). An online, condition monitoring system for linear axes is needed to help achieve the roadmap goals: decreased machine downtime, higher productivity, higher product quality, and enhanced knowledge about manufacturing processes (Teti et al., 2010).

Efforts to monitor the condition of linear axes components have utilized various sensors:

- Motor torque via current sensors (Li et al., 2014, Uhlmann et al., 2008, Zhou et al., 2009), accelerometers (Feng & Pan, 2012, Huang et al., 2010, Liao & Lee, 2009)
- Accelerometers, thermocouples, and analog controller outputs (torque, speed, and encoder position) (Liao & Pavel, 2012)
- Hall effect sensors (Garinei & Marsili, 2012)
- Piezoresistive thin films (Biehl, Staufenbiel, Recknagel, Denkena & Bertram, 2012, Möhring & Bertram, 2012)
- Piezoelectric ceramics (Ehrmann & Herder, 2013).

These attempts at condition monitoring of linear axes were limited in success, largely because both external sensors and built-in sensors have limitations. Built-in position sensors are usually highly accurate (Zhou et al., 2011), yet controller signals have problems such as low sample rate, limited sensitivity due to sensors being far from monitored components, and unwanted influences from multiple sources (Plapper & Weck, 2001). On the other hand, external sensors can be more direct and physically sensitive, but high costs and required bandwidths have impeded their application for online monitoring of linear axes (Zhou et al., 2009). Adding sensors to machine tools can also be very time-consuming with respect to setup, integration, and data communication.

In this paper, a new sensor-based method for diagnostics of machine tool linear axes is presented. The Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) project at the National Institute of Standards and Technology (NIST) developed a sensor-based method to quickly estimate the performance degradation of linear axes. External sensors are used for high-bandwidth direct or indirect measurements of changes in linear axis errors. The sensors are contained within a ‘sensor box’ for ease of installation and periodic use on a machine tool for data collection and analysis, e.g., within 5 min. The diagnostics and prognostics of the linear axes can be used to help optimize maintenance, production schedules, and ultimately part quality. The cost-effective sensors are expected to be an overall net positive when factoring in the expected savings in production losses and scrapped parts for a machine tool.

2. SENSOR BOX CONCEPT FOR METROLOGY

The goal of the new sensor-based method is to enable efficient monitoring of the change in positioning errors, and hence the change in tool-to-workpiece positioning performance, due to degradation of linear axes. This section outlines these errors, the concept of the sensor-based methodology, and the needed uncertainties of the method.

2.1. Straightness and Angular Errors

Even without degradation, the carriage of a linear axis translates and rotates due to imperfections as the carriage moves along the guideways of the linear axis. Figure 1 shows these six errors that change with axis degradation. As the carriage is positioned along the X axis, it encounters three translational errors from its nominal path: one linear displacement error \( E_{XX} \) in the X-axis direction and two straightness errors \( E_{XY} \) and \( E_{XZ} \) in the Y- and Z-axis directions. The carriage also experiences three angular errors \( E_{AX}, E_{BX}, \) and \( E_{CX} \) about the X-, Y-, and Z-axes.

![Figure 1. Translational and angular errors of a component commanded to move along a (nominal) straight-line trajectory parallel to the X-axis.](image-url)

A typical machine tool has three linear axes, which means that a total of 18 \((= 6 \times 3)\) translational and angular errors exist. These errors are major contributors to the position-dependent tool-to-workpiece errors.
2.2. Sensor Box Concept

Sensors can be used to measure changes in the straightness and angular errors due to degradation. Figure 2 shows a sensor box on a typical 3-axis machine tool with ‘stacked’ linear axes; the Z axis is on the X axis, which is on the Y axis. The sensor box is attached to the Z-axis slide, so that if any axis is moved, the sensor box moves and will detect motion. Accelerometers are used to detect translational errors, and inclinometers and rate gyroscopes are used to detect angular errors. Some properties of these sensors are outlined in Table 1.

Once collected, the sensor data is processed to yield the straightness and angular errors. Specifically, rate gyroscopes signals are integrated once to yield angular changes, and accelerometer signals are integrated twice to yield translational errors. Inclinometers may be used for direct measurement of angle from 0 Hz to about 2 Hz, as seen in Table 1. The reason for two types of angular sensors is that the inclinometer may measure low-frequency angular error terms with greater accuracy than the rate gyroscope.

Degradation may be tracked periodically by data collection during a fixed-cycle test (Garinei & Marsili, 2012, Huang et al., 2010, Liao & Lee, 2009, Verl et al., 2009, Zhou et al., 2009, Zhou et al., 2011). During a fixed-cycle test, the machine tool axes are commanded to move via the same program (the fixed cycle) with the machine tool initially in the same state (temperature, etc.) and undergoing the same nominal loads (cutting forces, if cutting occurs). The collected data is then processed, and the fixed-cycle results are compared to the previous results to determine the changes in straightness and angular errors. The deviations from one test to another are due to degradation, typically due to mechanical wear.

For the machine tool configuration highlighted in Figure 2, changes in the positioning errors could be estimated by using the data from the sensor box and the box’s position relative to the tool tip. Therefore, the sensor box is focused on tracking the effects of degradation of each linear axis on the machining performance. For 4- or 5-axis machine tools with rotary axes, the rotary axes would be held fixed during motion of the linear axes. Also, for a different machine configuration without 3-axis stacking, an additional sensor box on the worktable would be necessary.

Details of the fixed-cycle test and data processing for the determination of error changes will be described in later sections.

2.3. Tolerances for Errors

The sensor-based method depends on the available sensors, whose selection depends on the magnitude of errors to be detected and the accuracy with which they need to be identified. Small levels of degradation of linear axes are expected and allowed, but there are limits specified for axis errors. ISO 10791-2 (International Organization for Standardization, 2001) specifies the tolerances for linear axis errors of vertical machining centers. As shown in Table 2, the acceptable straightness error is limited to 20 μm and the acceptable angular error is limited to 60 μrad.

The measurement uncertainties must be less than the respective specified tolerances to measure the errors. The test uncertainty ratio (TUR), which is the ratio of the tolerance to

Table 1. Properties of sensors used in sensor box.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Bandwidth</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>0.02 Hz to 1700 Hz</td>
<td>2.9 (μm/s)/√Hz at 1 Hz to 0.4 (μm/s)/√Hz at 1 kHz</td>
</tr>
<tr>
<td>Inclinometer</td>
<td>0 Hz to 2 Hz</td>
<td>2.4 μrad</td>
</tr>
<tr>
<td>Rate gyroscope</td>
<td>0 Hz to 200 Hz</td>
<td>0.002 °/s√Hz</td>
</tr>
</tbody>
</table>

* frequencies correspond to half-power points, also known as 3 dB points
* maximum deviation at 0 Hz

Table 2. Tolerances for linear axis errors of vertical machining centers.

<table>
<thead>
<tr>
<th>Error</th>
<th>Tolerance*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straightness</td>
<td>20 μm</td>
</tr>
<tr>
<td>Angular (Pitch, Yaw, or Roll)</td>
<td>60 μrad</td>
</tr>
</tbody>
</table>

* for axes capable of 1 meter of travel, according to ISO 10791-2 (International Organization for Standardization, 2001)
the uncertainty of the measurement, should be sufficiently large. Typically, a TUR of at least 4:1 is recommended; the larger, the better for a measurement system. For the measurement system to be created, we will accept a TUR of at least 4:1 based on design constraints such as sensor cost and size. Thus, we will accept straightness and angular error measurement uncertainties of 5 μm and 15 μrad, respectively, based on the tolerances outlined in Table 2.

3. SENSOR-BASED METHODOLOGY

A sensor-based method was developed to satisfy the TUR constraint of 4:1 and a total cost of about US$5000 for sensors. This section summarizes the sensor box, the fixed-cycle test, and the sensor-based methodology for determination of changes in straightness and angular errors.

3.1. Sensor Box

Figure 3 presents the sensor box, which is composed of two inclinometers, one tri-axial rate gyroscope (three rate gyroscopes), and three accelerometers. Each sensor detects a component of the translational or angular errors seen in Figure 1. The relationships of the sensors to these error components are noted in Figure 3.

![Sensor Box Diagram](image)

Figure 3. Rendered image of sensor box with sensors.

The sensor box top is not shown in Figure 3, so the sensors and their placement can be seen. When the sensor box top is attached, a rubber seal between the box top and base ensure that the sensors are sealed for protection from machine tool environments (including fluids, metal chips, etc.).

3.2. Fixed-Cycle Test

Table 3 summarizes the fixed-cycle test for degradation metrology. For the fixed-cycle test, each of the axes is operated sequentially to move over its entire travel range at three constant speeds typical of linear axes: ‘Slow’ axis speed = 0.02 m/s (50 s to travel 1 m), ‘Moderate’ axis speed = 0.1 m/s (10 s to travel 1 m), and ‘Fast’ axis speed = 0.5 m/s (2 s to travel 1 m). Different axis speeds are used to account for the various sensor bandwidths and noise properties seen in Table 1, in order to minimize the measurement uncertainties of the estimated translational and angular errors. For example, the inclinometer requires a ‘slow’ speed due to its bandwidth of 2 Hz, while the accelerometer requires faster speeds to sense low spatial frequency motions due to its low cutoff frequency of 0.02 Hz. If data is collected for only the forward motion of each axis of a 3-axis machine tool, then the data collection time totals about 3 min (= 3 × (50 s + 10 s + 2 s)).

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Measurand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis Speed = 0.02 m/s</td>
<td>Rate Gyroscope: Angular errors, 0.1 mm to 2 mm wavelength</td>
</tr>
<tr>
<td>Axis Speed = 0.1 m/s</td>
<td>Rate Gyroscope: Angular errors, 2 mm to 10 mm wavelength</td>
</tr>
<tr>
<td>Axis Speed = 0.5 m/s</td>
<td>Accelerometer: Straightness errors, 100 mm to 10 m wavelength</td>
</tr>
</tbody>
</table>

Sensor data is collected, integrated (as needed), filtered, and processed to yield the error components noted in Figure 3. These ‘data fusion’ processes are based on the fact that signals generated by the same geometric errors can be decomposed into various frequency components via filtering and then added together to yield the original errors. As seen in Figure 4, each filtered sensor signal yields a portion of the same geometric error over different neighboring spatial frequency ranges. Because these frequency ranges border each other, the error components add together to result in the originating geometric errors with wavelengths down to 0.1 mm.

Specifically, the rate gyroscope signal is filtered with 2-pole Butterworth filters, integrated, and then summed to the raw inclinometer signal to yield the angular errors. The only exception is for the Z axis, which does not have an inclinometer (as indicated in Figure 3), so the rate gyroscope is used alone to yield $E_{CX}$. Also, the filtered outputs from the accelerometer signals collected at different speeds can be summed, with the resultant acceleration integrated twice to yield straightness errors. The sensors must have relatively low noise in order to minimize drift, especially for the straightness errors based on double integration.
Figure 4. Fixed-cycle test data analysis for (a) straightness errors and (b) angular errors.

4. SENSOR-BASED METHOD UNCERTAINTY

Uncertainty is inherent with physical measurements, and the sensor-based method is no exception. Various sources of uncertainty exist, including sensor misalignment, calibration, and nonlinearity, as well as modal vibrations that could influence the signals. However, this section focuses on the expected main sources of uncertainty to the straightness and angular error estimates: sensor noise and the data fusion process described in Section 3.2.

4.1. Uncertainty Contributions from Sensor Noise

Each sensor has specified noise levels that influence the recorded sensor values. When processed according to Figure 4, sensor noise contributes uncertainty to the straightness and angular errors. Table 4 and Table 5 summarize these uncertainty contributions, determined from numerical simulations based on product specifications (e.g., see Table 1) in which 500 trials were used for statistical purposes. For example, the 10-second long (for the ‘moderate’ speed) simulated noise signal for the rate gyroscope was sampled at 25.6 kHz, a possible experimental sampling rate. The root mean square (RMS) of the spectral density of the white noise was scaled to match the RMS of the spectral density (0.002 \( \mu \text{V/Hz} \)) of the sensor, as specified in the product datasheet. The simulated noise was band-pass filtered with 2-pole Butterworth filters with a lower cutoff frequency of 10 Hz and an upper cutoff frequency of 50 Hz. Next, the filtered angular velocity was integrated to determine the angular displacement noise. Out of 500 trials, the mean was negligible, so the standard uncertainty is approximately the RMS deviation. The largest angular displacement was shown to be 6.2 \( \mu \text{rad} \), and the RMS angular displacement was about 1.2 \( \mu \text{rad} \) for all trials, as seen in Table 5. Similarly, simulated acceleration signals were filtered and double-integrated to yield the translational displacement noises seen in Table 4.

Table 4. Straightness error uncertainties due to sensor noise.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Axis Speed</th>
<th>Filter</th>
<th>Expanded Uncertainty</th>
<th>Standard Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>Slow</td>
<td>Low-pass (5 Hz)</td>
<td>0.81 ( \mu \text{m} )</td>
<td>1.01 ( \mu \text{m} )</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>Moderate</td>
<td>Band-pass (2 Hz, 200 Hz)</td>
<td>0.14 ( \mu \text{m} )</td>
<td>0.029 ( \mu \text{m} )</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>Fast</td>
<td>Low-pass (5 Hz)</td>
<td>0.26 ( \mu \text{m} )</td>
<td>0.055 ( \mu \text{m} )</td>
</tr>
</tbody>
</table>

\( ^{a} \)‘Slow’ speed = 0.02 m/s, ‘Moderate’ speed = 0.1 m/s, and ‘Fast’ speed = 0.5 m/s

\( ^{b} \)defines an interval estimated to have a level of confidence of 99.8 percent

Table 5. Angular error uncertainties due to sensor noise.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Axis Speed</th>
<th>Filter</th>
<th>Expanded Uncertainty</th>
<th>Standard Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclinometer</td>
<td>Slow</td>
<td>Low-pass (2 Hz)</td>
<td>2.4 ( \mu \text{rad} )</td>
<td>1.4 ( \mu \text{rad} )</td>
</tr>
<tr>
<td>Rate gyroscope</td>
<td>Moderate</td>
<td>Band-pass (10 Hz, 50 Hz)</td>
<td>6.2 ( \mu \text{rad} )</td>
<td>1.2 ( \mu \text{rad} )</td>
</tr>
<tr>
<td>Rate gyroscope</td>
<td>Slow</td>
<td>Band-pass (10 Hz, 200 Hz)</td>
<td>7.3 ( \mu \text{rad} )</td>
<td>1.3 ( \mu \text{rad} )</td>
</tr>
</tbody>
</table>

\( ^{a} \)‘Slow’ speed = 0.02 m/s, ‘Moderate’ speed = 0.1 m/s, and ‘Fast’ speed = 0.5 m/s

\( ^{b} \)defines an interval estimated to have a level of confidence of 99.8 percent

\( ^{c} \)based on an assumed uniform distribution (NIST/SEMATECH, 2014)

The combined standard uncertainty over the full spatial spectrum due to sensor noise is equal to the square root of the sum of individual standard uncertainties listed in Table 4 or Table 5. Therefore, the combined standard uncertainty of the straightness error is 0.064 \( \mu \text{m} (= \sqrt{(0.015 \mu \text{m})^2 + (0.029 \mu \text{m})^2 + (0.055 \mu \text{m})^2} \) and the combined standard uncertainty of the angular error is 2.3 \( \mu \text{rad} (= \sqrt{(1.4 \mu \text{rad})^2 + (1.2 \mu \text{rad})^2 + (1.3 \mu \text{rad})^2} \). The combined expanded uncertainties for a coverage factor of \( k = 5 \), similar to those in Table 4 and Table 5, are 0.32 \( \mu \text{m} \) and 11.3 \( \mu \text{rad} \), respectively, for straightness and angular errors.

The uncertainty evaluations are based on Monte Carlo propagation of the contributions from the recognized sources of uncertainty. The resulting expanded uncertainties are the half-widths of coverage intervals that include 99.8 % of the Monte Carlo sample of values of the measurand. The corresponding coverage factor was obtained as the ratio between the expanded uncertainty and the standard uncertainty. The unusually large size of this factor (\( k = 5 \)) is attributable to the fact that the probability distribution of the measurand is markedly non-Gaussian.

Based on the tolerances of 20 \( \mu \text{m} \) and 60 \( \mu \text{rad} \) in Table 2, the TUR for noise-related straightness error is about 63:1 (=
20 μm / 0.32 μm) and the TUR for noise-related angular error is about 5:1 (= 60 μrad / 11.3 μrad). Because the TURs related to sensor noise satisfy the given constraint of 4:1, the sensors are acceptable.

4.2. Uncertainties of Sensor-Based Method

However, uncertainties of the straightness and angular errors are due to not only sensor noise, but also due to the data fusion process described in Section 3.2. Thus, the complete processes outlined in Figure 4 (with sensor noise included) were simulated for different randomly-generated straightness errors and angular errors within the tolerances (20 μm and 60 μrad) seen in Table 2. For any trial, the errors are generated in a process similar to a random-walk. Once generated, the simulated straightness and angular errors are considered to be the ‘reference’ errors, i.e., the ‘true’ errors, which can be compared to the ‘estimated’ errors resulting from the processes described in Section 3.2.

Figure 5(a) shows the three individual components of straightness error for one simulation that are summed to yield the estimated straightness in Figure 5(b). The ‘Fast’ axis-speed component is composed of the lowest frequency terms, while the ‘Slow’ axis-speed component is composed of the highest frequency terms.

The estimation of the straightness and angular errors could be improved with averaging the results of multiple runs for data collection. For one case, Figure 6(a) shows the estimated angular error resulting from the use of 5 runs for averaging, and Figure 6(b) shows how the maximum and RMS values of ΔError (= estimated angular error – reference angular error) change with the number of runs used for averaging. Figure 6(b) shows that the maximum difference and RMS values approach 4.6 μrad and 1.4 μrad, respectively, as the number of runs for averaging increases. Both values do not approach zero as the number of runs increases towards infinity, because the process of Figure 4(b) is not perfect with respect to filtering or data fusion.

![Figure 5](https://example.com/figure5.png)

Figure 5. Example estimation of straightness error: (a) Straightness error component for each axis speed rate and (b) reference straightness error versus the estimated straightness error.

For 100 simulations with different randomly-generated straightness errors (the ‘reference’ errors), the difference between the reference and estimated straightness errors was within ± 5.6 μm, and the RMS of the difference over the entire axis travel was typically around 0.97 μm.

![Figure 6](https://example.com/figure6.png)

Figure 6. (a) the average estimated angular error for 5 runs and (b) the maximum and RMS values of ΔError versus the number of runs.

Table 6 shows the uncertainties of the sensor-based method for both straightness and angular error estimations with various numbers of runs for averaging (1, 5, or 10).

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Runs for Averaging</th>
<th>Expanded Uncertainty&lt;sup&gt;b&lt;/sup&gt;</th>
<th>Standard Uncertainty&lt;sup&gt;a&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straightness</td>
<td>1</td>
<td>5.6 μm</td>
<td>0.97 μm</td>
</tr>
<tr>
<td>Straightness</td>
<td>5</td>
<td>4.1 μm</td>
<td>0.70 μm</td>
</tr>
<tr>
<td>Straightness</td>
<td>10</td>
<td>4.6 μm</td>
<td>0.65 μm</td>
</tr>
<tr>
<td>Angular</td>
<td>1</td>
<td>12.8 μrad</td>
<td>2.3 μrad</td>
</tr>
<tr>
<td>Angular</td>
<td>5</td>
<td>9.0 μrad</td>
<td>1.4 μrad</td>
</tr>
<tr>
<td>Angular</td>
<td>10</td>
<td>8.7 μrad</td>
<td>1.3 μrad</td>
</tr>
</tbody>
</table>

<sup>a</sup> for 100 simulations with different randomly-generated errors over a 1-m travel

<sup>b</sup> defines an interval estimated to have a level of confidence of 99 percent

Based on Figure 6(b) and Table 6, the number of runs should be no more than 5 runs (or 15 minutes of total data acquisition time for three axes), because more than 5 runs is time-
consuming with minimal gain in accuracy. This result is consistent with, and helps to support, international machining standards that utilize 5 runs in any direction (positive or negative) for averaging purposes, e.g., Section A.3.1 in ISO 230-2:2014 (International Organization for Standardization, 2014).

4.3. Method Limitations
Based on Table 2 and Table 6, the TUR for straightness error is about 5:1 (= 20 μm / 4.1 μm) and the TUR for angular error is about 7:1 (= 60 μrad / 9.0 μrad) for 5 runs used for averaging. Both TURs satisfy the given constraint of 4:1, so the process described in Section 3.2 is acceptable.

Nonetheless, the method is limited because neither the sensors nor the data fusion process described in Section 3.2 are perfect. Comparison of the straightness error uncertainties due to either noise (see Table 4) or the entire method (see Table 6) shows that the latter is dominant; the accelerometer noise is a minor contributor to measurement uncertainty. In fact, the major source of straightness error uncertainty is the limited sensor bandwidth; the lower cutoff frequency of the accelerometer is not 0 Hz but rather 0.02 Hz (3 dB). Hence, the spatial frequency of Figure 4(a) does not reach down to 0 mm⁻¹. Figure 7(a) shows how the main local difference between the reference and estimated straightness errors is basically a low-frequency shift.

5. Implementation of Sensor-Based Method
The new sensor-based methodology for diagnostics of machine tool linear axes must be tested, validated, and verified experimentally. This section outlines the means for testing the accuracy of the sensor-based method for the detection of straightness and angular errors.

5.1. Linear Axis Testbed
A linear axis testbed was designed for testing the sensor-based method. As seen in Figure 8, the testbed is composed of a linear slide with a travel length of 300 mm. The linear slide is driven by a direct current (DC) motor with a rotary encoder attached to the motor shaft for motion control. Position is detected with a resolution of about 5 µm, which is much smaller than the 0.1 mm resolution of the method (see Table 3 or Figure 4) to enable repeatable test results.

Sensor boxes move with the carriage: the ‘sensor box’ for the new method and other boxes for a commercial laser-based system. The main laser sensor box contains optical technology to achieve a straightness error uncertainty of ±0.7 μm and an angular error uncertainty of ±3.0 μrad for 300 mm of travel. Due to its accuracy and precision, the laser-based system is used for validation and verification of the sensor-based method results.

5.2. Experimental Method
The sensor-based method must be tested to determine its efficacy in measuring changes, due to degradation, in straightness and angular errors of linear axes. One possible approach to induce degradation signals is to physically wear the linear slide, shown in Figure 8. However, such an approach is potentially time-consuming, expensive, and not repeatable due to unpredictable wear patterns.
In contrast, we choose to experimentally simulate degradation by replacing the default ball bearings with those of different diameters, as illustrated in Figure 9. The linear slide contains four ‘blocks’ or ‘trucks’, each with recirculating balls that contact the rails to constrain the carriage along its nominally linear path. Initially, every ball has the same nominal diameter of approximately 3.972 mm. These default balls can be replaced with balls of smaller or greater diameter to induce straightness and angular error changes of the carriage. The change ($\Delta D$) of ball diameter is experimentally simple, quick, inexpensive, and repeatable.

Figure 9. Example of experimental simulation of linear axis degradation via changes ($\Delta D$) to ball diameters.

For example, Figure 9 shows how half of the balls can be replaced with balls that are 7 $\mu$m larger ($\Delta D = 7$ $\mu$m) and the other half can be replaced with balls that are 7 $\mu$m smaller ($\Delta D = -7$ $\mu$m). The net result is that the straightness errors, $E_{XX}$ and $E_{ZZ}$, will transition between about 5 $\mu$m and $-5$ $\mu$m as the carriage moves along the linear axis, for straightness error changes of about 10 $\mu$m. A variety of other ball configurations can cause translational or rotational changes of 20 $\mu$m or 60 $\mu$rad, respectively, which are the maximum acceptable errors according to Table 2. Therefore, patterns of balls of various diameters can be used to experimentally simulate error changes due to wear.

6. CONCLUSIONS

Manufacturers need quick and automated methods for continual diagnosis of machine tool linear axes without disruptions to production. Towards this end, a new sensor-based method was developed for linear axis diagnostics. The method uses a sensor box composed of inclinometers, accelerometers, and rate gyroscopes for high-bandwidth direct or indirect measurements of straightness and angular errors. When filtered and fused, the data yields seamless errors with wavelengths down to 0.1 mm. Simulations revealed that the multi-sensor-based method is capable of achieving test uncertainty ratios (TURs) of at least 4:1. The sensor-based method must be validated and verified. Thus, a linear axis testbed was designed to allow testing of the new method against a commercial laser-based system. Various degradation patterns can be experimentally simulated by simple substitution of the bearing balls with balls of smaller or greater diameter.

Future tests will reveal the effectiveness of the new sensor-based method. Once the method is verified for diagnostics of linear axes, further tests may show the value of certain metrics for prognostic purposes to estimate the RUL. If the data collection and analysis are integrated within a machine controller, the process may seem to be seamless. Automated diagnostics and prognostics of linear axes can be used to help optimize maintenance and ultimately part quality. Therefore, the method is expected to generate a net positive with respect to decreased production losses for a machine tool.

ACKNOWLEDGEMENTS

The authors thank the Fabrication Technology (FT) Group (NIST) for their outstanding contributions with the experimental setup. This work was also served by invaluable discussions with Andreas Archenti (KTH Royal Institute of Technology, Sweden).

REFERENCES


415
on DFNN performance degradation model. 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010 (pp. 1310-1314), August 10-12, 2010, Yantai, Shandong, China. doi: 10.1109/FSKD.2010.5569106


Biographies

Dr. Gregory W. Vogl is a Mechanical Engineer at the National Institute of Standards and Technology (NIST) located in Gaithersburg, Maryland. He received his B.S. (2000), M.S. (2003), and Ph.D. (2006) degrees in Engineering Mechanics from Virginia Tech, Virginia, USA. Currently, Greg is a member of the Prognostics and Health Management for Smart Manufacturing Systems (PHM4SMS) project, which seeks to develop a methodology, protocols, and reference datasets to enable robust real-time diagnostics and prognostics for smart manufacturing systems. Previously, he designed, fabricated, and experimented on microelectromechanical systems as a National Research
Dr. Brian A. Weiss has a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the Smart Manufacturing Operations Planning and Control program and the Project Leader of the Prognostics and Health Management for Smart Manufacturing Systems project within the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). Prior to his leadership roles in the SMOPAC program and the PHM4SMS project, he spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles; tactical applications operating on Android devices; advanced soldier sensor technologies; free-form, two-way, speech-to-speech translation devices for tactical use; urban search and rescue robots; and bomb disposal robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 & 2008), and the Jacob Rabinow Applied Research Award (2006).

Dr. Alkan Donmez is currently the Group Leader of the Production Systems Group as well as the Program Manager for the Measurement Science for Additive Manufacturing program in the NIST Engineering Laboratory. He has been with NIST for more than 25 years conducting and supervising research in advanced manufacturing sciences, including machine tool performance modeling and metrology, machining process metrology, as well as the recent efforts in metal-based additive manufacturing (AM). He has actively participated in national and international standard committees, developing machine tool performance testing standards, for more than 20 years. He has published more than 70 technical papers and reports in the area of machine tool metrology and manufacturing sciences. He has received various awards for his technical contributions, including R&D100, Applied Research Award of NIST, and Department of Commerce Silver and Bronze Medals.
Quality Control Based Tool Condition Monitoring
Amit Kumar Jain1, Bhupesh Kumar Lad2

1,2Reliability Engineering Lab, Discipline of Mechanical Engineering, IIT Indore, Indore MP 452020, India

1amitkjain23@gmail.com, 2bklad@iiti.ac.in

ABSTRACT
Quality control and tool condition monitoring are two most important aspects of machining process. This paper studies the correlation between tool wear and surface roughness to explore the possibility of modelling the interdependencies between these two aspects. An experimental study is presented in this paper to model the relationship between product quality parameter i.e. average surface roughness and tool wear. Current study reveals that there is a strong positive correlation between surface roughness and tool wear. To map this relationship an ensemble (random forest) fault estimation model is developed for identification and estimation of cutting tool health state. The results from fault estimation model are then used to provide guidelines for future process monitoring and developing dynamic quality control policy.

Keywords: Quality control, tool condition monitoring, surface roughness, tool wear, random forest.

1. INTRODUCTION
Advancement in intelligent manufacturing process has led to better product quality, increased flexibility and higher productivity (Wiendahl, Elmaraghy, Nyhuis, Zäh, Wiendahl, Duffie, and Brike, 2007, Wang, Wang, and Gao, 2013). Mainly these benefits are highly dependent on smooth operations of the various machine elements. Cutting tool is one such important element of the machining system (Zhou, Chen, Fuh, and Nee, 2000). In high speed machining process cutting tool usually suffers from rapidly increasing tool wear rate and the consequent degradation of workpiece surface finish as well as the drop in machined part dimensional accuracy. Also, in machining industry, 20% of the downtime of a machine tool is attributed to cutting tool failures (Kurada & Bradley, 1997). Therefore, Tool Condition Monitoring (TCM) plays a significant role in improving machine productivity, maintaining the quality and integrity of the machined part, minimizing material waste, and reducing manufacturing cost. Being the major cause of tool failure, identification and estimation of cutting tool health state is very important in the machining process (Zhong, Zhou, & Win, 2013). Many research works have been devoted to the methods that rely on the relationships between tool conditions and measurable signals of cutting forces, acoustic emission, vibration, current, etc. for tool wear detection. For example, Dimla and Lister (2000) analysed a relationship between measured signals (cutting force and vibration signals) and tool wear. Haber, Jiménez, Peres, and Alique, (2004) conducted an examination of tool wear monitoring in a machining process based on investigation of multiple signals. The analysis results discovered the relevance of cutting force and vibration signals signatures for tool wear development in high speed machining processes. Dimla (2000) carried out a comprehensive review of several methodologies for tool wear monitoring in machining using different sensor measurements.

In milling, “the cutting dynamics is governed by the interaction between tool structural vibrations and cutting forces” (Ruxu, Elbestawi, & Ullagaddi, 1992). Thus, the cutting force signal is reported to be the best indicator of tool conditions (Li, Lim, Zhou, Huang, Phua, Shaw, & Er, 2009). Although using cutting force signal is a promising method to monitor the tool condition, but it has some disadvantages. The major drawback of using force signals is the cost of the measurement device of a dynamometer and the big size of the dynamometer, which is not practical to mount with the workpiece (Zhong et al. 2013). Iulian and Dragos (2008) pointed out that “the dynamometer is costly, and the installation is rather inconvenient and can weaken the machine structure”. Similar types of disadvantages are with other measurable signals like acoustic emission, vibration etc. Also, use of any type of monitoring technique adds extra cost on the overall manufacturing cost, which is considerably high. An online methodology for tool condition monitoring without the application of measurable signals will cut down the complexity with its assembly, as well as the added expense of measurable signal monitoring system. Such type of methodology is not reported in the literature.
The other important aspect of machining process is quality control, as maintaining the quality of the machined surface is of prime importance. For this quality control tools (viz. control charts) are designed and used regularly to monitor the process. For example, Yang and Jeang (1994) developed a surface roughness monitoring and quality control method, combining statistical analysis and physics of the tool wear. Tangjitsitcharoen and Damrongthaveesak (2013) developed a methodology for online surface roughness and quality control of the estimated surface roughness in turning operation. Colosimo, Moroni, and Grasso (2010) proposed a methodology for modelling of a machining process and ongoing monitoring of its stability that is based on online collected data. However, implementation of quality control is time consuming, which decreases the profitability and expands the expense.

Quality control and tool condition monitoring are important part of machining process. Thus, developing a joint methodology, that not only maintains the quality but also performs tool condition monitoring, will be a highly profitable option. Identifying and mapping the correlation between tool wear and surface roughness will help in getting rid of measurable signal monitoring system and its associated expenses, the only expense associated will be the cost of quality control. The results from such relationship can be used to provide guidelines for efficient process monitoring and dynamic quality control. Thus, in a single expense, both the purpose of quality control and tool condition monitoring will be accomplished. Such type of methodology is not reported in the existing literature of current research. Therefore, the main contribution of this paper is in an attempt to explore the correlation between tool wear and surface roughness, and developing a methodology for joint consideration of quality control and tool condition monitoring. Such, quality control based tool condition monitoring methodology will lead to greater cost savings in overall manufacturing cost. Since, milling is one of the most complex and widely used machining operations; the same is selected in this study.

2. THEORETICAL PRELIMINARIES

2.1. Tool Wear

Tool wear can be stated as “the change in the shape from its original shape during a cutting process by gradual loss of the tool material” (Zhong et al. 2013). Tool wear in milling occurs at higher rate as the tool becomes dull. Due to which cutting forces and temperature increases and immediate loss of sharp edges occurs. After a certain point, tool wear can cause sudden failure of the cutting tool. (Tansel & McLaughlin 1993, Ertunc & Oysu, 2004). It can be illustrated in figure 1 by separating the wear stages as slight wear (regular stage of wear), moderate wear (micro breakage stage of wear) and worn-out as a function of tool life (Al-jonid, Jiayang, & Nurudeen, 2013, Wang, Yang, & Li, 2014). Tool wear affects the surface roughness of the workpiece, which is the main concern of a machining process. The power consumption from motors may also increase due to tool wear (Altintas & Yellowley, 1989, Zhang, Han & Chen, 1995). Thus, it is important to monitor and prevent the tool failure during cutting to achieve high product quality and efficient production.

![Tool Wear Stages](image)

**Figure 1.** Tool wear stages.

### 2.2. Surface Roughness

Surface roughness is defined as “the result of irregularities arising from the plastic flow of chips during the machining” (Lou, Chen & Li, 1999). The most widely used parameters for surface roughness measurements are average surface roughness (Ra), ten point height of irregularities (Rz) and maximum profile peak height (Rp) (Zhong et al. 2013). In this work, average surface roughness is mainly used to measure the surface roughness of workpieces. Average surface roughness (Ra) can be calculated using equation 1 (Lou et al. 1999).

$$R_a = \frac{1}{L} \int_0^L |Y(x)| \, dx \tag{1}$$

where, L = sampling length, and Y(x) = coordinate of the roughness profile curve.

### 2.3. Quality Control

Quality control is an important methodology for asserting standards in manufactured products by testing some samples from output against the specification. Techniques provided in quality control are online quality control methodology to screen an on-going production process. Control charts are most essential techniques of statistical process control. Figure 2 illustrates a typical quality control chart. “The control chart is a graphical display of a quality characteristic that has been measured from the sample versus the sample number or time” (Montgomery, 2007). The chart has a center line that presents mean value of the quality characteristics corresponding to the in-control state. Two other horizontal lines, called the Upper Control Limit (UCL) and the Lower Control Limit (LCL). These limits are set so that if the process is in control, all of the sample points will fall between them. As long as the point plots within the control limits, the process is assumed to be in control, and

419
no action is necessary. However, points that plot outside of the control limits is interpreted as evidence that the process is out of control, and investigation and corrective action are needed to detect and terminate the assignable cause for this behaviour.

In this study, the \( \bar{x} \) and R control charts are used, which are widely used to monitor the mean and variability of variables. In \( \bar{x} \) chart mean of samples are plotted in order to control the mean value of a variable. In R chart range of samples are plotted in order to control the variability of a variable.

![Control Chart](image)

Figure 2. A typical control chart.

3. METHODOLOGY

Details of the proposed methodology are given in following sub-sections.

3.1. Experimental Setup

In practice, tests and verifications of fault detection methods are easy to perform, because the faults can be easily simulated or introduced on the real industrial system. However, this is not the case for quality control methods where the change in quality characteristics is generally a consequence of a long and slow degradation of one or more components of the system. Thus, to test these methods, it is necessary to create the degradation through accelerated degradation tests of physical components and measure the quality characteristics throughout the life. For this purpose, an experimental setup is developed (see, figure 3). In the experiment, EMCO MILL E350 vertical milling machine is used as the test bed. A high speed steel 6mm flat end mill cutter with four cutting edges is selected for testing. Machining operation employed was face milling to create a flat plane surface on the workpiece, with constant operating conditions (feed = 300mm/min, speed = 1000RPM, depth of cut = 0.25mm) in dry state. After every cutting process, the surface roughness and tool wear is measured. A HANDYSURF E-25A/B tester was used to measure the finish surface and tool wear is measured. A HANDYSURF E-25A/B tester was used to measure the surface roughness. Toolmakers’ microscopy system was used to measure the tool wear of the milling cutter. During experiments utmost care has been taken while resetting the cutter, to minimize its effect on surface roughness. The setup is capable of providing real life data for quality control, as it covers in-depth quality aspects of machined products developed throughout the life of milling cutters.

3.2. Correlation between Surface Roughness and Tool Wear

Identification of correlation between surface roughness and tool wear will be of high significance. Thus, life tests are carried out to study the wear behaviour of milling cutters, i.e. all the cutters are run till it reaches a pre-defined level of wear. Two failure modes have been observed and recorded, viz. tool worn-out (if average wear value from four cutting edges reaches 0.746mm) and tool breakage. Figure 4 shows the tool wear measured in experiment with two different failure modes (worn-out and breakage) cutting tool. Average surface roughness (Ra) was also measured for each surface during the cutting processes. Figure 5 shows the average surface roughness of the workpiece and tool life of two cutters with different failure mode. The surface roughness value remains stable and small when the tool has very less wear. When the cutting tool reaches the failure state, the surface roughness value gradually increases, and then it significantly increases when the critical tool failure occurs.

From the experiments it is observed that identical cutting tools, even operated at same operating conditions, show different behaviour (because of inherent design variations), and may fail with different failure modes (worn-out and breakage). Such types of conditions significantly affect the performance of the process, for example see figure 5. It shows that average surface roughness values obtained from the two different cutters with different failure modes. The tool which failed from breakage is producing products with high average surface roughness from its initial age, while the worn-out tool is producing less rough products in its initial age and roughness increases as the tool reaches its end of life. If we see the wear pattern of both the tools in figure 4, it is clear that both the tools are having different wear behaviour. Thus, a correlation analysis is carried out; correlation measures the relationship between two variables (say, a and b). Correlation is used to determine whether the large values of first variables (say, a) are associated with the large values of second variables (say, b), and vice versa. Correlation coefficient is used to measure the strength of the relationship between two variables; this can be calculated using equation 2 (Zhong et al. 2013). The Pearson correlation coefficient (\( r \)) between the surface roughness and tool wear in case of tool breakage is found to be 0.859 and in case of worn-out tool is 0.807. This correlation study reveals that there is a strong positive relationship between the surface roughness and tool wear.

\[
 r = \frac{\Sigma(a_i-\bar{a})(b_i-\bar{b})}{\sqrt{\Sigma(a_i-\bar{a})^2}(\Sigma(b_i-\bar{b})^2)}
\]  

(2)
where, \( \bar{a} \) is the mean value of variable \( a \) (Average surface roughness), \( \bar{b} \) is the mean value of variable \( b \) (Tool wear).

3.3. Fault Estimation Model

As tool wear is the major cause of tool failure, identification and estimation of cutting tool health state is very important in the machining process, so that it can be replaced on timely manner. As shown from figure 4 and 5, tool wear and surface roughness exhibit a strong positive correlation. Thus, if we develop a relationship between surface roughness and tool wear, it will be of great interest. This relationship can be used for tool condition monitoring. A Fault Estimation Model (FEM) is developed to link one or more of the quality parameters like \( R_a \), \( R_p \) and \( R_z \) with the health state of the tool. Input to this fault estimation model will be quality parameters and output will be the current health state (stage I, stage II or stage III) of the tool. The prediction of current health stage will help in tool replacement decisions.

To develop an efficient fault estimation model, an ensemble classifier is needed. As a result, Random Forest (RF) is used to develop the fault estimation model. RF is utilized because of its high performance in modeling complex processes, unbiased estimate of the generalization error, high accuracy and fast build time (Liaw & Wiener, 2002). Originally proposed by Breiman (2001), the method adds an extra layer of randomness to the original bagging algorithm. It is more user friendly, intuitive, and is based on two parameters (the number of variables in the random subset at each node and the number of trees in the forest) only. Further, in contrast to most algorithms in literature (including discriminant analysis, support vector machines and artificial neural networks), it is dependent on the data values and is less sensitive to the values of the two parameters (Liaw & Wiener, 2002). Consequently, it is perfectly aligned to our needs, thus, we use this method to formulate the fault estimation model for cutting tools. In RF classifier each tree is constructed using the following methodology: Firstly, \( N \) number of training cases and \( M \) number of variables are taken in the classifier. \( m \) number of input variables are used to take decision at the node of the tree, here, \( m \) is kept lesser than \( M \). A training set is selected for this tree by choosing \( n \) times with replacement from all \( N \) available training cases. Rests of the cases are used to estimate the error of the tree, by predicting their classes. For each node of the tree, randomly \( m \) variables are selected, on which to base the decision at that node. Then, the best split based on these \( m \) variables in the training set is calculated. Each tree is fully grown and not pruned. For prediction, a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction. For more details regarding RFs, the interested reader can refer to Breiman (2001).

For the current study, random forest of 100 trees, each constructed while considering 1 random feature is used. The life data used here is drawn from experiments conducted on five milling cutters. Health states of the milling cutters are classified in three stages and their wear scopes are shown in table 1. No specific method or technique is available to decide wear scope. In the present work health states are defined based on the literature (Wang et al. 2014, Al-jonid et al. 2013) and physical observation of change in the surface roughness of the produced surface with tool degradation during experiments. The complete life dataset from milling cutters comprises of 321 numbers of the samples. Model should not be validated on the same data used to create the classifier. Accordingly, the K-fold cross-validation method was chosen (Stone, 1974) in this study. The original sample is partitioned into \( K \) disjoint subsamples. Of the \( K \) subsamples, a single subsample is retained as the validation data for testing the model, and the remaining (\( K-1 \)) subsamples are used as training data. The cross validation process is then repeated \( K \) times (the folds), with each of the \( K \) subsamples used exactly once as the validation data. Then, the \( K \) results from the folds are averaged to produce a single estimate of the classifier accuracy (Correa, Bielza, & Pamiés-Teixeira, 2009). In our model we chose \( K = 10 \). For performance assessment, accuracy of the testing results is calculated. Accuracy of a classification model is calculated as, the proportion of the total number of predictions that were correct (Wang et al. 2014). To check the applicability of developed model; computational time, that is the required time to learn and test the dataset is also computed. Moreover, to improve relevance and accuracy of prediction, an Advance Fault Estimation Model (AFEM) is also developed. In the advance model, with average surface roughness value two more parameters (\( R_p \) and \( R_z \)) are given as input. This advance fault estimation model can be used in the presence of extra information in terms of \( R_p \) and \( R_z \) in place of the fault estimation model to update the accuracy of the prediction. Table 2 shows the performance of both the developed models. Open source tool Weka (Version: 3.7.12) is used for training and testing the developed fault estimation models.

<table>
<thead>
<tr>
<th>Tool Wear Classification</th>
<th>Stage I (slight wear)</th>
<th>Stage II (moderate wear)</th>
<th>Stage III (worn-out)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wear value (mm)</td>
<td>&lt; 0.27750</td>
<td>0.27750 - 0.56775</td>
<td>&gt; 0.56775</td>
</tr>
</tbody>
</table>
Figure 3. Experimental setup (line diagram).

Figure 4. Tool wear versus tool life.

Figure 5. Average surface roughness versus tool life.

Table 2. Performance of fault estimation models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Estimation Model</td>
<td>70</td>
<td>0.13</td>
</tr>
<tr>
<td>Advance Fault Estimation Model</td>
<td>82</td>
<td>0.17</td>
</tr>
</tbody>
</table>

The results from developed fault estimation models are promising and show potential to be practically applied under industrial constraints in reasonable computational time.

3.4. Process Monitoring and Quality Control Policy

A CNC milling process is used for Mild Steel (MS) plate manufacturing with fixed dimensions (165x100x20mm).
Average surface roughness (Ra) of the plate in horizontal direction is an important quality characteristic. The average surface roughness value is in micron. We wish to establish a statistical control of the average surface roughness of the plate in this process using \( \bar{x} \) and R control charts. This will require setting of control charts limits. This is explained in the following sub-section.

### 3.4.1 Setting of \( \bar{x} \) and R Charts

In order to get statistical control limits for \( \bar{x} \) and R charts, common approach is to take some initial samples from the process considering the process was in control. In the current experiments, all the process related variables are kept constant, for example the operating conditions are kept constant throughout the process to achieve the desired dimensions. Similarly machine tool, workpiece material and the work environment etc. are same throughout the process. The only variable which changes periodically is the cutting tool, as it degrades with the time and eventually fails, thus it is to be replaced periodically. In order to get safe statistical control limit for \( \bar{x} \) and R charts for future use, data from in control process are required. In the current manufacturing scenario cutting tool is the only variable in the whole process which changes periodically (because of failures). The current study revealed that different failure modes of the cutting tools have significant effect on the product quality.

The control limits of the R chart are calculated as follows:

\[
UCL = D_4 \bar{R} \\
LCL = D_3 \bar{R}
\]

where, the constants \( D_3 \) and \( D_4 \) are tabulated based on sample size (for sample size of 5, \( D_3 = 0 \) and \( D_4 = 0.3251 \)) (Montgomery, 2007).

Since, the R chart indicates that the process variability is in control (see, figure 6); we may now construct the \( \bar{x} \) chart.

The center line is calculated as shown in equation 6.

\[
\bar{x} = \frac{\sum_{i=1}^{n} \bar{x}_i}{n}
\]

where, \( x_i \) = Mean of the i\textsuperscript{th} sample.

The control limits of the \( \bar{x} \) chart can be found out as follows:

\[
UCL = \bar{x} + A_2 \bar{R} \\
LCL = \bar{x} - A_2 \bar{R}
\]

where, the constant \( A_2 \) is tabulated based on sample size (for sample size of 5, \( A_2 = 0.577 \)) (Montgomery, 2007).

When the preliminary sample means are plotted on this chart as shown in figure 6, all the points are inside the control limits. Since, both the \( \bar{x} \) and R charts depict control, it means that the process is in control under stated levels. This set of safe control limits are adopted for monitoring future production. This completes the setting of \( \bar{x} \) and R charts limits for future use. The control charts shown here are made using Minitab (Version: 17.2.1). The conventional usage of the \( \bar{x} \) and R control charts is explained in the next section.

### 3.4.2 Conventional Process Monitoring and Quality Control Policy
Once a set of safe control limit is established, the conventional way is to use the control charts for monitoring future production. Figure 7 illustrates the working of conventional process monitoring and quality control policy. Additional samples from the process, each of sample size five from the process (with a new cutting tool) were collected after the control charts were established and the sample values of $\bar{x}$ and R are plotted on the control charts with sampling frequency of one hour. The control chart detected out of control process at 6th sample. As the control chart shows an out of control process, it means that an assignable cause has occurred at that time. Conventionally, the operator is directed to check process variables viz. cutting tool, process settings, calibration etc. and then make the adjustments in an effort to bring the process back into state of control. This conventional usage of control chart will only detect occurrence of assignable causes, also fixed sampling frequency or sample size were used throughout the monitoring. It will be of great interest if we are able to detect the reason for assignable cause and simultaneously able to vary the sampling frequency or sample size while monitoring the process for early detection of out of control process.

3.4.3 Fault Estimation Model Based Process Monitoring and Dynamic Quality Control Policy

For early detection of out of control process, fault estimation model based process monitoring and dynamic quality control policy is proposed. In this process, the mean surface roughness is monitored with a $\bar{x}$ control chart, and the process variability is monitored by R chart. Notice that if the R chart displays an out of control point, operating personnel are coordinated to contact process engineering instantly. The current manufacturing process is having only one controllable variable, cutting tool. In this scenario, the high chance of assignable cause may be tool health. Thus, the developed fault estimation model is linked with the control chart in such a way; the sample quality data is fed as input to the fault estimation model to know the current health state of the tool without stopping the production. The fault estimation model can give three types of indication about the health of cutting tool:

1. Tool is in stage I (Safe Zone)
2. Tool is in stage II (Partial Safe Zone)
3. Tool is in stage III (Worn-out Zone).

Based on the output from fault estimation model some guidelines are proposed for each health stage of the cutting tool for process monitoring and dynamic quality control. When the health state of the cutting tool is identified as stage I (the stage I of the cutting tool indicates only slight wear have been occurred in the tool, and the tool is in safe zone), the process monitoring is continued with initial sampling frequency or sample size. As the fault estimation model indicate the shift in the health state of cutting tool from stage I to stage II, it means that moderate wear is now present in the tool and this can be the reason of assignable cause in near future. Being in partial safe zone, it’s not wise to discard the tool, here the decision on varying the sampling frequency or sample size is needed to be taken for early detection of out of control process in future. As the tool health state is identified as stage III (tool is now in worn-out zone), this indicate that tool wear will soon can cause out of control process, thus here further decision on varying the sampling frequency or sample size can be made for very early detection of out of control process. Also, as the tool is reached to its failure zone, tool replacement decision can be taken in a timely manner, and this will also eliminate the faulty product development and reduce losses because of tool failure viz. power consumption etc. Based on these guidelines smart decisions on quality improvement (cost of inspection can be managed efficiently), and timely tool replacement can be taken efficiently before tool failure.

![xbar and R Control Charts](image)

Figure 6. $\bar{x}$ and R charts.
Figure 7 illustrates the working of fault estimation model based process monitoring and quality control policy, applied on the same process data as used for conventional process monitoring and quality control policy. Additional samples from the process, each of sample size five from the process are fed as input to the fault estimation model to know the current health state of the cutting tool. From the results of fault estimation model it is identified that the current health state of the cutting tool is reached to stage II at third sample. According to the proposed guidelines, the decision on varying the sampling frequency is taken for future monitoring. Sampling frequency from the fourth sample is changed to half an hour from one hour for early detection of out of control process. With this change the control chart is now able to detect the out of control process early. The control chart detected out of control process at fourth sample. Table 3 shows the performance of fault estimation model based usage of control chart.

Table 3 Fault estimation model based usage of control chart.

<table>
<thead>
<tr>
<th>Sampling frequency</th>
<th>1 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault estimation model</td>
<td>Input</td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt; sample</td>
<td>Stage I</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt; sample</td>
<td>Stage I</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt; sample</td>
<td>Stage II</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision on change in sample frequency from 4&lt;sup&gt;th&lt;/sup&gt; sample onwards</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Sampling frequency</td>
</tr>
<tr>
<td>Out of control process detection</td>
</tr>
</tbody>
</table>

4. COMPARISON OF CONVENTIONAL AND FAULT ESTIMATION MODEL BASED PROCESS MONITORING AND QUALITY CONTROL POLICY

Table 4 and figure 7 shows the comparison of performance of conventional and fault estimation model based control chart policy in terms of product produced till detection of the out of control process. Till actual occurrence of out of control process twenty nine products were produced. Whereas, in conventional usage of control chart, total sixty products were produced from the process till the detection of out of control process. However, only thirty five products were produced from the process till the detection of out of control process through fault estimation model based usage of control chart. It is clear that the fault estimation model based process monitoring and dynamic quality control policy is capable of detecting out of control process very early than conventional policy. With the help of fault estimation model based control chart usage, we are able to reduce the number of faulty product development. As the difference between the products produced before the detection of out of control process is thirty one from conventional policy with actual occurrence, this is considerably high. Consequently, only six products were produced till the detection of out of control process from the fault estimation model based usage of control chart.

Figure 7. Illustration of conventional and fault estimation based process monitoring and quality control polices.
Table 4 Comparison of performance of conventional and fault estimation model based usage of control chart.

<table>
<thead>
<tr>
<th>Products produced till out of control process detection</th>
<th>Actual occurrence of out of control process</th>
<th>Conventional way of usage of control chart</th>
<th>Fault estimation model based usage of control chart</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
<td>60</td>
<td>35</td>
</tr>
</tbody>
</table>

This joint methodology will lead to online monitoring of the production process as well as serve the purpose of tool condition monitoring. The developed quality control based condition monitoring methodology is of high importance for manufacturing industries in improving the performance of their machining process as well as reducing the overall manufacturing cost.

5. Conclusion

This paper explores the correlation between tool wear and surface roughness and utilizes the same for dynamic quality control and efficient tool replacement decisions. The major contributions of this paper are as follows:

1. An experimental study is carried out, which revealed that strong positive correlation exists between tool wear and surface roughness.

2. An ensemble (random forest) based fault estimation model is developed to map the relationship between surface roughness and tool wear.

3. Guidelines for process monitoring and quality control based on the results of fault estimation model are proposed. These guidelines will lead to efficient quality improvement as well as timely tool replacement decisions.

4. The fault estimation model based process monitoring and dynamic quality control policy is capable for early detection of out of control process than conventional usage of control charts.

The results of this study will promote and enable the establishment of a quality control based intelligent predictive monitoring system to estimate the useful life of the tools and detect the surface degradation prior to costly failure and damage to high valued workpieces.

Acknowledgement

This research work is funded by Department of Science and Technology (DST), India, Project no. YSS/2014/000320. We acknowledge the support.

References


**Biographies**

**Amit Kumar Jain** is working as research scholar in School of Engineering (Discipline of Mechanical Engineering) at the Indian Institute of Technology Indore, India. He received his Masters of Technology in Maintenance Engineering from Maulana Azad National Institute of Technology Bhopal, India. His research interest includes prognosis, reliability engineering.

**Bhupesh Kumar Lad** is an assistant professor in School of Engineering (Discipline of Mechanical Engineering) at the Indian Institute of Technology Indore, India. He received his Ph.D. from Indian Institute of Technology Delhi, India. His research interest includes reliability, prognosis, computer aided manufacturing process planning.
Adaptive Multi-scale PHM for Robotic Assembly Processes

Benjamin Y. Choo¹, Peter A. Beling², Amy E. LaViers³, Jeremy A. Marvel⁴, and Brian A. Weiss⁵,

¹²³ University of Virginia, Charlottesville, Virginia, 22904, USA
bycj@virginia.edu
beling@virginia.edu
alaviers@virginia.edu

⁴⁵ National Institute of Standards and Technology, Gaithersburg, Maryland, 20899, USA
jeremy.marvel@nist.gov
brian.weiss@nist.gov

ABSTRACT

Adaptive multiscale prognostics and health management (AM-PHM) is a methodology designed to support PHM in smart manufacturing systems. As a rule, PHM information is not used in high-level decision-making in manufacturing systems. AM-PHM leverages and integrates component-level PHM information with hierarchical relationships across the component, machine, work cell, and production line levels in a manufacturing system. The AM-PHM methodology enables the creation of actionable prognostic and diagnostic intelligence up and down the manufacturing process hierarchy. Decisions are made with the knowledge of the current and projected health state of the system at decision points along the nodes of the hierarchical structure. A description of the AM-PHM methodology with a simulated canonical robotic assembly process is presented.

1. INTRODUCTION

Prognostics and Health Management (PHM) refers to a class of techniques and methods that enable condition monitoring of a physical machine or functional process. PHM encompasses health monitoring of a system; provides diagnostic information including what is at fault, why the fault occurred, and how the fault can be remedied; and offers prognostic intelligence as to when a system or process is going to degrade to various states that may include going out of specification or failure.

A manufacturing system is a complex system-of-systems with a hierarchical structure. A manufacturing system hierarchical structure is described as a facility consisting of multiple assembly/fabrication lines that are further divided into work cells or work stations which are further divided into multiple machines consisting of components (Hopp & Spearman, 2008). One challenge in PHM for manufacturing is that in most applications data gathering and analysis is limited to the component level. For example, prognostic intelligence for machines, such as robots or machine tools, typically does not propagate beyond the boundaries of the machine even though the failure of a single component may lead to failure of other components or to system-wide effects.

The use of PHM technologies in manufacturing operations continues to experience growth driven by advances in sensor, computing, and communications technologies, and in machine learning and other data analytic techniques. An increased interest in PHM within manufacturing is also reflected in recent academic literature. Yoon, He, and Van Hecke (2014) applied PHM to an additive manufacturing process for improved fault diagnosis and quality control. Philippot, Marang, Gellot, Ptn, and Riera (2014) suggest a fault tolerant control structure for manufacturing plant control. The self-aware machine platform for application in a manufacturing shop floor proposed by Liao, Minhas, Rangarajan, Kurtoglu, and de Kleer (2014) provides a richer set of PHM information, including predicted component wear and real-time anomaly detection to the shop supervisor. However, there is a notable absence of methodologies to support the development of agile and flexible PHM systems in smart manufacturing environments (Peng, Dong, & Zuo, 2010).

Ideally, PHM would be available at the system level, including prognostic intelligence being propagated up the hierarchical structures that relate components to machines, machines to work cells, and work cells to production lines. Model-based diagnostic methods that have been developed for hierarchical aerospace systems may be applied to hierarchical manufacturing systems. For example, Narasimhan and Brownston (2007) suggested a general
framework for stochastic and hybrid model-based diagnostics for aerospace systems. Feldman, de Castro and van Gemund (2013) proposed a decision support framework for satellite systems that uses active testing to increase diagnostic accuracy. Biswas and Mahadevan (2007) also proposed a framework for system health management that includes fault detection, fault identification, and adaptive control for aerospace applications. In the manufacturing domain, Celik, Lee, Vasudevan, and Son (2010) applied a dynamic data-driven framework on a supply chain system to perform multi-fidelity simulation. Ferri, Rodrigues, Gomes, de Medeiros, Galvo, and Nascimento (2013) have suggested a method for achieving system-level PHM by propagating the remaining useable life (RUL) along the fault tree structure of the manufacturing system model. This is a positive step in creating a methodology for achieving system-level PHM within Smart Manufacturing based on the system model and component-level PHM.

To address the existing gap in providing PHM for hierarchical manufacturing systems, we propose a methodology termed Adaptive Multiscale PHM (AM-PHM). The AM-PHM methodology is designed to support PHM in Smart Manufacturing Systems (SMS). AM-PHM is characterized by its incorporation of multi-level, hierarchical relationships and PHM information gathered from a manufacturing system. AM-PHM utilizes diagnostic and prognostic information regarding the current health of the system and constituent components, and propagates it up the hierarchical structure. By doing so, the AM-PHM methodology creates actionable diagnostic and prognostic intelligence along the manufacturing process hierarchy. This information includes the predicted health state upon completion of a task. The AM-PHM methodology allows for more intelligent decision-making to increase efficiency, performance, safety, reliability, and maintainability.

AM-PHM, at a given level along the system hierarchy, uses operational profiles from adjacent, higher-level operational profiles. These profiles describe the production goals under consideration by the decision-makers (e.g., operators and supervisors) at the higher level. In addition to the traditional workload, bill of materials, and requirements of the manufacturing process, the operational profile may have a focused objective such as minimizing cost or maximizing reliability. One instantiation of the AM-PHM concept may be as an AM-PHM module situated at every node along the hierarchical structure. The AM-PHM module gathers PHM information from subordinate systems or components and makes a decision ideal for the task corresponding to the operational profile. The AM-PHM module then creates operational profiles for its subordinate AM-PHM modules while producing diagnostic and prognostic information for its higher-level subsystem.

An example robotic assembly process is selected to show the effectiveness of the AM-PHM methodology. In today’s manufacturing world, the finished products/goods are becoming more complex as machines with increased capabilities are being deployed to the manufacturing floor. One example is the utilization of the industrial robot.

Worldwide, the manufacturing landscape has experienced extensive growth in the development and deployment of new robotic technologies. Paired with the introduction of newer, cheaper, and more reliable sensing technologies, the capabilities of robotic systems have improved in a relatively short amount of time. Processes that were historically performed by manual labor may now be accomplished using robots. As such, the use of robots outside of the automotive and electronics industries is on the rise (Orcutt, 2014).

Global manufacturing initiatives are stressing the development and integration of smart manufacturing technologies in modernized manufacturing facilities. Such technologies are seen as key to maintaining economic stability within an increasingly competitive global market (Holdren et al, 2011).

Robotic assembly is expected to be a principle application of robotics in manufacturing (Marvel & Falco, 2012). Historically, mechanical assembly has been addressed by manual labor. However, advancements in robotic perception, force control, and kinematic dexterity have enabled robotics to be viable options for assembly applications. This expands the traditional application suite of material handling, painting, and welding that have been more typical of robotic operations in manufacturing. Moreover, with the introduction of collaborative robot technologies, the expansion of robotics is expected to positively impact manufacturing processes that remain largely manual in nature (Marvel, 2014).

With the anticipated integration of robots into both new and preexisting manufacturing lines, the quality of PHM will directly influence the effectiveness of interoperability and system performance. This is particularly true when humans are expected to work alongside robotic collaborators, where robot performance also impacts safety. Should a robotic system experience a failure, it is expected to do so in a safe, reliable manner that does not negatively impact its environment, process, or collaborators. Moreover, the road to recovery must be clearly established and easy to implement. This necessitates significant advancements in the quality and dissemination of robotic PHM.

The remainder of the paper is organized as follows. Section 2 examines the current state of PHM capabilities and standards in manufacturing. Section 3 presents the AM-PHM methodology including the proposed AM-PHM features for describing the health state of systems. Section 4 discusses two example implementations of the AM-PHM methodology as applied to a test robotic assembly production line scenario. Section 5 concludes the paper by highlighting the significance of AM-PHM in manufacturing.
2. CURRENT STATE OF PHM IN SMART MANUFACTURING

PHM technologies in manufacturing systems reduce time and costs for maintenance of products or processes through efficient and cost-effective diagnostic and prognostic activities. In 2010, a comprehensive review was conducted of prognostic and diagnostic methodologies for condition-based maintenance (CBM) that presented the existing strategies within four categories: physical models, knowledge-based models, data driven models, and combination (hybrid) models (Peng et al., 2010). This review highlighted many specific methods across four categories (Hidden Markov Models, Bayesian network-related methods, Fuzzy Logic, Principal Components Analysis) along with their successes and limitations. No method stood out as being sufficient to provide both diagnostic and prognostic intelligence at multiple levels. This review demonstrated that for every method’s strength, there was at least a single weakness. Similarly, another review of existing methods for manufacturing systems was conducted in 2012 that focused on comparing time-based maintenance (TBM) and condition-based maintenance (CBM) (Ahmad & Kamaruddin, 2012). TBM, commonly referred to as preventative maintenance, is typically simpler to implement (in that maintenance is scheduled based upon a specific unit of time; e.g., cycle time) while CBM, sometimes termed predictive maintenance, may ultimately be more cost effective if a process’s or equipment’s health data accurately reflects its current state and allows a machine to run longer until maintenance (as compared to a TBM schedule). The challenge in CBM is gathering sufficient data to make a reasonably accurate prediction.

Product PHM (providing health monitoring, diagnostics, and/or prognostics for a finished system; e.g., automobile, aircraft, power generation station) is more widespread as compared to process PHM (providing health monitoring, diagnostics, and/or prognostics to a system that integrates one or more pieces of equipment to complete a task; e.g., assembly process, welding process, machining process) (Batzel & Swanson, 2009) (Holland Barajas, Salman, & Zhang, 2010) (Hu & Koren, 1997) (Shen, Wan, Cui, & Song, 2010). Likewise, PHM techniques have been developed and applied more widely at component/equipment levels, yet some work has occurred at the higher/system levels. For example, innovative methods have been developed to support various machining operations (Al-Habaibeh & Gindy, 2000) (Altintas, Verl, Brecher, Uriarte, & Pritschow, 2011) (Biehl, Staufenbiel, Recknagel, Denkena, & Bertram, 2012) (Borisov, Fletcher, Longstaff, & Myers, 2013). System-level PHM methods have also been developed, yet seem to be focused in their applicability and/or limited in capability (Barajas & Srinivasa, 2008) (Dutta, Jize, Maclise, & Goggin, 2004) (Hofmeister, Wagoner, & Goodman, 2013).

Vogl et al. (2014) conducted a detailed review of existing standards that were designed to help guide implementation of PHM in manufacturing. Specifically, many of the current PHM standards were developed within the International Organization for Standardization (ISO) and focus primarily on condition monitoring and diagnostics (ISO, 2002) (ISO, 2003) (ISO, 2012). Few standards include discussion of prognostics (ISO, 2004). Most standards fall into one of two categories; standards that are very specific and only applicable to a few processes and standards that are very broad that may lack guidance for applications. Likewise, no standard has been developed that offers the flexibility to be applied at multiple hierarchical levels of a complex system to promote effective PHM practices.

3. ADAPTIVE MULTISCALE PHM FOR SMART MANUFACTURING

A manufacturing system hierarchical structure can be described as a facility consisting of multiple assembly/fabrication lines which are further divided into work cells or work stations which are further divided into multiple machines (Hopp & Spearman, 2008). For this paper, the hierarchical structure of the facility, assembly line, work cell, and machine will be used as a primary example, although there exists more complex methods of describing a manufacturing facility.

Information is passed down in the form of orders, schedules, bills of materials, or control signals between each hierarchical level of the system. The job of the subordinate system is to follow the tasks assigned by the higher-level node. Historically, maintenance policies for machines have been based on usage time or workload, as static policies defined in these terms can be estimated through historical data and experience. An effort to modify this approach into a feedback system where the health state of the machine or component is considered in making maintenance decisions emerged only recently. (National Institute of Standards and Technology, 2015) However, health state information is often confined to the component or machine level and is not propagated up to the system level.

On the other end of the spectrum, the system-level approach to analyzing a manufacturing system has resulted in generalized risk and fault analysis methods such as fault tree analysis (FTA) and failure mode and effects analysis (FMEA) (SAE International, 2009). Also, modeling software tools such as SysML have been used to describe the system structure including interoperability and interdependency between components of the system (Wünsch, Lüder, & Heinze, 2010).

The AM-PHM methodology is designed to provide decision-makers with enhanced information on the current and predicted health state of the decision-maker’s subsystems. Figure 1 depicts the AM-PHM methodology for a simple hierarchical manufacturing structure.
Figure 1. Conceptual representation of AM-PHM

For AM-PHM, a decision-maker is not limited to the machine operator. Rather, it refers to any person or machine such as the control unit of a manufacturing robot or the supervisor of an assembly line that is responsible for making decisions that can influence the outcome of the system. The point at which the decision-maker resides in the hierarchical structure is called the decision point within the AM-PHM methodology. Conceptually, an AM-PHM module resides at every decision point of the hierarchical structure of the manufacturing system.

A hierarchical manufacturing system refers to a manufacturing system in which multiple levels exist. For each level, the higher-level nodes encompass the lower-level nodes. In this level structure, the parent nodes have control over the states of its subsystems while subsystems do not have direct control over the states of its parent nodes. Examples of the hierarchical structure may be a SysML description or a fault tree structure of the manufacturing system. Another example may be a treelike description of the physical setup of a manufacturing system consisting of assembly lines, work cells, and machines.

For the example structure shown in Figure 1, an order is placed to the Facility Manager with the number of products requested, product requirements, and expected finish date. The order information and the operational directive are passed onto the facility level AM-PHM module. The directive refers to a particular set of attributes or objectives that the decision-maker would like to focus on. For example, the decision-maker may be interested in reducing the time, cost, risk, or wear in maximizing the utilization rate.

The facility level AM-PHM module reports the health information of the facility to the Facility Manager. PHM information on the subsystem is needed for effective directive-driven decisions to be made. The PHM information from the subsystem is processed at each AM-PHM module. This results in health metrics that appropriately represent the current and future state of the system. These health metrics may include remaining usable life of the system, expected health state upon completion, nature of fault, and proposed solutions.

The AM-PHM module also creates operational profiles once all aforementioned information is gathered. Each operational profile is designed to control the subsystems with a focused directive. The operational profile also contains the projected health information for the subordinate systems such as projected health upon completion. The decision-maker may now choose from the set of operational profiles that fit within the constraints handed down from its superior nodes.

The Facility Manager selects the operational profile that best fits the directive and order requirements. Once the operational profile is chosen, the set of instructions contained within that operational profile are handed down to the subordinate AM-PHM module and a similar process repeats itself. For the example in Figure 1, the selected operational profile containing the number of products needed to be produced by each production line and operational directive is passed down to the Assembly Line level.

A similar process is now repeated at the Assembly Line level. The Assembly Line Manager takes the operational profile handed down from the Facility level and selects an appropriate operational profile. The operational profile handed down to the Work Cell level contains information such as number of products produced for a particular work cell and bill of materials needed for the processing of the order.

A similar process is repeated for the Machine level. For the Machine level the operational profile contains machine operation parameters and the AM-PHM information contain data such as the aggregated wear for critical components.

Although the simplified scenario depicted in Figure 1 is convenient for initial discussion of the AM-PHM concept, the concept may be expanded for more general SMS...
environments in which there exists an extensive hierarchy of processes and components.

Additional features that better describe the current health state at a particular juncture of the system are needed for the AM-PHM system to be helpful to the decision-maker. The newly suggested features are (a) greatest wear, (b) average wear, (c) health balance score, (d) probability of successful completion, and (e) estimated health state upon completion.

(a) Greatest wear represents the most extreme wear in percent from all the wear states of all the subordinate components. This gives an idea of the state of the most worn component of the system.

(b) Average wear represents the arithmetic weighted mean of the wear in percentage of all the subordinate components. This metric represents the overall average health state of the system. The average on its own may not reveal much information but in conjunction with the greatest wear and the health balance score, this helps to describe the health state of the all the components of the subsystem. Different components contribute differently to the overall performance of a manufacturing system. There are established methods such as FTA, FMEA, Hierarchical Holographic Modeling (HHM), and Risk Filtering, Ranking, and Management (RFRM) that may be used to analyze the weight of each component to different failures. The differing importance of a component is included as a weighted coefficient.

(c) Health balance score is the standard deviation of the wear state of each of the subordinate components at a given node. This metric indicates degree of concentration of wear of the system. A higher number would indicate that wear values vary greatly among components, while a smaller number would indicate that the system has similar wear along most of its components.

(d) Probability of successful completion is the probability that the component will complete the given operational profile with the current state of health. This gives decision-makers an idea of the success rate or confidence involved with a given solution.

(e) Estimated health state upon completion refers to the expected final state of health for all metrics involved in AM-PHM. This is used to show a predicted picture of the overall state of health at the point of completion of the assigned task.

The average wear, health balance score, and probability of successful completion may be further customized so that each of the components carry different weights. This means the proposed metrics can focus on certain components depending on its importance within the overall structure of the system.

One notable point for the proposed features is that the basis for the usefulness of these metrics lies on the assumption that the PHM information from the component level is accurate to a certain degree. An accurate wear model is necessary for the health metrics to be useful.

4. AM-PHM IMPLEMENTATION IN SMART MANUFACTURING ROBOTIC ASSEMBLY

An example assembly line involving multiple robots is described in this section. The AM-PHM methodology is applied to the canonical manufacturing process simulation. The canonical process is a generalized test case of the example assembly line and includes related assumptions. This simplified test case, including its simulated results, highlights the usefulness of the AM-PHM implementation. The structure and the trend of the numbers involved are reasonable in real manufacturing settings. The use of robotic arms in industry is widespread as stated in Snyder (1985) and the trend of the drill wear in the canonical example simulation follow the model by Kadigama, Abou-El-Hossein, Noor, Sharma and Mohammad (2011).

The example hierarchical structure of a manufacturing environment consists of a single assembly line with multiple work cells, each of which has multiple machines, each comprised of multiple components. The operational profiles flow from the higher-level block to the lower-level blocks in the AM-PHM framework. The PHM information is reported from the lower-level blocks up to the higher-level block. However, both the operational profile and the PHM information are processed appropriately for each level.

The specific information that is listed in the operational profiles and the PHM reports differ depending on the block's location in the hierarchical structure. For example the operational profile generated by the assembly line for each work cell will resemble a bill of materials; the operational profile generated by the work cell for each machine will resemble a process instruction; and the operational profile generated by the machine to its components will be close to a set of control signals.

The operational profile generator of the AM-PHM module at each level must translate the task it receives from the higher-level AM-PHM module into a task that can be understood by the subordinate level. Similar concepts apply to the PHM information at each stage. The PHM information from the component to the machine will include RUL of replaceable parts, while the PHM information from the machine to the work cell includes more information on the tradeoffs involved with different operational profiles. Finally, the PHM report from the work cell to the assembly line would include more information on the probability of successful completion and the overall health state of the work cell. The AM-PHM module must process the PHM information it receives from the lower levels and provide value-added, level appropriate information for the upper level.
Two different examples of AM-PHM are given. The first example is focused on a simple AM-PHM structure with simple operational profiles and a PHM report involving only RUL. This structure may be implemented if the nature of the task performed at an assembly line does not require sophisticated PHM capabilities or if changing the existing system model and fault tree structure is not desired. The deployment of AM-PHM into the existing assembly line model is minimally invasive and most likely will not affect the overall structure of the fault tree.

The second example is a more sophisticated AM-PHM system. This is needed if the assembly line handles a more complex process involving many different machines with interdependencies and interoperability. The downside is that the implementation may become more complicated and the use of AM-PHM may impact the existing fault tree structure of the system model.

In the canonical example, a robot with two drilling arms is used to drill holes into a box. The left and right drilling arm are each responsible for drilling holes into the left and right side of the box, respectively. A SysML model of the drilling robot is presented in Figure 2. The corresponding FTA diagram of the drilling robot is shown in Figure 3. Only the flank wear of the drill bit component on each arm is considered for the simplified AM-PHM example, as flank wear is one of the common wears exhibited in drilling (Kadirgama et al. 2011). It is important to note that one drilling arm may perform the job of the other drilling arm with the penalty of reduced production rate.

In real-world manufacturing systems, there are many factors such as material properties, work piece structure, and machine characteristics that are carefully considered when selecting machining parameters. Machining parameters are optimized to best fit the particular manufacturing process. However, in a complex system-of-systems, optimization based on one feature means there is a trade-off with other features. Also, for a particular process there is a range of acceptable machining parameters rather than one fixed operating point (Furness, Wu, and Ulsoy, 1996). Drill bit manufacturers recommend a range of feed rates and cutting speeds for their drill bits (Sandvik Coromant, 2005).

When the parameters for a process are selected, the model [for the process] does not account for the fact that the system may change as the machine experiences wear in its components. The wear of the components, such as the flank wear of the drill bit, affects the characteristics of the system. Thus, the optimal operating parameters may need adjustment to account for the change in the system caused by the deteriorating health state of the machine.

For the canonical process example simulation, simplifications are made to emphasize the effect of the AM-PHM methodology and to reduce the complexity of the example. The drilling robot is tasked to drill 100 holes on the left and right side of the box. The left and right drilling arm each drill on their respective sides, simultaneously. Though there are several different types of wear involved with the drill bit, only the flank wear occurring on the cutting edge of the drill bit is considered.

The work piece is made of Nickel alloy with a Brinell hardness of 200. The production line has identified an acceptable and stable range of operating parameters. The cutting speed is between 100 m/min to 180 m/min. The feed rate is between 0.1 mm/rot and 0.2 mm/rot. Each hole has a cutting depth of 1.5 mm and the drill diameter is 10 mm. Expected tool life is different for different combinations of cutting speed and feed rate and follows the values stated by Kadirgama et al. (2011).

The drill bit is considered completely worn and reached its replacement point when there is 0.3 mm of flank wear. The RUL or tool life depends on the machining parameters and the replacement threshold for the drill bit. Tool life also differs depending on the size and geometry of the drill bit. Thus, to provide a more comparable quantitative figure for the amount of wear, the wear is presented as a percentage. The wear percentage is calculated by dividing the remaining tool life by the tool life for a new tool.

![Figure 2. SysML description of drilling robot](image-url)
In this case, the actual implementation of the AM-PHM is done through the use of FTA as an intermediate, semi-automated step of linking system-level hierarchical information and component-level health information. Only the RUL given at each level is used as the source of health information.

The work cell is tasked to build 20 boxes. The starting wear state of the individual drill bits are 85 % worn for the left drill in Robot 1 and new for all other drill arms. The default operating speed is set to a feed rate of 0.2 mm/rot and 120 m/min. This results in a wear rate of 15 % per minute for the drill bit and a production rate of 5 boxes per minute. The component-level RUL is calculated based on this initial condition. The component-level RULs show that for Robot 1, the left arm has an RUL of 1 minute and the right arm has an RUL of 6.6 minutes. For Robot 2, both the left and right arm has an RUL of 6.6 minutes. This information is propagated along the hierarchical structure according to the rules. Robots 1 and 2 each result in RULs of 1 minute and 6.6 minutes, respectively. The decision to distribute the load to the two robots is made based on the production targets and RUL information by the work cell operator. A work load of five boxes is assigned to Robot 1 and a work load of 15 boxes is assigned to Robot 2. The job takes three minutes to complete and the final RUL upon completion for each robot is 0 and 3.6 minutes, respectively. The complete results including additional information on the health of the work cell are presented in Table 1.

The result shows that the system has an RUL of 3.6 minutes. This is information previously unattainable to the decision-maker. Utilization of the RUL information enables more efficient use of the components of the manufacturing system. The advantage of this degree of PHM reflection is that at any point in the hierarchical structure the same RUL calculation method can be applied again reducing the complexity of implementation. The upper-level RUL is calculated using simple multiplication and comparison process. This begins by converting the fault tree (consisting of logic AND and OR gates) to a sum of products (SOP) expression. Once the SOP expression for the system is obtained, the system RUL is calculated by multiplying the probability distribution of the RULs for the product terms of the expression. The next step is to select the appropriate RUL for the sum portion of the expression. The system RUL ends up highlighting the set of components that are contributing to the nearest expected system failure. However, the system RUL does not contain health information on the other components of the system that are not directly tied to the upcoming failure. This limits the range of intelligent decisions that can be made.
4.2. Full Implementation of AM-PHM

A more sophisticated implementation of the AM-PHM concept would be to introduce additional features that help convey timely information on the health state of the system. The new features used in this example are the health balance score, average wear state, worst wear state, and estimated wear state upon completion. An order to make five boxes was given to the work cell as with the previous example. For the starting health state, only the right drill arm's wear state is at 75% while all other components are new.

The PHM information from a subordinate component is conveyed to the upper-level AM-PHM module. The collected PHM information is processed to produce the PHM information at the current node. The cutting speed and feed rate parameters are changed to a different operating point within the stable and acceptable range. Work load is changed and the expected results are calculated for all the different parameters. The drill bit wear trend follows the model suggested by Kadirgama et al. (2011). The production rate is changed by adjusting the cutting speed and feed rate which effects the wear rate of the drill bit. According to Furness, Wu and Ulsoy (1996) the feed and speed have relatively small effects on the drill hole quality and that the drilling feed and speed is limited by factors such as drill wear. The drill speed parameters may be adjusted within a certain confine without significantly affecting the hole quality. The final decision is made from the set of choices that best fits the operational directive. The results for this simulation are given in Tables 2 and 3.

For the case in Table 2, the work cell was handed down orders to produce 20 boxes with a directive of minimum health balance. Low balance score means that the components are at a similar state of health and may be used to align maintenance points for the components. The chosen operational profile distributes a load of five boxes for the first robot and 15 boxes for the second robot. However, the cutting speed is adjusted to 100 m/min and the feed rate is also adjusted to 0.1 mm/rot. The production rate is slowed down to 2.1 box/min as a result which reduces the wear of Robot 1’s drill bits to 0.02 mm/min or 6.6% of its tool life per minute. This results in the production taking approximately 2.1 minutes.

For the case in Table 3, the work cell is also ordered to produce 20 boxes but with a directive of minimum time. The operational profile chosen suggests a cutting speed of 180 m/min and a feed rate of 0.2 mm/rot. The production rate is increased to 7.6 boxes per minute at the cost of seeing 0.1 mm of flank wear per minute or 33% of reduction in tool life per minute. The left drill bit reaches its failing point after 30 seconds and the right drilling handle the job of drilling holes on the left side as well which reduces the production rate for Robot 1. Production is completed in 1.5 minutes at an increased cost on the wear of the drill bits.

The AM-PHM methodology is being applied in a simulated environment that is designed to resemble real-world hierarchical manufacturing systems. The canonical example simulation is based on real-world drill bit wear trends. For simplicity, in this paper, tool life is only dependent upon the operating parameters since the material stays consistent. The AM-PHM suggests operating points by optimizing a weighted cost function. The cost function includes all the health related features. The weight used in the cost function is only dependent upon the parameters are based on existing stable operating conditions to ensure system stability.

Table 1. AM-PHM based manufacturing results using RUL

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production Rate of Robot 1 (box/min)</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RUL of Robot 1 (min)</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Production Rate of Robot 2 (box/min)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>RUL of Robot 2 (min)</td>
<td>6.6</td>
<td>5.6</td>
<td>4.6</td>
<td>3.6</td>
</tr>
<tr>
<td>Produced (box)</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>RUL of Work Cell 1 (min)</td>
<td>6.6</td>
<td>5.6</td>
<td>4.6</td>
<td>3.6</td>
</tr>
</tbody>
</table>

Table 2. AM-PHM results based on maximum mean health

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production Rate of Robot 1 (box/min)</td>
<td>2.1</td>
<td>2.1</td>
<td>2.1</td>
<td>-</td>
</tr>
<tr>
<td>RUL of Robot 1 (min)</td>
<td>15.15</td>
<td>14.15</td>
<td>13.15</td>
<td>13</td>
</tr>
<tr>
<td>Production Rate of Robot 2 (box/min)</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>-</td>
</tr>
<tr>
<td>RUL of Robot 2 (min)</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Produced (box)</td>
<td>0</td>
<td>9.7</td>
<td>19.4</td>
<td>20</td>
</tr>
<tr>
<td>RUL of Work Cell 1 (min)</td>
<td>15.15</td>
<td>14.15</td>
<td>13.15</td>
<td>13.15</td>
</tr>
</tbody>
</table>

435
The canonical simulation used in this example is based on models from literature. In the future the AM-PHM methodology will be applied to real-world data some of which is obtained from actual production facilities. The real data will also include a more detailed wear model in which the wear rate is also dependent on additional factors such as current state of wear and material properties.

Table 3. AM-PHM result using maximum health balance and minimum time

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>Production Rate of Robot 1 (box/min)</th>
<th>RUL of Robot 1 (min)</th>
<th>Production Rate of Robot 2 (box/min)</th>
<th>RUL of Robot 2 (min)</th>
<th>Produced (box)</th>
<th>RUL of Work Cell 1 (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.6</td>
<td>3</td>
<td>7.6</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>5.7</td>
<td>2</td>
<td>7.6</td>
<td>2</td>
<td>15.2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3.8</td>
<td>1</td>
<td>7.6</td>
<td>1</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>1</td>
</tr>
</tbody>
</table>

5. CONCLUSION

The concept of Adaptive Multiscale PHM for manufacturing was introduced in this paper. The AM-PHM methodology calls for the AM-PHM module at each decision point along the hierarchical structure to receive operational profiles outlining the job requirements and report back performance and health estimates appropriate for the upper level.

The AM-PHM is demonstrated on a canonical test manufacturing scenario simulation. Directive oriented decisions were made in the simulation by using additional information on the health of the system in addition to knowledge on the system hierarchical model. The AM-PHM shows promising results as it enables manufacturing work cells to adapt to changing machine conditions.

Further development of the AM-PHM methodology will continue. A modified work cell canonical process is in development. This model is based on a real-world manufacturing facility. A canonical process work cell simulator capable of simulating continuous wear of the components is being developed. The AM-PHM will be tested using this simulation environment and will be compared against other existing PHM based decision-making policies. The results of the different policies will be compared using quantitative measures such as time, monetary cost and Overall Equipment Effectiveness (OEE).

REFERENCES


**Biographies**

**Benjamin Y. Choo** is in the Ph.D program of the Systems and Information Engineering Department at the University of Virginia (UVa). He received his B.S. and M.S. degree from the Electrical Engineering Department at Yonsei University, Korea in 2005 and 2007 respectively. He received his M.E degree in Electrical Engineering from UVa in 2012. His research interests include manufacturing systems, machine learning and 3D depth sensors.

**Dr. Peter A. Beling** is an associate professor in the Department of Systems and Information Engineering at the University of Virginia (UVA). Dr. Beling received his Ph.D. in Operations Research from the University of California at Berkeley. Dr. Beling’s research interests are in the area of decision-making in complex systems, with emphasis on adaptive decision support systems and on model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including prognostics and health management, mission-focused cybersecurity, and financial decision-making. He is active in the UVA site of the Broadband Wireless Applications Center, which an Industry-University Cooperative Research Center sponsored by the National Science Foundation.

**Dr. Amy E. LaViers** is an Assistant Professor in Systems and Information Engineering and Director of the Robotics, Automation, and Dance Lab at the University of Virginia. She aims to extract useful features from human movement for robotic applications, such as, endowing co-robots the ability to work alongside human workers in manufacturing plants. Her research began at Princeton University where she earned a certificate in Dance and B.S.E. in Mechanical and Aerospace Engineering. She went on to complete a M.S. and Ph.D. in Electrical and Computer Engineering at the Georgia Institute of Technology.

**Dr. Jeremy A. Marvel** is a project leader and research scientist in the Intelligent Systems Division of the National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Dr. Marvel received his Ph.D. in 2010 in computer engineering from Case Western Reserve University in Cleveland, OH. Since joining the research staff at NIST, he has established the Collaborative Robotics Laboratory, which is engaged in research dedicated to developing test methods and metrics for the performance and safety assessments of collaborative robotic technologies. His research focuses on intelligent and adaptive solutions for robot applications, with particular attention paid to human-robot collaborations, multi-robot coordination, safety, perception, self-guided learning, and automated parameter optimization. Jeremy is currently engaged in developing measurement science methods and artifacts for the integration and application of robots in collaborative assembly tasks for manufacturing.
Dr. Brian A. Weiss has a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the Smart Manufacturing Operations Planning and Control program and the Project Leader of the Prognostics and Health Management for Smart Manufacturing Systems project within the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). Prior to his leadership roles in the SMOPAC program and the PHM4SMS project, he spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles; tactical applications operating on Android devices; advanced soldier sensor technologies; free-form, two-way, speech-to-speech translation devices for tactical use; urban search and rescue robots; and bomb disposal robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 & 2008), and the Jacob Rabinow Applied Research Award (2006).
On Accurate and Reliable Anomaly Detection for Gas Turbine Combustors: A Deep Learning Approach

Weizhong Yan\textsuperscript{1} and Lijie Yu\textsuperscript{2}

\textsuperscript{1}General Electric Global Research Center, Niskayuna, New York 12309, USA
yan@ge.com

\textsuperscript{2}General Electric Power & Water Engineering, Atlanta, Georgia 30339, USA
Lijie.Yu@ge.com

\begin{abstract}
Monitoring gas turbine combustors’ health, in particular, early detecting abnormal behaviors and incipient faults, is critical in ensuring gas turbines operating efficiently and in preventing costly unplanned maintenance. One popular means of detecting combustors’ abnormalities is through continuously monitoring exhaust gas temperature profiles. Over the years many anomaly detection technologies have been explored for detecting combustor faults, however, the performance (detection rate) of anomaly detection solutions fielded is still inadequate. Advanced technologies that can improve detection performance are in great need. Aiming for improving anomaly detection performance, in this paper we introduce recently-developed deep learning (DL) in machine learning into the combustors’ anomaly detection application. Specifically, we use deep learning to hierarchically learn features from the sensor measurements of exhaust gas temperatures. And we then use the learned features as the input to a neural network classifier for performing combustor anomaly detection. Since such deep learned features potentially better capture complex relations among all sensor measurements and the underlying combustors’ behavior than handcrafted features do, we expect the learned features can lead to a more accurate and robust anomaly detection. Using the data collected from a real-world gas turbine combustion system, we demonstrated that the proposed deep learning based anomaly detection significantly indeed improved combustors’ anomaly detection performance.

Deep learning, to the best of our knowledge, has not been used for any PHM applications, however. It is our hope that our initial work presented in this paper would shed some light on how deep learning as an advanced machine learning technology can benefit PHM applications and, more importantly, can stimulate more research interests in our PHM community.

1. INTRODUCTION

A combustion system is a critical component of gas turbines that burns fuel air mixture to create thrust or power. A heavy-duty industrial combustor typically operates under high temperature and high flow rate conditions that introduce significant thermodynamic stress to combustor components. Imbalanced fuel distribution and combustion instabilities are the main causes of different combustors’ abnormalities, including fuel nozzle faults, liner cracks, transition piece defects, excessive vibration due to acoustic waves and heat release oscillations, and non-compliant emissions (Allegorico & Mantini, 2014). Those abnormalities, if not detected early, could lead to catastrophic combustor failures or lean blowout, which trigger turbine trips; those abnormalities could also adversely affect the life of hot gas path components, or result in higher NOx and CO emissions. Consequently, reliably detecting abnormal behaviors and incipient faults earlier is important in ensuring gas turbines operating efficiently and in preventing costly turbine trips.

Combustor anomaly detection is technically challenging because gas turbine combustors are an extremely complex system, of which the operating conditions are heavily dependent on many factors, such as, machine type, fuel used, ambient conditions, and equipment aging.

Monitoring the exhaust gas temperatures measured at the gas turbine exhaust section is a popular means for detecting the combustor abnormalities (Allegorico & Mantini, 2014). Exhaust temperature profiles provide valuable information.
about thermal performance of gas turbines and combustors, thus can be indicative to combustor health conditions.

Traditionally, for combustor anomaly detection, knowledge-based rules are applied to the exhaust temperature profiles. Such knowledge-based rules not only have inadequate detection performance (detection rate and false alarm rate), but also are laborious in designing and developing the rules. Aiming for more accurate and robust detection of combustors’ incipient faults, thus for reducing unplanned downtimes and operation costs, in recent years we at GE have been pursuing advancing our anomaly detection technologies from the traditional knowledge-based rules to knowledge-augmented data-driven approaches. Specifically for combustor anomaly detection, we have explored different data-driven, machine learning technologies, such as SVM, random forests, and neural networks. Using advanced machine learning modeling techniques has made certain degree of improvement in detection performance, but not as significantly as we would like. We observed that it is the feature engineering, a process of extracting appropriate features or signatures from raw sensor measurements, which made bigger difference in combustors’ detection performance.

In our early work we handcrafted a set of features based on domain and engineering knowledge of gas turbine combustors. Using such handcrafted features for our anomaly detection models yielded better detection performance than directly using raw exhaust temperatures for combustors’ anomaly detection problems; however, handcrafting features is a manual process that is very much problem-specific and un-scalable. Thus it would be of great value if somehow we can automate the feature generation process. Deep learning (DL) is a sub-field of machine learning that involves learning good representations of data through multiple levels of abstraction. By hierarchically learning features layer by layer, with higher-level features representing more abstract aspects of the data, deep learning can discover sophisticated underlying structure and features. In recent years deep learning has attracted tremendous research attention and proven outstanding performance in many applications including image and video classification, computer vision, speech recognition, natural language processing, and audio recognition (Arel et al., 2010).

Inspired by the success of deep learning in many other domains, in this paper we explore how deep learning can benefit PHM applications in general and combustor anomaly detection applications in particular. Broadly speaking deep learning has two types: supervised and unsupervised. Unsupervised feature learning, i.e., using unlabeled data to learn features, is the key idea behind the self-taught learning framework (Raina et al., 2007). Unsupervised feature learning is well suited for machinery anomaly detection since for PHM applications abundant unlabeled data are available and easily accessible, while accurately labeling industrial data is costly and, often time, impossible due to uncertainty of true events.

Deep learning, to the best of our knowledge, has not been used for any PHM applications, despite its success in many other domains. Our initial work presented in this paper can hopefully shed some light on how deep learning, as an advanced machine learning technology, can benefit PHM applications and, more importantly, our work here can hopefully stimulate more research interests in our PHM community.

The remaining of the paper is organized as follows. Section 2 provides related work on both anomaly detection and feature engineering & feature learning as well. We then give details on our methodology of using deep learning for combustor anomaly detection in Section 3. Use case study and its results are given in Section 4. We conclude our paper in Section 5.

2. RELATED WORK

2.1. Anomaly detection

Anomaly detection, a technique for finding patterns in data that do not conform to expected behavior, has been extensively used in a wide range of applications, such as fraud detection in credit card and insurance industries, intrusion detection in cyber-security industry, fault detection in industrial analytics, to name a few. Survey papers, for example, Chandola et al. (2009), provide a comprehensive review of different anomaly detection methods and applications.

Anomaly detection has been actively applied to different PHM applications including: aircraft engine fault detection (Tolani et al., 2006), wind turbine fault detection (Zaher et al., 2009), locomotive engine fault detection (Xue & Yan, 2007), marine gas turbine engine (Ogbonnaya et al., 2012), and combined cycle power plants (Arranz et al., 2008), to name a few.

There are a few studies specifically on combustor anomaly detection. For example, Mukhopadhyay and Ray (2013) used symbolic time series analysis for detecting lean blow-out in gas turbine combustors. The time series data they used for analysis were optical sensor data from the photomultiplier tube (PMT). In the work by Chakraborty et al. (2008), the tailpipe wall friction coefficient was proposed as the failure precursor to flame out of thermal pulse combustors and several data-driven techniques (information theory, symbolic dynamics and statistical pattern recognition) were applied to pressure oscillation signals for estimating the friction coefficient of the tailpipe wall. One work that mostly relates to our study in this paper is by Allegorico and Mantini (2014). Similar to ours, they also performed combustor anomaly detection based on exhaust temperature thermocouples. They formulated the anomaly
detection as a classification problem and used traditional neural networks and logistic regression as the classifiers. However, they didn’t do any feature engineering to extract features. Rather they directly used the exhaust temperature profile as the inputs to classifier, which showed a reasonable detection performance on the small dataset the authors picked, but may not generalize well in real applications.

2.2. Feature engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data (Brownlee, 2014). Feature engineering is arguably a critically important task in developing predictive solutions (Domingos, 2012); and at the same time it is also a challenging but the least well-studied topic in machine learning and data-mining (Brownlee, 2014). That is because feature engineering is a very much problem-specific, manual process that is typically performed by machine learning experts in conjunction with domain experts.

As features are highly application dependent, there is almost no universal feature set that works well for all applications. Over the years, though, many application domains do have developed a number of application-specific features that are popularly used. For example, frequency of each word in the bag-of-words for document classification, scale-invariant feature transform (SIFT) for object recognition (Lowe, 1999), and Mel-frequency cepstral coefficients (MFCC) for speech recognition (Davis and Mermelstein, 1980), and defect frequencies for vibration analysis. These commonly used features serve as a good starting point for feature engineering.

In literature, publications specific on feature engineering are very sparse as stated by Brownlee (2014) that “feature engineering is another topic which doesn’t seem to merit any review papers or books, or even chapters in books”. Recently there are a few attempts on developing feature engineering tools that aim for facilitating the feature engineering task. For example, Anderson et al. (2014) proposed a feature engineering development environment that allows the user to write feature engineering code and evaluate the effectiveness of the engineered features. Heimerl et al. (2012) developed FeatureForge tool that uses interactive visualization for supporting feature engineering for natural language processing.

Feature engineering for PHM applications also attracts researchers’ attention. For example, Yan et al. (2008) provided a survey on feature extraction for bearing PHM applications.

2.3. Feature (representation) learning

Feature learning, also called representation learning, is a sub-field of machine learning where the focus is to learn a transformation of raw data input to a representation that can be effectively exploited in machine learning tasks. Feature learning becomes an active research topic in recent years as deep learning or deep representation learning becomes a hot research topic [NIPS (2014), ICML (2013), and ICLR (2015)]. Deep representation learning has created great impact in the areas such as speech recognition (Deng et al., 2010), object recognition (Hinton et al., 2006), and natural language processing (Collobert et al., 2011). Deep representation learning employs deep learning architecture for feature learning. By stacking up multiple layers of shallow learning blocks, higher layer features learned from lower layer features represent more abstract aspects of the data, and thus can be more robust to variations.

Feature learning can be broadly categorized into unsupervised and supervised learning groups (Wikipedia, 2015). Supervised representation learning includes primarily the traditional multi-layer neural networks and supervised dictionary learning. Unsupervised representation learning, a key idea behind the self-taught learning framework (Raina et al., 2007), covers more techniques, ranging from traditional methods such as PCA, ICA, and k-means, to advanced methods such as autoencoders, RBM, and sparse coding. Unsupervised representation learning has several advantages. For example, the explicitly learned features can be used for different prediction models. Unsupervised representation learning can also be an important component of transfer learning (Bengio, 2011). Successful feature learning algorithms and their applications can be found in recent literature using a variety of approaches, such as RBMs (Hinton et al., 2006), autoencoders (Hinton & Salakhutdinov, 2006), sparse coding (Lee et al., 2007), and K-means (Coates et al., 2011). The most popular building blocks include autoencoder and restricted Boltzmann machines (RBM). Denosing autoencoders (DAE), a variant of classic autoencoders, and its deep counterpart, stacked denoising autoencoders (SDAE) (Vincent et al., 2010), have been used as a representation learning algorithm for several applications, for example, for pose-based action recognition (Budiman et al., 2014), for tag recommendation (Wang et al., 2015), and for handwritten digits recognition (Vincent et al., 2010). SDAE has not been used for PHM applications, however.

3. METHODOLOGY

For combustor anomaly detection concerned in this paper, we adopt unsupervised representation learning scheme. Under this scheme, features are explicitly learned unsupervisingly (without class labels) and the explicitly learned features are then used as input for a separate supervised model (classifier). There are different shallow learning blocks that can be stacked up to form deep feature learning structures. For combustor anomaly detection concerned in this paper we adopt the SDAE proposed by Vincent et al. in 2010 as the unsupervised representation...
learning algorithm, which has the denoising autoencoder (DAE), a variant of autoencoder (AE), as its shallow learning blocks. The main reason we chose SDAE is that denoising autoencoders can learn features that are more robust to input noise and thus useful for classification. The features learned from the SDAE are then taken as the input to a separate NN classifier, extreme learning machine (ELM), for anomaly detection. Figure 1 illustrates both the SDAE for deep feature learning and the ELM for classification for combustor anomaly detection. Both SDAE and ELM are described in detail as follows.

![Overall structure of unsupervised feature learning for combustor anomaly detection](image)

**3.1. SDAE for unsupervised feature learning**

Stacked denoising autoencoder (SDAE), introduced by Vincent et al (2010), is a deep learning structure that has denoising autoencoder (DAE) as its shallow learning blocks. DAE is a variant of classic autoencoder (AE). While details can be found in many references, we provide a brief description of AE and DAE as follows.

An auto-encoder (AE), in its basic form, has two parts: an encoder and a decoder. The encoder is a function that maps an input $x \in \mathbb{R}^d$ to hidden representation $h(x) \in \mathbb{R}^h$, that is, $h(x) = f(Wx + b_h)$, where $f$ is a nonlinear activation function, typically a logistic sigmoid function. The decoder function maps hidden representation $h$ back to a reconstruction $y$: $y = s_g(W'h + b_y)$, where $s_g$ is the decoder’s activation function, typically either the identity function (yielding linear reconstruction) or a sigmoid function.

Autoencoder training consists of finding parameters $\theta = \{W, b_h, b_y\}$ that minimize the reconstruction error on a training set of examples, $D$. That is: $J_{AE}(\theta) = \sum_{x \in D} L(x, g(f(x)))$, where $L$ is the reconstruction error.

The reconstruction error, $L$, can be the squared error $L(x,y) = -\sum_{i=1}^{d}(x_i - y_i)^2$ when $s_g$ is linear; or the cross-entropy loss $L(x,y) = -\sum_{i=1}^{d} x_i \log(y_i) + (1 - x_i)\log(1 - y_i)$ when $s_g$ is the sigmoid.

To prevent autoencoders from learn the identity function that has zero reconstruction errors for all inputs, but does not capture the structure of the data-generating distribution, it is important that certain regularization is needed in the training criterion or the parametrization. A particular form of regularization consists in constraining the code to have a low dimension, and this is what the classical auto-encoder or PCA do.

The simplest form of regularization is weigh-decay which favors small weights by optimizing the following cost function: $J_{AE+wd}(\theta) = \sum_{x \in D} L(x, g(f(x))) + \lambda \sum_{ij} W_{ij}^2$

Another form of regularization is by corrupting input $x$ during training the autoencoder. Specifically, corrupting the input $x$ in the encoding step, but still to reconstruct the clean version of $x$ in the decoding step. This is called denoising autoencoder (DAE). The goal here is not for denoising of input signals per se. Rather denoising is advocated as a training criterion such that the extracted features will constitute better high-level representation.

Vincent et al (2010) discussed three ways to corrupt inputs: 1) additive isotropic Gaussian noise: $\tilde{x} = x + \mathcal{N}(0, \sigma^2 I)$; 2) masking noise: a fraction $n$ of the elements of $x$ (chosen at random for each example) is forced to 0; and 3) salt-and-pepper noise: a fraction $n$ of the elements of $x$ (chosen at random for each example) is set to their minimum or maximum possible value (typically 0 or 1) according to a fair coin flip. While the additive Gaussian noise is a natural choice for real valued inputs, the salt-and-pepper noise is a natural choice for input domains which are interpretable as binary or near binary such as black and white images or the representations produced at the hidden layer after a sigmoid squashing function. The masking noise is equivalent to turning off components that have missing values. Thus DAE is trained to fill-in the missing data, which forces the extracted features to better capture the dependence among the all input variables.

**3.2. ELM for classification**

For combustor anomaly detection problem concerned in this paper, we use SDAE to learn features, which are then used as the input to the extreme learning machine (ELM) classifier. ELM is a special type of feedforward neural networks introduced by Huang, et al. (Huang et al., 2006).
Unlike in other feedforward neural networks where training the network involves finding all connection weights and bias, in ELM, connections between input and hidden neurons are randomly generated and fixed, that is, they do not need to be trained; thus training an ELM becomes finding connections between hidden and output neurons only, which is simply a linear least squares problem whose solution can be directly generated by the generalized inverse of the hidden layer output matrix (Huang et al., 2006). Because of such special design of the network, ELM training becomes very fast. ELM has one design parameter, i.e., the number of hidden neurons. Studies have shown that the ELM prediction performance is not too sensitive to the design parameter, as long as it is large enough, say 1000, which simplifies ELM design and makes ELM a more attractive model. Numerous empirical studies and recently some analytical studies as well have shown that ELM is an efficient and effective model for both classification and regression (Huang et al., 2012). Once again the ELM classifier takes feature learned from SDAE as the inputs and outputs probabilities of abnormality for gas turbine combustors.

4. CASE STUDY AND RESULTS

4.1. The business application

The asset of interest for our study is a combustor assembly used in heavy-duty industrial gas turbines. The combustor assembly is composed of a plural of individual combustion chambers. Fuel and compressed airflow is mixed and combusted in each combustion chamber, then the expanded hot gas is guided through a hot gas path along a number of turbine stages to derive work. A number of thermal couples (TC) are arranged in the turbine exhaust system to measure the exhaust gas temperature. The number of TC measurements varies depending on the turbine frames being monitored. It is a standard practice using TC temperature profile to infer combustor health condition. A typical TC temperature profile after mean normalization is shown in Figure 2.

4.2. Data description

Our database has several years of data sampled at once-per-minute. For demonstration purpose, in this study, we use several months of data for one turbine. Specifically, we use three months of event-free data and four months of data where 10 events occurred somewhere in the four-month window. After filtering out bad data points and those data points corresponding to part load condition (TNH<95%), we end up with 13,791 samples before the POD events (these samples are event-free and are considered to be normal), 300 samples for the POD events, and 47,575 samples after the POD events. The number of thermocouples for this turbine is 27, which is equal to the number of combustor cans.

In this study, we treat the 13,791 samples before the POD events as event-free (normal) data for unsupervised feature learning. And we use the rest of data (both POD events and event-free data) for training and testing the classifier.

4.3. Model design

For unsupervised feature learning, we use 2-layer SDAE. While DAE1 has 30 hidden neurons, DAE2 has 12 hidden neurons (See Figure 1). Activation functions for hidden neurons of both DAEs are sigmoid function. The noise rate is 0.2. The learning rate and momentum are 0.02 and 0.5, respectively. The number of epochs for learning is 200 for both DAEs. DAEs are implemented in Matlab R2014a.

The ELM classifier, as discussed in Section 3, has one design parameter, that is, the number of hidden neurons. Generally setting the number of hidden neurons to a large number, i.e., 1000 in this study.

As described in the previous section, our data is highly imbalanced between normal and abnormal classes (with majority-to-minority ratio of approximately 150), which deserves a special attention in classifier modeling. In literature there are many different strategies handling imbalanced data. He and Garcia (2009) provided a comprehensive review of different imbalance learning strategies. In this paper we take advantage of ELM’s capability of weighting samples during learning.

4.4. Results

To demonstrate effectiveness of unsupervised feature learning for combustor anomaly detection, we compare classification performance between using the learned features and using knowledge-driven, handcrafted features. Remember we use the identical setting of the ELM classifier for the comparison. In other word, using different feature sets is the only difference between the two designs in comparing classification performance. We use ROC curves as the classification performance measure for comparison. We employ 5-fold cross-validation for model training and validation. To obtain more robust comparison we run the 5-fold cross-validation 10 times, each time with different randomly splitting of 5 folds of the data.

The handcrafted features are primarily simple statistics calculated on TC profiles. These simple statistics essentially
capture engineering knowledge of combustor TC profiles associated with different combustor states (healthy and fault). The 12 handcrafted features are listed in Table 1 below.

<table>
<thead>
<tr>
<th>ID</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DWATT</td>
<td>Raw turbine load</td>
</tr>
<tr>
<td>2</td>
<td>TNH</td>
<td>Raw turbine speed</td>
</tr>
<tr>
<td>3</td>
<td>MAX</td>
<td>Max TCs</td>
</tr>
<tr>
<td>4</td>
<td>MEN</td>
<td>Mean TCs</td>
</tr>
<tr>
<td>5</td>
<td>STD</td>
<td>Standard deviation of TCs</td>
</tr>
<tr>
<td>6</td>
<td>MED</td>
<td>Median of TCs</td>
</tr>
<tr>
<td>7</td>
<td>DIF</td>
<td># diff b/w positive &amp; negative TCs</td>
</tr>
<tr>
<td>8</td>
<td>ZR</td>
<td>Zero crossing</td>
</tr>
<tr>
<td>9</td>
<td>KR</td>
<td>kurtosis</td>
</tr>
<tr>
<td>10</td>
<td>SK</td>
<td>skewness</td>
</tr>
<tr>
<td>11</td>
<td>M3S</td>
<td>Max of 3-pt sum</td>
</tr>
<tr>
<td>12</td>
<td>M3M</td>
<td>Max of 3-pt median</td>
</tr>
</tbody>
</table>

The ROCs for the 10 runs of 5-fold cross-validation using the handcrafted features are shown in blue in Figure 3.

From the ROC comparison in Figure 3, one can see clearly that the deep learned features give significant better classification performance than the handcrafted features do. Also from the ROCs one can see that using the deep learned features yields smaller variation in ROCs than using the handcrafted features. For example, when false positive rate (1-specificity) is at 1%, the mean and the standard deviation of the true positive rates (sensitivity) for both the deep learned features and the handcrafted features are approximately 0.99 ± 0.01 and 0.96 ± 0.02, respectively.

The 12 learned features are shown in Figure 4. Unlike the handcrafted features, each of which is a numerical number, the learned features are patterns representing the data (TC profiles) underlying structures. As the result, the learned features are more powerful in representing the data, thus performing better in classification. The ROCs for the 10 runs of 5-fold cross-validation using the learned features are shown in red in Figure 3.

5. CONCLUSION

Accurately detecting gas turbine combustor abnormalities is important in reducing O&M costs of power plants. Traditional rule-based anomaly detection solutions are inadequate in achieving the desired detection performance. Adopting more advanced machine learning technologies as a means of improving combustors’ detection performance is in great need. Realizing that generating good features is both a critically important and challenging task in developing machine learning solutions, in this paper we attempt to leverage recently developed unsupervised representation learning, a key part of deep learning, for finding more salient features from raw TC measurements for achieving more accurate and robust combustor anomaly detection. More specifically, we want to know if representation learning, which has approved to be effective in many other applications, can be an effective feature generation means for PHM applications. By applying SDAE we demonstrated that deep feature learning could effectively generate features from the raw time-series TC measurements, which thus improved combustor anomaly detection.

Unsupervised representation learning, or deep learning in general, has proven to be an effective ML technology in
other domains, but has not been used for any PHM applications. It is our hope that our initial work presented in this paper can shed some light on how deep learning as an advanced machine learning technology can benefit PHM applications and stimulate more research interests in our PHM community. In future we would like to conduct more thorough studies of combustor anomaly detection by using more real-world data. We would also like to explore other deep learning methods other than SDAE for combustor anomaly detection and other PHM applications as well.

**ACKNOWLEDGEMENT**

We would like to thank our colleague, Dr. Johan Reimann, for insightful discussion of deep learning technology over the course of this study.

**REFERENCES**


Brownlee, J. (2014). Discover feature engineering, how to engineer features and how to get good at it. machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/


Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN), Orlando, FL, August 12-17, 2007.

BIographies

Weizhong Yan, PhD, PE, has been with the General Electric Electric Power Generation since 1998. Currently he is a Principal Scientist in the Machine Learning Lab of GE Global Research Center, Niskayuna, NY. He obtained his PhD from Rensselaer Polytechnic Institute, Troy, New York. His research interests include neural networks (shallow and deep), big data analytics, feature engineering & feature learning, ensemble learning, and time series forecasting. His specialties include applying advanced data-driven analytic techniques to anomaly detection, diagnostics, and prognostics & health management of industrial assets such as jet engines, gas turbines, and oil & gas equipment. He has authored over 70 publications in referred journals and conference proceedings and has filed over 30 US patents. He is an Editor of International Journal of Artificial Intelligence and an Editorial Board member of International Journal of Prognostics and Health Management. He is a Senior Member of IEEE.

Lijie Yu, PhD, has joined GE since 2011. She started her career at the GE Global Research Center at Niskayuna, NY as a Computer Scientist. During that time, she built experience developing advanced algorithms for intelligent systems across multiple GE businesses. She is specialized in signal processing, machine learning, information fusion, and equipment health management. Currently she is senior analytics engineer at GE Power and Water, supporting power generation equipment remote monitoring and fleet management. She is holder of 7 awarded patents and authored 15 publications. She is also a certified Black Belt.
Adapting nearest neighbors-based monitoring methods to irregularly sampled measurements

Inês M. Cecílio¹, James R. Otewill², and Nina F. Thornhill³

¹,³ Centre for Process System Engineering, Department of Chemical Engineering, Imperial College London, London SW7 2AZ, UK

i.cecilio09@imperial.ac.uk
n.thornhill@imperial.ac.uk

² ABB Corporate Research Center, ul. Starowiślna 13a, 31-038 Kraków, Poland

james.otewill@pl.abb.com

ABSTRACT

Irregularly spaced measurements are a common quality problem in real data and preclude the use of several feature extraction methods, which were developed for measurements with constant sampling intervals. Feature extraction methods based on nearest neighbors of embedded vectors are an example of such methods. This paper proposes the use of a time-based construction of embedded vectors and a weighted similarity metric within nearest neighbor-based methods in order to extend their applicability to irregularly sampled measurements. The proposed idea is demonstrated within a method of univariate detection of transient or spiky disturbances. The result obtained with an irregularly sampled measurement is benchmarked by the original regularly sampled measurement. Although the method was originally implemented for off-line analysis, the paper also discusses modifications to enable its on-line implementation.

1. INTRODUCTION

One of the early steps in the PHM architecture is feature extraction from raw sensor data. Feature extraction aims to retain only the information that is relevant for classification and diagnostics, thus reducing the dimensionality of the raw data space and the risk of misclassification (Russell et al., 2000). Examples of features extracted for classification and diagnostics include the Hotelling T-square statistics, wavelet coefficients, non-linearity of the time series (Thornhill, 2005), and occurrence of spiky disturbances (Cecílio et al., 2014).

A common challenge to carry out feature extraction in real systems is the quality of the measurements available. A usual quality problem is that the interval between samples in a measurement is not constant. Figure 1 shows an example of such irregularity in a measurement from a real gas processing plant. This irregularity may arise for instance from problems with data communication. Another cause is data compression, which is done after sampling in order to save memory. Compression is done either by eliminating samples or by substituting the values of the samples by a constant value, for example, the average over a period.

Several methods for feature extraction were developed for measurements with constant sampling intervals $\Delta t$ and are not applicable to measurements such as those in Figure 1. For instance, several methods obtain the spectral information of the measurements from their Fourier transforms and wavelet decomposition. However, both techniques assume that the measurement samples are taken at regular intervals. Applications of these in the process monitoring were given by Thornhill et al. (2002); Choudhury et al. (2004); Tangirala et al. (2007); Zang & Howell (2007); Babji & Tangirala (2010). Methods that use cross-correlation, for example to extract time lags (Bauer & Thornhill, 2008), also require regularly-sampled data.

This paper focuses on feature extraction methods that use nearest neighbors of embedded vectors. Embedded vectors are segments of a time series. Nearest neighbors are the segments from a time series which are most similar to a ref-
ference segment (Chandola et al., 2009). Nearest neighbor-based methods have been used successfully to extract the non-linearity of a time series (Thornhill, 2005), time lags (Stockmann et al., 2012), and the occurrence of transient or spiky disturbances (Cecílio et al., 2014). However, none of these methods is directly applicable to irregularly sampled measurements. The reason is that the nearest neighbors approach implies measuring the similarity between embedded vectors. The conventional similarity measures are defined between two ordered sequences $p$ and $q$ which have the same number $m$ of samples and whose samples are synchronized. For example, the Euclidean distance metric which is used in the references mentioned above is defined as

$$d(p, q) = \sqrt{\sum_{i=1}^{m} (p_i - q_i)^2}.$$  \hspace{1cm} (1)$$

However, the segments represented by the embedded vectors normally span a constant interval of time. Therefore, in irregularly sampled measurements those segments will have a varying number of samples and varying intervals between samples. Hence, the conventional similarity measures are not directly applicable.

The contribution of this paper is to reformulate the construction of embedded vectors and the computation of similarity for the case of irregularly sampled measurements. As a result, the new formulation extends the applicability of methods based on nearest neighbors of embedded vectors.

The paper is structured as follows. Section 2 provides background on the analysis of irregularly sampled time series and time-based construction of embedded vectors. Section 3 explains the proposed techniques to construct embedded vectors and to compute similarity in the case of a measurement with irregular sampling rate. These techniques are applicable in the context of any nearest neighbor-based method. For brevity, section 4 demonstrates the techniques in one particular method, which detects and identifies transient disturbances (Cecílio et al., 2014). The demonstration uses the same case study as Cecílio et al. (2014) in order to have a benchmark for the results. Section 5 closes with conclusions.

2. Background

2.1. Analysis of irregularly sampled time series

Research in irregularly sampled time series is commonly found in domains such as astronomy (Scargle, 1989; Bos et al., 2002), finance (Zumbach & Müller, 2001), and geophysics (Rehfeld et al., 2011).

Weighting methods are one of the type of methods to analyse irregularly sampled time series. They generalize measures, such as distance and correlation, which are conventionally defined for pairs of aligned samples (Rehfeld et al., 2011). This paper uses a weighting method because nearest neighbor-based techniques require a similarity measure.

The conventional implementation of distance and correlation measures is illustrated in Figure 2a. For the case of the distance measure, the arrows in the figure indicate that only differences between aligned samples are considered. Instead, the weighting method calculate differences between all possible pairs of samples, and weights each difference according to the time misalignment between the pair (Rehfeld et al., 2011). This idea is illustrated in Figure 2b for sample $p_{10}$ and sample $q_6$. Larger weights are represented in the figure by darker tones on the arrows. The weighting function is such that the more aligned samples are, the more their difference counts towards the distance metric.

The weighted version of the Euclidean distance metric is defined as

$$d(p, q, w) = \sqrt{\sum_{i=1}^{n_p} \sum_{j=1}^{n_q} w_{i,j} (p_i - q_j)^2}.$$  \hspace{1cm} (2)$$

where $w_{i,j}$ is the weight attributed to the difference between sample $p_i$ of time series $p$ and sample $q_j$ of time series $q$. Examples of weight functions $w$ found in the literature are sinc and Gaussian functions (Rehfeld et al., 2011). In particular, the Gaussian function (equation (3)) is a positive function which decays smoothly to zero, and is symmetric with relation to the time misalignment $(t_i - t_j)$ between samples $p_i$ and $q_j$. 

![Figure 2a: Conventional implementation: only pairs of aligned samples.](image-url-a)

![Figure 2b: Weighting method exemplified for samples $p_{10}$ and $q_6$: all possible pairs of samples, with each pair weighted according to time misalignment. Larger weights are represented by darker tones.](image-url-b)
\[ w_{i,j} = w(t_i, t_j) = \frac{1}{\sqrt{2\pi L}} \exp \left( -\frac{(t_i - t_j)^2}{2L^2} \right) \] (3)

Since a distance metric should be non-negative and symmetric, the Gaussian function is a relevant alternative for a weighting function. The Gaussian weighting function has a width parameter \( L \) which determines the rate of decay of the weight values \( w_{i,j} \) with the time misalignment between the two samples.

Other methods to analyse irregularly sampled time series include: (i) reconstruction methods, (ii) spectral transforms, and (iii) ARMA model fitting (Rehfeld et al., 2011).

Reconstruction methods resample the time series into a regular time grid and then apply existing methods developed for regularly sampled time series. Common techniques of resampling include linear and spline interpolation, regression, and approximation by the value of the closest point closest in time (Lall & Sharma, 1996).

A common spectral transform for irregularly sampled time series is the Lomb-Scargle Fourier transform (Scargle, 1989). It determines the spectrum of a measurement from a least squares fit of sine curves to the time series of the measurement. It is suitable for measurements with periodic components and no outliers (Stoica et al., 2009). The wavelet transform can also be computed for irregularly sampled time series if implemented through the lifting scheme (Sweldens, 1998).

Fitting autoregressive-moving-average (ARMA) models to a time series involves determining the coefficients of the ARMA model. To determine the coefficients from irregularly sampled time series, research focuses on adapting estimation algorithms such as maximum-likelihood estimation (Isaksson, 1993) and the Burg algorithm (Bos et al., 2002).

### 2.2. Time-based construction of embedded vectors

Embedded vectors were originally defined for regularly sampled time series (Kantz & Schreiber, 2003). They refer to segments from a time series with a fixed number \( m \) of samples, with each embedding vector lagging the previous by \( \delta \) samples (Kantz & Schreiber, 2003). Figure 3a illustrates the selection of three embedded vectors from a symbolical time series represented by dots, with \( m = 5 \) and \( \delta = 2 \). Embedded vectors are commonly used in the analysis of nonlinear time series.

Cecílio et al. (2015) proposed an alternative approach to the construction of embedded vectors motivated by the integrated analysis of measurements with fast and slow sampling rates. That paper imposed the same time span \( M \) for all embedded vectors and a lag of a constant number \( \Delta \) of time units for the different measurements. This way, the embedded vectors of the different measurements would be synchronized.

### 3. METHODS

This section explains the techniques to construct embedded vectors and to compute similarity in the case of a measurement with irregular sampling rate. These techniques extend the applicability of time series analysis methods based on nearest neighbors of embedded vectors.

#### 3.1. Embedded vectors

Consider a time series \( X \) of sample values \( x(t_i) \) (equation (4a)) which are ordered according to the time sequence \( T \) of strictly increasing sampling instants \( t_i \) (equation (4b)).

\[
X = \{ x(t_1), x(t_2), \ldots, x(t_n) \} : t_1 < t_2 < \cdots < t_n \quad (4a)
\]

\[
T = \{ t_1, t_2, \ldots, t_n \} : t_1 < t_2 < \cdots < t_n \quad (4b)
\]

An embedded vector \( x_r \) is defined as a segment of the time
series \( X \) which spans \( M \) time units. The number of samples is variable and is here denoted as \( m_r \). Furthermore, embedded vector \( x_r \) lags the previous \( x_{r-1} \) by a constant number \( \Delta \) of time units. The construction of embedded vectors from \( X \) is represented in Figure 3b.

Additionally, for each embedded vector \( x_r \), a time vector \( t_r \) should be created to arrange the time instants of each sample in \( x_r \), that is, \( t_r = \{ t_{r,1}, t_{r,2}, \ldots, t_{r,m_r} \} \).

It should be noted that the embedded vectors of a measurement cannot be arranged in an embedding matrix, as conventionally done with regularly sampled measurements (Thornhill, 2005; Cecilio et al., 2014). This is due to the different number of samples in each embedded vector, as illustrated in Figure 3b.

### 3.2. Similarity

Each pair of embedded vectors \( x_r \) and \( x_s \) is then compared using the weighted Euclidean distance metric

\[
d(x_r, x_s, w)^s = \sqrt{\sum_{i=1}^{m_r} \sum_{j=1}^{m_s} w_{i,j} (x_{r,i} - x_{s,j})^2}
\]

where \( i \) and \( j \) represent the indices of the samples in \( x_r \) and \( x_s \), respectively. This equation is a scaled version of the weighted Euclidean distance metric presented in equation (2). The aim of scaling is to have a metric \( d(x_r, x_s, w)^s \) which is independent of the number of samples in \( x_r \) and \( x_s \).

The weighting function \( w_{i,j} = w(t_{r,i} - t_{s,j}) \) is defined as in equation (3), and depends on the time instants of the samples of \( x_r \) and \( x_s \). The width parameter \( L \) may be optimized or used as suggested in (Rehfeld et al., 2011), that is,

\[
L = \frac{\Delta t}{4}
\]

where \( \Delta t \) is the mean value of the sampling intervals in measurement \( X \).

### 4. APPLICATION TO UNIVARIATE DETECTION OF TRANSIENT DISTURBANCES

This section demonstrates the proposed formulation of embedded vectors and similarity measure within a method for extracting the moment of occurrence as well as intensity of transient disturbances (Cecilio et al., 2014). In electrical circuits, transient disturbances include voltage spikes, which can be an indication of unbalanced power grid as well as a cause of degradation of sensitive electronics (Bevrani, 2009). In rotating tools, transient disturbances can indicate abnormal shock and vibration levels.

#### 4.1. Univariate detection of transient disturbances

In Cecilio et al. (2014), transient disturbances were formally defined as infrequent and short-lasting deviations of a measurement from its underlying trend. The extraction of the moment of occurrence and intensity of these features was formulated as an anomaly detection problem, and solved using nearest neighbors of embedded vectors.

The implementation of the method can be summarized in the following steps.

1. Embedded vectors with a fixed number \( m \) of samples and \( \delta \) samples of lag are generated from a time series \( X \).
2. Each embedded vector is then compared to every other embedded vector, using the Euclidean distance metric.
3. An anomaly index \( a_i \) is then attributed to each embedded vector \( x_r \) as the \( k \)th smallest distance between \( x_r \) and every other embedded vector. This is denoted by \( d_{k} \), the distance to its \( k \)th nearest neighbor.
4. An anomaly index vector \( a_i \) is formed by the sequence of anomaly indices \( a_i \) of each embedded vector.
5. A threshold based on the statistics of \( a_i \) distinguishes embedded vectors which capture the transients from embedded vectors which capture periods of normal operation.

In the following, the two initial steps of the method will be replaced by the alternative formulation of embedded vectors and similarity proposed in section 3. With this formulation the method of univariate detection of transient disturbances can now be applied to irregularly sampled time series.

#### 4.2. Case study

The proposed method uses a measurement from Cecilio et al. (2014) in order to have a benchmark for the results. The measurement represents the shaft speed of a compressor during 20 seconds, and was obtained from a gas compressor rig located at ABB Corporate Research Center, Kraków, Poland. The shaft speed was measured at a regular rate of 1 kHz, therefore its measurement had to be manipulated in order to have an irregularly sampled time series for illustration of the proposed method. This was done by randomly eliminating samples from the original measurement, which also resulted in a decrease in the total number of samples from 20,000 samples to approximately 400. The time instants of the retained samples were stored. Figure 4 shows the speed measurement after this manipulation. Figure 5 shows a close-up to highlight the irregular spacing between samples.

Two step changes, around 5 and 11 s, were imposed in the drive of the compressor by changing its speed set-point, resulting in the two transients seen in the figure. The objective of the proposed method is to detect those transients.
ANNUAL CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2015

Figure 4. Compressor speed measurement from (Cecílio et al., 2014). The measurement was manipulated in order to have an irregularly sampled time series. The values are normalized by the initial value.

Figure 5. Close-up on the compressor speed measurement to highlight the irregular spacing between samples.

4.3. Results

Figure 6a shows the anomaly index vector $a_i$ computed from the measurement in Figure 4. As in Cecílio et al. (2014), $a_i$ was normalized by its median so that $a_i = 1$ now approximates the average anomaly index of non-anomalous embedded vectors.

The positive detection of the two transients is indicated by the fact that the embedded vectors which correspond to transient disturbances have anomaly indices above the detection threshold, which is represented by the dashed line in Figure 6a. The figure shows that the construction of embedded vectors and similarity measure suggested in section 3 are able to cope with the sampling irregularity and achieve the desired detection.

Figure 6b shows the result obtained by Cecílio et al. (2014) with the original regularly sampled measurement. The figure clearly shows the similarity between the two results. This demonstrates the potential of the proposed formulation for the analysis of irregularly sampled time series with nearest neighbor-based methods.

4.4. Comment on the use of the method for real-time monitoring

The techniques proposed in this paper to construct embedded vectors and to compute similarity are applicable in both on-line and off-line analysis methods. In section 4, the techniques were demonstrated as part of an off-line analysis because the transients detection method proposed in Cecílio et al. (2014) was originally implemented in that way. However, the concept of transients detection with nearest neighbors is amenable to on-line implementation, and this section discusses possible approaches.

The crucial point for on-line implementation is the computational cost of comparing every embedded vector to all other embedded vectors.

One way to reduce this cost is the following. When new samples arrive and a new embedded vector is formed, that embedded vector is compared against a finite number $N_H$ of past embedded vectors. This amounts to $N_H$ computations of distance between two vectors. The anomaly index $a_i$ of that embedded vector comes from one search operation amongst those $N_H$ distances.

The efficiency of the distance computations can also be improved. The anomaly index $a_i$ only uses one piece of information out of the $N_H$ distances calculated for each embedded vector. However, these $N_H$ distances can also be used to identify tight clusters of embedded vectors. As a result, every time a new embedded vector is formed it needs only be compared to the centroid of each cluster instead of all the embedded vectors that form that cluster.

These modifications in the implementation should enable the on-line implementation of the transients detection, which is better suited for PHM applications.

5. CONCLUSIONS

This paper presented an adaptation to methods based on nearest neighbors to enable their application to measurements with irregular sampling rates. The first two steps of these methods normally involve the construction of embedded vectors and a similarity assessment. With irregular sampling rates the conventional construction of embedded vectors and similarity measure cannot be applied. The proposed techniques comprise a time-based formulation for embedded vectors, and a weighted distance metric to assess the similarity between the embedded vectors. These techniques can substitute the first two steps of conventional nearest neighbors methods.

The new techniques were demonstrated within a method of detection of transient or spiky disturbances, which had been developed for regularly sampled measurements. The case study showed that the new formulation achieves results in an irregularly sampled time series on a par with the results ob-
Figure 6. Normalized anomaly index vector. The dashed line indicates the detection threshold.

Open questions about the proposed idea include:

- studying if, and under which conditions, the weighted Euclidean metric converges to conventional Euclidean metric,
- determining the statistical behaviour of the anomaly index vectors in order to attribute a confidence level to the selected threshold, in the case of the detection methods,
- optimizing the width parameter $L$, and re-evaluating the parameter optimisation done for methods with regularly sampled measurements, and
- analysing the sensitivity of the methods to the distribution of samples in the measurement.

The paper also discussed possible modifications to enable the on-line implementation of the techniques.

**ACKNOWLEDGMENT**

The authors gratefully acknowledge the financial support from the Portuguese Foundation for Science and Technology (FCT) under Fellowship SFRH/BD/61384/2009 and the Marie Curie FP7-IAPP project “REAL-SMART - Using real-time measurements for monitoring and management of power transmission dynamics for the Smart Grid”, Contract No: PIAP-GA-2009-251304.

**REFERENCES**


Comparison and ensemble of temperature-based and vibration-based methods for machinery prognostics

James Kuria Kimotho¹ and Walter Sextro²

¹,² Department of Mechatronics and Dynamics, University of Paderborn,

james.kuria.kimotho@uni-paderborn.de
walter.sextro@uni-paderborn.de

ABSTRACT

This paper presents a comparison of a number of prognostic methods with regard to algorithm complexity and performance based on prognostic metrics. This information serves as a guide for selection and design of prognostic systems for real-time condition monitoring of technical systems. The methods are evaluated on ability to estimate the remaining useful life of rolling element bearing. Run-to failure vibration and temperature data is used in the analysis. The sampled prognostic methods include wear-temperature correlation method, health state estimation using temperature measurement, a multi-model particle filter approach with state equation parameter adaptation utilizing temperature measurements, prognostics through health state estimation and mapping extracted features to the remaining useful life through regression approach. Although the performance of the methods utilizing the vibration measurements is much better than the methods using temperature measurements, the methods using temperature measurements are quite promising in terms of reducing the overall cost of the condition monitoring system as well as the computational time. An ensemble of the presented methods through weighted average is also introduced. The results show that the methods are able to estimate the remaining useful life within error bounds of ±15%, which can be further reduced to ±5% with the ensemble approach.

1. INTRODUCTION

In the last decade, maintenance focus has shifted towards prognostic health management where maintenance action is taken based on the current health state of a system and its estimated remaining useful life. In addition, new technical systems, referred to as self-optimizing mechatronic systems with the ability to adaptively control reliability have been developed (Sondermann-Wölke & Sextro, 2010; Meyer & Sextro, 2014). These systems are able to react to changed operating conditions or faults within the system through behavior adaptation based on multi-objective optimization and consequently require accurate estimation of the current health state and the remaining useful life. The overall objective of this approach is to increase reliability, availability and safety of technical systems.

A number of methods for estimating the remaining useful life (RUL) have been proposed. These methods can be divided into three broad categories: 1) reliability based, which rely on failure times of similar units, 2) model based, which rely on mathematical models based on physics of failure and 3) data driven methods, which rely on raw sensory data obtained from a system during operation (Tobon-Mejia, Medjaher, & Zerhouni, 2011). Reliability based methods are the simplest to employ since they do not require condition monitoring data. However, their accuracy is relatively low, especially for systems subjected to varying operating conditions and consequently displaying varying lifetimes. Model-based methods though found to be very accurate, are system or component specific and are not easily adaptable to different systems. In addition, due to the complexity of modern day systems, the system models are very complex and computationally intensive. Data driven methods have received considerable efforts since they can be adapted to different systems. Data driven methods employ mainly statistical based algorithms such as support vector machines (SVM) and hidden Markov models (HMM) or artificial intelligence methods such as artificial neural networks (ANN) (Tobon-Mejia et al., 2011). These algorithms require a lot of data for training and this involves conducting run-to-failure tests to generate the training and validating data. The performance of these algorithms also depend on suitability of the features extracted from the raw data (Kimotho & Sextro, 2014a). For classification of health states, good features should demonstrate separability between different health states while for regression approach where the features are mapped to a function (either a health index or RUL), then the features should have the ability to capture the degradation trend, preferably monotonic change. For rotat-
ing machinery, the most common condition monitoring data utilized are vibration signals.

Temperature measurements have been recognized as a condition monitoring tool but their application in estimating the remaining useful life has not been fully realized. One of the identified limitations of temperature in machinery diagnosis and prognosis is the inability to identify faults at the development stage. However, this limitation can be overcome by strategic positioning of the temperature sensors. A number of sensors for condition monitoring of technical systems have been developed. A wireless temperature sensor for condition monitoring of bearings operating through thick metal plates was proposed by (Gupta & Peroulis, 2013). The sensor consists of a temperature-sensitive permanent magnet which is attached to the inner ring of the bearing, thus allowing the bearing temperature to modulate the produced magnetic field. Joshi et al (Joshi, Marble, & Sadeghi, 2001) demonstrated the application of radio telemetry for bearing cage temperature measurement for use in condition monitoring of bearings. The cage telemetry was found to capture faults such as loss of lubrication much faster than housing thermocouple. Brecher et al, (Brecher, Fey, Hassis, & Bonerz, 2014) demonstrated the use of a customized telemetry system for measuring a bearing’s inner ring temperature for high speed applications. The analysis showed that the inner ring temperature was vital in accurately monitoring the health of the bearing. These developments could prove useful in enhancing fault identification and estimation of remaining useful life in bearings as well as reducing the overall cost of the condition monitoring system.

In this paper, five methods for estimating the remaining useful life of bearings are evaluated and compared in terms of performance based on prognostic metrics and computational time. The first approach involves correlating wear with temperature rise due to frictional heating. The method uses run-to-failure temperature measurements to obtain coefficients which can be used with the temperature measurements at any given time to estimate the remaining useful life. The second approach involves estimating the health states of a bearing from the temperature measurements and deducing the remaining useful life from the current health state. The third approach involves the application of multi-model particle filter with model parameter adaptation to propagate a health index derived from temperature measurements to a predetermined threshold. The fourth approach involves estimating the health states of a degrading component using features extracted from vibration measurements (Kimotho, Sondermann-Wölke, Meyer, & Sextro, 2013). Classification algorithms are used to identify the current health state. The probability of each health state together with the percentage remaining useful life at each health state of similar units are utilized in estimating the remaining useful life. The last approach involves mapping features extracted from vibration measurements to the remaining useful life at any given time. Regression algorithms are used in this approach (Kimotho & Sextro, 2014a). The last two methods based on vibration signals have been previously discussed in (Kimotho, Sondermann-Wölke, et al., 2013; Kimotho & Sextro, 2014a) and are only briefly introduced for comparison and ensemble purposes.

All the methods are evaluated using ball bearing run-to-failure data for training and truncated run-to-failure data for testing obtained from the 2012 PHM data challenge (Nectoux, Medjaher, Ramasso, Morello, & Zerhouni, 2012). The data is obtained through highly accelerated run-to-failure experiments conducted at three different operating conditions shown in Table 1. Due to the highly accelerated degradation, the data sets are characterized by high variability in experiment durations, ranging from 1 to 8 hours. In this work, only data sets from test 1 are analyzed.

### 2. Prognostic Methodologies

The following subsections outline different methodologies that have been evaluated on their suitability to estimate the remaining useful life of technical systems. An ensemble approach of combining the estimations of different approaches is also explored.

#### 2.1. Wear - Temperature Correlation - Method 1

During operation, rolling element bearings encounter resistance to rotation which consist of rolling and sliding friction. This resistance occurs at the rolling contacts, contact areas between the rolling elements and the cage, as well as the guiding races (Harris & Kotzalas, 2006). The frictional forces perform work which is dissipated in form of heat, consequently increasing the bearing temperature. The frictional heat generated depends on the applied load, rotational speed, the type and size of bearing, properties and quantity of lubricant as well as the rate of heat dissipation. The rise in temperature reduces the viscosity of the lubricant which leads to a reduction in the lubricant film thickness. This results to higher asperity contact, increased heat generation due to increased friction and consequently increased wear (Joshi et al., 2001). Wear results to continued loss of geometric accuracy of the rolling and gradual development of other faults such as micro-pitting (Harris & Kotzalas, 2006). Since it is assumed that wear can be prevented by proper attention to the bearing, no considerable effort has been made to estimate the remain-

<table>
<thead>
<tr>
<th>Test</th>
<th>Speed (rpm)</th>
<th>Load (kN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1800</td>
<td>4.0</td>
</tr>
<tr>
<td>2</td>
<td>1650</td>
<td>4.2</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>5.0</td>
</tr>
</tbody>
</table>
ing useful life of bearings related to wear and change in temperature (Harris & Kotzalas, 2006). Johnson (Johnson, 1985) investigated the temperature produced by frictional heating in sliding contact by examining the temperature produced in a half space by a heat source which moves on the surface. The maximum temperature occurs towards the rear of the heated zone which has the longest exposure, as shown in Figure 1 (Johnson, 1985). For a bearing rotating at constant speed, frictional heat is generated at all contact points leading to an overlap in the maximum temperature throughout the bearing. This would result to an almost constant temperature distribution. Therefore, the temperature will not be a function of position but the factors mentioned previously. The bearing operating temperature will also depend on the balance between the heat generated and the heat removed from the bearing through conduction, convection and radiation. In most cases the temperature of the bearing increases rapidly during initial operation and then increases slowly to a steady state temperature as shown in Figure 2. The rate of rise depends on the rate of heat removal. When a bearing is run continuously to failure, at constant speed, the rate of temperature rise changes rapidly initially, and then decreases to an almost constant value. Just before failure, the temperature rise again changes rapidly. This behavior is shown in Figure 2. The behavior is also consistent with degradation of bearings, where degradation is high initially, followed by gradual degradation and finally high degradation towards failure. This indicates that temperature can be used to effectively track degradation of components occasioned by wear.

Assuming that the heat removal rate is constant for a given system, then it is possible to track bearing degradation from temperature measurements. Since wear is approximately proportional to the work done by the frictional forces which give rise to frictional heat, then it may be assumed to be directly proportional to the temperature rise in the component. This relationship can be formulated as shown below:

\[
\frac{\Delta \dot{m}}{\Delta A} \propto \frac{\Delta P_R}{\Delta A} (T(t) - T_i),
\]

(1)

where \(\Delta \dot{m}\) is the material removal rate, \(\Delta P_R\) is frictional power, \(T(t)\) is the current temperature, \(T_i\) is the room temperature and \(\Delta A\) is the contact area. Taking the initial temperature as the datum for temperature change and considering that wear is approximated by the material removal rate yields

\[
m_{EOL} = \rho \Delta A \Delta z = \int_0^{t_{EOL}} \dot{m}(t) dt = k \int_0^{t_{EOL}} \Delta T(t) dt,
\]

(2)

where \(\rho\) is the density of the material, \(\Delta A\) is the contact area, \(\Delta z\) is the approximate wear depth, \(t_{EOL}\) is the time at the end of life of the component, \(k\) is proportionality constant, \(m_{EOL}\), is the allowable mass that can be removed through wear before a component is considered to have failed. The ratio of allowable mass to the proportional constant can be obtained from the training data as by

\[
\frac{m_{EOL}}{k} = \int_0^{t_{EOL}} \Delta T(t) dt.
\]

(3)

Considering the current time, \(t_c\), Eq. (3) can be rewritten in cumulative form as:

\[
\frac{m_{EOL}}{k} = \int_0^{t_c} \Delta T(t) dt + \int_{t_c}^{t_{EOL}} \Delta T(t) dt.
\]

(4)

During testing or online prognosis, the second term of Eq. (4) is unknown. This term is proportional to the remaining allowable wear before the component fails. Therefore this factor can be referred to as the remaining wear coefficient. Eq. (4) can be rearranged as:

\[
\int_{t_c}^{t_{EOL}} \Delta T(t) dt = \frac{m_{EOL}}{k} - \int_0^{t_c} \Delta T(t) dt.
\]

(5)

The term \(\int_{t_c}^{t_{EOL}} \Delta T(t) dt\) can be used as a health index \(HI\), defined as follows

\[
HI = \int_{t_c}^{t_{EOL}} \Delta T(t) dt \cdot \frac{\frac{m_{EOL}}{k}}{\frac{m_{EOL}}{k}}.
\]

(6)

Division by \(\frac{m_{EOL}}{k}\) normalizes the health index such that \(HI\)
is within the range $0 \leq HI \leq 1$, with $HI = 1$ for a healthy component and $HI = 0$ for a failed component. Figure 3 shows the health index of the two bearings used to generate the training data sets. The term $\frac{m_{EOL}}{k}$ is computed from the training data sets which contain run-to-failure temperature measurements. The health index of the test bearing at the current time is computed using Eq. (6), with $\frac{m_{EOL}}{k}$ obtained from the training data sets. To obtain the time to end of life, $t_{EOL}$ of the test bearing, a polynomial curve of order 2 is fitted to the calculated health index and extrapolated to the point where the health index is zero. The RUL can then be calculated as shown in Figure 4. The performance of the method can be evaluated through prognostic performance metrics such as percentage error computed as follows

$$\text{Error} = \frac{\text{RUL}_{\text{actual}} - \text{RUL}_{\text{estimated}}}{\text{RUL}_{\text{actual}}} \times 100,$$  

where $\text{RUL}_{\text{actual}}$ is the actual remaining useful life and $\text{RUL}_{\text{estimated}}$ is the estimated RUL. Table 2 shows the % Error computed for the different test bearings. The results show that the proposed method is a promising approach in prognostics with the estimation error below 15%.

![Figure 3. Health index computed from the training bearings.](image)

2.2. Health State Estimation using Temperature Measurements - Method 2

This method involves estimating the health states of a degrading component using temperature measurements. Based on the current health state, a health index can be defined from which the remaining useful life can be estimated. k-means clustering algorithm is employed to discretize the temperature data into a number of clusters and an energy factor for each health state is calculated from the temperature change using Eq. (8)

$$\left(\frac{\Delta m_{EOL}}{k}\right)_{HSi} = \int_{t_{HS_{i-1}}}^{t_{HS_i}} \Delta T(t) dt.$$  

A health index at each health state is then obtained as follows

$$HI_{HSi} = 1 - \frac{\int_{t_{HS_{i-1}}}^{t_{HS_i}} \Delta T(t) dt}{\left(\frac{\Delta m_{EOL}}{k}\right)_{HSi}}.$$  

A total of 4 health states were identified from the training data. For test data, the current health state is estimated using classification algorithms such as support vector machines, which output the health state probability. Figure 5 shows a plot of the correlation coefficient for bearing 1_6. The right hand axis shows the normalized temperature change curve. To obtain the time to end of the current health state, a polynomial curve of order 2 is fitted to the calculated health index and extrapolated to the point where the health index is zero. This point signifies the transition to the next health state or end of life if the current health state is the last health state (HS4 in this case). The remaining useful life can then be cal-

![Figure 4. RUL prediction of bearing 1.7.](image)

![Figure 5. Health states of bearing 1.6.](image)

<table>
<thead>
<tr>
<th>Test bearing</th>
<th>Actual RUL (h)</th>
<th>Estimated RUL (h)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing_1_4</td>
<td>0.094</td>
<td>0.085</td>
<td>11.82</td>
</tr>
<tr>
<td>Bearing_1_5</td>
<td>0.446</td>
<td>0.400</td>
<td>10.45</td>
</tr>
<tr>
<td>Bearing_1_6</td>
<td>0.392</td>
<td>0.334</td>
<td>14.89</td>
</tr>
<tr>
<td>Bearing_1_7</td>
<td>2.103</td>
<td>1.983</td>
<td>5.71</td>
</tr>
</tbody>
</table>

Table 2. % Error of RUL estimation of the proposed method.
culated using Eq. (10)

\[ RUL = (t_{HSi} - t_{ci}) + \sum_{j=i+1}^{n} \frac{RL_j}{1 - RL_j} \cdot t_{HSi}, \]  

(10)

where \( t_{ci} \) is the current time in health state \( i \), \( t_{HSi} \) is the time at the end of health state \( i \), \( RL_j \) is the percentage of remaining useful life of health state \( j \), obtained from the training data and \( n \) is the total number of health states. If the current health state is the last health state, then \( RL_j = 0 \) and Eq. (11) reduces to

\[ RUL = (t_{HSi} - t_{ci}). \]  

(11)

Figure 6 shows the plot of the calculated health index and fitted health index for bearing 1,6, which was identified to be in health state 4. Table 3 shows the performance evaluation of this method based on the percentage error, calculated from Eq. (7). One advantage of this method is the ability to identify current health state, which would be important for maintenance action recommendation or for adaptive systems utilizing discrete health states to adapt the operating regime. In addition, the method can be adapted to systems with varying operating conditions by defining different wear factors based on the operating regime and health state.

Table 3. % Error of RUL estimation of the proposed method.

<table>
<thead>
<tr>
<th>Test bearing</th>
<th>Actual RUL (h)</th>
<th>Estimated RUL (h)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing 1,4</td>
<td>0.094</td>
<td>0.083</td>
<td>12.70</td>
</tr>
<tr>
<td>Bearing 1,5</td>
<td>0.446</td>
<td>0.485</td>
<td>-8.74</td>
</tr>
<tr>
<td>Bearing 1,6</td>
<td>0.392</td>
<td>0.343</td>
<td>12.53</td>
</tr>
<tr>
<td>Bearing 1,7</td>
<td>2.103</td>
<td>2.162</td>
<td>-2.79</td>
</tr>
</tbody>
</table>

2.3. Temperature based Particle Filter Approach with Parameter and Model Adaptation - Method 3

Particle filter is a general Monte Carlo (Sampling) method for estimating the state of a system that changes over time using a sequence of noisy measurements obtained from the system (Arulampalam, Maskell, N., & T., 2002). The state of the system is considered to evolve according to

\[ x_k = f(x_{k-1}, t_{k-1}, t_k) + n_k, \]  

(12)

where \( x_k \) is the state of the system at time \( k \) and \( f \) is the transition function that propagates \( x_{k-1} \) to \( x_{k-1} \), and \( n_k \) is the process noise. The state vector is assumed to be unobservable and its information is only obtained through noisy measurements of its observation \( z_k \) which is obtained by

\[ z_k = g(x_k) + \nu_k, \]  

(13)

where \( g \) is the observation model and \( \nu_k \) is the measurement noise. The filtering process involves the estimation of the state vector at time \( k \), given all the measurements up to time \( k \), denoted by \( z_{1:k} \). From a Bayesian setting, this problem involves recursively calculating the distribution \( p(x_k|z_{1:k}) \) which is done in two steps (Arulampalam et al., 2002).

1. Prediction Step, where \( p(x_k|z_{1:k-1}) \) is computed from the filtering distribution \( p(x_{k-1}|z_{1:k-1}) \) at time \( k - 1 \) as follows:

\[ p(x_k|z_{1:k-1}) = \int p(x_k|x_{1:k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}, \]  

(14)

where \( p(x_{k-1}|z_{1:k-1}) \) is assumed to be known due to recursion and \( p(x_k|x_{k-1}) \) is given in Eq. (12) (Arulampalam et al., 2002). The distribution \( p(x_k|z_{1:k-1}) \) is known as a prior over \( x_k \) before receiving the most recent measurement \( z_k \).

2. Update step, where the prior is updated with the new measurement \( z_k \) using Bayes’ rule to obtain the posterior over \( x_k \)

\[ p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \]  

(15)

The computations in the prediction and update steps Eqs. (14-15) can be done using approximation methods such as Monte Carlo sampling (Arulampalam et al., 2002). A detailed description of this approach can be found in (Arulampalam et al., 2002).

When using this approach for prognostics, there is no new measurement available and hence the update step is not carried out. The system state is propagated until a predefined threshold is reached. This approach has been employed in prognostics of various technical systems such as batteries (Lee, Cui, Rezvanizaniani, & Ni, 2012; Xing, Miao, Tsui, & Petch, 2011), fuel cells (Jouin, Gouriveau, Hissel, Pera, & Zerhouni, 2014; Kimotho & Sextro, 2014b), gears (He, Bechhoefer, Dempsey, & Ma, 2012) and bearings (Wang & Gao, 2013).
In the context of this work, a health index is selected as the state of the system ($x_k = HI_k$) and defined by normalizing the temperature change such that $0 \leq HI \leq 1$ based on the maximum and minimum values of the training data which consists of run-to-failure temperature measurements as shown in Eq (16)

$$HI = \frac{\Delta T - \Delta T_{\text{min}}}{\Delta T_{\text{max}} - \Delta T_{\text{min}}}. \quad (16)$$

Two state equation were selected based on the temperature trend. The first part of the curve was approximated using a logarithmic equation while the second part was approximated using an exponential equation. The transition point was taken as the point where the rate of change of the health index (filtered using a kernel-based smoother) is zero, that is, $\frac{dHI}{dt} = 0$. Figure 7 shows the selection of state equations based on training data set from bearing 1_1. The selected state equations are shown below

$$f_1 = \alpha \cdot \ln \left( \frac{t_k}{t_{k-1}} \right) + HI_{k-1}, \quad (17)$$

$$f_2 = HI_{k-1} \cdot \exp(\beta(t_k - t_{k-1})), \quad (18)$$

where, $\alpha$ and $\beta$ are state equation parameters to be fitted from the training data.

To evaluate the performance of the approach on the training data and the suitability of the selected state equation parameters, the available training data is truncated at different fractions of the component’s lifetime. The health index is then computed and propagated until it reaches a threshold. The RUL is then calculated as shown in Figure 8. The process is done for several runs and a statistical value such as mean or median is taken as the overall RUL as shown in Figure 8. Figure 9 shows the performance of the approach based on the training data at different truncation intervals. Most of the particle filter approaches employed in literature use single state equation parameters when propagating the state of the system. However, as seen in Figure 9, due to the non-linearity of the degradation trend, it is difficult to obtain parameters that are able to track degradation of the systems accurately throughout its lifetime. The parameters are accurate at certain degradation stages and inaccurate at others. This limitation can be addressed by adapting the parameters to the rate of degradation. Figure 10 shows the RUL estimation with state equation parameter adaptation based on the rate of change of the health index. With this approach, the estimated RUL is approximately within ±10% confidence bounds at all degradation stages. Once suitable state equation parameters have been identified, the method is then applied to truncated test data or data acquired in real-time. This involves normalizing the change in temperature of the test data using Eq. (16) together with the maximum and minimum values obtained from the training data. The particles are propagated with resampling until the available data is exhausted after which the model is used to propagate the health index up to the threshold. Figure 11 shows estimation of RUL through this approach for bearing 1_7. The method was applied to other test bearings and a performance analysis conducted and presented in Table 4. With this approach, errors less than 15% can be attained. The approach is more robust since measurement and
the lifetime of the system. This can be done by employing clustering algorithms to identify the classes within the features. Algorithms such as k-means and self-organizing maps (SOM) neural networks can be employed to identify the health states in the features. Figure 12 is a feature plot (skewness and clearance factor) showing clustering of data for different health states during degradation of a ball bearing. Machine learning algorithms are then trained to map the input

2.4. Prognostic Approach based on Health State Estimation - Method 4

Most technical systems undergo through a series of discrete health states before failure, with east health state indicating the severity of faults or degradation. RUL estimation based on this approach involves extracting relevant features from the raw run-to-failure data and discretizing the features into a number of clusters representing different health states within

state propagation uncertainties are taken into account. The prediction accuracy also increases as the component nears failure due to resampling.

Table 4. % Error of RUL estimation using particle filter approach

<table>
<thead>
<tr>
<th>Test bearing</th>
<th>Actual RUL (h)</th>
<th>Estimated RUL (h)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing 1_4</td>
<td>0.094</td>
<td>0.106</td>
<td>-12.50</td>
</tr>
<tr>
<td>Bearing 1_5</td>
<td>0.446</td>
<td>0.433</td>
<td>2.99</td>
</tr>
<tr>
<td>Bearing 1_6</td>
<td>0.392</td>
<td>0.417</td>
<td>-6.38</td>
</tr>
<tr>
<td>Bearing 1_7</td>
<td>2.103</td>
<td>2.008</td>
<td>-4.52</td>
</tr>
</tbody>
</table>

Figure 10. Estimated RUL at different fractions of lifetime with parameter adaptation.

Figure 11. RUL estimation for bearing 1_7.

Table 5. % Error of RUL estimation of the proposed method

<table>
<thead>
<tr>
<th>Test bearing</th>
<th>Actual RUL (h)</th>
<th>Estimated RUL (h)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing 1_4</td>
<td>0.094</td>
<td>0.086</td>
<td>9.03</td>
</tr>
<tr>
<td>Bearing 1_5</td>
<td>0.446</td>
<td>0.438</td>
<td>1.87</td>
</tr>
<tr>
<td>Bearing 1_6</td>
<td>0.392</td>
<td>0.411</td>
<td>-4.81</td>
</tr>
<tr>
<td>Bearing 1_7</td>
<td>2.103</td>
<td>2.123</td>
<td>-0.97</td>
</tr>
</tbody>
</table>

Figure 12. Clustering of data according to the health state.

features to the fault classes (health states). The workflow of this approach is summarized in Figure 13. The probability of a data point belonging to a given class is then computed from which the remaining useful life is calculated as follows:

\[
RUL_c = \frac{t_c}{1 - \sum_{i=1}^{N} P_i \cdot RL_i} \cdot \sum_{i=1}^{N} P_i \cdot RL_i - d_{c},
\]

where \( t_c \) is the current time, \( P_i \) is the probability of the system being in health state \( i \), such that \( \sum_{i=1}^{N} P_i = 1 \), \( N \) is the total number of health states, \( RL_i \) is the historical fractional RUL of similar systems and \( d_{cj} \) is the duration of stay in the current health state. Various classification algorithms such as support vector machines (SVM), random forests (RF), neural networks (NN) can be employed with this approach. A data point is assigned the class with the highest probability. Figure 14 shows the health state probability for training bearing 1_1 as it degrades with time. Table 5 presents the performance analysis of this approach based on the works of the authors in (Kimotho, Sondermann-Woelke, et al., 2013; Kimotho & Sextro, 2014b).
2.5. Mapping Extracted Features to RUL - Method 5

If the run-to-failure data condition monitoring data and corresponding failure times of similar units are available, then machine learning algorithms can be applied to map the extracted feature to the remaining useful life as shown in Figure 15. Machine learning algorithms such as support vector regression, regression trees, neural networks, extreme learning machines, etc, using regression approach can be employed. This approach is very sensitive to the trendability of the data and as such data processing approaches and feature selection to obtain monotonically changing features should be employed (Kimotho & Sextro, 2014a).

Kimotho et al (Kimotho & Sextro, 2014a) developed a prognostic approach for non-trending data. The approach involves applying an autoregressive model to the extracted features to obtain a monotonically changing feature which is used as the input to extreme learning machine (ELM) algorithm. Normalized RUL of similar units is used as the target in order to cater for units with varying lifetimes. The RUL at the current time is then computed through Eq. (20) which is derived from Figure 16. Given the current time, $t_c$, and the normalized RUL, $F_c$, the estimated remaining useful life $RUL$ can be obtained by similar triangles as follows:

$$RUL = t_{EOL} - t_c = \frac{t_c}{1 - F_c}F_c.$$  

(20)

ELM is a single layer feedforward neural network that utilize generalized inverse matrix operation to compute the weights output weights in a single calculation while the input weights are randomly generated (Huang, Zhu, & Siew, 2006). As a consequence, it does not require adaptation of weights and biases thus significantly reduces the computation time for both training and testing. Table 6 shows the performance analysis of this approach based on the works of (Kimotho & Sextro, 2014a).
3. DISCUSSION

Results show that all the presented approaches estimate the RUL of ball bearings within error bounds of \( \pm 20\% \). However in prognostics, late estimations are usually undesirable since the system may fail before scheduled maintenance. Therefore when computing performance weights of different approaches, a method of penalizing late predictions more than early predictions should be employed. One such method is using the exponential weighting method shown below (Gouriveau et al., 2014) with

\[
w = \begin{cases} 
\exp(-\ln(0.5) \cdot (\frac{\text{Error}}{20})) & \text{if Error} < 0 \\
\exp(\ln(0.5) \cdot (\frac{\text{Error}}{20})) & \text{if Error} \geq 0.
\end{cases}
\] (21)

The maximum score is 1 for the case when Error = 0. Based on this performance evaluation criteria, the performance of the discussed methods (methods 1-5 in subsections 2.1-2.5) is presented in Table 7. Also in Table 7 is the computation time for each method. The computation time includes time taken to extract relevant features and to estimate the remaining useful life but does not include time for feature selection and algorithm training or parameter identification. This is because during testing or real-time prognostics, the trained model or model parameters depending on the algorithm have already been obtained. Table 7 shows that machine learning algorithms utilizing vibration measurements yield the best performance. In particular, method 4 that involves health state estimation is suitable for systems that undergo through various stages of degradation before failure. This method can further be coupled with diagnosis to identify the fault type, location and size at each health state. This information would be of great importance in selecting the most suitable model to use as well as reducing the maintenance time. However, the method requires much longer computation time compared to the wear-temperature correlation methods as well as the complexity of the algorithms involved, both for feature extraction and for machine learning algorithms. Diagnosis is much more difficult to conduct with temperature measurements.

RUL estimation can be improved by sensor data fusion or ensemble of several methods either through simple mean or weighted mean of the RUL as shown in Eq. (22)

\[
\text{RUL}_{\text{ens}} = \frac{\sum_{i=1}^{n} w_i \text{RUL}_i}{\sum_{i=1}^{n} w_i},
\] (22)

where RUL\(_i\) is the RUL estimated by method \(i\) and \(w_i\) is the weight of method \(i\), which can be taken as the mean score of each method in Table 7. For the case of simple mean, \(w_i = 1\). Table 8 shows the effect of combining all or a number of the methods described using weighted mean and simple mean. All possible combinations of the algorithms were evaluated and a combination of three methods (2-3-4) yielded the best results. This shows that an ensemble of a number of algorithms, especially with data fusion from different sensors yields a more robust prognostic approach. Combination of all temperature-based methods yields predictions within a 10% error bound, with all the results being early predictions.

4. CONCLUSION

Five approaches to prognostics of technical systems have been presented. Three methods are empirical based and utilize temperature measurements for prognosis while two methods are data-driven and utilize vibration measurements for diagnosis. The methods have been evaluated on their accuracies to estimate the remaining useful life as well as their suitability for real-time prognostics based on temperature and vi-
Table 7. Performance evaluation of the presented methods

<table>
<thead>
<tr>
<th>Applied Method</th>
<th>Bearing</th>
<th>Time (s)</th>
<th>Error (%)</th>
<th>Time (s)</th>
<th>Error (%)</th>
<th>Time (s)</th>
<th>Error (%)</th>
<th>Time (s)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Method 1</td>
<td></td>
<td>Method 2</td>
<td></td>
<td>Method 3</td>
<td></td>
<td>Method 4</td>
</tr>
<tr>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
<td>_______________</td>
</tr>
<tr>
<td>L4</td>
<td>0.64</td>
<td>1.65</td>
<td>0.64</td>
<td>1.74</td>
<td>0.18</td>
<td>0.35</td>
<td>1.74</td>
<td>0.73</td>
<td>1.74</td>
</tr>
<tr>
<td>L5</td>
<td>0.30</td>
<td>1.68</td>
<td>0.30</td>
<td>1.85</td>
<td>0.90</td>
<td>1.11</td>
<td>1.85</td>
<td>0.94</td>
<td>1.11</td>
</tr>
<tr>
<td>L6</td>
<td>0.65</td>
<td>1.68</td>
<td>0.65</td>
<td>1.87</td>
<td>0.41</td>
<td>1.10</td>
<td>1.87</td>
<td>0.51</td>
<td>1.10</td>
</tr>
<tr>
<td>L7</td>
<td>0.68</td>
<td>1.64</td>
<td>0.68</td>
<td>1.78</td>
<td>0.86</td>
<td>1.90</td>
<td>1.78</td>
<td>0.87</td>
<td>1.90</td>
</tr>
<tr>
<td>mean w</td>
<td>0.69</td>
<td>0.56</td>
<td>0.56</td>
<td>0.86</td>
<td>0.76</td>
<td>0.63</td>
<td>0.66</td>
<td>0.86</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 8. Performance evaluation of ensemble of prognostic methods using weighted mean

<table>
<thead>
<tr>
<th>Applied Method</th>
<th>1-2-3-4-5</th>
<th>1-2-3</th>
<th>2-3-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4</td>
<td>0.43</td>
<td>0.86</td>
<td>0.57</td>
</tr>
<tr>
<td>L5</td>
<td>2.19</td>
<td>0.93</td>
<td>0.43</td>
</tr>
<tr>
<td>L6</td>
<td>7.41</td>
<td>0.77</td>
<td>1.36</td>
</tr>
<tr>
<td>L7</td>
<td>2.72</td>
<td>0.91</td>
<td>2.34</td>
</tr>
<tr>
<td>mean w</td>
<td>0.87</td>
<td>0.92</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Vibration measurements yield better results and can be used for diagnosis and prognosis simultaneously. However, vibration analysis is computationally intensive and calls for relatively complex data driven algorithms for estimating remaining useful life. In addition, data acquisition and processing requirements for temperature sensors are less complex than those of vibration sensors such as accelerometers. With a multifunctional data acquisition system, it is possible to combine sensor information in order to build a low cost as well as a more robust condition monitoring system. This would give rise to the possibility of reducing the number of accelerometers within the system. A condition monitoring system utilizing a single accelerometer for each bearing and two temperature sensors, one as a reference sensor and the other to track the temperature rise of the bearing should be explored. For a more robust prognostic approach, ensemble of different prognostic approaches and in particular the use of sensor data fusion is necessary. The presented results show that fusion of methods utilizing the different sensor data improves the RUL prediction accuracy significantly. Some remaining issues that can be explored include the possibility of using dynamic weights during ensemble. Since the methods perform differently at different time intervals, an ensemble approach utilizing the higher weights at different estimation time intervals should be explored. Integration of uncertainties such as future loading conditions, uncertainty in measurements and uncertainty in model selection and propagation need to be incorporated in the proposed methods. The possibility of reducing the number of vibration sensors in a condition monitoring system for rotating machinery and having a temperature sensor on each component subjected to wear should also be explored. This way the vibration data can be used both for detecting faults and for prognosis while the temperature sensor can be used in locating the faulty component. The end result would be a robust, low cost condition monitoring system for rotating machinery as well as technical systems with wear related failures.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the German Academic Exchange Service (DAAD) and the Ministry of Higher Education Science and Technology (MoHEST), Kenya.

REFERENCES


**Biographies**

James Kuria Kimotho studied mechanical engineering at the Jomo Kenyatta University of Agriculture and Technology, Kenya. Since 2012, he is with the research group Mechatronics and Dynamics at the University of Paderborn. His research focuses on prognostics and health management of mechatroninc systems.

Walter Sextro studied mechanical engineering at the Leibniz University of Hanover and at the Imperial College in London. Afterwards, he was development engineer at Baker Hughes Inteq in Celle, Germany and Houston, Texas. Back as research assistant at the University of Hannover he was awarded the academic degree Dr.-Ing. in 1997. Afterward he habilitated in the domain of mechanics under the topic Dynamic contact problems with friction: Models, Methods, Experiments and Applications. From 2004-2009 he was professor for mechanical engineering at the Technical University of Graz, Austria. Since March 2009 he is professor for mechanical engineering and head of the research group Mechatronics and Dynamics at the University of Paderborn.
Separated two-phase flow model of cryogenic loading operation

Dmitry G Luchinskiy1, Ekaterina Ponizovskaya-Devine2, Michael Khasin3, Dogan Timucin4, Jarred Sass5, Jose Perotti6, Barbara Brown7

1,2 Mission Critical Technologies, Inc., El Segundo, CA, 94035, USA
dmitry.g.luchinsky@nasa.gov

2,3 SGT, Inc., Greenbelt, MD, USA
ekaterina.v.ponizovskayadevine@nasa.gov
michael.khasin@nasa.gov

4 NASA Ames Research Center, Moffett Field, CA, 94035, USA
dogan.timucin@nasa.gov

5,6,7 Kennedy Space Center, Kennedy Space Center, FL, 32899, USA
jose.m.perotti@nasa.gov
jared.p.sass@nasa.gov
barbara.l.brown@nasa.gov

ABSTRACT

We present results of development of separated two-phase cryogenic flow model motivated by NASA plans to mature technology of autonomous cryogenic management on the ground and in space. The solution algorithm is based on the nearly-implicit scheme. We discuss the stability, speed, and accuracy of the algorithm in the context of applications to online health management of cryogenic loading operation. We present the results of validation of the model by comparison with the experimental data obtained during chilldown of the horizontal transfer line obtained at National Bureau of Standards and at the cryogenic testbed in Kennedy Space Center. We demonstrate a good agreement of the model predictions with the experimental data.

1. INTRODUCTION

Modeling boiling two-phase flow is of great interest to many aerospace applications (Konishi & Mudawar, 2015). Fast, time-accurate predictions of cryogenic two-phase flow are especially important for autonomous control of cryogenic propellant loading on the ground and in space (Robert, William, Kelly, & Evelyn, 2012).

Predicting the dynamics of boiling two-phase flows is a longstanding challenging problem (Prosperetti & Tryggvason, 2007; Ishii & Hibiki, 2010). The problem becomes even more complicated when analysis of the cryogenic flow is required under reduced gravity conditions, because there is a severe shortage of useful correlations (Konishi & Mudawar, 2015).

At the same time a number of efficient algorithms (TRACE5, 2007; RELAP5:1, 2012; Nourgaliev & Christon, 2012; Berry et al., 2014) and advanced correlation relations (Choi et al., 2009; RELAP5:4, 2012) for analysis of multi-phase flows have been developed during recent years. It is of great interest to verify if the current state of art techniques in modeling multiphase flows can be effectively applied to the on-line control and integrated health management of cryogenic loading operations.

Here we present the results of development of separated two-phase flow model of cryogenic flow in transfer line using nearly implicit algorithm (RELAP5:1, 2012). We provide details of the algorithm and discuss the stability, speed, and accuracy. We present results of verification and validation of the model using two sets of experimental data obtained during chilldown of the cryogenic transfer line at National Bureau of Standards and at Kennedy Space Center. In conclusions, we summarize obtained results, briefly outline future work and discuss possible applications.

Dmitry Luchinsky et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
2. Model

We model cryogenic loading and chilldown using Wallis equations (Wallis, 1969) for a one-dimensional separated two-phase flow. The model consists of a set of conservation laws for the mass, momentum, and energy for the gas

\[
\left( \alpha \rho_g \right)_t + \frac{1}{A} \left( \alpha \rho_g u_g \right)_x = \Gamma_g
\]

\[
\left( \alpha \rho_g u_g \right)_t + \frac{1}{A} \left( \alpha \rho_g u_g^2 \right)_x + \alpha \rho_g \sin \theta = -\alpha \rho_g \sin \theta
\]

\[
-\tau_{wg} \frac{\dot{w}_g}{A} - \tau_{lg} \frac{\dot{l}_g}{A} + \Gamma_g u_{lg}
\]

\[
\left( \alpha \rho_l e_g \right)_t + \frac{1}{A} \left( \alpha \rho_l u_l \right)_x = -\frac{1}{\Lambda} \left( p \alpha \rho_l u_l + \dot{G}_g \right)_x
\]

\[
-\rho \alpha \frac{\dot{w}_l}{A} + \dot{q}_{lw} \frac{\dot{w}_l}{A} + \dot{q}_{il} \frac{\dot{l}_l}{A} = \Gamma_l h_{lg} + \Gamma_w h_{wg}
\]

and for the liquid

\[
\left( \beta \rho_l u_l \right)_t + \frac{1}{A} \left( \beta \rho_l u_l^2 \right)_x + \beta \rho_l \sin \theta - \tau_{wl} \frac{\dot{w}_l}{A} - \tau_{ll} \frac{\dot{l}_l}{A} - \Gamma_w u_{wl}
\]

\[
\left( \beta \rho_l e_l \right)_t + \frac{1}{A} \left( \beta \rho_l e_l \right)_x = \frac{1}{\Lambda} \left( p \beta \rho_l u_l \right)_x + \rho \beta - \rho \alpha \frac{\dot{w}_l}{A} + \dot{q}_{lw} \frac{\dot{w}_l}{A} + \dot{q}_{il} \frac{\dot{l}_l}{A} = \Gamma_l h_{lg} - \Gamma_w h_{wg}
\]

coupled to the equation for the wall temperature

\[
\rho_w c_w d_w \frac{\partial T_w}{\partial t} = H_{wg} \left( T_g - T_w \right) + H_{wl} \left( T_l - T_w \right) + H_{amb} \left( T_{amb} - T_w \right).
\]

Each phase of the fluid is characterized by its own void fraction \( \alpha_k \), density \( \rho_k \), temperature \( T_k \) (energy \( e_k \)), and velocity \( u_k \), where index \( k \) takes values \( g \) for the gas/vapor phase and \( l \) for liquid phase. A closed system of equations can be obtained assuming local pressure values for the two phases are equal \( p_g = p_l = p \) and that the source terms on the right-hand sides of the balance equations are exclusive algebraic functions of state and flow parameters. In addition, each phase is characterized by the mass flow rate \( \Gamma_k \) (negative for the liquid), heat flux to the dry wall \( q_{pw} \) and to the wetted wall \( q_{lw} \), heat flux at the interfaces \( q_{g(l)(l)} \), pressure losses at the dry and wetted wall \( \tau_{g(l)(w)} \) and at the interface \( \tau_l \), and relative velocity at the interface \( u_{il} \). The geometry of the sections is determined by the length \( L \) of the section, cross-section area \( A \), height \( y \), wetted \( l_{wl} \), dry \( l_{wg} \), and interface \( l_i \) perimeters.

The heat transfer to the wall is characterized by the heat transfer coefficient from gas (liquid) to the wall \( H_{g(l)w} \) and the heat transfer coefficient from the environment to the wall \( H_{amb} \). \( T_{g(l)} \) is the temperature of the gas (liquid), and \( \rho_w \), \( c_w \), and \( A_w \), are wall material density, specific heat, and cross-section area.

The set of equations (1) - (3) is closed by the volume conservation condition and equations of state

\[ \alpha_g + \alpha_l = 1, \quad \rho_g(\alpha) = \rho_p(\alpha) \left( \rho, e(\alpha) \right) \]

We tested a number of algorithms that solve equations (1) - (3) in various approximations (Luchinsky, Smelyanski, & Brown, 2014a, 2014c; Hafiychuk et al., 2014). Here we present details of the nearly implicit algorithm, which was developed following closely the ideas of (RELAP5-3.2, 2012), and delivers one of the best performance in terms of speed, level of details, and accuracy.

3. Algorithm

The calculations of the velocities, pressure, and provisional values of densities and energies at the first sub-step of the algorithm is the key to the stable performance of the nearly-implicit method. These calculations are designed to break limitations of the acoustic and material CFL and to increase implicitness of the method. This sub-step is structured as a predictor of the fractional time step technique. The CFL limitations are lifted by increased implicitness of calculations of the new velocities and pressures in the system. The implicitness is further increased by estimations of the provisional values of the densities, energies, heat and mass transfer coefficients.

The first step of the algorithm can be briefly summarized as follows:

- Solve expanded equation with respect to pressure in terms of new velocities;
- Solve momenta equations written in the form of block tri-diagonal matrix for the new velocities;
- Find new pressure;
- Find provisional values for energies and void fractions using expanded equations;
- Find provisional values of mass fluxes and heat transfer coefficients using provisional values of temperatures obtained.

At the second step new (corrected) values of the densities, void fractions, and energies are found by solving the unexpanded conservation equations for the phasic masses and energies using provisional values for the heat and mass fluxes in source terms. The solution is reduced to independent solution of four tri-diagonal matrices. The values of pressure and velocities in these matrices are taken at the new time step.

We now provide some further details of the algorithm.

3.1. First step of the algorithm

To find new pressure we first express it in terms of new velocities. To do so we formally solve the following set of expanded conservation equations discretized on the main grid (the notation convention for the grid is shown in Fig. 1)
The sum density equation
\[
\alpha_n^L \frac{d \rho_n^{L+1}}{dt} + \frac{\rho_n^L}{\rho_l^L} \frac{d \rho_l^{L+1}}{dt} + \frac{d \alpha_n^{L+1}}{dt} (\rho_g - \rho_l)^n_L + \\
\Delta t \left( \frac{\rho_n^L}{\rho_l^L} \right)_{n+1}^{L+1} (\rho_{g,1} - \rho_l^L) + \\
\Delta t \left( \frac{\rho_n^L}{\rho_l^L} \right)_{l+1}^{L+1} (\rho_g - \rho_{l,l}) = 0.
\] (4)

The difference density equation
\[
\alpha_n^L \frac{d \rho_n^{L+1}}{dt} - \frac{\rho_n^L}{\rho_l^L} \frac{d \rho_l^{L+1}}{dt} + \frac{d \alpha_n^{L+1}}{dt} (\rho_g - \rho_l)^n_L + \\
\Delta t \left( \frac{\rho_n^L}{\rho_l^L} \right)_{n+1}^{L+1} (\rho_{g,1} - \rho_l^L) - \\
\Delta t \left( \frac{\rho_n^L}{\rho_l^L} \right)_{l+1}^{L+1} (\rho_g - \rho_{l,l}) = 2 \Gamma_n^L g_L.
\] (5)

The gas energy equation
\[
(\rho_e + p)_g^L \frac{d e_g^L}{dt} + (\alpha \rho_e)_g^L \frac{d e_l^L}{dt} + \\
\Delta t \left( (\alpha \rho_e + p)_g^L \right)_{g,l} (\rho_{g,l}^L)_{g,l+1} + \\
(\alpha \rho_e)_g^L \frac{d e_g^L}{dt} = \left[ H_{gw}^L (T_w^L - T_{g}^{L,n+1})_L S_{gw} + \\
H_{ig}^L (T_{g,l}^{n+1} - T_{g}^{L,n+1})_L S_{ig} + \\
\Gamma_{g,L}^n \left( \rho_{g,L}^n + \rho_{g,L}^n \right) A \frac{\Delta t}{V}
\right].
\] (6)

and the liquid energy equation
\[
-(\rho e_l + p)_L^L \frac{d e_l^{L+1}}{dt} + (\alpha \rho e)_l^L \frac{d e_l^{L+1}}{dt} + \\
\Delta t \left( (\alpha \rho e + p)_l^L \right)_{l,L} (\rho_{g,l}^L)_{l,L+1} + \\
(\alpha \rho e)_l^L \frac{d e_l^{L+1}}{dt} = \left[ H_{lw}^L (T_w^L - T_{l}^{L,n+1})_L S_{wl} + \\
H_{il}^L (T_{l,l}^{n+1} - T_{l}^{L,n+1})_L S_{il} + \\
\Gamma_{gL}^L \left( \rho_{g,L}^n \right)_L A \frac{\Delta t}{V}
\right].
\] (7)

Here \(\eta_{l,L+1} = A_{ij} n_{ij}^n / V_L\).

Using the following expansion for the densities
\[
d \rho_{g,L} \approx \frac{d \rho_{g,L}}{d \rho_{g,L}} \rho_{g,L}^n + \frac{d \rho_{g,L}}{d \rho_{g,L}} \rho_{g,L}^n,
\]
and keeping temperatures at the previous time step the equation (4) – (7) can be written in the matrix form as follows
\[
A_n^L \begin{bmatrix} 
\frac{d \rho_g}{dt} \\
\frac{d \rho_l}{dt} \\
\frac{d e_g}{dt} \\
\frac{d e_l}{dt} \\
d p \\
d e 
\end{bmatrix}^{n+1} = a_n^L d \rho_{g,L+1}^{n+1} + b_n^L d \rho_{L,L+1}^{n+1} + c_n^L d e_{L+1}^{n+1} + e_x^n.
\] (8)

The matrix and vector elements in equation (8) are defined in Appendix A.

By solving (8) for \(d \rho_{L,L+1}^{n+1}\), we have
\[
d \rho_{L,L+1}^{n+1} = \left( A_n^L \right)^{-1} (a_n^L d \rho_{g,L+1}^{n+1} + b_n^L d \rho_{L,L+1}^{n+1} + c_n^L d e_{L+1}^{n+1} + e_x^n),
\] (9)

where \((A_n^L)^{-1}\) is the 4-th row of the inverted matrix \(A_n^L\).

By using (9) to eliminate new pressure from momenta equations we obtain block tri-diagonal matrix equation for the new velocities in the form
\[
\begin{bmatrix}
B_1 & C_1 \\
A_2 & B_2 & C_2 \\
\vdots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
A_N & B_N & C_N
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_N
\end{bmatrix}
= \begin{bmatrix}
k_1 \\
k_2 \\
\vdots \\
k_N
\end{bmatrix}
\] (10)

where coefficients of \(2 \times 2\) matrices \(A, B,\) and \(C\) and vectors \(x_m = [d u_{g,m}^{n+1}, d u_{l,m}^{n+1}]^T\) and \(k_m = [n_{2m-1}, n_{2m}]^T\) are given in Appendix B.

3.2. Second step

The new velocities and pressure and provisional values for the temperatures can be substituted into unexpanded equations of the mass and energy conservations (first and last equations in (1) and (2), see (RELAPS5.1, 2012) for further details). All four equations can be re-written in the standard tridiagonal form
\[
a_L d u_{L,L+1}^{n+1} + b_L d u_{L,L}^{n+1} + c_L d u_{L,L+1}^{n+1} = S_L,
\] (11)

and solved efficiently using standard solvers. The explicit form of the conservative terms \(d u_{L,L}^{n+1}\), source terms \(S_L\), and matrix coefficients \(a_L = -\lambda_{L+1}^{n+1} \gamma_{L+1}^{n+1}, b_L = 1 + \lambda_{L+1}^{n+1} \gamma_{L+1}^{n+1} - \lambda_{L+1}^{n+1} \gamma_{L+1}^{n+1},\) and \(c_L = \lambda_{L+1}^{n+1} \gamma_{L+1}^{n+1}\) is given in Appendix C.

The structure of these terms is clear from the equations (1) and (2). Here we provide as an example the structure of the \(S_L\) term for the first equation (1)
\[
S_{g,L} = \frac{\Gamma_{g,L}^n \Delta t}{V_L} - (\alpha \rho)^L_{g,j+1} \gamma^{n+1}_{g,j+1} + (\alpha \rho)^L_{g,j} \gamma^{n+1}_{g,j}.
\] (12)

Note that coefficients \(a_L, b_L,\) and \(c_L\) for the tridiagonal matrix are the same for each pair of density and energy equations.

Figure 1. The notation convention for the centers of the control volumes on the main and staggered grid.
Calculation of the updated densities, void fractions, and energies completes the time step of integration of the semi-implicit algorithm.

3.3. Correlations

The thermodynamic properties of the cryogenic flow are modeled using multiple correlations for the source terms. The heat transfer and pressure losses correlations are based on the flow pattern recognition. Currently we are using the flow map introduced for refrigerants by Wojtan et al. (Wojtan, Ursenbacher, & Thome, 2005). It is a modification of the map by Kattan et al (Kattan, Thome, & Favrat, 1998) and it is based on the Steiner map (Steiner & Kind, 2010). The latter map determines transitions between flow regimes as a relation between fundamental hydrodynamic numbers and geometrical parameters of the flow.

The two-phase friction pressure drop \( \frac{dp}{dz} \) is defined using Lockhart-Martinelli correlations (Chisholm, 1967). The pressure losses between the phases are partitioned (RELAP5:1, 2012) as follows

\[
\tau_{wg} \dot{w}_g = \alpha_g \left( \frac{dp}{dz} \right)_{2\phi} \left( \frac{1}{\alpha_g + \alpha_l Z^2} \right),
\]

\[
\tau_{wl} \dot{w}_l = \alpha_l \left( \frac{dp}{dz} \right)_{2\phi} \left( \frac{Z^2}{\alpha_l + \alpha_g Z^2} \right).
\]

Here \( Z^2 \) is given by

\[
Z^2 = \left( f_{g} Re_{g} \rho_g u_l^2 \frac{\alpha_{wl}}{\alpha_l} \right) / \left( f_{g} Re_{g} \rho_g u_l^2 \frac{\alpha_{wg}}{\alpha_g} \right),
\]

friction factor \( f_{g}(l) \) is approximated using Churchill formula (Churchill, 1977). Coefficients \( \alpha_{wl} \) and \( \alpha_{wg} \) depend on the flow pattern (RELAP5:1, 2012).

The interface drag is given by

\[
\tau_{sg} = -\tau_{tl} = \frac{1}{2} C_D \rho_g \left( u_g - u_l \right) \left( u_g - u_l \right),
\]

where interfacial drag coefficient \( C_D \) depends on the flow pattern (TRACE5, 2007).

The heat transfer correlations are subject of extensive research (Tong & Tang, 1997; Faghri & Zhang, 2006) and will be considered in more details elsewhere. Here we briefly outline the framework for the development of the correlation module. The heat fluxes at the wall and at the interface are defined as follows

\[
\dot{q}_{wg} = H_{wg} \left( T_w - \bar{T}_g \right); \quad \dot{q}_{lg} = H_{lg} \left( \bar{T}_{l,s} - \bar{T}_g \right);
\]

\[
\dot{q}_{wl} = H_{wl} \left( T_w - \bar{T}_l \right); \quad \dot{q}_{tl} = H_{tl} \left( \bar{T}_{l,s} - \bar{T}_l \right).
\]

Here the heat transfer coefficients \( H_{wg(l)} \) and \( H_{lg(l)} \) have to be defined for each flow regime.

The total interfacial mass flux per unit volume \( \Gamma_g = \Gamma_{wg} + \Gamma_{lg} \) in equations (1) and (2) are defined in terms of the heat fluxes as follows at the wall

\[
\Gamma_{wg} = f_{cor} \frac{\dot{q}_{wl}}{h_g - h_l^s};
\]

and at the interface

\[
\Gamma_l = \frac{\dot{q}_{tl} + \dot{q}_{lg} l_i}{h_g - h_l^s};
\]

where the enthalpies are given by the following expression

\[
h_g - h_l^s = \begin{cases} h_{g,s} - h_l, & \Gamma > 0 \\ h_g - h_{l,s}, & \Gamma < 0 \end{cases}.
\]

The heat transfer coefficients \( H_{wg(l)} \) and \( H_{lg(l)} \) are defined using heat transfer correlations at the wall and at the interface that depend on the flow pattern, mass flow rate, and flow quality. For example, in the single phase flow the heat transfer is determined by the largest of four possible heat transfer coefficients corresponding to laminar and turbulent, forced and natural convection (see e.g. (TRACE5, 2007; Nellis & Klein, 2009; Holman, 1989)).

Another important example is the heat transfer correlations for the horizontally stratified flow that should recognize nucleate, transition, and film boiling regimes. At present the flow boiling correlations in horizontally stratified regime are defined as corrections to the pool boiling correlations. In this context the correlations are focused on the estimations of the onset of nucleate boiling, critical heat flux, and minimum film boiling parameters and corresponding corrections due to forced convection.

The onset of nucleate boiling is correlated using ideas of (Frost, Dzakovic, & American Society of Mechanical, 1967) (see also (Ghiaasiaan, 2008)). The critical heat flux is modelled using Griffith flow corrections (Griffith, 1975) to Zuber correlations (Zuber & Tribus, 1958). The onset of film boiling is estimated using Iloeje flow corrections (Iloeje, Plummer, Rohsenow, & Griffith, 1982) to Berenson correlations (Berenson, 1961). The correlations for two other regimes recognized by the present version of the present model include annular and mist flow and follow (TRACE5, 2007; RELAP5:4, 2012) in micro-gravity.

The ability to include the full range of correlations for the two-phase non-equilibrium boiling flow is an important advantage of the model especially in the view of the possible applications of the model to the autonomous control of cryogenic fluid management.

4. Stability and Speed

The resulting very efficient and fast computational scheme is a variation of the algorithm developed in (RELAP5:1, 2012). It involves inversion of \( N 4 \times 4 \) matrices, solution
of \((N-1)\) tree-block-diagonal matrix equation, solution of four \(N \times N\) tridiagonal matrix equations, and \(N \times m\) explicit computations. The implicits of the nearly-implicit scheme (RELAP5:1, 2012) is elevated to break both acoustic and material CFL (Nourgaliev & Christon, 2012), yet the algorithm resolves both types of waves and scales linearly with the number of control volumes.

However, a number of inherent instabilities can severely slow down computations and sometimes prevent the convergence of the scheme. Specifically, the basic Wallis (Wallis, 1969) model (1), (2) is known to be non-hyperbolic and stiff (Staedtke, 2006). It also lacks positivity and subject to the instabilities due to the phase appearance-disappearance (Nourgaliev & Christon, 2012; Cordier, De-gond, & Kumbaro, 2014).

The non-hyperbolicity of the model is related to the existence of a pair of complex eigenvalues whenever \(u_g \neq u_l\). It can be corrected using variety of the methods including e.g. addition of virtual mass term (RELAP5:1, 2012; Staedtke, 2006).

The stiffness of the model is related to the nature of the source terms and results in the set of eigenvalues for the \(4 \times 4\) matrices \(A^g\) that may differ by twelve orders of magnitude. Numerical experience shows that coding explicit solution for the inverted \(4 \times 4\) matrices at the first step of the algorithm can substantially improve its performance.

### 4.1. Lack of positivity

The lack of positivity and the phase appearance-disappearance problems could not be completely eliminated in the existing methods of solution. It is important to note that the corresponding instabilities are not related to the incomplete implicitness of the method and are sever even in fully implicit versions (Bestion, 2000). It is related to the fact that for small values of the volume fraction of the disappearing phase and for large enough time steps the mass fluxes related to the mass exchange between phases may exceed the quantity of the remaining mass.

To mitigate this problem limiters on the values of pressure and a time step control are introduced to the algorithm. The problem can also be partially mitigated by using smoothers discussed in the following section.

### 4.2. Phase appearance disappearance

The most significant stability issue is related to the phase appearance disappearance (Bestion, 2000; Nourgaliev & Christon, 2012; Cordier et al., 2014). Although both mathematical and physical roots of this instability are clearly established no general solution of the problem was proposed so far.

In the limit of vanishing volume fraction of one of the phases, the two phases almost decouple and the minority phase obeys a pressureless gas dynamics system (Cordier et al., 2014). The physical origin of this problem can be traced back to the large density perturbation due to gravitational instability (Zeldovich, 1970). (see (Bouchut, Jin, & Li, 2003)).

A more formal approach reveals that in the limit of vanishing phase this system becomes non-hyperbolic because the Jacobian of the flux matrix is not diagonalizable and most shock-capturing schemes, which require a complete basis of eigenvectors, breakdown in this limit (Cordier et al., 2014). Although no general satisfactory solution was found so far, the problem can be partially mitigated by using smoothing scheme following e.g. recommendations by Liou (Chang & Liou, 2007) and adjust temperature, velocity, and density according to the following expression

\[
\phi_{adj} = g(x)\phi_d + (1 - g(x)) \phi_c, \tag{13}
\]

where

\[
g(x) = x^2 (2x - 3); \quad \text{and} \quad x = \frac{\alpha_d - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}},
\]

Here “d” stands for disappearing phase and “c” for conducting phase. The values of the minimum and maximum void fraction, for which smoothing (13) is applied are established using numerical experimentation and set currently at the \(1 \times 10^{-7}\) and \(1 \times 10^{-2}\) respectively.

We note that discussion of the instability due to the lack of positivity in (Bestion, 2000) is related to density-energy coupling, while discussion of the singularity due to the phase disappearance in (Cordier et al., 2014) is related to the density-momentum coupling. In practice, both types of instabilities can be coupled to each other. An analysis shows that conditioning of the matrix \(A_x\) in the first step of the algorithm has significant impact on the overall performance of the nearly implicit scheme and its stability properties. The results of this analysis will be presented elsewhere.

### 5. Verification and Validation

In the view of possible applications of the algorithm to the online integrated health management of cryogenic systems the verification and validation (V&V) of the algorithm using a large set of experimental data becomes especially important. For the validation we are using experimental data obtained at National Bureau of Standards (Brennan, Brentari, & Smith, 1966) and at the experimental loading system at Kennedy Space Center (Robert et al., 2012). Below we present some results of the V&V, some additional information can be found in (Luchinsky, Smelyanskiy, & Brown, 2014b).

#### 5.1. Loading system

The sketch of the model of the cryogenic loading system developed at KSC for testing autonomous regimes of operation is shown in Fig. 2. The system consists of the storage tank
Figure 2. Sketch of the finite volume model of the system built in SINDA-FLUINT. Different colors correspond to the pipes of different diameters. Location of the main valves and sensors is indicated by arrows.

(left) and the external tank (right). The tanks are connected by a pipeline. A number of control valves and sensors can be seen along the loading path highlighted by colors. The location of the main input valve and of the pump are indicated in the figure. The cryogenic fluid is nitrogen. The characteristic speed of sound in the gas phase is about \( a_g = 200 \) m/sec and is about \( a_l = 700 \) m/sec in the liquid phase. The characteristic transient time of the pressure equilibration is less than 1 sec.

The system includes multiple control (damp) valves that remotely regulate flow during chilldown and loading. Temperature and pressure sensors located at several places along the system monitor the state of the flow. Importantly, the temperature of the flow is measured in the middle of the pipe. The real time accurate monitoring of complex nominal and off-nominal flow regimes during the loading and remote control of the flow by the valves make current system a unique and interesting experimental testbed well suited for the development of autonomous loading operations.

5.2. Pressure waves

The important feature of the nearly-implicit algorithm (RELAP5.1, 2012) is the ability to resolve both pressure and material waves. In this section we verify that the algorithm can resolve pressure wave propagation in the system.

To simplify the interpretation of the obtained results the following boundary conditions were specified. All the dump valves are closed. All the internal line control valves are opened. The pipes are initially filled with nitrogen gas at the temperature \( T = 300K \) and pressure 1 atmosphere. The storage tank is filled with nitrogen gas at the same temperature. The pressure in the storage tank is 3.5 atm and in the vehicle tank 0.99 atm. The input and output valves are closed.

At the time instant \( t = 0 \) sec input and output valves are opened to 5% of their area allowing for a small shock wave propagation in the tube. The pressure in the shocks is less than 0.01 atm different as compared to the initial pressure in the pipes. These weak normal shock waves propagate in the pipes at nearly speed of sound.

The results of the integration of the model with 295 control volumes and time step \( \Delta t = 0.025 \) ms under given initial conditions are shown in the Fig. 3. The first snapshot taken at 0.0125 sec shows propagation of the shock waves away from the input and output valves towards the middle of the pipe. The velocities behind the shock waves on both sides of the system are positive because the pressure gradients on the input and output valves have the same direction. At the same time, the shock waves propagate in the opposite directions with velocity approximately 350 m/sec, which is very close to the speed of sound \( (a = 353 \) m/sec) in gaseous nitrogen at 300 K.

In the second snapshot taken at 0.01925 sec the wave front on the left encounters the transition from the 6 inch diameter pipe to the 3 inch diameter pipe at the distance approximately 12.5 m from the entrance. At this location the wave front splits and two waves: Transmitted wave continues to move forward, while reflected wave is propagating in the opposite direction.

Similar splitting occurs about 7 ms earlier at the location of another junction between the 2 inch and 4 inch pipes approximately 10 m away from the exit. Both locations and the direc-
tions of the wave propagation are indicated by black vertical arrows.

The third snapshot taken at 0.02375 sec shows how the wave front on the left is approaching the third junction located approximately 17 m from the entrance. This process continues until the velocity distribution in the pipe approaches quasi-equilibrium in approximately a few hundred milliseconds. The last snapshot (d) shows the nearly equilibrium distribution of the pressure in the pipe.

Examples of simulations of material waves can be found in (Luchinsky et al., 2014b). Below we provide two examples of validation of the algorithm.

5.3. Chilldown test by the National Bureau of Standards

The first example of the model validation is based on a well-known set of experimental data obtained for chilldown of horizontal transfer line at National Bureau of Standards (Brennan et al., 1966).

The vacuum jacketed line was 61 m long. The internal diameter of the copper pipe was 3/4 inches. Four measurement stations were located at the distance 6, 24, 42, and 60 m from the input valve. In the particular experiment, which was considered in this work the working liquid was nitrogen and pressure in the storage tank was 4.2 atm.

The comparison with the experimental data is shown in Fig. 4. The top figure shows model predictions for the fluid temperature as compared to the experimental time-traces. The bottom figure shows similar comparison for the wall temperature.

It can be seen from the figures that there are three different regions in the pipe. In the first region the pipe cools down to the liquid temperature in 30 sec and this part of the pipe is filled with liquid. In the second region the temperature is gradually changing towards liquid temperature during first 90 sec. In this region the dryout transition occurs. In the remaining part of the pipe the temperature stays high for a long period of time, indicating that the front of cold gas and the transition region are slowly moving along the pipe towards exit with the speed approximately 0.5 m/sec.

It can also be noticed from the figure that a sudden temperature drop occurs in the system whenever that fluid and wall temperature cool down to approximately 130 K. This temperature drop corresponds to the transition from the film boiling regime to the intermittent regime of boiling, when the heat transfer to the wall is sharply rising towards its maximum value, corresponding to the critical heat flux.

It can be seen from the figure that the model can reproduce all the experimentally observed features mentioned above. Further improvements of the model predictions in the transition region can be achieved by using more accurate correlations for various flow regimes as will be discussed in details elsewhere.

5.4. Chilldown cryogenic testbed

Our last example of the validation of the separated model is based on the comparison of the model predictions with the experimental time traces obtained during chilldown of the transfer line of the cryo-testbed at KSC. The schematics of the KSC cryo-testbed is shown in the Fig. 2.

There are two main features that differentiate this case from the previous example. Firstly, the complex geometry and a large number of components with minor losses and heat leaks render modeling of this system a challenging problem. Sec-
Figure 6. Model predictions (green lines) for the pressure in the pipes are shown in comparison with the experimental time-traces (black lines) at 6 different locations along the transfer line.

 Secondly, the flow during chilldown is controlled by the bleed valves, in-line control valves, and by the storage tank pressure. As a result, the chilldown dynamics is very different from the previous case, where it was mainly controlled by the heat transfer from the boiling flow to the pipe walls.

The chilldown at the cryo-testbed was achieved in three steps (stages). Initially the input valve is open, allowing some liquid to flow into the hot pipes. This liquid immediately evaporates creating high pressure gas, blocking further liquid flow, and keeping the rest of the pipe hot. This stage can be recognized as a small drop in the fluid temperature in the Fig. 5 and the first pressure jump in the Fig. 6 (a), (b), and (c).

The second step begins when the second valve in the middle of the transfer line is opened. Simultaneously, a few bleed valves are opened creating enough suction for the cold flow to fill in half of the transfer line. This stage can be recognized by the temperature drop in Fig. 5 (a) - (d). Note, that the temperature remains high in the rest of the pipe until the end of this stage as can be seen in the Fig. 5 (e) and (f). One can also recognize this stage of chilldown by the pressure jump in Fig. 6 (d), while in the rest of the pipe the pressure stays low until the end of this stage (see Fig. 6 (e) and (f)).

The last chilldown step begins when the third line valve is open allowing the fluid to flow through the entire transfer line. This step can be identified as a fast temperature drop in Fig. 5 (e) and (f). It can also be seen as pressure jump in the Fig. 6 (e) and (f).

The comparison of the model predictions (green lines in the Fig. 5 and 6) with the experimental time traces (black lines) shows that the model can quite accurately reproduce all three stages of the chilldown. We note that the model integration is fast. The integration of nearly 3000 sec of real time shown in the figures takes several seconds on laptop.

6. CONCLUSION

To summarize we note that the developed separated model of the two-phase cryogenic flow allows for the fast and time accurate predictions of the chilldown dynamics in large scale systems with complex valve control. These capabilities of the model pave the way for its on-line applications to the integrated health management as will be discussed in details in a separate presentation.

We note also that the fast integration of the model using a variation of the nearly implicit algorithm (RELAP5:1, 2012) opens a possibility of development of a set of optimization tools that can be used to optimize the loading regimes and speed up the development and design of cryogenic transfer lines under normal and reduced gravity conditions. The development and testing of such optimization tools is currently under way and will be a subject of future presentations.

Importantly, the separated model allows one to incorporate and verify a wide range of correlations available for cryogenic boiling flows. Once the model is equipped with the set of optimization tools many of the unknown parameters of those correlations can be learned from the experimental data under various complex flow conditions required for the development of autonomous cryogenic loading operations in micro-gravity.

ACKNOWLEDGMENT

This work was supported by the Advanced Exploration Systems and Game Changing Development programs at NASA HQ.

NOMENCLATURE

- $u$: velocity
- $T$: temperature
- $p$: pressure
- $e$: specific energy
- $h$: specific enthalpy
- $H$: heat transfer coefficient
- $g$: gravity
- $Re$: Reynolds number
- $t$: time
- $\Delta t$: time step
- $A$: cross-sectional area
- $S$: wall surface area
- $V$: volume of the control volume
- $l$: perimeter
- $x$: coordinate along the pipe
- $y$: height of the control volume
- $\dot{q}$: heat flux
- $c$: specific heat
Greek:
\( \alpha \) gas void fraction
\( \beta \) liquid void fraction
\( \rho \) density
\( \tau \) wall shear stress
\( \Gamma \) mass flux per unit volume

Subscript
\( g \) gas
\( l \) liquid
\( w \) wall
\( i \) interface

References


RELAP5:1. (2012). Relap5-3d code manual volume I: Code structure, system models, and solution methods (Computer software manual No. INEEL-EXT-98-
00834). INEL.
TRACE5. (2007). Trace v5.0 theory manual field equations, solution methods, and physical models [Computer software manual].

**Biographies**

**Dmitry G. Luchinsky** is a senior research scientist in MCT Inc. He obtained his MSc and PhD in physics in Moscow working on nonlinear optics of semiconductors. He is an author of more than 100 publications. He has been on a number of occasions a Royal Society Visiting Fellow and a NASA visiting scientist. He worked as a senior scientific researcher in VNII for Metrolological Service (Moscow, Russia) and as a Senior Research Fellow in Lancaster University (Lancaster, UK). His research interests include nonlinear optics, stochastic and chaotic nonlinear dynamics, dynamical inference, fluid dynamics, ionic motion. His research is currently focused on theory and CFD of gas dynamics and cryogenic flows.

**Ekaterina Ponizovskaya-Devine** Dr. Ekaterina Ponizovskaya-Devine is a research scientist in SGT since 2009. Her current work is related to integrated health management and control of cryogenic propellant loading systems and two-phase cryogenic flow modeling. She obtained her PhD at Moscow institute of Physics and Technology in Physics, working on stochastic resonance in non-equilibrium electron-hole plasma. She was a post-doc in Spanish National Research Council, working in photonics, nanotechnology, negative index materials. In 2005 she had Visiting Scholar position at HP Lab working in nanotechnology. Her research interests are physics based models, two-phase flow, nanotechnology, physics model based fault diagnostics, optimization and control.

**Michael Khasin** Dr Michael Khasin is a Senior Researcher in SGT Inc., working at NASA Ames Research Center. He holds B.Sc. in Physics (Honors Program), 2001, M.Sc. in physics, 2003, and PhD in chemical physics, 2008, from the Hebrew University of Jerusalem. As a postdoctoral researcher at Michigan State University, Massachusetts Institute of Technology, and University of Michigan, Michael worked on the theory of non-equilibrium systems and their control and transport in disordered system. His research interests include applied physics, nonlinear and stochastic dynamics, large fluctuations and their control in non-equilibrium systems and efficient simulation of complex dynamics. He has been an invited speaker to many international and national conferences and workshops. Currently his research is focused on the theory of heat transfer, fluid dynamics, and fluid structure interaction.

**Dogan A. Timucin** is a Research Scientist in the Intelligent Systems Division at NASA Ames Research Center. He holds B.S. (1989), M.S. (1991), and Ph.D. (1994) degrees in electrical engineering, and has been with NASA since 1995. His research interests are in the general areas of classical electrodynamics and optics, statistical physics, nonlinear dynamics, Bayesian inference, and numerical optimization.

**Appendix A**

The elements of the matrix $A_x^n$ can be written in the following form (index ($j$) enumerates columns of the matrix $A_x^n$)

$$A_x^{(1)} = \begin{bmatrix} \alpha_{g,L}^n (\partial \rho)_{g,L}^n + \rho_{g,L}^n \alpha_{g,L}^n (\partial \rho)_{g,L}^n \\ 0 \\ \alpha_{g,L}^n (\partial \rho)_{g,L}^n \end{bmatrix}$$

$$A_x^{(2)} = \begin{bmatrix} -\beta_{g,L}^n (\partial \rho)_{g,L}^n \\ \beta_{g,L}^n (\partial \rho)_{g,L}^n \\ 0 \\ \beta_{g,L}^n (\partial \rho)_{g,L}^n \\ \beta_{g,L}^n (\partial \rho)_{g,L}^n \end{bmatrix}$$

$$A_x^{(3)} = \begin{bmatrix} \rho_{g,L}^n + \rho_{g,L}^n \\ (\partial \rho)_{g,L}^n \\ (\partial \rho)_{g,L}^n \\ (\partial \rho)_{g,L}^n \\ (\partial \rho)_{g,L}^n \\ -p_{g,L}^n - p_{g,L}^n \\ \rho_{g,L}^n - p_{g,L}^n \end{bmatrix}$$

$$A_x^{(4)} = \begin{bmatrix} \alpha_{g,L}^n (\partial \rho)_{g,L}^n - \beta_{g,L}^n (\partial \rho)_{g,L}^n \\ (e \alpha_{g,L}^n)_{g,L}^n \\ \beta_{g,L}^n (\partial \rho)_{g,L}^n \\ \alpha_{g,L}^n (\partial \rho)_{g,L}^n + \beta_{g,L}^n (\partial \rho)_{g,L}^n \end{bmatrix}$$

Columns of the vector-multipliers for gas velocities on the
right hand side of the eq. (8) have the form
\[ a^n_x = - \left[ \begin{array}{c} \frac{n_{g,j}^n}{g,j+1} + \rho_n^{g,j+1} \\ \frac{n_{g,j}^n}{g,j+1} \end{array} \right] \cdot n_{g,j+1}^n; \]
\[ b^n_x = \left[ \begin{array}{c} \frac{n_{g,j}^n}{g,j} + \rho_n^{g,j} \\ \frac{n_{g,j}^n}{g,j} \end{array} \right] \cdot n_{g,j}^n; \]

Vector-columns for the liquid velocities are
\[ c^n_x = \left[ \begin{array}{c} \frac{n_{g,j}^n}{g,j+1} \frac{n_{g,j}^n}{g,j+1} + \rho_n^{g,j+1} \\ \frac{n_{g,j}^n}{g,j+1} \frac{n_{g,j}^n}{g,j+1} \end{array} \right] \cdot n_{g,j+1}^n; \]
\[ d^n_x = \left[ \begin{array}{c} \frac{n_{g,j}^n}{g,j} \frac{n_{g,j}^n}{g,j} + \rho_n^{g,j} \\ \frac{n_{g,j}^n}{g,j} \frac{n_{g,j}^n}{g,j} \end{array} \right] \cdot n_{g,j}^n; \]

where the coefficients \( n_{g,j}^n \) are of the form
\[ n_{g,j}^n = \tilde{\alpha}_{g,j} n \Delta t A_j^j / V_L; \quad n_{g,j}^n = \tilde{\alpha}_{i,j} n \Delta t A_j^j / V_L. \]

Finally, the free vector in eq. (8) is written as follows
\[ e^n_x = \left[ \begin{array}{c} \frac{n_{g,j}^n}{g,j+1} \frac{n_{g,j}^n}{g,j+1} + \rho_n^{g,j+1} \\ \frac{n_{g,j}^n}{g,j+1} \frac{n_{g,j}^n}{g,j+1} \end{array} \right] \cdot n_{g,j+1}^n + b^n_x n_{g,j}^n + c^n_x n_{g,j+1}^n + d^n_x n_{g,j}^n. \]

APPENDIX B
The equations for velocities solved in the nearly-implicit-algorithm have the following form (RELAP5:1, 2012)
\[ \alpha \rho_g u_{g,t} + \beta \rho_g u_{g,t} + \frac{\alpha \rho_g}{2} \left( u_g^2 \right)_x + \frac{\beta \rho_g}{2} \left( u_g^2 \right)_x + p_x = -\rho_m z_{x} - p_{g} F_{w,g} - \beta \rho_g u_{F_{w,L}} - \Gamma_f \left( \rho_g - \rho_g \right), \]
\[ u_{g,t} = u_{g,t} + \frac{\alpha \rho_g}{2} \left( u_g^2 \right)_x + \frac{\beta \rho_g}{2} \left( u_g^2 \right)_x + \left( \frac{p_x}{\rho_g} \right), \]
\[ u_{f} F_{w}-u_{f} F_{w}-u_{f} = \Gamma_f \left( u_{f} - u_{f} \right) + \frac{\alpha \rho_g}{\rho_m F_{i}} \left( u_{g} - u_{g} \right) + \rho_m F_{i} \left( u_{g} - u_{g} \right) + \frac{\rho_m}{\alpha \rho_g} \rho_m F_{i} \left( u_{g} - u_{g} \right) \]

Applying equation (9) to momenta equations in (1) and (2) we obtain the explicit form of the matrix coefficients in (10) given below. For the coefficients of the \( A_j \) matrix
\[ a^n_{k,11} = -\bar{\alpha}_{g,j} \bar{u}_{g} \left( u^2 \right)^{g}_{L-1} - a_{p2,L-1} \Delta t; \]
\[ a^n_{k,12} = -\bar{\alpha}_{g,j}^2 \bar{u}_{g} \left( u^2 \right)^{L-1}_{L-1} - a_{p4,L-1} \Delta t. \]

For the coefficients of the \( B_j \) matrix we have
\[ b^n_{k,11} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} - \left( a_{u} \right)^{n}_{L-1} - (a_{p1,L-1}) \Delta t; \]
\[ b^n_{k,12} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} - \left( a_{u} \right)^{L-1}_{L-1} - (a_{p4,L-1}) \Delta t. \]

For the coefficients of \( C_i \) matrices we have
\[ c^n_{k,11} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} + a_{p1,L} \Delta t; \]
\[ c^n_{k,12} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} + a_{p4,L} \Delta t; \]
\[ c^n_{k,21} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} + \left( a_{u} \right)^{L-1}_{L-1} \Delta t; \]
\[ c^n_{k,22} = \left( \bar{\alpha}_{g,j} \bar{u}_{g} \right)^{L}_{L} + \left( a_{u} \right)^{L-1}_{L-1} \Delta t. \]

The free vector has the form
\[ \frac{n_{L}}{\Delta t} = \rho_{m,j} \left( \frac{(\bar{\alpha}_{g,j})}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j; \]
\[ \frac{n_{L}}{\Delta t} = \rho_{m,j} \left( \frac{(\bar{\alpha}_{g,j})}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j; \]
\[ \frac{n_{L}}{\Delta t} = \rho_{m,j} \left( \frac{(\bar{\alpha}_{g,j})}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j; \]
\[ \frac{n_{L}}{\Delta t} = \rho_{m,j} \left( \frac{(\bar{\alpha}_{g,j})}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j - \left( \frac{\bar{\alpha}_{g,j}}{\bar{\alpha}_{g,j}} \right) \left( \frac{u^2}{L-1} \right) \Delta x_j; \]

476
APPENDIX C

The unexpanded equations of the energy and mass conservation have the same structure (11). We have for gas density

\[ dU^{n+1}_L = d \left( (\alpha \rho)^{n+1}_g \right), \]

\[ S_g = \tilde{\Gamma}^{n+1}_g L \Delta t - (\alpha \rho)^{n+1}_{g,j} \gamma^{n+1}_{g,j}, \]

for liquid density

\[ dU^{n+1}_L = d \left( (\alpha \rho)^{n+1}_l \right), \]

\[ S_L = -\tilde{\Gamma}^{n+1}_l L \Delta t - (\alpha \rho)^{n+1}_{l,j} \gamma^{n+1}_{l,j}, \]

for gas energy

\[ dU^{n+1}_L = d \left( (\alpha \rho e)^{n+1}_g \right), \]

\[ S_L = \left( (\alpha \rho e)^{n+1}_{g,j} + (\alpha \rho)^{n+1}_g p^{n+1}_L \right) \gamma^{n+1}_{g,j} - \left( (\alpha \rho)^{n+1}_{g,j} p^{n+1}_L \right) \Delta t, \]

and for the liquid energy

\[ dU^{n+1}_L = d \left( (\alpha \rho e)^{n+1}_l \right), \]

\[ S_L = \left( (\alpha \rho e)^{n+1}_{l,j} + (\alpha \rho)^{n+1}_l p^{n+1}_L \right) \gamma^{n+1}_{l,j} - \left( (\alpha \rho)^{n+1}_{l,j} p^{n+1}_L \right) \Delta t. \]

Coefficients \( \lambda_j^{(n+1)} \) and \( \mu_j^{(n+1)} \) given by the expressions for upwind values for densities

\[ \tilde{\lambda}_j^{n+1} = 1 + (s_{u,j} + z_{u,j} \cdot s_{p,j}), \]

\[ \tilde{\mu}_j^{n+1} = 1 - (s_{u,j} + z_{u,j} \cdot s_{p,j}), \]

and energies

\[ \lambda_j^{n+1} = \tilde{\lambda}_j^{n+1} - z_{u,j} z_{p,j} \frac{p_{g(l),L-1} - p_{g(l),L}}{p_{g(l),L-1} + p_{g(l),L}}, \]

\[ \mu_j^{n+1} = \tilde{\mu}_j^{n+1} + z_{u,j} z_{p,j} \frac{p_{g(l),L-1} - p_{g(l),L}}{p_{g(l),L-1} + p_{g(l),L}}. \]
Modeling and Prediction of Criminal Activity Based on Spatio-Temporal Probabilistic Risk Functions

Paulina Flores, Mario Vergara, Pablo Fuentes, Francisco Jaramillo, David Acuña, Aramis Perez, Marcos Orchard

1 Universidad de Chile, Department of Electrical Engineering, Av. Tupper 2007, Santiago, Chile
pflores@ing.uchile.cl
mario.vergara.l@ing.uchile.cl
pablo.fuentes@ing.uchile.cl
francisco.jaramillo@ing.uchile.cl
davacuna@ing.uchile.cl
aramis.perez@ing.uchile.cl
morchard@ing.uchile.cl

ABSTRACT

Security forces need to model risk patterns associated with criminal activity to study cause-effect relationships and predict new crimes. In this regard, criminal risk models are important to obtain relevant information for better resource allocation and prevention of future crime activity. This paper proposes a method to model and predict future criminal activity based on spatial probabilistic risk functions and a characterization of their temporal evolution as new data become available. This method uses geo-referenced information of public services (e.g., shopping centers, banks) and criminal incidents to approximate the prior risk function as a Gaussian Mixture Model (GMM). Temporal evolution of crime activity is characterized using an algorithm that is based on Sequential Monte Carlo Methods and Importance Sampling. This algorithm incorporates information from new measurements, in a recursive manner, to approximate the posterior spatial probabilistic risk function by updating particle positions in the map. Finally, we propose a novel prediction scheme for criminal activity that uses Gaussian fields centered on hypothetical future criminal events, which are sampled from a GMM that characterizes the spatial distribution associated with recent crime activity. The optimum number of centroids for each Gaussian kernel is evaluated using Silhouette algorithm. The time index related to each hypothetical future crime event is probabilistically characterized using an exponential distribution. Results using real data show that the majority of future events occur within risk modeled zones, information which can be used for resource allocation and improvement of intervention plans.

1. INTRODUCTION

Around the world, order and security forces focus on monitoring criminal incidents that occur in a determined area and time period. Location technologies and services have made it possible to include geo-referenced information, used by analysts to find spatio-temporal patterns in reported events, so as to predict when and where a new crime will occur.

Many techniques and models have been developed to achieve this objective. Among the most used we can find Hot-Spots theories (Eck, Chainey, Cameron, & Wilson, 2005); where criminal incidents are located on a plane, forming clusters that are assumed to be invariant for any prediction horizon. Unfortunately, this technique fails to reflect changes in crime patterns as the environment changes. So as to avoid this problem, more sophisticated statistical models have been developed. For instance, in Liu & Brown (2003) a point-pattern-based transition density model is implemented, which depends on geographic and demographic information. In Xue & Brown (2006) and in Smith & Brown (2007), a model based on spatial decisions has been developed: criminals are assumed to choose places that can be modeled in terms of profit maximization, which depends on geographic and demographic information. In Xue & Brown (2006) and in Smith & Brown (2007), a model based on spatial decisions has been developed: criminals are assumed to choose places that can be modeled in terms of profit maximization, which depends simultaneously on the gain in committing the crime and the likelihood of being arrested. Other model can be reviewed at Brown, Dalton, & Hoyle (2004) and Rodrigues, Diggle, & Assuncao (2010). The disadvantage of such models is that they do not directly incorporate the temporal component, and when it is modeled using time series (for example), space-time interactions are not considered (Ivaha, Al-Madfai, Higgs, & Ware, 2007). Recent studies have developed some approaches that apply generalized additive models (Generalized Additive Models, GAM) to combine spatial and temporal data, as well as diverse characteristics, for prediction (Wang & Brown, 2012a).
This paper proposes a mixed approach, which considers Gaussian fields and spatial probabilistic risk models based on information of public services associated with an area of interest and criminal event data. The concept of posterior probability distribution is used to yield a spatial notion of crime risk. The time component is incorporated via exponential models and the sequential inclusion of information from new crimes committed within the same area, updating the prior risk distribution. This scheme provides an estimate for the probability of occurrence for a crime in a certain area and time, conditional to nearby public services and events that have occurred in the past.

The article is structured as follows. Section 2 presents a theoretical background focused on concepts such as Gaussian Mixture Models, Important Sampling, Resampling, and Model Evaluation Methods. Section 3 presents the proposed methodology for modeling, temporal characterization, and prediction of criminal events. Section 4 focuses on the analysis of generated results. Finally, conclusions and future work are presented in Section 5.

2. Theoretical Background

This section presents an overview of the main concepts that will help to understand our proposal for spatio-temporal models, and the associated solution for crime prediction. These concepts include Gaussian Mixture Models, Bayesian Inference and Monte Carlo integration methods such as Particle Filtering, Importance Sampling and Resampling algorithms. Finally, a brief discussion on ad-hoc performance measures is included.

2.1. Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) are parametric probability density functions widely used in the literature due to their capability for approximating multimodal probability density distributions. They are defined as a weighted sum of single Gaussian (Yu, Sapiro, & Mallat, 2011) distributions as stated in Eq. (1):

\[ G(x) = \sum_{i=1}^{M} \alpha_i \cdot f_i(x), \]  

(1)

where \( x \) is a D-dimensional random vector, \( \alpha_i \) are the mixture weights satisfying \( \sum_{i=1}^{M} \alpha_i = 1 \), and \( f_i(x) \) are multivariate Gaussian distributions of dimension D, given by:

\[ f_i(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^1/2} \exp \left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right). \]  

(2)

where \( \mu_i \) and \( \Sigma_i \) are the mean vector and the covariance matrix of the \( i \)-th Gaussian of the mixture.

As mentioned before, GMMs are parameterized by their mixture weights, mean vectors, and covariance matrices. The usual calculation of these parameters, conditional to a data set, is done by applying the Expectation Maximization (EM) algorithm, which is an iterative method that provides maximum likelihood or Maximum a Posteriori (MAP) estimates of these parameters.

There are three types of GMMs, depending on the choice of covariance matrices. In the first case, there is one covariance matrix per Gaussian component (nodal covariance). In the second case, the covariance matrix is the same for all Gaussian components (grand covariance). The third form is a single covariance matrix shared by the overall Gaussian model (global covariance) (Reynolds, 1995). In this paper we use two of these forms, the nodal model covariance and global covariance.

2.2. Bayesian Inference and Monte Carlo Integration

A common problem to be solved when dealing with uncertain dynamical systems is the manner in which statistical inference can be performed. In this regard, Bayesian approaches are well suited, as they provide a general framework for estimating hidden dynamical variables of a system through sequential update. These approaches involve two stages that are executed sequentially after a new measurement is available. The first stage consist of computing a prior probability density function (PDF); task that requires a stochastic representation for state transitions in the dynamic system. The second stage incorporates new measurements into the analysis by correcting the prior PDF through their likelihood, thus yielding a posterior PDF.

Mathematically, the evolution in time of the state sequence is considered as the set of \( N_x \)-dimensional vectors \( x_{0:k} = \{x_i, \ i = 0, ..., k\} \), and the observable events (or measurements) as the set of \( N_y \)-dimensional vectors \( y_{1:k} = \{y_i, \ i = 1, ..., k\} \). In Bayesian theory, the posterior distribution defined by \( p(x_{0:k}|y_{1:k}) \) is decomposed in terms of the prior distribution \( p(x_{0:k}) \), its likelihood \( p(y_{1:k}|x_{0:k}) \), and the evidence \( p(y_{1:k}) \) as:

\[ p(x_{0:k}|y_{1:k}) = \frac{p(y_{1:k}|x_{0:k}) \cdot p(x_{0:k})}{p(y_{1:k})}. \]  

(3)

Under Markovian assumptions, the posterior distribution in Eq. (3) can be expressed recursively (S. a. Doucet, 2001) as:

\[ p(x_k|y_{1:k}) = \frac{p(y_k|x_k) \cdot p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \]  

(4)

The prior distribution is then decomposed as stated in the Chapman-Kolmogorov equation:
\[ p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) \cdot p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (5) \]

The recurrence relation established by Eqs. (4)-(5) defines the optimal Bayesian solution for the filtering problem. However, this relation cannot be determined analytically in general, and a close-form solution can only be found in a restrictive set of cases (e.g., the well-known Kalman filter for linear, and Gaussian systems). In these cases, assuming a Gaussian distribution with unbiased, and consistent, estimates for the mean and covariance matrix of the prior PDF, the filter can then optimally derive the mean and covariance matrix of the posterior PDF. In nonlinear systems, though, the posterior PDF is not necessarily Gaussian (Arulampalam, Maskell, Gordon, & Clapp, 2002). In this regard, we consider in this article a class of sub-optimal nonlinear Bayesian algorithms that allow better characterization of the posterior PDF in dynamic, non-Gaussian systems: Particle Filters (PF). In PF, the key idea is to represent the posterior density function by a finite set of weighted random samples \( \{x_i, w_i\}_{i=1}^{N_s} \) in order to perform statistical inference.

### 2.3. Importance Sampling

The estimation of the posterior distribution requires to determine the prior distribution, the evidence, and its likelihood. As the likelihood is usually known, and the evidence corresponds to a normalizing constant, the most difficult task correspondsto the computation of the prior PDF \( p(x_k|y_{1:k-1}) \) described in Eq. (5). However, that expression includes an integral for probability densities that do not have a closed-form in general.

As the Eq. (5) involves the computation of an intractable integral, the idea of a sample-based approximation seems to be suitable. Nevertheless, it is usually hard to sample from the distribution \( p(x_k|y_{1:k}) \) at any time \( k \). Importance Sampling (IS) (Bergman, 1999) solves this problem by sampling for an alternative PDF, which is known in the literature as importance distribution and is denoted by \( q(x_{0:k}|y_{1:k}) \). The support of this distribution must, at least, include the support of \( p(x_{0:k}|y_{1:k}) \). Moreover, as the samples are drawn from an alternative distribution a weight is associated to them, and thus

\[ p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^{N_s} w_i^k \delta(x_{0:k} - x_i^k). \quad (6) \]

The challenge is to compute weights adequately. It is assumed that \( p(x_{0:k}|y_{1:k}) \propto \pi(x_{0:k}|y_{1:k}) \) is difficult to sample, but \( \pi(x_{0:k}|y_{1:k}) \) can be evaluated. Let \( x^i \sim q(x_{0:k}|y_{1:k}), \ i = 1, ..., N_s \) be samples that are easily generated from the importance density \( q(\cdot) \) (Arulampalam et al., 2002), then:

\[ w_i^k = \frac{p(x_i^k|y_{1:k})}{q(x_i^k|y_{1:k})} \propto \frac{\pi(x_i^k|y_{1:k})}{q(x_i^k|y_{1:k})}. \quad (7) \]

The art of IS is about choosing the importance distribution \( q(\cdot) \) which approximates \( p(\cdot) \) as closely as possible. This is the principal factor that affects the performance of this approach (Candy, 2007). Furthermore, if this condition does not hold, a degeneracy phenomenon appears yielding sample impoverishment and thus, undesirable inefficiencies in the method.

#### 2.4. Resampling

A common problem with IS is the degeneracy phenomenon, where after a few iterations all but one particle have negligible weight. In (A. Doucet, Godsill, & Andrieu, 2000) it has been shown that the degeneracy phenomenon is impossible to avoid.

Resampling is a method for addressing the effects of the degeneracy phenomenon in order to reduce them. The basic idea is to eliminate particles with low weights and concentrate on particles with higher weights. The algorithm (Arulampalam et al., 2002) consists of drawing a new set of \( N_s \) particles \( \{x^*_k\}_{i=1}^{N_s} \) by resampling (with replacement) from:

\[ p(x_k|y_{1:k}) = \sum_{i=1}^{N_s} \omega^*_k \delta(x_k - x^*_k), \quad (8) \]

so that \( P[x^*_k = x^i_k] = w^i_k \). The new samples are i.i.d, so they are equally weighted and \( \omega^*_k = \frac{1}{N_s} \). The method is described in Algorithm (1).

#### 2.5. Exponential distribution

Recently, an analysis over real-crime data has demonstrated that the frequency of crime (robbery, thefts, or burglaries) events over a geographic area, and considering a fixed time interval, follows an exponential distribution (Furtado, Melo, Coelho, Menezes, & Belchior, 2008).

Thus, assuming that a criminal event is equally likely to occur in a small time interval (no matter how far or near the last criminal event occurred), then it is possible to model the criminal events rate by an exponential distribution (Blumstein, Cohen, Roth, & Visher, 1986), as it is the only continuous distribution that possesses this memory-less property.

The exponential distribution can be parametrized by its rate \( 1/\beta \) and is given by:

\[ h_\beta(x) = \begin{cases} \frac{1}{\beta} \cdot \exp\left(-\frac{1}{\beta} x\right) & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (9) \]
ANNUAL CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2015

Algorithm 1: Resampling Algorithm

Input: A set of particles with degeneracy phenomenon 
\[ \left( x_i^*, w_i^*, v_i^* \right)_{i=1}^{N_s} \]

Output: A new set of particles with \( w_k \) weight 
\[ \left( x_k^*, w_k^*, v_k^* \right)_{k=1}^{N_s} \]

1. Initialize the CDF: \( c_1 = 0 \);
2. for \( i=2:N_s \) do
3. Constructed CDF: \( c_i = c_{i-1} + w_i^* \);
4. end
5. Start at the bottom of CDF: \( i = 1 \);
6. Draw a starting point: \( u_i \sim \mathcal{U}\left[0, N_s^{-1}\right] \);
7. for \( j=1:N_s \) do
8. Move along the CDF: \( u_j = u_1 + N_s^{-1}(j-1) \);
9. while \( u_j > c_i \) do
10. \( j^* = i + 1 \);
11. end
12. Assign sample: \( x_k^{*j*} = x_i^* \);
13. Assign weight: \( w_k^* = N_s^{-1} \);
14. Assign parent: \( i^j = i \);
15. end

where the maximum likelihood estimator of \( \beta \) is:
\[ \hat{\beta} = \frac{1}{n} \sum_{i=1}^{n} x_i \] (10)

and \( x_i \) are i.i.d. samples from \( h_\beta(x) \). In this paper, the probabilistic characterization of criminal incidents rates related to a determined area is used for prediction purposes.

2.6. Model evaluation

To evaluate the performance of the proposed risk model, this research effort compares high-probability areas predicted by the model with the number of crimes that actually occur in those areas. A model is said to be good as long as the number of incidents that occur within a fixed time period are proportional to the predicted for that area. The characterization of risk model performance at a time \( t_j \) is given by the curve that relates the High-Risk Percentage (HRP\( \theta \)) vs. True Incident Percentage (TIP\( \theta \)), a method proposed by (Wang & Brown, 2012b) in which:

\[ HRP_{\theta} = \frac{\left\|\left\{a_{i, \theta} | \mathbb{P}(inc_{a_{i, \theta}, t_j} = 1) > \theta\right\} \right\|}{\left\|\left\{a_{i}\right\}\right\|} \] (11)

\[ TIP_{\theta} = \frac{\left\|\left\{inc_{a_{i, \theta}, t_j} = 1 | a_{i, \theta} \subset \left\{a_{i} | \mathbb{P}(inc_{a_{i}, t_j} = 1) > \theta\right\} \right\|}{\left\|\left\{inc_{a_{i, \theta}, t_j} = 1\right\|\right\|} \] (12)

where \( \left\| \cdot \right\| \) is the cardinality of a set, \( \theta \in [0, 1] \) is a threshold and \( \mathbb{P}(inc_{a_{i}, t_j} = 1) \) is the probability that criminal incidents happen in a area subdivision \( a_i \) and a time window \( t_j \). In this case, HRP represents the percentage of high-risk areas predicted by the model, whereas TIP represents the incidents from a test set that took place within the high-risk areas. Both measures are computed for different \( \theta \) and plotted against each other obtaining a graphic similar to the operating characteristic curve (ROC) (Fawcett, 2006). If many crimes incident take place in high-risk areas, a curve closer to the upper left corner is expected. In the opposite case, a curve similar to a linear relationship is expected.

To measure the model quality, we use the concept of Area Under the Curve (AUC). This area takes values between 0.5 and 1, corresponding to the worst and the best possible cases, respectively.

3. PROPOSED METHODOLOGY

The proposed methodology provides a probabilistic characterization of criminal activity, using for this purpose a set of samples that are distributed in the space (geographic area) accordingly to a risk function. Furthermore, it also includes a mechanism to model the temporal evolution of the sample spatial distribution, where samples are reallocated sequentially as soon as the notification of new criminal incidents is available. Also, a prediction strategy is presented to evaluate the risk level within a specific area and future time period.

3.1. Required Information

For the generation of probabilistic risk models associated with criminal incidents, it is necessary to analyze three types of information sources: (i) definition of a geographic area of interest, (ii) geo-referenced data of public services (e.g., banks, supermarkets), and (iii) geo-referenced data of criminal events. These items are described below.

3.1.1. Definition of Area of Interest (A)

The definition of the area \( A \subset \mathbb{R}^2 \), for which the probabilistic model is needed, depends on various factors such as: main motivation of the study, processing capacity of the machine, available information, among others.

3.1.2. Geo-referenced Data of services

Within \( A \), it is possible find many services that define places where people naturally gather (e.g., hospitals, schools, parks, supermarkets, pharmacies, among others). As soon as the area \( A \) is defined, it is necessary to note the amount of services that are included on it. When the area under study is large and it contains few services (located in a sparse manner), then it would be more difficult to implement the concept of Hot-Spot for criminal risk. However, if the area is very small and it contains several services, then the risk function could be characterized almost as a constant.
3.1.3. Geo-referenced Data of Criminal Events (d_j)

Criminal events are understood as any type of crime that occurred within a time interval \( T \subseteq \mathbb{R}^+ \). A probabilistic model requires a considerable amount of data to have statistical validity. For the case in which this paper is framed; this “considerable amount” will depend on the area of interest. The type of crimes for which the model will be generated may vary: they can incorporate many types of crime events (when requiring a general risk model for a particular area), or they can be focused on a special set of crime events (high social impact crimes, such as homicide, burglary, robbery, among others).

3.2. Generation of Spatial Risk Models

For the generation of a probabilistic model that achieves the characterization of the risk associated with criminal activity, it is of paramount importance to relate the current services to the occurrence of criminal events.

This part of the methodology is divided into a sequence of steps, which are listed below and summarized as a flowchart in Figure 1.

3.2.1. Definition of Area of Interest

Area \( A \subseteq \mathbb{R}^2 \) where services and criminal events that are representative for the case study are defined (Figure 2.A).

3.2.2. Positioning of the Events and Services in the Area

The information of the set of criminal events denoted by \( \{d_j\}_{j=1}^{D} \), and the set of services \( \{s_j\}_{j=1}^{M} \) that exist in \( A \) during the time interval \( T \) (Figure 2.B), is provided in terms of spatial positioning via Global Positioning System (GPS) coordinates, preferably.

3.2.3. Definition of Service Risk Influence Range

Let \( r_i \subseteq \mathbb{R}^+ \) be the risk influence range associated with a service \( s_i \) such that \( r_i \) defines the radius of a circle centered on the corresponding service coordinates. They were considered identical for all services (Figure 2.C) in this particular study.

3.2.4. Search for Events Associated with Each Service \( s_i \)

Let the location of a crime \( d_j \) be given by the coordinates \( \vec{x}_d = (d_{jx}, d_{jy}) \in \mathbb{R}^2 \), and let the location of a service \( s_i \) be given by the coordinates \( \vec{x}_s = (s_{ix}, s_{iy}) \in \mathbb{R}^2 \). The relationship between \( d_j \) and \( s_i \) will depend on

\[
\text{dist}_{ij} = \sqrt{(d_{jx} - s_{ix})^2 + (d_{jy} - s_{iy})^2},
\]

where the crime \( d_j \) is said to be “associated” with a service \( s_i \) if it is fulfilled that \( \text{dist}_{ij} \leq r_i \). Repeating this procedure for all crimes \( \{d_j\}_{j=1}^{D} \), a new set of events that are associated with a particular service will be obtained, defined as \( \mathcal{D}_i = \{\vec{x}_d | \text{dist}_{ij} \leq r_i \} \).

3.2.5. Calculation of the Associated Risk

Once the sets \( \{\mathcal{D}_i\}_{i=1}^{M} \) are defined, then the elements of \( \mathcal{D}_i \) are used to adjust the parameters of the \( i \)-th component of a GMM that is described in Eq. (2). In fact, if \( ||\mathcal{D}_i|| \) denotes the number of elements in the set \( \mathcal{D}_i \), then

\[
\mu_i = \frac{1}{||\mathcal{D}_i||} \sum_{x_j \in \mathcal{D}_i} x_j,
\]

\[
\Sigma_i = \frac{1}{||\mathcal{D}_i||} \sum_{j=1}^{||\mathcal{D}_i||} (x_j - \mu_i)(x_j - \mu_i)^T,
\]

where \( x_j \in \mathcal{D}_i, \ j = 1, ..., ||\mathcal{D}_i|| \). Therefore, the risk associated to the \( i \)-th service is assumed to distribute following a Gaussian probability density determined by the aforementioned parameters.

3.2.6. GMM Model Generation

As each service has its own risk probability density, the risk function that covers the whole area of services will be de-
Figure 2. A) Event localization scheme within the area of interest; B) Events and services; C) Events, services and ranges defined for each service.

A GMM, as stated in Eq. (1). Hence,

\[ PDF_{\text{prior}}(\vec{x}) = \sum_{i=1}^{M} \alpha_i \cdot f_i(\vec{x}), \] (16)

where

\[ \alpha_i = \frac{1}{M} \Rightarrow \sum_{i=1}^{M} \alpha_i = 1. \] (17)

Finally, a GMM of equally weighted components is obtained for describing a prior PDF of the criminal risk of a particular area of interest.

Figure 3. Representation through level curves of a GMM that characterizes the criminal risk.

3.3. Characterization of Temporal Evolution of Spatial Risk Distribution

To allow adaptation (update) of the risk spatial distribution, we propose to represent this distribution by a set of samples. These samples can then be reallocated in space accordingly to new notifications of criminal activity, procedure which can be understood as the computation of a posterior distribution.

Additionally, in the absence of new events, the samples may also change their position in order to incorporate underlying uncertainty sources, property that requires a risk prediction model capable of propagate uncertainty along a period of time.

This part of the methodology is divided in two main stages: off-line and on-line, as it is depicted in Figure 4.

During the off-line stage, a set of samples (particles) are arbitrarily located at certain positions and weighted, following the Importance Sampling methodology. Then, a resampling procedure is carried out to obtain a new set of equally weighted samples.

During the on-line stage, a temporary evolution strategy defines a dynamic equation for the movement of the particles according to the inclusion of new observations (sequential incorporation of new criminal events). Therefore, every time that a new criminal notification arrives, some particles will be attracted to the area where the event was reported. Once the positions of the particles have been updated, a posterior distribution is obtained by fitting a GMM (each particle is associated with a Gaussian bi-variate probability distribution).

Finally, a different strategy is required to generate predictions for the evolution of the posterior distribution. A strategy for prediction is presented here which employs the same methodology for reallocating particles used when criminal events were registered. The additional feature of this strategy is that it simulates future criminal activity via sampling procedures, using a GMM denoted \( GMM_{\text{pred}} \). This GMM is fitted using only recent criminal activity. The maximum number of step-ahead predictions is defined equal to the number of registered crimes events used for fitting \( GMM_{\text{pred}} \). Besides this, the time step between two simulated crimes is obtained by sampling from an exponential distribution, as described in Eq. (9), whose parameter \( \beta \) is set to the average time of data registers that were used to fit \( GMM_{\text{pred}} \).

3.3.1. Spatial Risk Distribution

The spatial risk distribution (prior distribution) defined and calculated in Section 3.2 is used (Figure 5.A).

3.3.2. Important Sampling

In order to have a better way to manipulate the spatial risk distribution with lesser computational cost, Important Sampling is used. Then, the prior distribution is empirically approximated using a set of sample (particles).

3.3.3. Resampling

After applying important sampling, it is necessary to ensure the existence of more particles in high risk probability areas, as well as less particles in low-risk probability areas.
Moreover, it is highly desired every particle to have the same weight. Due to this, Resampling (Section 2.4) must be applied (Figure 5.B).

3.3.4. New Observation (Crime Event)

Geo-referenced data of new crime events is chronologically stored and used by the Temporal Evolution module.

3.3.5. Temporal Evolution (Posterior PDF)

Temporal evolution of the posterior risk function is based on the movement of the particles as new observations keep arriving. Thus, it becomes necessary to define a time-varying model to represent the manner in which the particles will move.

The model must meet three requirements: i) Particles located far away from the criminal event should not be significantly affected, ii) Particles located nearby the criminal event should maintain their proximity to it, iii) Particles located at reasonable distance from the criminal event should move towards the observation, since the number of particles located in a determined area is an indicator of the associated risk (Figure 5.C). Following these guidelines, the transition model is defined as:

\[ x(k) = x(k - 1) + f(d) \{ y(k) - x(k - 1) \} + w(k), \]  

(18)

where:

- \( x(k) \) : Particle position at \( k^{th} \) time instant.
- \( x(k - 1) \) : Particle position at the previous time instant.
- \( y(k) \) : Observation (crime event) at \( k^{th} \) time instant.
- \( w(k) \) : Process noise.
- \( f(d) \) : Non-linear function which depends of the distance \( d \) between the observation and the particle.

The function \( f(d) \) is defined as:

\[
g(d) = \frac{1}{2\pi |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (\vec{x} - \vec{y}) \Sigma^{-1} (\vec{x} - \vec{y}) \right)
\]

(19)

\[
f(d) = \frac{g(d)}{\max(g(d))}
\]

(20)

The covariance matrix \( \Sigma \) is a design parameter and depends of the area of interest.

3.3.6. Posterior PDF

Once the temporary evolution is implemented using a set of recent criminal activity records, a new GMM can be approximated to obtain a criminal risk spatial distribution. Therefore, each particle becomes to the centroid of a Gaussian bi-variate distribution, and the variance corresponds a design parameter.

3.3.7. Prediction Module

The prediction strategy assumes that there should not be major changes on the spatial position of the criminal activity in the short term. This is, we consider that the crimes are distributed according to a stationary probability density. Hence, the historical data of recent criminal events is used to update the
posterior risk PDF, as shown in Figure 6.A in color blue. The procedure then uses these events to generate a GMM representing the risk PDF associated with recent criminal activity, denoted by $GMM_{pred}$, as shown in Figure 6.B.

Now, to propagate uncertainty throughout time, future crime events are simulated by sequentially drawing samples from $GMM_{pred}$, which are coloured in black in Figure 6.C. Future temporal evolution is then characterized by the movement of particles when driven by these simulated events.

To consider the temporary component associated to the prediction (prediction time step, prediction horizon) the time between observations is modeled as an exponential random variable, whose rate estimated as the inverse of the average time registered between criminal felonies observations that were used to fit $GMM_{pred}$.

3.3.8. Predicted Risk PDF

Once the prediction stage is finished, a Gaussian kernel is centered at each particle, in the same manner as when characterizing the posterior risk PDF, and a GMM is approximated to obtain a criminal risk spatial distribution.

4. RESULTS

The proposed methodology is applied to an interesting case of study, which considers actual records of criminal activity over a populated urban area. The database includes:

- Geo-referenced information on 4262 public services. These services mainly relate to: stores, banks, bars, fire stations, liquor stores, automatic teller machines (ATMs), police stations, shopping centers, schools, parks, hospitals, clinics, among others. Each record is labeled and has its respective GPS coordinates (latitude and longitude).
- Geo-referenced information of criminal incidents that occurred within the area (23109 records).

For analysis purposes, only robbery offenses with force are considered in this study (crimes that are classified as events with "high social impact"). Among those, 1870 of 2240 records are considered part of the training set, and will be used to characterize the prior risk distribution. From the remaining events, 185 are used to test the filtering stage, and 185 are used to validate the proposed risk prediction approach.

4.1. Prior Spatial Risk Probabilistic Function

Geo-referenced information on 4262 public services is used to generate different prior distributions for the spatial risk probabilistic function, considering for this purpose three values for the service influence range: three, five, and seven blocks respectively; see Figure 7, Figure 8, and Figure 9. Using the AUC measure described in Section 2.6, results show better risk characterization when the influence range is set to three blocks.

4.2. Posterior Spatial Risk Probability Function

Once the prior spatial risk distribution is determined, 185 crime events are sequentially used to compute the posterior spatial risk probabilistic function. This procedure, described in Section 3.3.5, requires first to obtain samples from the prior distribution. Figure 10 shows the obtained results when applying an importance sampling and resampling to particles that are first allocated using an arbitrary grid over the area of interest. The grid is generated considering 23 and 16 subdivisions for the horizontal and vertical dimensions, respectively. As a result, after the resampling procedure, some coordinates in the grid contain more than one particle.
The characterization of the posterior spatial risk probabilistic function is based on the position of particles after incorporating the impact of crime events that are sequentially registered during the filtering stage. The particle movement is governed by the Gaussian attraction field described by Eq. (19) and Eq. (20), where the covariance matrix is given by:

$$\Sigma = \begin{bmatrix} 1.5 \cdot 10^{-5} & 0 \\ 0 & 1.5 \cdot 10^{-5} \end{bmatrix}$$ \hspace{1cm} (21)

During the update (or filtering stage), we identify those particles that are significantly affected by the appearance of criminal events. This is done by setting a threshold for the attraction force $f(d)$ equal to 0.6. In this case, 276 particles are in the aforementioned condition. Those are the particles that will modify their position during the prediction stage. This is done to minimize the computation effort associated with the prediction stage.

After sequentially incorporating 185 criminal events into the analysis, the posterior distribution is built by using a GMM, as explained in Section 3.3.6, every particle is used as the centroid of a Gaussian kernel (Figure 11). The predictive capability of the posterior spatial risk distribution is evaluated using HRP and TIP measures accordingly to Eq. (11) and Eq. (12), for different values of influence ranges and grid sizes; see Table 1.

<table>
<thead>
<tr>
<th>Influence Range</th>
<th>HRP</th>
<th>TIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.923</td>
<td>0.859</td>
</tr>
<tr>
<td>4</td>
<td>0.924</td>
<td>0.857</td>
</tr>
<tr>
<td>5</td>
<td>0.911</td>
<td>0.862</td>
</tr>
<tr>
<td>6</td>
<td>0.915</td>
<td>0.865</td>
</tr>
<tr>
<td>7</td>
<td>0.923</td>
<td>0.855</td>
</tr>
</tbody>
</table>

Table 1. AUC considering [3,7] for particle influence range and [1,10] for grid resolution in the posterior spatial risk function (in blocks).
4.3. Predicted Spatial Risk Probability Function

Once the posterior spatial risk distribution is obtained by using 185 criminal events to update the position of 368 particles (Figure 11), we proceed to generate the GMM Hot-Spot distribution related to recent criminal activity. After applying clustering analysis to recent criminal activity, and testing the number of clusters using the Silhouette algorithm, three clusters as found as the optimal choice for the centroids of the Hot-Spot GMM (Figure 12). This Hot-Spot GMM, associated with recent criminal activity, is used to simulate future criminal activity. This is done by sampling 185 events from the GMM (using a combination of Importance Sampling and Resampling algorithms); see Figure 12.

Figure 12. Hot-Spot GMM for recent criminal activity with 3 centroids, and 185 simulated events that are used for prediction purposes.

The inclusion of temporal information related to each simulated crime event is characterized by an exponential distribution with $\beta = 113.09$ [minutes] (Figure 13). This exponential distribution is sampled 185 times to assign time intervals between each simulated crime event (Figure 15).

Figure 13. (a) Histogram is constructed considering recent criminal activity (185 events). This information is used to fit an exponential distribution. (b) 185 samples are obtained from this exponential distribution to obtain time intervals related to simulated crime events.

As a result, the simulated 185 future crime events define a prediction window of 2 weeks approximately. These events are used to modify the position of particles, according to Gaussian attraction fields (Eq. (19)). As a result, the predicted spatial risk function is obtained (Figure 14).

Figure 14. Predicted Spatial Risk Function (GMM after 185 prediction steps).

To evaluate the performance of the predicted spatial risk function, HRP and TIP measures are calculated according to Eq. (11) and (12). In this regard, the AUC is computed assuming different influence ranges for particles (when building the GMM), diagonal covariance matrices, and different grid sizes (measured in blocks); see Table 2. Figure 15 shows the HRP vs. TIP curve with the best AUC=0.93.

Table 2. AUC considering [3,7] for particle influence range and [1,10] for grid resolution in the posterior spatial risk function (in blocks).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.932</td>
<td>0.913</td>
<td>0.906</td>
<td>0.906</td>
<td>0.902</td>
<td>0.895</td>
<td>0.882</td>
<td>0.893</td>
<td>0.884</td>
<td>0.874</td>
</tr>
<tr>
<td>4</td>
<td>0.932</td>
<td>0.911</td>
<td>0.904</td>
<td>0.903</td>
<td>0.902</td>
<td>0.893</td>
<td>0.882</td>
<td>0.893</td>
<td>0.885</td>
<td>0.876</td>
</tr>
<tr>
<td>5</td>
<td>0.924</td>
<td>0.922</td>
<td>0.914</td>
<td>0.914</td>
<td>0.907</td>
<td>0.904</td>
<td>0.893</td>
<td>0.895</td>
<td>0.893</td>
<td>0.886</td>
</tr>
<tr>
<td>6</td>
<td>0.916</td>
<td>0.916</td>
<td>0.908</td>
<td>0.907</td>
<td>0.901</td>
<td>0.898</td>
<td>0.887</td>
<td>0.894</td>
<td>0.895</td>
<td>0.889</td>
</tr>
<tr>
<td>7</td>
<td>0.919</td>
<td>0.918</td>
<td>0.909</td>
<td>0.908</td>
<td>0.904</td>
<td>0.901</td>
<td>0.890</td>
<td>0.893</td>
<td>0.893</td>
<td>0.885</td>
</tr>
</tbody>
</table>

5. DISCUSSION

One of the main objectives behind the implementation of sequential approaches for the characterization of the posterior risk distributions is to be able to understand changes in patterns of criminal activity. In this regard, the performance of the “filtering” stage strongly depends on two aspects: (i) sampling strategies, and (ii) the definition of the attraction field that will modify the position of particles as a function of the appearance of new criminal events.

Regarding the former aspect, it is first necessary to determine the impact of sampling strategies when trying to characterize the prior spatial risk function. Although it is possible to use importance sampling and define weights for particles whose location is determined by samples from an uniform distribution (over the area of interest), we propose instead to assign...
weights proportional to the prior risk function to particles that are allocated on an arbitrarily defined grid.

The first problem found when sampling from a bi-variate uniform distribution is to define the amount of particles. A small number of particles results in a scarce representativeness of criminal focuses related to the prior risk function. In our case study, this strategy misses the greatest criminal focus, the one located near the point [-33.52, -70.6] (see Figure 16). Opposite case happens when considering a large number of particles: risk is overestimated because some particles are located in areas where originally there is no criminal risk. Moreover, the processing time for the prediction stage is proportional to amount of sampled particles and thus, the algorithm losses computational efficiency (Figure 17).

The second problem is the confidence degree associated with samples whose location is generated via uniform sampling. Although the amount of particles may be high enough (avoiding the lack of particles in specific areas, but ensuring reasonable processing times), it is not guaranteed that particles will satisfactorily approximate the original prior distribution. It is observed that the implementation of importance sampling strategies (with 370 particles), where the position of particles is obtained from a bi-variate uniform distribution, leads to different prior risk functions performing particle resampling; see Figure 18.

In this regard, the proposed method, where the position of particles is defined by an arbitrary grid and where the weights are assigned proportional to the prior risk function, avoids the two problems described above if resampling is used as a method for ensuring adequate risk characterization. The use of a grid allows to explore the area in a more intuitive manner and, furthermore, resampling allows to efficiently represent criminal risk within the whole area of interest.

The second aspect to be considered in the analysis is related...
to the definition of the function \( f(d) \) given by Eq. (20), where the covariance matrix of \( g(d) \), given by Eq. (19), is a design parameter that defines the strength of the particle movement as a function of the distance between the particle and the new criminal event. Our proposal considers that the covariance matrix is diagonal. If the value of diagonal elements is four times greater than the magnitude of the process noise variance, then particles that are located far away from the observation approach abruptly. As a result, after just a few iterations, all particles would converge to Hot-Spot centroids thus losing the capability of uncertainty characterization and representativeness.

On the other hand, if values of diagonal elements in the covariance matrix are equal to the process noise variance, then particles tend stay unaffected by the appearance of new crime events. This two situations provide boundaries that need to be considered in the algorithm design. The matrix given by Eq. (21) provided good results in our case study. This matrix basically represented an influence range of eight blocks around the criminal event.

The implementation of these suggestions allowed to obtain a posterior spatial risk distribution that significantly improves the characterization of criminal activity over time. Using performance measures such as HRP y TIP, Eq. (11) and Eq. (12), it is possible to obtain AUC over 0.9 (Table 1) indicating that the model provides consistent information on the recent crimes that occurred within the area of interest.

In terms of the analysis of future criminal activity, the prediction stage plays a fundamental role. The prediction window (equivalent to the number of prediction steps) should be set accordingly to the number of crimes used for estimating the Hot-Spots distribution. In other words, if 185 crime events are used to compute the posterior risk function, then it is only safe to make predictions between 1 and 185 steps-ahead in time. Although it is possible to make long-term predictions using a risk function solely based on recent criminal activity (in this case, 185 events), these predictions would be biased. The latter, because information associated with Hot-Spot distributions would discard prior knowledge and would be (in that case) based on a much reduced spatio-temporal window.

Regarding the inclusion of the temporal variable in the prediction stage, it is difficult to establish regular time periods between crimes, since criminal activity occur at irregular time intervals. Thus, the temporal analysis shown in Figure 13 is justified. Additionally, to calculate the prediction time, it is only necessary to compute the sum of realizations from an exponential distribution: a method that is simple and computationally efficient. In this case considering 185 criminal incidents, the time between events was satisfactorily characterized using an exponential distribution with parameter \( \beta = 113.09 \) [minutes], resulting in a prediction window of approximately 2 weeks.

Analyzing the data provided in Table 2 it must be noted that, independently of the influence range associated with each particle, as the resolution becomes smaller the AUC of the prediction model improves. This result is intuitive since it implies that crime could be better predicted if every block is monitored independently. However, police resources are limited, and there is an optimum AUC subject to that constraint. However, there is an optimum influence range in terms of the model performance.

6. Conclusion

This article provides a method to characterize the evolution in time of criminal risk in a specific area. A case study with real data that includes location of public services and criminal incidents is also presented. The novel methodology for quantifying risk of criminal events uses a particle-based empirical representation. Two different stages are distinguished in this method: off-line and on-line. The former considers the location of services and 1870 crimes that occurred during a time period of 6 months, yielding a probabilistic characterization of the risk using prior knowledge of historical crimes in the area. The on-line stage approximates the posterior spatial risk distribution using a sequence of 185 new crimes. This task is done by sequentially updating the location of samples (particles), using concepts of importance sampling and resampling. In addition, a strategy for criminal risk prediction is presented. For this prediction strategy, a GMM is fitted using historical registers of recent criminal activity. This GMM is used to simulate 185 future crime events that help to explore the evolution in time of particle positions. Each of these simulations has an associated time of occurrence, modeled by an exponential distribution.

The sequential estimation of a posterior distribution expressed as the movement of samples in the space requires the adjustment of some parameters. One of them is the noise variance (see Eq. (18)), which accounts for the uncertainty in the movement of the samples (it determines how far can a particle move from its initial position on each iteration). For a more realistic reallocation, this hyperparameter should take into account several factors like rate of samples per area and time dependence, to name a few. In this study, each criminal event is considered equally important and thus, each one is given the same importance in terms of the effect on particle movement. Future work will consider different types of crimes in order to cover a wider and more realistic scenario.

For the implemented prediction stage, it is important to consider the temporal analysis of processed criminal incidents, because that allows to generate predictions at irregular time periods; constituting a major advantage over methods where prediction assumes equally spaced time intervals. It is important to note that in this case we were able to generate predictions for two weeks in advance, but it is also possible to
provide those results in terms of days, and even police shifts. Also, the prediction can be improved even more considering updates of the Hot-Spot distribution every fixed set of incoming criminal incidents. Finally, we should emphasize the importance of methodologies implemented since they become a good complement to the police, helping in managing its resources to cover the areas of high criminal risk.

ACKNOWLEDGMENT

This work has been partially supported by FONDECYT Chile Grant Nr. 1140774. The work of Pablo Fuentes has been supported by research grant CONICYT-PCHA / Magister Nacional / 2015-22150857. The work of Francisco Jaramillo has been supported by a Ph.D scholarship from CONICYT, Chile. CONICYT-PCHA/ Doctorado Nacional/ 2014-21140201. The work of Aramis Pérez was supported by the University of Costa Rica (Grant for Doctoral Studies) and CONICYT-PCHA/ Doctorado Nacional/2015-21150121.

REFERENCES


BIOGRAPHIES

B.Eng. Paulina V. Flores was born in Santiago, Chile. She received her degree in Electrical Engineering from Universidad de Chile in 2014. She participated in 1st International Summit on Scientific Criminal Analysis. She currently works in electrical power distribution market, in the design and implementation of fraud and meter malfunction detection models (Chilectra, ENEL Group). Her current interest include NTL Detection, Big Data Analysis and Business Intelligence.

B.Sc. Mario A. Vergara was born in Santiago, Chile in 1992. He received the B.Sc degree in Electrical Engineering at Universidad de Chile in 2014. Currently he is a M. Sc. student at the same institution under Dr. Marcos E. Orchard supervision. His research interests include Bayesian signal processing, non-linear filtering and elements of information theory.

B.Sc.Pablo F. Fuentes was born in Santiago, Chile in 1992. He is a student of Electric Engineering from the Faculty of Physical and Mathematical Sciences, Universidad de Chile. He is currently preparing his M. Sc. in fault diagnosis and
prognosis, receiving support from Conicyt under Dr. Marcos E. Orchard supervision. He is interested in investigating criminal risk characterization and its temporal evolution within the area of Santiago, Chile.

**B.Sc. Francisco Jaramillo** received the B.Sc. degree in Electronics Engineering from Universidad de La Frontera, Temuco, Chile, in 2009. Currently he is a doctorate student at the Department of Electrical Engineering at the University of Chile under Dr. Marcos E. Orchard supervision. His research interests include machine learning, control systems, and estimation and prognosis based on Bayesian algorithms with applications to nitrogen removal in pilot-scale Sequencing Batch Reactors for Wastewater Treatment Plants.

**B.Sc. David E. Acuña** was born in Santiago, Chile. He received his B.Sc. degree in Electrical Engineering (2014) from Universidad de Chile. He currently pursues a M.Sc. degree in Electrical Engineering at the same university and is a Research Assistant at the Lithium Innovation Center (Santiago, Chile). His research interests include signal processing, non-linear filtering, and prognostics and health management for energy storage devices.

**M. Sc. Aramis Pérez** is a Research Assistant at the Lithium Innovation Center (Santiago, Chile) and professor at the School of Electrical Engineering at the University of Costa Rica. He received his B.Sc. degree (2002) and Licentiate degree (2005) in Electrical Engineering from the University of Costa Rica. He received his M.Sc. degree in Business Administration with a General Management Major (2008) and also he is a M.Sc. candidate in Industrial Engineering from the same university. Currently he is a doctorate student at the Department of Electrical Engineering at the University of Chile under Dr. Marcos E. Orchard supervision. His research interests include parametric/non-parametric modeling, system identification, data analysis, machine learning and manufacturing processes.

**Dr. Marcos E. Orchard** is Associate Professor with the Department of Electrical Engineering at Universidad de Chile and was part of the Intelligent Control Systems Laboratory at The Georgia Institute of Technology. His current research interest is the design, implementation and testing of real-time frameworks for fault diagnosis and failure prognosis, with applications to battery management systems, mining industry, and finance. His fields of expertise include statistical process monitoring, parametric/non-parametric modeling, and system identification. His research work at the Georgia Institute of Technology was the foundation of novel real-time fault diagnosis and failure prognosis approaches based on particle filtering algorithms. He received his Ph.D. and M.S. degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He received his B.S. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile. Dr. Orchard has published more than 50 papers in his areas of expertise.
Active Mission Success Estimation through PHM-Informed Probabilistic Modelling

Adam R. Short1 and Douglas L. Van Bossuyt2

1,2Colorado School of Mines, Golden, Colorado, 80401, USA
ashort@mines.edu
dvanboss@mines.edu

Abstract

Prognostics and Health Management (PHM) techniques have traditionally been used to analyze electrical and mechanical systems, but similar techniques can be adapted for less mechatronically-focused processes such as crewed space missions. By applying failure analysis techniques taken from PHM, the probability of success for missions can be calculated. Extensive work has been conducted to predict space mission failure, but many existing methods do not take full advantage of modern computing power and the potential for real-time calculation of mission failure probabilities. The Active Mission Success Estimation (AMSE) method is developed in this paper to track and calculate the probability of failure as the mission progresses, and is intentionally adaptable for shifting mission objectives and parameters. This form of mission modelling takes a broader view of the mission and objectives, and develops statistical probability models of success or failure for multiple possible choice combinations that is used to inform real-time decisions and maximize probability of mission success. A case study of a generalized crewed Mars mission that has turned into a survival scenario is considered where an astronaut has been left behind on the surface and must survive for an extended period of time before undertaking a long-distance journey to a new launch site for rescue and return to Earth. The AMSE method presented here aims to establish real-time probabilistic modeling of decision outcomes during an active mission and can be used to inform mission decisions.

1. Introduction

Risk analysis and space exploration have a long and intertwined history of development. With initial modern rocket efforts of Goddard (Goddard 1920) and others, and the start of the Second World War, the space exploration era and risk analysis of complex systems both began. During the space race between the Soviet Union and the United States of America, tools such as Probabilistic Risk Assessment (PRA) (Kumamoto, Henley, and J 1996) were developed to closely examine complex system risk in a probabilistic and quantitative manner. Recently, a renewed focus on understanding risk during the early design phases of complex systems has received special attention (Douglas Lee Van Bossuyt 2015; Douglas L Van Bossuyt 2013; Douglas Van Bossuyt 2012; Van Bossuyt, Tumer, and Wall 2013). However, relatively little work has been done to develop real-time risk-informed decision support tools for missions that are actively occurring. Current risk modeling and analysis methods require significant adjustment and reanalysis when an unforeseen event occurs that can delay critical risk information during a rapidly developing scenario.

By changing the way the mission is modelled and analyzed to be more modular and actively recalculating risk as the mission progresses, the probability of success can be more accurately estimated and decision points with many multiple options can be analyzed to help inform mission command decisions to increase probability of mission success. This paper presents the Active Mission Success Estimation (AMSE) method that provides timely risk information to inform decisions being made in crisis situations with rapidly evolving circumstances.

Performing AMSE requires that all critical components of the mission be modelled thoroughly and modularly to enable rapid rearranging of model elements to evaluate potential decision outcomes to estimate mission success probabilities. To effectively represent the mission, a novel form of functional modelling was developed where environments nest around the system of interest and are used to determine what hazards are present in the environment that can damage the system. Mission tasks are then developed and analyzed by configuring tasks to represent both internal and external system risks including the effects of the nested functional modeling environment.

For the purpose of this paper, a case study is considered of a
single astronaut attempting to survive alone on Mars until rescue can arrive (Weir 2011), where the single astronaut is the system of interest.

1.1. Specific Contributions

This paper contributes the novel AMSE method for real-time assessment of risk during a mission using PHM techniques and functional modelling to provide decision-makers with immediate and up-to-date risk information during critical decision points. The AMSE method utilizes a novel form of nested functional modelling to analyze the effects of various layers of environmental protections. These protections could either protect the subject directly or be layered around each other. In the case study the effects of various environmental protections such as a permanent base, a passenger rover, or an Extravehicular Mobility Unit (EMU) were considered. AMSE provides quick and active estimation of current mission success, as well as projecting probable success based upon potential decision options. Through the active analysis of mission success probability during important decision points, the probability of success for the mission can be maximized. Additionally, the modular nature of AMSE allows for quick adaption to unexpected mission parameters. While AMSE was developed with space mission applications in mind, it could easily be adapted to analyze any complex system.

2. Background

AMSE relies upon several topics including PHM techniques, decision theory, and functional modeling. Traditional mission success estimation relies upon difficult to configure methods such as Probabilistic Risk Assessment (PRA) (Modarres, Kaminskiy, and Krivtsov 2011), (Mohaghegh, Kazemi, and Mosleh 2009) or Worst Case Analysis (WCA) (Ye 1997), (Nassif, Strojwas, and Director 1986). PRA, WCA, and related techniques are very successful in analyzing potential foreseeable failure scenarios but have difficulty in situations where rapid reconfiguration of the model is necessary in unanticipated rapidly changing situations, such as those faced by the astronaut in the hypothetical case study presented in this paper.

2.1. Space Mission Engineering

Space mission engineering is the process of establishing and refining mission parameters in order to reach broadly defined mission objectives (Wertz, Everett, and Puschell 2011a). Generally, space mission engineering aims to reduce time, cost, and risk associated with a space mission. One long-standing issue facing space mission engineering is the Space Spiral phenomenon where increasing cost of missions leads to longer mission schedules and reduced new mission frequency that leads to a higher demand for mission reliability that in turn leads back to a higher mission cost. This leads to ever-increasing expenses and timeline delays for many space missions. One way that the Space Spiral can be combatted is by reducing the amount of time and energy that goes into the space mission engineering process through development of new techniques that can reduce time and decrease risk, leading to lower costs.

2.2. Functional Modelling

Functional modelling describes a variety of techniques used to represent the function of a system, often including many sub-functions representing work done in the system on flows that represent energy, material, or information passing between and being transformed by functions and sub-functions. In addition to internal flows, input and output flows enter and exit the system boundaries. One popular form of functional modelling is Flow Block Diagrams or Functional Flow Diagrams (FFD) (Blanchard, Fabrycky, and Fabrycky 1990), (Böhm and Jacopini 1966). FFD is very good at modelling systems where there are direct linear flows between various functions and a clear system input and output exists. However, many existing methods of functional modelling suffer when the system is less linear, leading to tangled networks of flows and functions that are impractical to analyze or provide an inaccurate representation of the reality of a situation. Recently work has been done on the development and modelling of systems to model failure propagation through uncoupled systems (O’Halloran, Papakonstantinou, and Van Bossuyt 2015). Uncoupled failure propagation refers to systems where failure can be exported from one sub-system to another sub-system through three-dimensional space instead of being limited to propagation along system flows.

2.3. Space Mission Risk and Success Assessment

Space mission risk assessment can take many different forms, each having its own advantages and disadvantages, but generally risk assessment techniques tend to build on a foundation of probabilistic modelling. One method for risk estimation is the use of a hazard rate, $\lambda$, in an exponential distribution, Eq. (1), to calculate the expected survivability rate of a space mission (Wertz, Everett, and Puschell 2011b).

$$S(t) = e^{-\lambda t}$$

(1)

The expected survival can then be determined to find the expected failure rate function, Eq. (2).

$$F(t) = 1 - S(t) = 1 - e^{-\lambda t}$$

(2)

While the failure rate or a related function appears in many risk assessment methods, many additional complex techniques for evaluating risk of failure to a system exist. One example is the Failure Flow Identification and Propagation (FFIP) method (Kurtoglu, Tumer, and Jensen 2010; Kurtoglu and Tumer 2008) that uses a function block...
diagram (Stone and Wood 2000) structure and analyzes how failure flows through a system. This technique can be enhanced to enable mission control and autonomous decision making through the application of Failure Flow Decision Functions (FFDF) (Short and Van Bossuyt 2015c) that determine what the best a worst ways for a failure to propagate through a system are. Space mission risk assessment can also be applied to control of autonomous systems in order to maximize mission success while minimizing human work hours (Short and Van Bossuyt 2015a), (Short and Van Bossuyt 2015b). Many of the existing methods, while generally robust, have a lengthy setup analysis process. The lengthy and resource-intensive setup of existing methods makes active assessment of dynamic situations infeasible when relying on established methods.

2.4. Prognostics and Health Management

Prognostics and Health Management (PHM) is used to predict and prevent failures in mechatronic systems (Sheppard, Kaufman, and Wilmering 2014). Many methods exist for PHM analysis, each with their own strengths and weaknesses, making them more or less advantageous for particular applications (Hutcheson et al. 2015), (Balaban et al. 2013). The process of making decisions based on PHM information is referred to as Prognostic-Enabled Decision Making (PDM) (Sweet et al. 2014) and can be used to decide which act presents the optimal level of risk and reward within a system. This can be an incredibly useful tool in analysis with PHM because it can be used to calculate the potential damage that could be caused to a system by one component failing.

An essential element of PHM analysis is the development of mathamatic models of physical systems such as mobility, control systems, structures, or power as well as the hazards that face the systems. These models are a necessary piece of PHM because they offer a prediction of the results of taking an action on the physical state of the system. The application of PHM techniques could be further extended by considering their effects on the wellbeing on a person in the system. In this case, a person can be treated similarly to traditional hardware with equations estimating the probability of survival based on a variety of mission-specific factors.

2.5. Human Exploration of Mars

Recently there has been resurgence in interest in sending human explorers on a mission to Mars. This has taken the form of planned missions from leading organizations such as the National Aeronautics and Space Administration (NASA) (Daines 2015), increased interest in popular culture (Weir 2011), (Sneider 2015), and combinations of the two (“Mars One” 2015). While autonomous rovers and satellites have gathered a large amount of information on Mars, there are still many unknowns to discover and a large number of serious problems remain to be solved. One such problem is the lengthy flight time between Earth and Mars that not only presents psychological risks from extended isolation (Gushin et al. 1998), but also presents danger from the large amount of radiation exposure along the way (Hellweg and Baumstark-Khan 2007). Once the astronauts have arrived, radiation on the surface has shown to be present, but at less hazardous levels than some earlier estimates. However, there are still environmental risks from extreme weather (B. A. Cantor 2007), (B. Cantor, Malin, and Edgett 2002), potentially unexpected hazardous terrain (Lakdawalla 2015), health effects from reduced gravity (Horneck et al. 2003), (Marty et al. 2009), difficulty of communication with Earth due to signal delay, and various other hazards. The hazards are all compounded by external rescue or repair being exceptionally difficult in the case of an emergency.

3. METHODOLOGY

The AMSE method is based on an object-oriented form of functional modelling, risk assessment techniques derived from FFIP and related methods, and concepts from decision theory to evaluate a mission’s current state and potential for success. The core of AMSE is a Survival Rate function for the system of interest that examines a number of variables to determine the probability of the system’s survival. These variables can be greatly affected by the performance of tasks, which in this context refer to actions that affect resources and sub-systems health and often take up some amount of time. The definition of tasks in the context of AMSE will be expanded upon later in this section.

To have a structure to build the analysis upon, a modified form of FDF is developed and implemented where instead of functions being lined up and having flows pass from one to another, systems of interest are nested within each other, with each later augmenting flows passing through their barriers. The system can be visualized as a series of nested shapes, each representing a different sub-system/environment. At the center of the system is the critical system which represents a system (or systems) that are critical to mission success. For the case study presented in the next section, the critical subsystem is an astronaut stranded on Mars. The AMSE nested functional model is shown in Figure 1.
In addition to the sub-systems’ effect on the flows, tasks can also affect the flows and the state of external resources. Tasks are actions that can be taken to control the system in a way that is either desirable or undesirable. For the purpose of the case study discussed in the next section, many tasks are actions taken by the astronaut to increase their probability of survival and mitigate risk. Tasks can be combined in order to create a Task Plan for a length of time that can be used modularly to reduce the amount of work necessary to represent a series of actions. The Task Plans are then put into chronological order to create the Mission Plan, which represents all of the tasks that will be taken over the course of the mission including driving, eating, working, sleeping, and downtime. An example of the Mission Plan structure can be seen below.

- **Mission Plan**
  - Task Plan Week 1
    - Task A
    - Task B
    - Task C
  - Task Plan Week 2
    - Task A
    - Task D
  - Task Plan Week 3
    - Task C
    - Task D
    - Task A

An overview of setting up AMSE analysis is now presented:
1) A nested functional model of the mission system must be developed. 2) A mathematic model is developed to represent the flows and sub-systems from the visual functional model representation. 3) The critical sub-systems or flows must be identified. The critical sub-systems will be the primary focus of the mission analysis so models associated closely with them should be thoroughly developed to ensure accuracy. 4) The general mission plan should be define clearly. This includes a general mission objective, as well as any major secondary objectives, or necessary actions. 4) Task Modules should be developed that are necessary to complete all of the mission objectives, as well as account for downtime as necessary. 5) Task Modules are then used to construct the Task Plans which represent periods of time, such as a few hours, a day, a few weeks, years, or any other desirable amount of time. All necessary tasks must be included in the Task Plan and all time must be accounted for. The resolution of the Task Plan however can be either very detailed or very broad depending on the circumstances of the analysis. 6) Organize the Task Plans into a final Mission Plan. Finally analysis can be performed on the mission. Figure 2 graphically shows the AMSE method.

The AMSE method has been implemented into a MATLAB environment.

---

Figure 1. Simplified functional diagram of the system focusing on the astronaut (top). Simplified functional model of the Surface Exploration Vehicle (bottom).

The primary flows between the sub-systems are resources and energy. Some examples include food, heat, physical forces, or information. These flows are used to calculate the current and future states of the sub-systems. When passing through a sub-system, a flow can be affected by that sub-system to either be increased, reduced, or transformed before being passed on to the next layer of sub-system. Eventually the magnitude of the flows, health of sub-systems, and current time are used to calculate the health of the system.
While this list is not a comprehensive list of all of the dangers that face the astronaut, it does account for hazards identified to have a substantial effect on survival. At the beginning of the mission, the astronaut weighs 85 kg and is considered to be 180 cm tall for the purpose of modelling calorific intake needed and energy stored on their body.

Mars has temperatures ranging between -143 °C and 35 °C, an air pressure of 0.6 kPa (0.006 atm), surface radiation around 215 μGy/day, and an atmosphere that is approximately 96% carbon dioxide (Mahaffy et al. 2013). The astronaut must depend upon the available protection of existing equipment to avoid several forms of mission ending fatalities.

The Martian base protects from most radiation and provides breathable air at a comfortable temperature. However the base is immobile and at one point in the mission will need to be abandoned when the astronaut transits across Mars to an extraction point for rescue from the surface. The Martian base is powered by solar cells on its exterior surface.

The EVA suit has a suit port that acts as an airlock and allows for the astronaut to get in and out of the suit through the SEV which in turn docks to the Martian base. The suit can regulate temperature and offers approximately 8 hours of breathable air. While the suit protects to a small degree from radiation, it does not protect from large amounts of radiation which increases the health risk of using the EVA suit on missions. Additionally, it is hard to perform tasks in the suit thus causing calorific use while in the suit to increase.

The SEV is a 6-wheeled vehicle that can reach speeds of up to 25 km/h and can dock with the Martian Base (NASA.org 2015). The SEV has a rechargeable battery for power and can be charged from the Martian base’s power or solar cells directly. The two SEVs have parts that are interchangeable if necessary and can carry up to 1000 kg each. Driving the SEV burns approximately 170 kilocalories per hour.

At the Martian base there is enough food for 400 sols of 1500 calorie meals. Since 1500 calories does not allow for a very large amount of activity and rescue cannot arrive until sol 550, more food will have to be cultivated. Using the potatoes, the astronaut will grow additional food to extend food stores in an attempt to avoid starvation ending the mission. After initial setup time, the potato farm will produce 850 calories of food per day. However, the potatoes will have to be tended which also expenses calories.

While much sleep and rest will be necessary to conserve energy and reduce risk due to starvation, scientific work will still be performed by the astronaut over the course of the mission. The time spent in the Martian environment is incredibly valuable and while the astronaut is stranded there,

4. CASE STUDY

To demonstrate the effectiveness of AMSE for space mission assessment, a case study is presented of an astronaut in a solo survival scenario on Mars. The astronaut has been left behind by their crew due to unforeseen circumstances, and now must survive for 550 Martian days (Martian days or sols are approximated to be 24.6 hours) until a rescue vehicle can arrive. Additionally, over the last 45 sols of the mission, the astronaut must travel approximately 4200 km to reach the extraction point and be rescued. An ideal mission plan will be considered, as well as an adapted mission plan selected from several possible options after an unexpected event.

At the beginning of the mission, the astronaut has access to 1 Martian base, 2 surface exploration vehicles (SEV) (Bagdigan and Stambaugh 2015), 4 extra-vehicular activity suits (Boyle et al. 2012), 400 sols’ worth of meal rations, and several raw potatoes capable of growth.

The model of the astronaut takes into account radiation exposure, calorific intake and usage, time since sleep, physical injury, and exposure to extreme temperatures.
they should attempt to collect as much scientific information and perform as many experiments as possible. The physical intensity of this work will remain low to reduce energy expenditure, but two hours of work will be performed per day when possible.

An overview of the Mission Plan is shown below.

- **Sol 1-5**
  - Start Potato farm
- **Sol 6-14**
  - Assess current situation for survival
- **Sol 15**
  - Test drive rover
- **Sol 16 – 500**
  - Establish routine
    - Farm potatoes
    - Perform Experiments
  - Perform extended EVAs every eight weeks
    - Maintain equipment
    - Perform external experiments
  - Perform extended SEV missions every eight weeks
    - Test modifications to SEV
    - Prepare for long drive
  - Eat extra meals and rest once every four weeks
    - Replenish lost nutrients
    - Conserve energy
- **Sol 500-545**
  - Drive to extraction point in SEV
    - Drive 4 hours a day
    - Stop to recharge batteries using solar cells
- **Sol 546-550**
  - Prepare for rescue
  - Conserve energy when possible

In addition to the planned mission analysis, this case study is run on a scenario where half the food is lost at sol 350. AMSE is then used to determine how to properly re-ration food in order to achieve acceptable probabilities of survival which is defined as a mean instantaneous survival rate of 0.95 or higher. A survival rate of 0.95 is chosen for the minimum survival rate, because it is known to be achievable as the rate is below the control mission plan’s mean instantaneous survival rate, but is still high enough that many Mission Plans would fail to reach the rate if Mission Plans are not constructed carefully.

A loss of food is chosen as a focus for the case study because it presents a very direct risk to the system (astronaut) and there are multiple actions that can be taken to address the problem.

5. **RESULTS AND DISCUSSION**

AMSE is able to model the mission scenario described in Section 4 and provide an assessment of probability of success, as well as allow for rapid assessment of problems as they arise and allow for expedient information to be generated that can inform mission command decisions. General mission tasks are modelled for an astronaut attempting to survive alone on Mars while awaiting rescue. The astronaut is required to survive 550 sols and travel over 4200 km. The biggest risk that arises in the system is the risk due to starvation, because food supplies are very limited. The survival rates for the control mission are show below in Figure 3.

![Instantaneous Survival Rate vs Time](image_url)

**Figure 3.** Instantaneous Survival Rate vs Time shows the probability of survival at a particular moment (top). Probability of Surviving the Mission, shows the probability that the astronaut will survive the entire mission as related to time in the mission (bottom).

One notable feature of the plots in Figure 3 is that there are spikes in the probability of mission survival, especially around 225 sols where a local maximum is present. This is unexpected because it was assumed going into the analysis that the rate of survival would only go up over time. The cause of this phenomenon is dangerous driving and EVA tasks that momentarily increase the risk presented to the
astronaut, leading to the characteristic spikes in the second plot. There are similar spikes in the Instantaneous survival rate plot also occurring in line with EVA and SEV missions.

The real test of AMSE is in the food loss section of the case study analysis. In this case, half of the current food supplies are lost on sol 350. If no mitigating action is taken, the astronaut will likely begin to run out of food by sol 503 and will starve to death within seven to fourteen sols with near guaranteed death by starvation by sol 543. The survival rates for the potato loss scenario with no rationing are shown below in Figure 4. The Probability of Surviving the Mission vs Time is shown in the bottom plot of Figure 4. This plot represents the calculated rate of surviving the rest of the mission at a given time. If the astronaut is likely to survive the mission, the probability of survival should approach 1 as the mission time runs out. This is a result of there being less potentially hazardous events between the astronaut and the end of the mission.

While the astronaut can survive through a 1150 kilocalorie diet and by limiting the amount of physical work the astronaut does for two weeks, the astronaut will still lose a dangerous amount of weight. The 85 kg astronaut that started the mission will drop down to approximately 50 kg in this scenario. Half of the weight lost is lost in the final 150 sols of the mission. For comparison, under the conditions of the control scenario the astronaut only loses 23 kg, dropping down to 62 kg. The astronaut is very likely to starve to death at around 48 kg under these conditions. Figure 6 shows a plot of the astronaut’s estimated weight over time.

By reducing the rations to 1115 calories per day and spending a few weeks resting and reducing physical work tasks, the food can be stretched through the end of the mission. The results are shown in Figure 5.

Figure 5. Instantaneous Survival Rate (top) and Mission Survival Rate (bottom) after losing half of the food on sol 350. Food is rationed to 1115 calories per day.

Figure 4. Instantaneous Survival Rate (top) and Mission Survival Rate (bottom) after losing half of the food on sol 350. No food rationing occurs. Mission failure is assured by Sol 543.
A final piece of the results that deserve attention is the mission survival rate near the beginning of the mission vs the end. Near the beginning of the mission the survival rate was much lower in general and near the end the survival rate should approach 1. The general trend results from the fact that the beginning of the mission still has many potential problems that may arise and lead to the loss of the astronaut and subsequent mission failure. The mean mission success value for the control mission is found to be around 0.06 which means a 6% rate of astronauts in this scenario being successfully rescued.

6. CONCLUSION AND FUTURE WORK

The AMSE method is shown to be a viable tool for mission success estimation. The method is much more dynamic than existing methods due to its object-oriented approach to mission design and assessment, and is able to accurately model complex mechatronic systems and their interactions with a human sub-system through the application of PHM techniques and methodology.

While the current revision of the AMSE method is fast and effective, there is still room for improvement. To make the method more accessible, software should be developed for AMSE that has an improved user interface and leads the user more directly through the method to ensure that the analysis is performed completely and correctly. Additionally, a future AMSE case study should be applied to large missions with more individuals and equipment as well as a higher degree of model fidelity. While the analysis presented in the case study tracks the use of multiple SEVs and EVA suits, there is only one human component to consider. Modelling a mission team dynamic may be very interesting and improve upon the model as a tool for mission planning and assessment. The resolution of the models themselves may also be increased to give a better understanding of the nuances of a mission from emergent system behavior. Currently the Martian base, SEV, EVA, and astronaut all have between four and 12 parameters that affect their state, but these can be increased and linked more complexly to allow for methods such as FFIP and FFDF to be built directly into the analysis to determine what factors may more directly affect critical sub-systems. A final expansion of AMSE is a built-in optimization toolkit for hassle-free solving of problems such as food and resource rationing over the course of the mission.

AMSE has shown to be an effective tool for rapid and effective mission planning and assessment. The object oriented structure of the method allows for rapid construction and analysis of mission alternatives. This will enable more dynamic and informed mission planning to be performed. Additionally the AMSE method shows a great deal of potential for future improvement and development.

ACKNOWLEDGMENTS

The authors wish to thank the staff of the mechanical engineering department and graduate school at Colorado School of Mines, and Andy Weir for inspiration for the paper. The authors look forward to the next time Mark Watney ends up on the wrong end of an aborted mission.

REFERENCES


Aircraft Line Maintenance Planning Based on PHM Data and Resources Availability Using Large Neighborhood Search

Wlamir Olivares Loesch Vianna1, Leonardo Ramos Rodrigues1 and Takashi Yoneyama2

1EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil
wlamir.vianna@embraer.com.br
leonardo.ramos@embraer.com.br

2ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil
takashi@ita.br

ABSTRACT
Maintenance planning has become a topic of great interest among researchers and industry practitioners in recent years, since it directly impacts the availability and the lifecycle cost of systems. In the aviation industry, maintenance planning becomes even more relevant due to the high availability expectations from aircraft operators and the high costs incurred when an aircraft becomes out of service. For this reason, some minor maintenance activities are carried out near the gate, between two consecutive flight legs. These activities are referred to as aircraft line maintenance. Planning line maintenance activities is critical because a problem in the execution of line maintenance may lead to flight delays and even flight cancellations. This paper presents a methodology for aircraft line maintenance planning including both the troubleshooting tasks and the repair activities. The proposed methodology uses a Large Neighborhood Search (LNS) algorithm in order to find the most appropriated time and location to perform line maintenance activities. The algorithm considers the precedence relation between a troubleshooting task and its respective repair activity, as well as the dispatchability constraints included in the MEL (Minimum Equipment List). Resources availability such as spare parts, equipments and personnel are taken into account, as well as the risk of occurrence of an AOG (Aircraft on Ground) event, estimated from PHM (Prognostics and Health Monitoring) data. An AOG event is an event that leads to a flight cancelation. The optimization goal is to minimize the Expected Cost of Repair (ECR) considering both delay and AOG expenses. A numerical example is presented to illustrate the application of the proposed methodology.

1. INTRODUCTION
An airline flight operations department needs an efficient planning system in order to successfully manage its expensive assets. Considering that, operational research plays a major role in the airline industry’s tactical planning (Barnhart, Belobaba & Odoni, 2003). Most applications in the literature deal with the following areas (Sarac, Batta & Rump, 2006):

- The schedule preparation, where airlines identify a list of flight legs along with departure and arrival times;
- The fleet assignment problem;
- The aircraft routing area; and
- The disruption recovery problem, whose objective is to react to all operational disruptions.

Also, several maintenance planning applications have been proposed in these areas. According to Papakostas, Papachatzakis, Xanthakis, Mourtzis and Chryssolouris (2010), although the increasing progress, most of these approaches have limitations. GO/NOGO decisions are not directly supported and the academic demonstrators so far supporting this kind of functionality have limited intelligence without concurrently taking into consideration parameters such as possible flight delay, cost consequences and actual Remaining Useful Life (RUL) of aircraft systems and components.

Wlamir Vianna et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
Many applications deal with the troubleshooting optimization problem (Langseth & Jensen, 2003; Lin, 2012; Pernesta, Nyberg & Warnquist, 2012; Kalagnanam & Henrion, 1990), which consists of defining the best test sequence in order to optimize the balance between the test cost and the probability that the test will be helpful to isolate the fault. Basically, all non-trivial troubleshooting domains are NP-hard (Vomlelová, 2003), specially those considering operation limitations such as turnaround time (TAT) and resources availability.

This paper proposes a methodology for defining a good line maintenance strategy using a LNS (Large Neighborhood Search) algorithm that takes into account the troubleshooting tasks, the flight plan, resources availability and the health condition of components.

The remaining of this paper is organized as follows. Section 2 describes the formulation of the problem. The proposed methodology is discussed in detail in Section 3. A numerical example to illustrate the application of the proposed methodology is presented in Section 4. Concluding remarks and future research opportunities are given in Section 5.

2. Problem Formulation

Airline maintenance activities can be divided into 3 groups: shop maintenance, hangar maintenance and line maintenance. This paper is focused on line maintenance, which is defined by the FAA (2013) as "any unscheduled maintenance resulting from unforeseen events, or scheduled checks where certain servicing and/or inspections do not require specialized training, equipment or facilities".

For the events that are not related to an AOG situation (i.e., events that do not compromise the aircraft minimal safety requirements and do not cause a flight cancelation), a decision must be made in order to define the sequence and most appropriate time to execute the troubleshooting tasks and the repair activities in order to fix the failure. To accomplish that, the variables described below are considered.

A problem \( D \) that consists in defining the time and location for a group of troubleshooting tasks and repair activities to be performed, needs to be solved. There is a set of faults \( F = \{F_1, \ldots, F_m\} \) that describes all possible causes for the problem. Each fault \( F_i \in F \) has an associated probability \( p_i \) that represents the probability of problem \( D \) to be caused by fault \( F_i \). Also, a troubleshooting task \( T_i \) with duration \( d(T_i) \) and a repair activity \( R_i \) with duration \( d(R_i) \) are associated to each fault \( F_i \).

There is a set of flight legs \( L = \{L_1, \ldots, L_n\} \) that describes the next flights of the aircraft under consideration. For each flight leg \( L_j \in L \), a turnaround time \( TAT(L_j) \) describes the time interval available to perform maintenance activities between the \( j \)-th flight leg and the next one.

A binary matrix \( CT(T_j, L_j) \) indicates if the base where the aircraft will be after flight leg \( L_j \) has all the resources (materials, maintenance personnel, etc.) required in order to execute the troubleshooting task \( T_i \). Similarly, a binary matrix \( CR(R_j, L_j) \) indicates if the base where the aircraft will be after flight leg \( L_j \) has all the resources required in order to execute the repair activity \( R_i \).

A matrix \( P_{AOG}(F_i, L_j) \) defines the probability of an AOG event to be caused by each fault \( F_i \) at each flight leg \( L_j \). Each element of this matrix is estimated based on PHM data. Details on how these estimates are made are presented in Section 4.

A cancellation cost \( C_{AOG} \) defines the cost incurred if an AOG event occurs. A delay cost per minute \( C_{delay} \) defines the cost incurred for each minute of delay.

Furthermore, the following assumptions are considered:

- Problem \( D \) is always caused by a single fault \( F_i \in F \).
- New faults or changes in the parameters of matrix \( P_{AOG}(F_i, L_j) \) are never introduced during the execution of troubleshooting or repair tasks.
- Every AOG event results in a flight cancellation while every task execution that exceeds the available time \( TAT(L_j) \) results in a flight delay.
- The troubleshooting task \( T_i \) can isolate only fault \( F_i \in F \).
- The repair activity \( R_i \) is effective only for fixing fault \( F_i \in F \).
- Maintenance activities can not be executed in parallel.
- Maintenance activities can not be split to be carried out in two or more different flight legs.
- The flight plan is fixed independent of task allocation.

The optimization problem consists of defining the following variables:

- The troubleshooting task plan \( TP = \{TP_1, \ldots, TP_m\} \), where \( TP_i \) indicates the most appropriate time to execute the troubleshooting task \( T_i \).
The repair task plan $RP = \{RP_1, ..., RP_m\}$, where $RP_i$ indicates the most appropriate time to execute the repair task $R_i$.

$TP$ and $RP$ are related to the flight legs $L$. It implies that $TP_i$ and $RP_i$ have integer values. For example, if $TP_3 = 3$, it means that the troubleshooting task associated to fault $F_2$ will be carried out after the third flight leg.

A time availability matrix $A$ must be built. Each element $A_{ij}$ of matrix $A$ contains the amount of time in $TAT(L_j)$ that is not used by any troubleshooting task or repair activity, considering that problem $D$ was caused by fault $F_i$. The expressions to calculate the elements of matrix $A$ are shown in Eq. (1).

$$A_{ij} = \begin{cases} TAT(L_j) - \sum d(T_i) \cdot \delta_{ij} & j < RP_i \\ TAT(L_j) - d(R_i) - \sum d(T_i) \cdot \delta_{ij} & j = RP_i \\ TAT(L_j) & j > RP_i \end{cases} \quad (1)$$

where $\delta_{ij}$ is a binary variable that assumes the value "1" if $TP_j = j$ and the value "0" otherwise.

The Expected Cost of Repair considering that the problem $D$ was caused by fault $F_i$, $ECR_i$, is calculated as shown in Eq. (2).

$$ECR_i = C_{AOG} \cdot P_{AOG}(i, RP_i) + C_{\text{delay}} \sum_{j=1}^{RP} \max \{0; -A_{ij}\} \left[1 - P_{AOG}(i, j)\right] \quad (2)$$

Finally, the Total Expected Cost of Repair, $TECR$, is obtained according to Eq. (3).

$$TECR = \sum_{i=1}^{m} ECR_i \cdot p_i \quad (3)$$

The $TECR$ is the objective function of the optimization problem. The problem constraints are listed below.

- $CT(T_i, TP_i) = 1$, for $i = 1, ..., m$
- $CR(R_i, RP_i) = 1$, for $i = 1, ..., m$
- $RP_i \geq TP_i$, for $i = 1, ..., m$

The two first constraints are related to the resources availability and base limitations to execute each troubleshooting task and repair activity, while the last constraint is related to the precedence relation between a troubleshooting task and its respective repair activity.

Once $TP$ and $RP$ are defined, maintenance activities are carried out according to Algorithm 1.

**Algorithm 1: Plan Execution**

1. $fault \leftarrow 0$
2. for $j \leftarrow 1, n$ do
3. \hspace{1em} for $i \leftarrow 1, m$ do
4. \hspace{2em} if $TP_j = j$ then
5. \hspace{3em} Execute $T_i$
6. \hspace{2em} if $F_i = true$ then
7. \hspace{3em} $fault \leftarrow i$
8. \hspace{2em} end if
9. \hspace{2em} end if
10. \hspace{1em} if $RP_j = j$ and $fault = i$ then
11. \hspace{2em} Execute $R_i$
12. \hspace{1em} end if
13. \hspace{1em} end for
14. \hspace{1em} return

3. Optimization Methodology

Considering the complexity of the current problem (NP-Complete), a heuristic method will be used and therefore an optimal solution is not guaranteed. In this paper, constraint programming (CP) and local search (LS) will be used to solve the optimization problem described in the previous Section. Although this method has a higher change of being stuck in a local minima solution compared to other more robust optimization methods such as Simulated Annealing and Genetic Algorithms, the decision to choose this method was made due to its low computational cost and relative easy implementation. The use of several methods to solve similar problems can be found in literature. Papakostas et al. (2010) used a multi-criteria mechanism for deferring maintenance actions. Langseth and Jensen (2003) presented a greedy heuristic algorithm for fault diagnosis. Optimal partitioning (Ottosen & Jensen, 2011) and Bayesian networks (Pernesta et al., 2012) have also been used to solve maintenance planning problems.

The efficiency of a troubleshooting task is used by many authors in maintenance action solutions (Ottosen & Jensen, 2011). The efficiency of a troubleshooting task is defined in Eq. (4).

$$\text{eff}(T_i) = \frac{p(T_i)}{d(T_i)} \quad (4)$$
The efficiency ranks the troubleshooting tasks in order of the most probable action with less cost, or duration in this particular case. Tasks with higher efficiency values are carried out previous to others with lower efficiency values. One good strategy to begin with is to perform the tasks in the efficiency order as soon as possible with no delay. In other words, $A_{ij} \geq 0$ for every $F_i$ and $L_j$. The same strategy is applicable to the repair tasks plan. Although this strategy does not consider the AOG costs, it is a good starting point for the optimization problem.

Considering the starting point strategy described above, the combination of LS and CP and the search space of the problem, a Large Neighborhood Search (LNS) optimization algorithm was implemented.

LNS is a heuristic algorithm that generally starts with a feasible solution and iteratively tries to obtain a better solution by searching the “neighborhood” of the current solution. A critical issue in the design of a neighborhood search algorithm is the choice of the neighborhood structure, i.e., the manner in which the neighborhood is defined. At the same time, the larger the neighborhood, the longer it takes to search the neighborhood at each iteration (Ahuja, Ergunb, Orlinc & Punnend, 2002). In this paper, the neighborhood was defined based on two operations:

- **Swapping**: Exchange the execution time of two tasks.
- **Shifting**: Anticipate or postpone a task.

The first operation evaluates if allocating the time reserved for one task to another one with different efficiency reduces the AOG risk with the possible impact of increasing delay costs. The second operation evaluates if executing a task before improves costs by reducing AOG risks with the possible impact of increasing delay costs.

Considering the neighborhood structure defined for the problem, the proposed methodology finds a solution according to Algorithm 2.

Function “Initial Solution” generates the troubleshooting and repair plans ($TP$ and $RP$) based on the task efficiency rank as previously discussed. Function “Swap” generates all possible strategies by swapping two tasks at a time. Function “Shift” generates all possible strategies by shifting each task to all available times. Function “TECR” estimates the Total Expected Cost of Repair for each alternative. This algorithm estimates $TECR$ for all possible neighbors considering all tasks swapping and shifting and, if any neighbor presents a lower $TECR$ compared to the current solution, this neighbor is selected as the new current solution. The process is repeated until there is no neighbor with lower $TECR$ compared to the current solution.

In the proposed methodology, only the swap between two tasks and the shift of one task was considered. It can limit the search area. The neighborhood search is executed only for the current best solution and the algorithm always converges for the same strategy independent of how many trials are made for the same inputs.

### 4. Numerical Example

In this Section, an example of application of the proposed methodology is presented. In this example, we suppose that a “Bleed 1 Fail” message associated to the failure of an aircraft bleed system 1 happened and a decision should be made in order to isolate and repair the fault. There are six faults that may cause the problem:

1. Shutoff Bleed Valve.
2. Transitory Condition (Spurious Message).
3. Torque Motor Controller.
5. Cross Bleed Valve.

The probability vector $p$ and tasks duration used in this example are:

\[
p = [0.45 \ 0.30 \ 0.10 \ 0.05 \ 0.05 \ 0.05]
\]

\[
d[T_1...T_6] = [90 \ 10 \ 15 \ 60 \ 80 \ 70]
\]

\[
d[R_1...R_6] = [0 \ 80 \ 80 \ 0 \ 0 \ 0]
\]

The faults whose troubleshooting task results in repairing the fault has a repair duration equals to 0. To illustrate this situation consider the following example: The troubleshooting tasks to isolate the Shutoff Bleed Valve includes replacing the valve to verify whether the fault is removed and, if that is the case, no more activities are required.
The delay cost $C_{\text{delay}}$ and the AOG cost $C_{\text{AOG}}$ were arbitrarily chosen and defined as 300 and 120,000, respectively.

The variables associated to the flight plan and bases resources used in this example are:

$$TAT = [40, 37, 33, 55, 814, 62, 66, 34, 730, 25, 839, 33, 45]$$

$$CT = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$

$$CR = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$

Monte Carlo simulation was used to generate a set of 5,000 realizations of the bleed system 2 degradation index evolution and the associated time of failure. Here, the time of failure is associated to the time when the degradation index reaches 100%.

The AOG probability is associated to the probability of failure of bleed system 2, considering that Bleed System 1 is unavailable and the aircraft must not be dispatched with both bleed systems failed. In order to estimate the probability distribution of a failure in the bleed system 2 to occur, the PHM algorithm proposed by Gomes, Ferreira, Cabral, Galvão & Yoneyama (2010) was used. Figure 1 shows the Bleed Valve from System 2 degradation indexes for this method over the last 30 days previous to the fault message.

The degradation indexes were used in order to estimate the coefficients of a linear curve using a least square regression method with a confidence level of 95%. The coefficients and the covariance matrix obtained from the data shown below.

$$b_0 = 72.07$$
$$b_1 = 1.66$$

$$\text{cov}_{b_0,b_1} = \begin{bmatrix}
2.90 & 0.15 \\
0.15 & 0.01 \\
\end{bmatrix}$$

where $b_0$ and $b_1$ are the coefficients of the linear model $y = b_0 + b_1 \cdot x$. In this model, the dependent variable $y$ is the degradation index and the independent variable $x$ is the time.

The set of time of failures obtained from the Monte Carlo simulation was then used to fit a Weibull distribution for the Remaining Useful Life (RUL) of bleed system 2.

The probability of an AOG event to occur due to a bleed system 2 failure associated to each leg was estimated by inserting the date of each flight at the Cumulative Distribution Function (CDF) of the Weibull distribution. Figure 2 shows the CDF obtained for this example.

![Figure 1: Bleed System 2 past degradation indexes.](image1.png)

![Figure 2: CDF obtained from the Weibull curve fitting.](image2.png)
These probabilities are associated to every fault except fault \( F_2 \) (Transitory Condition), which does not turn the Bleed System unavailable. Considering the CDF presented in Figure 2 and the TAT values, the probability matrix associated to each failure, \( P_L \), is:

\[
P_L = \begin{bmatrix}
0.28 & 0.29 & 0.30 & 0.35 & 0.43 & 0.46 \\
0.51 & 0.53 & 0.63 & 0.65 & 0.80 & 0.84 & 0.87
\end{bmatrix}
\]

The matrix \( P_{AOG} \) for the example is:

\[
P_{AOG} = \begin{bmatrix}
P_L & \text{zeros}(13,1) & P_L' & P_L' & P_L' & P_L'
\end{bmatrix}
\]

After all variables are defined, the proposed methodology was used and the Total Expected Cost of Repair was estimated. The troubleshooting plan, \( TP \), and repair plan, \( RP \), obtained for this example were:

\[
TP = [6\ 1\ 1\ 4\ 8\ 13]
\]

\[
RP = [6\ 1\ 6\ 4\ 8\ 13]
\]

\[TECR = 11,376\]

Comparisons were made to two other possible strategies. The first one, which will be called "Method A", is a conservative strategy that recommends executing all tasks at the first possible base, even if it causes a flight delay. The second one, which will be called "Method B", recommends not executing any preventive action. Instead, it recommends executing a corrective repair action once the failure has happened. It makes the Total Expected Cost of Repair to be equal to the AOG cost \((TECR = C_{AOG})\). The Total Expected Costs of Repair estimated from the application of the methods are summarized in Table 1.

<table>
<thead>
<tr>
<th>Optimization Method</th>
<th>TP</th>
<th>RP</th>
<th>TECR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNS</td>
<td>[6 1 1 4 8 13]</td>
<td>[6 1 6 4 8 13]</td>
<td>11,376</td>
</tr>
<tr>
<td>Method A</td>
<td>[1 1 1 1 1 1]</td>
<td>[1 1 1 1 1 1]</td>
<td>48,150</td>
</tr>
<tr>
<td>Method B</td>
<td>N/A</td>
<td>N/A</td>
<td>120,000</td>
</tr>
</tbody>
</table>

From these results it is possible to see how effective the proposed methodology was in this example. It found a much cheaper solution compared to the other strategies.

5. CONCLUSIONS

This paper presented an aircraft line maintenance planning methodology including both the troubleshooting tasks and the repair activities to be carried out during the turn-around time. Resources availability and flight plan were considered. The probability of an AOG event to occur was also taken into account, based on RUL estimates obtained from a PHM system. A Large Neighborhood Search (LNS) algorithm was used in order to optimize the expected cost of repair (ECR), considering delay and AOG costs. A numerical example was presented to illustrate the application of the proposed methodology. The results showed that the proposed methodology is promising, although it does not guarantee that the optimal TECR will be found.

Improvements in the proposed methodology could be made by implementing more robust neighborhood definitions. Some operations that could be used in the neighborhood definition include compounded swaps, cyclical shifts, assignments/matching (Ahuja et al., 2002), optimal partitions (Ottosen & Jensen, 2011) and multi-start search algorithms.

Future work opportunities include evaluating the proposed methodology with real data, proposing new operations for the neighborhood definition. Another direction for future research is to evaluate the performance of different local search algorithms such as Simulated Annealing, Tabu Search, Ant Colony Optimization and Genetic Algorithms to solve the troubleshooting optimization problem. Also, adding new constraints to the optimization problem in order to consider other aspects such as safety and risk requirements is an interesting direction for future research.

The assumptions adopted in this paper may not consider some real operational situations such as the occurrence of multiple failures, the execution of troubleshooting tasks in parallel and the flexibilization of flight plans. Proposing a more robust optimization troubleshooting model that considers all these operational situations is also a topic for future research in this area.

ACKNOWLEDGEMENT

The authors acknowledge the support of FINEP (grant 1498/07).
REFERENCES


BIographies

Wlamir Olivares Loesch Vianna holds a bachelor’s degree on Mechanical Engineering (2005) from Universidade de São Paulo (USP), Brazil, and Master Degree on Aeronautical Engineering (2007) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2007. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications in aeronautical systems.

Leonardo Ramos Rodrigues holds a bachelor’s degree in Electrical Engineering from Universidade Federal do Espírito Santo (UFES, 2003), Brazil, a Masters’s Degree (2008) and a Doctorate’s Degree (2013) in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, Brazil). He is with EMBRAER S.A. since 2006, working as a Development Engineer focusing on PHM technology applications. His current research interests are the application of health monitoring techniques for electronic components and the usage of PHM information for inventory optimization.

Takashi Yoneyama is a Professor of Control Theory with the Electronic Engineering Department of ITA. He received the bachelor’s degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in medicine from Universidade de Taubaté, Brazil, and the Ph.D. degree in electrical engineering from the University of London, U.K. (1983). He has more than 300 published papers, has written four books, and has supervised more than 70 theses. His research is concerned mainly with stochastic optimal control theory. He served as the President of the Brazilian Automatics Society in the period 2004-2006.
Cost-benefit Analysis of Prognostics and Condition-based Maintenance Concepts for Commercial Aircraft Considering Prognostic Errors

Nico B. Hölzel1 and Volker Gollnick2

1,2DLR - German Aerospace Center, Air Transportation Systems, Hamburg, 21079, Germany
nico.hoelzel@dlr.de
volker.gollnick@dlr.de

ABSTRACT
This paper provides a lifecycle cost-benefit analysis of the use of Prognostics and Health Management (PHM) systems in future or present commercial aircraft. The approach considers individual aircraft component’s failure behavior, prognostic performance levels including prognostic errors, and condition-based maintenance (CBM) concepts. The proposed methodology is based on a discrete-event simulation for aircraft operation and maintenance and uses an optimization algorithm for the planning and scheduling of condition-based maintenance (CBM) tasks. In the study, a 150-seat short-/medium-range aircraft equipped with PHM and subject to a CBM program is analyzed. The simulation results are evaluated from an operational and economic perspective. The analysis results can support the derivation of technical and economic requirements for prognostic systems and CBM planning concepts.

1. BACKGROUND
In general, prognostic systems provide early detection of the precursor (and/or incipient) fault condition of a component and are capable to predict its remaining useful life (RUL) (Engel et al., 2000). In addition, the fault isolation and identification capabilities of PHM contribute to a reduction of no-fault-founds (NFFs) and support the trouble shooting process (Leao et al., 2007). The implementation of PHM in commercial aircraft can help to reduce operational interruptions due to unscheduled maintenance events.

Significant reductions in maintenance downtimes and costs can be obtained when today’s periodic, preventive maintenance is transformed towards a predictive (i.e. condition-based) maintenance strategy. The major expected benefits in this case are substitutions of preventive inspection tasks and reductions of waste of (component-) lives. This will lead to reductions of overall maintenance cost and downtimes. These benefits are also known as the realization of maintenance credits.

But a CBM concept leads to an increased planning complexity and therefore requires a different maintenance planning approach in order to achieve the aimed goals of a PHM and CBM implementation (Hölzel et al., 2014).

Besides the solving of technical challenges of prognostics one important prerequisite of an implementation is the provision of a reliable cost-benefit assessment of the onboard use of PHM. Such an analysis must be able to capture all relevant impacts of the technology on aircraft operation and maintenance over the aircraft lifecycle.

Economic assessments of PHM applications have been discussed by many authors (e.g. Banks et al., 2005; Feldman et al., 2009; Leao et al., 2007; Sandborn & Wilkinson, 2007; Scanff et al., 2007). Typical measures are lifecycle costs (LCC) or return-on-investment (ROI) estimates of the implementation costs and the potentials for cost avoidance (e.g. Banks et al., 2005). Leao et al. (2007) developed a cost-benefit analysis (CBA) methodology for PHM applied to legacy aircraft. Their approach is capable to conduct assessments from an aircraft manufacturer’s or operator’s perspective, but it requires many inputs from technical analyses and PHM specialists. Sandborn and Wilkinson (2007) have proposed a lifecycle cost approach which includes a maintenance planning model and considers various uncertainties with regard to PHM systems. While the model provides a detailed picture of the usefulness of PHM on component or sub-system level, it does not cover additional impacts and interactions on overall system (i.e. aircraft) level.

Both levels of analysis, component and overall system level, are needed, when a profound CBA of PHM with particular attention on the implementation of CBM should be provided. A cost-benefit approach has to cover the relevant impacts of PHM on component or sub-system level and
should consider the corresponding uncertainties. This component level must then be integrated on aircraft level, in order to analyze the effects of PHM and CBM in a realistic aircraft operation scenario.

In real world applications, no prognostic system will operate completely perfect. Therefore, uncertainties and prognostic performance levels including probabilities of false prognoses (false positives) and missed failures (false negatives) have to be considered (Saxena et al., 2010). Previous analyses have shown that the prognostics performance level has a significant impact on the added value of a PHM system (Hölzel et al., 2012).

When assessing technologies and processes with impacts on the air transportation system level, all phases of the life cycle and interdependencies with other system elements have to be considered. This is true for the assessment of PHM, since new maintenance concepts influence maintenance cost and aircraft availability (and thereby aircraft utilization). The use of a discounted cash-flow method is required to take into account the time value of money when assessing an aircraft or technology concept over its entire lifecycle.

The overall benefits of a PHM application depend on the criticality of the monitored item (in terms of safety and operational reliability of the aircraft), the prognostic performance levels and both the current and novel maintenance concept. Therefore, a detailed modeling and analysis of all relevant factors and economic conditions is needed.

2. GOAL OF STUDY

The goal of this study is to propose an appropriate method for analyzing the economic potentials of a PHM and CBM implementation in existing and future commercial aircraft. The applied methodology should facilitate informed decision making in the design or acquisition phases of PHM systems.

The applied approach should consider all phases in aircraft lifecycle and include the following benefits of PHM deriving from the capability to provide advanced warnings of failures and predictions of the RUL:

1. Reduction of unscheduled maintenance events due to failures (and NFFs) of items/components.
2. Enabling CBM: Transition from preventive to condition-based maintenance measures.

To consider uncertainties in component failure behavior, the methodology used in the study should be based on individual component failure probability functions. Prognostic errors (i.e. false alarm rates and missed failure rates) have to be included to account for imperfect sensors or prognostic algorithms. The selected approach should be able to simulate the impacts of PHM systems and a CBM concept in a realistic aircraft operation scenario. The simulation results are then evaluated in a lifecycle cost-benefit model.

The approach is demonstrated in a case study to show the potential economic benefits of a PHM/CBM concept from an airline perspective including possible prognostic errors and uncertainties in technical failure behavior.

3. METHODOLOGY

This chapter gives an overview of the generic aircraft lifecycle analysis method used for this study and describes the specific assessment approach for the CBA of PHM and CBM concepts.

3.1. Aircraft Lifecycle Analysis Approach

At DLR, the lifecycle cost-benefit model AIRTOBS (Aircraft Technology and Operations Benchmark System) was developed to enable a holistic economic assessment of aircraft technologies’ already in a conceptual design phase.

The model captures time and cost aspects in aircraft lifecycle, is generic in nature and is feasible for economic assessments of various aircraft technologies and operation concepts from an operator’s perspective. Apart from the assessment of prognostic concepts (Hölzel et al., 2012; Hölzel et al., 2014), studies on aircraft with natural laminar flow (Wicke et al., 2012) or intermediate stop operation concepts (Langhans et al., 2010) have been conducted.

It models all economic relevant parameters along the aircraft life cycle. The aircraft operational lifecycle is initiated by the acquisition of an aircraft and ends with the decommissioning. The model includes aircraft specific parameters (e.g. acquisition cost, fuel consumption, seating capacity, crew size, and aircraft specific charges), operational aspects (e.g. route network, maintenance concepts and costs, and ticket prices), as well as global boundary conditions (e.g. fuel price trend, annual inflation rate). AIRTOBS focuses on the perspective of an aircraft operator and includes methods to account for costs and revenues.

An overview of AIRTOBS is shown in Figure 1. It consists of three main modules. The Flight Schedule Builder (FSB) generates a generic aircraft lifecycle flight schedule based on airline route data assuming full aircraft availability (i.e. no maintenance). Routes are considered based on the aircraft cycle time including flight time, taxi and runway operation times, and turnaround time. This flight schedule has the character of a basic mission plan for a single aircraft. It does not include any maintenance downtimes.

1 In this context, the term technologies can represent aircraft, systems, components, or aircraft operational and maintenance concepts.
The mission plan serves as the fundament for the Maintenance Schedule Builder (MSB). The MSB executes a discrete-event simulation of the flight operation and maintenance events along the aircraft lifecycle. The MSB uses input data from maintenance databases for the modeling of scheduled and unscheduled maintenance events, including airframe, engine and component maintenance.

The Lifecycle Cost-Benefit (LC2B) module calculates all costs and revenues on the basis of the simulated aircraft operation and maintenance using pre-defined cost and revenue models. All values are escalated over the aircraft lifecycle to account for inflation, before they can be summarized as net present value (NPV). It can be calculated as given in Eq. (1), where $C_0$ is the initial investment (i.e. aircraft price) and $C_i$ is the cash-flow in the $i$-th year. The discount rate $r$ represents the rate of return that could be achieved with equivalent investment alternatives in the capital market (Brealey, Myers, & Franklin, 2006). In business practice, a company or industry weighted average cost of capital (WACC) is often used as discount rate.

$$NPV = -C_0 + \sum \frac{C_i}{(1 + r)^i}$$ (1)

The NPV is one among many other metrics that are calculated in AIRTOBS and can be used for the comparative evaluation of aircraft technologies and operational concepts.

The presented simulation and assessment tool AIRTOBS is modeled in MATLAB®. Each module requires specific input data and can be configured with regard to analysis goals and needs. Aircraft type and operator specific XML-files are used to configure and control the analyses.

3.2. Applied Assessment Approach

The economic analysis in this paper follows the assessment approach as outlined in Figure 2. At its core, the approach is based on the discrete-event simulation of aircraft operation including the optimization algorithm for maintenance planning provided by AIRTOBS. The desired economic performance indicators are calculated with the LC2B module.

A CBA is realized by comparing the system under assessment (i.e. aircraft equipped with PHM and subject to a CBM program) with a pre-defined baseline (i.e. reference aircraft without PHM and subject to a conventional maintenance program).

The analysis requires a large amount of input data:

- PHM system: specification of covered failure modes of sub-systems or components, corresponding prognostic performance levels and costs,
- Reference aircraft: aircraft data, scheduled maintenance program, MEL, component failure behavior, etc.,
- Maintenance capacities at considered airports: number of mechanics, hangar slots, capabilities, etc.,
- Flight schedule and aircraft rotation plan,
- Operational and boundary conditions: ticket prices, labor cost, inflation, etc.

Based on the specified PHM system and a selected aircraft with its corresponding failure probability density functions (PDFs) a lifecycle simulation of technical failures is conducted. This process results in RUL values (in case of a successful prognosis) and the generation of unscheduled maintenance events (in case of failure not predicted by the PHM system).
Preventive maintenance tasks derived from the aircraft’s maintenance planning document (MPD) and CBM tasks initiated by the estimation of RULs are subject to an integrated maintenance planning and optimization process. Available maintenance capacities at different airports, the planned flight missions and rotation plan form the constraints of the optimization problem. The optimizer identifies a valid and efficient maintenance plan.

The planned flight missions are derived from a flight schedule (generated by the FSB module for a selected airline’s operation concept).

The discrete-event simulation then models the flight operation and maintenance in aircraft lifecycle based on simulated unscheduled events and calculated scheduled (preventive and condition-based) maintenance events and the corresponding aircraft downtimes.

Finally, the overall economic analysis is conducted using the LC2B module of AIRTOBS.

Parametric studies will show the sensitivities of prognostic performance levels, CBM implementation and maintenance planning constraints with regard to the benefits of an operator’s point of view. From these studies, it is possible to derive essential requirements for prognostic systems and CBM concepts, e.g. minimum performance levels, maximal costs for acquisition and operation and minimum maintenance capacities, under given conditions.

### 3.3. Modeling of Maintenance Events and PHM Impacts

This section describes the modeling of maintenance events and the logic how the impacts of PHM on scheduled and unscheduled maintenance are implemented in the MSB module as depicted in Figure 1. The maintenance modeling is realized as discrete-event simulation based on the planned flights in aircraft lifecycle.

#### 3.3.1. Scheduled Maintenance

Scheduled maintenance is considered depending on discrete, interval-based events. Intervals are specified by flight hours (FH), flight cycles (FC), and calendar time (years, months, days). Each event has a specific ground time, during which the flight schedule is adjusted while producing time discrete costs to the airline. To account for operating experience and maturity effects in maintenance, maturity curves are provided within the model. The maintenance schedule created by the MSB follows a traditional block check concept for heavy maintenance. Line maintenance checks are modeled on task-oriented basis and can thereby be subject to a dynamic planning process.

#### 3.3.2. Unscheduled Maintenance

An unscheduled event is characterized by a technical failure and/or a fault message (of a diagnostic system), fault report (of crew or maintenance), or a finding (due to an inspection). It can be followed by one or more component removals taking place in aircraft line maintenance. A component removal results in a shop maintenance event and the installation of an airworthy (new or repaired) component.

Modeling of unscheduled maintenance requires knowledge of the failure behavior of the respective components or systems. When sufficient historic data are available, (parametric or non-parametric) failure distribution functions can be calculated (Hölzel et al., 2012). The presented approach uses discrete component lifetimes randomly drawn from the estimated failure distribution functions to model unscheduled removals on component or sub-system level over the aircraft lifecycle (Figure 3).

![Figure 3. Modeling of component lifetimes.](image-url)
identical first three digits (ATA 3D Chapter, i.e. sub-system level) (Hölzel et al., 2012).

NFF\(^2\) events are modeled based on the NFF probabilities per FH that have been calculated from in-service data. The occurrence of an NFF event leads to an unscheduled component removal. The result is an early end of the current lifetime of a component, marked with a star in Figure 4 a. The beginning of the subsequent component lifetime is brought forward to the date of the NFF event, as shown in Figure 4 b. All other future component lifetimes are pulled forward correspondently (Hölzel et al., 2012).

Using the previously (by the FSB module) created lifetime flight schedule, unscheduled events are simulated based on component failure behavior, aircraft related mean times to repair (MTTR) and maintenance man-hours, i.e. downtime and man-hours needed for replacement of a component or LRU. Component removals produce costs for labor and material. Furthermore they can result in flight delays or cancellations depending on the minimum equipment list (MEL), the MTTR, and the planned aircraft turnaround time. Delays are modeled as a reduction in aircraft availability and a cost element that covers passenger compensations and accommodation.

Unscheduled failures not meeting the MEL-conditions can cause a flight cancellation when the remaining availability is not adequate to execute all planned flights of the respective day. In addition, a delay time threshold can be defined, which enforces a cancellation when a delay exceeds the threshold.

To consider the influences of maintenance strategies and component reliabilities on spare parts provisioning, related inventory costs are modeled. Overall LRU inventory costs are modeled based on estimated component quantities to meet a desired service level and the total carrying cost (capital and inventory cost). The estimated component quantities are calculated based on the aircraft utilization, quantities per aircraft, mean times between unscheduled removals (MTBURs), repair turnaround times, and fleet size (Khan et al., 1999).

3.4. Impacts of PHM and Prognostic Errors

An implementation of prognostics in aircraft systems can lead to a variety of operational and economic benefits as described before. In this study, the following benefits of PHM are in focus:

1. Reduction of unscheduled events due to failures (and NFFs) of items/components.
2. Enabling CBM: Transition from preventive to condition-based maintenance measures with corresponding influence on aircraft downtimes and maintenance cost.

The underlying effect mechanisms of prognostics on aircraft maintenance are modeled in different ways.

\[\text{Figure 4. Modeling of NFFs and prognostic false alarms.}\]

Impending failures that are successfully predicted by the prognostic system no longer result in unscheduled events. Instead, a CBM task is generated with the estimated RUL as latest due date. Those CBM tasks are subject to the maintenance planning process described in the following section 3.5. It is assumed that NFF events of components monitored (covered) by PHM can be avoided completely.\(^3\)

Depending on the prognostic performance level (described in a PHM model) an impending failure can be detected

\(^2\) An item removal is classified as NFF when no fault is exhibited during subsequent acceptance test (James et al., 2003).

\(^3\) In reality, there are many different reasons for NFF events. It is expected that only a portion of these events can actually be prevented by the use of PHM.
Two types of prognostic errors are taken into account:

1. False alarm: Prognostic system detects an impending failure, although no failure is impending, or system reports impending failure early.

2. Missed failure: Prognostic system does not detect an impending failure or detects it late.

The modeling of prognostic errors is shown in Figure 4 and Figure 5. The occurrence of PHM false alarms (marked with a cross) in the a/c lifecycle is modeled in the same way as an NFF. Each failure of an item that is initially covered by PHM can evolve into a missed failure with a certain probability (Figure 5). A missed failure event has the same consequences as a failure not covered by PHM.

For the sake of simplification and generalization, the task codes are summarized to six task code groups (TCG) within the model as shown in Table 2. TCG 1 to 3 reflect tasks, which are potentially redundant (obsolete), if a PHM system covers the contained tasks. It is assumed that the prognostic system is able to automatically carry out a certain fraction of the check- or inspection-tasks in a continuous or non-continuous manner. The fraction of tasks covered by a PHM system can be adjusted with the task redundancy parameter $P_{TR}$. The parameter $P_{TR}$ implies that it is possible to eliminate the corresponding scheduled maintenance task from the MPD under consideration of certification requirements.

The probabilities of false alarm and missed failure events depend on the performance level of the PHM system and are input values of the model. The operational consequences of an undetected failure are modeled based on the MEL and the planned flight operation.

The potential impact of PHM on preventive, scheduled maintenance tasks depends on its task code. Scheduled maintenance tasks can be assigned to a variety of different task codes (Airbus, 2007) as listed in Table 1. While tasks with some task codes could become obsolete if a PHM system is used, prognostics have no influence on other scheduled tasks listed in the scheduled maintenance program (MPD).

If a significant fraction of scheduled tasks can be eliminated through a PHM implementation, this reduces the total workload and potentially also the aircraft downtime of a maintenance check. Without special consideration of the minimum duration of certain tasks (“shortest path”), the influence of PHM on aircraft downtimes can be estimated as shown in Eq. (2).

\[ t_{DT,new} = t_{DT,0} \left(1 - P_{TR} \cdot r_{routine} \cdot r_{TR}\right) \]

\[ t_{DT,new} \text{ resulting maintenance downtime} \]

\[ P_{TR} \text{ task redundancy parameter} \]

\[ r_{routine} \text{ fraction of routine tasks covered} \]

\[ r_{TR} \text{ fraction of reactive tasks covered} \]

---

4 It is assumed that an individual PHM system is designed to detect certain (incipient) fault conditions. Due to different uncertainties in detecting the correct fault condition and predicting the RUL it may occur that the prognostic algorithm misses an impending failure.
maintenance downtime without PHM impact (reference case)

\( P_{TR} \) task redundancy parameter

\( r_{TR} \) ratio of routine tasks potentially redundant in case of PHM use

\( r_{routine} \) ratio of routine task man-hours to complete man-hours of check

It is assumed that preventive maintenance tasks related to TCG 4 have to be carried out less frequently when the corresponding items are monitored by PHM. This means, the former limited service life of the item is extended through the use of PHM depending on the actual condition. Since no component degradation models are available for this study, the influence of PHM on service life is modeled with the interval escalation parameter \( P_{TR} \), which is assumed as input value and can be varied in a parameter variation.

In addition to routine activities, scheduled checks also comprise large amounts of non-routine tasks. Detected findings result in non-routine activities (i.e. repairs or replacements of the respective items), when the degradation may reach a critical state prior to the next preventive inspection. It is assumed that a certain part of these non-routine tasks can be conducted at a later time, the respective items are subject to a CBM strategy (and monitored by PHM). These tasks are summarized in TCG 5. The last task code group (TCG 0) includes non-routine (e.g. findings that are critical for flight safety and thus have to be repaired immediately) and other tasks (e.g. cabin refurbishments and paintings) to which a PHM system has no influence.

### 3.5. Condition-based Maintenance Planning

The planning of aircraft maintenance is the allocation of maintenance tasks (i.e. objects) that must be carried out on specific aircraft to maintenance capacities (i.e. bins). Combinatorial problems of this character are of higher complexity and are very similar to the elementary bin-packing problem (Fukunaga et al., 2007; Bohlin, 2010). Since the aircraft maintenance planning, as discussed in this paper, considers more variables and constraints as the “simple” bin packing problem, it is very likely to be NP-hard\(^5\). Although the problem might not be solved in polynomial time, solutions can efficiently be verified, e.g. by using a branch-and-bound algorithm (Korte et al., 2006; Schröder, 2011).

In the proposed approach, each ground time of an aircraft (turnaround times and overnight stays) is regarded as a maintenance opportunity. It is the goal to minimize aircraft maintenance costs and to utilize existing maintenance opportunities efficiently while aircraft rotation planning and limited maintenance capacities are considered. This is achieved by appropriate grouping (packaging) of maintenance tasks, while considering technical (maintenance intervals or RULs determined by a PHM system) and organizational restrictions. The process of task packaging reduces the number of maintenance events and allows an efficient use of maintenance opportunities. But it leads to waste of life when items are maintained earlier than required or tasks are performed before due date (Hölzel et al., 2014).

In this study, preventive scheduled and condition-based maintenance activities are subject to the maintenance planning optimization. The maintenance optimization is designed as a dynamic planning approach that responds to varying maintenance needs and airline operation during aircraft lifecycle. This is achieved by splitting the operating lifecycle into shorter planning periods (e.g. four or eight weeks) that are run through sequentially. Compared to a single optimization covering the complete lifecycle, this procedure leads to a significantly reduced computation time (due to the reduction of the optimization problem) and reflects the reality in a better way.

The CBM planning function is implemented in the AIRMAP module (as shown in Figure 1). AIRMAP is based on a mathematical formulation of the maintenance planning problem as described in Hölzel et al. (2014). The planning problem has been formulated on a fleet level to model the competition of a number of aircraft for limited maintenance resources. The applied optimization approach can be characterized as depth-first-search branch-and-bound algorithm. The resulting task packaging and maintenance scheduling process is illustrated in Figure 6. The figure shows due dates (marked with an “X”) for a number of tasks (“Task 1” to “Task n”) in two random periods in aircraft life. For each planning period, the algorithm searches for a cost-minimal maintenance plan in an iterative process. The resulting maintenance events are marked with vertical dotted lines. The distances between the time of an event and the due dates of the allocated tasks represent the waste of life (expressed in FH). Due to the limitation of maintenance capacities and individual costs and man-hours of the tasks, it can be feasible to allocate a task to an event other than the nearest (e.g. allocation of second due date of “Task 5” to “Event 2” in Figure 6).

It is possible that the optimizer cannot allocate tasks, which are due shortly after the beginning of a new period because of a lack of maintenance opportunities. To avoid this, the user of the optimizer can define a buffer period that forces the algorithm to allocate the respective tasks in the preceding period (e.g. the third execution of “Task 1” is allocated to “Event 3” in Figure 6).

The optimizer plans maintenance events for planning periods sequentially (beginning with aircraft entry into service). The algorithm takes into account only those tasks

---

\(^5\) NP-hard describes a class of problems in computational complexity theory.
that are due in the current planning period. All other tasks are moved to the next planning period.

AIRMAP submits the best plan found to the MSB module (as depicted in Figure 1), which then simulates the performed maintenance events and fulfilled flight missions over the complete aircraft lifecycle schedule as basis for the economic assessment in the LC2B module.

4. ANALYSIS

In this chapter, the data and assumptions used for the analysis and the applied parameter variation are described. Afterwards, the analysis results are presented and discussed.

The following case study is intended to demonstrate that the proposed analysis approach is suitable to assess the overall benefits and costs of the use of PHM and CBM planning in aircraft lifecycle. A focus will be put on the investigation of the operational and economic impact of prognostic errors and the statistical variance of the overall results due to the probabilistic modeling in the aircraft lifecycle simulation. While the results provide no answers regarding the suitability of specific PHM approaches or system architectures, they make it possible to derive technical and economic requirements for those in a subsequent step.

4.1. Data and Assumptions

Studies following the proposed assessment approach require extensive data, which is usually – at least partially – considered confidential by airlines and maintenance, repair & overhaul (MRO) companies. For this reason, the authors have preferably used publicly available information only or have derived the required data under use of assumption from this information.

An aircraft similar to an Airbus A320 will be used as a reference in this study. This applies to the typical aircraft operation, the maintenance program and all recurring and non-recurring costs as well as expected revenues in the operational lifecycle of this type of aircraft. It is assumed that aircraft configurations to be assessed in this study have the same technology level as today’s A320 aircraft, but with PHM installed.

The following sections describe the data and assumptions made for the aircraft operation, scheduled and unscheduled maintenance, and relevant operational boundary conditions.

4.1.1. Aircraft Lifecycle and Operations

An operating lifecycle of 25 years is assumed in this study. The aircraft is operated by a full-service network carrier on a short-range rotation with a daily utilization of 8.75 FH. Table 3 shows details of an assumed aircraft operation.

Table 3. Aircraft operational data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating days/week</td>
<td>[d]</td>
<td>7</td>
</tr>
<tr>
<td>Night curfew</td>
<td>[h]</td>
<td>7</td>
</tr>
<tr>
<td>Flights per day</td>
<td>[FC]</td>
<td>7</td>
</tr>
<tr>
<td>FH/FC</td>
<td>-</td>
<td>1.25</td>
</tr>
<tr>
<td>Taxi time per FC</td>
<td>[h]</td>
<td>0.3</td>
</tr>
<tr>
<td>Turn-around time</td>
<td>[h]</td>
<td>0.75</td>
</tr>
<tr>
<td>Block fuel</td>
<td>[kg]</td>
<td>4,000</td>
</tr>
</tbody>
</table>
4.1.2. Aircraft Maintenance and PHM Application

The modeling of unscheduled maintenance events in this study follows the approach as described in section 3.3.2. A total of 25 aircraft subsystems are considered in the study. The failure behavior of each subsystem is described by an individual non-parametric failure distribution function. It is assumed, that 15 of the 25 subsystems are potential candidates for a PHM implementation. The assumed prognostic performance levels are defined in section 4.2.

A simplified task-based maintenance program has been modeled as reference maintenance program. It is equivalent to the real A320 maintenance program in terms of man-hours and cost as described in Table 4 (Hölzel et al., 2014). It has been derived from the A320 MPD and more realistic cost data and estimates of the related man-hours published by Aircraft Commerce (2006).

The maintenance events outlined in Table 4 cover routine and non-routine tasks as well as cabin refurbishments and typical volume of work resulting from Airworthiness Directives (AD) and Service Bulletins (SB).

The modeled reference maintenance program, referred to as equivalence maintenance program in the following, consists of two parts:

1. Task-based (equalized) concept for short and medium interval tasks (former Service Check, A-Check, and C-Check),

2. Block checks for long interval tasks (former IL- and D-Check).

Transit & Pre-flight Checks can be performed at any airport and do not require an additional maintenance downtime. That is why these checks are not considered for the composition of an equivalence maintenance program and in the following maintenance planning and optimization process.

The modeled equivalence maintenance program consists of 12 short interval and 80 medium interval tasks, which represent the maintenance man-hours and task code groups shown in Table 5 over the lifecycle of 25 years. The short interval tasks are characterized by intervals between 80 and 1000 FH. The intervals of the medium interval tasks range from 4,000 to 14,000 FH.

It is assumed that the 6- and 12-year heavy maintenance checks (former IL-/D-check) will persist as block check events. As a consequence, an interval extension of one task of a heavy maintenance check does not lead to an interval escalation of the total check, unless the intervals for all tasks of the checks are being extended accordingly.

### Table 5. Equivalence maintenance program – Part 1 (equalized check events).

<table>
<thead>
<tr>
<th>TCG</th>
<th>Short interval</th>
<th>Medium interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MH</td>
<td>Ratio</td>
</tr>
<tr>
<td>Routine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1,898</td>
<td>8.4 %</td>
</tr>
<tr>
<td>2</td>
<td>2,451</td>
<td>10.9 %</td>
</tr>
<tr>
<td>3</td>
<td>1,193</td>
<td>5.3 %</td>
</tr>
<tr>
<td>4</td>
<td>8,798</td>
<td>39.1 %</td>
</tr>
<tr>
<td>Non-routine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3,568</td>
<td>15.9 %</td>
</tr>
<tr>
<td>0</td>
<td>4,597</td>
<td>20.4 %</td>
</tr>
<tr>
<td>Sum</td>
<td>22,505</td>
<td>100 %</td>
</tr>
</tbody>
</table>

Analysis of long interval tasks (6-/12-year check tasks and other tasks with intervals longer than generic C-check interval) show that about 89 % account for TCG 1 to 3, which could be subject to task elimination. Only 9 % of the tasks account for TCG 4, which could be subject to interval escalation. The following analysis considers in connection with the block check events only the potential PHM impact of task redundancy, which accounts for almost 90 % of the routine work. The part 2 of the modeled equivalence maintenance program is summarized in Table 6.

### Table 6. Equivalence maintenance program – Part 2 (remaining block check events).

<table>
<thead>
<tr>
<th>TCG</th>
<th>IL-Check</th>
<th>D-Check</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MH</td>
<td>Ratio</td>
</tr>
<tr>
<td>Routine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>941</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1,092</td>
<td>89 %</td>
</tr>
<tr>
<td>3</td>
<td>5,963</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>821</td>
<td>9 %</td>
</tr>
<tr>
<td>other</td>
<td>183</td>
<td>2 %</td>
</tr>
<tr>
<td>Sum</td>
<td>9,000</td>
<td>100 %</td>
</tr>
<tr>
<td>Non-routine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2,500</td>
<td>50 %</td>
</tr>
<tr>
<td>0</td>
<td>2,500</td>
<td>50 %</td>
</tr>
<tr>
<td>Sum</td>
<td>5,000</td>
<td>100 %</td>
</tr>
</tbody>
</table>

The applied generic modeling approach allows the comparison of a current maintenance program with any potential or future maintenance program without having
described all maintenance tasks precisely. Particularly in early design stages of new aircraft, the proposed methodology could be a viable option to estimate the impact of alternative maintenance concept early on.

4.1.3. Operational Boundary Conditions

A summary of the relevant economic input data used in the analysis is given in Table 7. Assumed ticket prices for economy (EC) and business class (BC) influence airline revenues in the lifecycle CBA.

Table 7. Summary of economic and operational data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Fiscal year</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket price - EC</td>
<td>[US$]</td>
<td>2008</td>
<td>111</td>
</tr>
<tr>
<td>Aircraft price $C_0$ (incl. 35% discount)</td>
<td>[Mio. US$]</td>
<td>2008</td>
<td>50</td>
</tr>
<tr>
<td>Labor rate (maintenance)</td>
<td>[US$/MH]</td>
<td>2009</td>
<td>70</td>
</tr>
<tr>
<td>Fuel price (fuel price scenario)</td>
<td>[US$/gal]</td>
<td>2013</td>
<td>2.49</td>
</tr>
<tr>
<td>Delay cost</td>
<td>[US$/min/pax]</td>
<td>2009</td>
<td>0.63</td>
</tr>
<tr>
<td>Average inflation</td>
<td>[1/year]</td>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>Discount rate $r$</td>
<td>[-]</td>
<td></td>
<td>0.08</td>
</tr>
<tr>
<td>Calibration factor revenues</td>
<td>[-]</td>
<td></td>
<td>0.929</td>
</tr>
</tbody>
</table>

The initial investment cost $C_0$ is assumed as 50 Mio. US$ (aircraft list price in 2008 less an assumed price discount of 35 %). This study does not provide cost estimates for the development and implementation of PHM systems. The goal is to derive maximum acceptable investment costs for PHM systems from the analysis results. Therefore, no additional fix costs for an airplane equipped with PHM are considered. The delay costs of 0.63 US$/ per passenger per minute include costs of passenger compensation and rebooking for missed connections, but also considers the costs of potential loss of revenue due to future loss of market share as a result of lack of punctuality (Eurocontrol, 2007). The internal rate of return $r$, which is used for the discounted cash-flow calculation, is assumed at 8 %. The reference aircraft (see 3.2) has been calibrated with a calibration factor of 0.929 affecting the ticket revenues to an airline internal rate of return of 12 % after 10 years of operation.

4.2. Parameter Variation

The prognostic and CBM concepts to be evaluated in this study are not implemented in commercial aircraft yet. Thus, it is difficult to estimate actual performance characteristics of such concepts on aircraft operational level today. By conducting parameter variations it is possible to analyze the sensitivities of selected parameters with regard to the benefits of an operator’s point of view.

The five selected parameters and their values are depicted in Table 8. The parameter $P_{UEP}$ (“unscheduled event prevention”) describes the portion of component or subsystem failures for which a specific prognostic system can report imminent failures, without consideration of false alarms and missed failures (see also section 3.4). $P_{UEP}$ can range from 0 to theoretical 100 percent, which means that the respective percentage of the total number of impending failures of the 15 selected subsystems will be predicted. To limit the computing times, the $P_{UEP}$ rates for each of the 15 subsystems are assumed to be identical in all analyses of this study. The parameter $P_{FA}$ (“false alarms”) is defined as a probability of occurrence per FH. The “missed failure rate” ($P_{MF}$) is modeled as a ratio of failure event covered by the PHM system. The “task redundancy” rate ($P_{TR}$) is the percentage of preventive maintenance tasks that can potentially be eliminated if a PHM system is used to monitor the respective item (see also section 4.1.2). The “interval escalation” rate describes the factor by which preventive maintenance intervals may be extended if the corresponding item is monitored by a PHM system.

Table 8. Parameter space for analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{UEP}$</td>
<td>unscheduled event prevention</td>
</tr>
<tr>
<td>$P_{FA}$</td>
<td>false alarms [1/FH]</td>
</tr>
<tr>
<td>$P_{MF}$</td>
<td>missed failure rate</td>
</tr>
<tr>
<td>$P_{TR}$</td>
<td>task redundancy</td>
</tr>
<tr>
<td>$P_{IE}$</td>
<td>interval escalation</td>
</tr>
</tbody>
</table>

It is important to mention that the parameters $P_{UEP}$ and $P_{TR}$ are modeled independently although an actual PHM system may contribute to both underlying benefits.

The parameter space as defined in Table 8 results in 3,750 separate analyses (for a full factorial experiment), which have been conducted. In this study, each analysis consists of 100 independent simulation runs (Monte Carlo simulations) to account for the stochastic behavior of the unscheduled maintenance module (due to the probabilistic modeling of the component failure behavior and the PHM impacts). The number of Monte Carlo simulations can be understood as a number of simulated aircraft in a fleet, while certain interdependencies within the fleet (e.g. competition of a number of aircraft for limited maintenance capacities) are neglected in this study. Each simulated aircraft comprises an individual failure behavior. The arising variances of analysis results are discussed at the end of section 4.3.
4.3. Analysis Results

The performed analyses provide technical-operational and economic results. All results describe values for the operative lifecycle of a single aircraft. The impacts of PHM on unscheduled maintenance and aircraft operation are shown first. Then, the economic results from an airline perspective are presented. An impression of the variance of the simulated results due to the applied probabilistic modeling approach will be given at the end of this section.

Figure 7. Unscheduled component removals depending on $p_{UEP}$ and $p_{FA}$ ($p_{MF} = 0$).

Figure 7 depicts the impact of PHM on the number of unscheduled component removals in a/c lifecycle depending on the prognostic performance. In this study, a use of PHM leads to a reduction of unscheduled events from 5,400 to 4,250 in the optimal case (i.e. use of a perfect PHM). Depending on the false alarm rate, the reduction can be smaller or the number of removals can even increase to in case of very values of $p_{FA}$. The reduction of NFFs leads to a decrease of the number of total events, while false alarms cause additional removals. Possible missed failures have no effect on the number of component removals.

As mentioned before, an unscheduled event results in a technical delay, when a failure (or NFF) is not covered by a PHM system, the MEL is not fulfilled, and MTTR exceeds the available time during a/c turn-around. It can be seen from Figure 8 that the number of technical delays can be reduced by 420 in the best case. But if false alarm rates are very high, the number of delays can increase. Combinations of very high $p_{FA}$ and relatively low prevention rates of unscheduled events (e.g. $p_{UEP} = 0.25$) seem to be critical. In these cases, an unreliable prognostic algorithm induces a high amount of subsequent failures (and thereby potential delays) not covered by PHM.

Figure 8. Impact of PHM false alarms and missed failure rate on technical delays
(solid line: $p_{MF} = 0$, dashed line: $p_{MF} = 25\%$).

As outlines in the beginning, a central goal of a PHM and CBM implementation is to improve the aircraft availability in order to increase the utilization. Both effects, the reduction of unscheduled events and the elimination of redundant tasks, can contribute to higher aircraft utilization. Figure 10 shows that – even without a change in the aircraft operation concept – up to 675 additional flight cycles could be realized in aircraft lifecycle.

Figure 9. Impact of task redundancy ($p_{TR}$) and unscheduled event prevention ($p_{UEP}$) on aircraft utilization.

Under the assumptions of this study, the avoidance of unscheduled events enables up to 485 additional flight cycles. Another 190 flights can be realized by shortening the maintenance downtimes for IL- and D-Checks in case of $p_{TR} = 1$. This picture changes in the case of high false alarm rates as shown in Figure 10.
Figure 10. Impact of PHM false alarms and missed failure rate on aircraft utilization for $p_{TR} = 0$ (solid line: $p_{MF} = 0$, dashed line: $p_{MF} = 25\%$).

While total operating and maintenance cost in a/c lifecycle can increase due to an increase in utilization, an appropriate metric to evaluate the effect of PHM is direct maintenance cost (DMC) per FH. Figure 11 shows the potential reduction of DMC/FH for a varying $p_{TR}$ and different $p_{UEP}$.

The introduction of a CBM concept has influences on the amount of maintenance man-hours performed on a planned basis. Figure 12 and Figure 13 show the impacts of a variation of the parameters $p_{TR}$ and $p_{IE}$ on man-hours for equalized maintenance events (planned in AIRMAP). The absolute level of man-hours at $p_{Cov} = 1$ (Figure 13) is about 17,000 hours higher (over the lifecycle) than at $p_{Cov} = 0$ (Figure 12). The component removal events covered by PHM are responsible for this different level of man-hours. The shape of the curves is identical in both cases. In the reference case (without PHM), this workload has to be carried out instead on a reactive basis.

The following figures describe the highest aggregated economic results of the presented study. The monetary benefit of an aircraft operator, expressed as NPV, is shown for different variations and combinations of the five selected parameters.

Figure 14 presents the impacts of $p_{UEP}$ on NPV for different $p_{MF}$ in combination with $p_{TR} = 0$ and $p_{TR} = 1$. The range of NPV improvements can vary by around 3 million US$ in this case. An extremely unreliable prognostic system that produces high numbers of false alarms can cause tremendous extra maintenance costs and reduced aircraft utilization with corresponding decreases of the operator NPV (Figure 15).
Figure 14. Impact of unscheduled event prevention ($p_{UEP}$) and missed failures on NPV (solid line: $p_{TR} = 0$, dashed line: $p_{TR} = 1$).

The simulated results for all variations of $p_{UEP}$, $p_{TR}$, and $p_{IE}$ are shown in Figure 16. Each of the five parts of the figure shows the impacts of the task redundancy rate and the interval escalation factor on airline NPV with the respective PHM coverage rate. It can be seen that the maximum benefit of an interval escalation (i.e. the difference of NPV for $p_{IE} = 0\%$ and $p_{IE} = 100\%$ in each subfigure) accounts for around 0.85 million US$. The maximum overall increase of NPV that could be realized under given assumptions is 5.6 million US$ (as depicted in Figure 16 e). Although it is highly unlikely that a PHM-coverage of 100% for the selected systems could be achieved at an acceptable price, the results show the range of potential benefits. The increase in NPV by a certain PHM/CBM configuration is at the same time the upper limit of the acquisition cost of such a system.

Figure 15. Impact of unscheduled event prevention ($p_{UEP}$) and prognostic missed failures on NPV (solid line: $p_{TR} = 0$ and $p_{IE} = 0$, dashed line: $p_{TR} = 1$ and $p_{IE} = 1$).

Figure 16. Impact of unscheduled event prevention ($p_{UEP}$), task redundancy ($p_{TR}$), and interval escalation rates ($p_{IE}$) on NPV.
which could be accepted.

Since each of the analysis results is a mean value of 100 simulations the values may be subject to significant variances. A graphic evaluation of the variance of the simulation results indicates a relatively high selectivity of the individual analyses. This means e.g. that a PHM system with $p_{UEP} = 0.5$ very likely leads to less component removals than a system with $p_{UEP} = 0.25$ even in a realistic operational scenario. The distributions of the number of component removals – as depicted in Figure 17 – show small overlaps between the different values for $p_{UEP}$.

![Figure 17. Variation of simulated unscheduled component removals for different PHM coverage rates.](image1)

Figure 17. Variation of simulated unscheduled component removals for different PHM coverage rates.

![Figure 18. Variation of simulated airline NPVs for different prognostic false alarm rates (for $p_{UEP} = 0.5$).](image2)

Figure 18. Variation of simulated airline NPVs for different prognostic false alarm rates (for $p_{UEP} = 0.5$).

Figure 18 shows again relatively small variances of the simulated results. But a significant overlapping of the results exists for the perfect PHM and the smallest false alarm rate. That means it cannot be guaranteed that an aircraft equipped with the (theoretical) perfect PHM will perform better over the lifecycle than a system with a small false alarm rate. A considerable overlapping of the results can be observed also for the simulated NPV values depending on different $p_{UEP}$ as shown in Figure 19.

![Figure 19. Variation of simulated airline NPVs for different PHM coverages (perfect PHM).](image3)

Figure 19. Variation of simulated airline NPVs for different PHM coverages (perfect PHM).

It becomes clear that due to the many stochastic factors acting here (and in reality), a considerable uncertainty exists with regard to the value of a PHM use that can be expected in a real-world aircraft operation.

5. CONCLUSION AND OUTLOOK

In this paper we have presented an integrated approach to model the impacts of PHM and CBM planning from an aircraft lifecycle perspective considering prognostic performance levels and errors. The integration of the CBM planning approach in a lifecycle cost-benefit model allows the economic assessment of a PHM and CBM implementation in future aircraft. The application of the assessment approach can deliver valuable requirements for the future development of PHM and CBM concepts and demonstrate its consequences for operators and MROs.

The analysis results show that benefits by a PHM implementation can only be expected, if a very detailed examination is made. Especially high false alarm rates have the potential to cause an economic deterioration compared to the reference system.

Since the general assessment approach is generic in nature, it can be adapted to all kinds of technologies and types of aircraft. For the analysis of a different aircraft type, AIRTOBS must be configured with the aircraft and operator specific XML-file. Furthermore, the corresponding maintenance program and the failure behavior of the systems under consideration must be provided.

At present, the assessment approach is limited to a single aircraft analysis. An extension of AIRTOBS on a fleet-level
is in preparation to allow full use of the maintenance planning and optimization approach implemented in AIRMAP, i.e. scheduling maintenance tasks and planning capacities for a fleet of different aircraft types on an airline’s network. While currently one aircraft lifecycle is simulated at the time, an analysis on fleet level requires the simultaneous simulation of multiple aircraft in order to capture the interdependencies within the fleet. An analysis on a fleet-level allows an even more realistic assessment of PHM, while it is expected that this will result into a lower economic benefits per aircraft. This is because several aircraft compete for limited maintenance resources, leading to less efficient solutions of the CBM planning process.

ACKNOWLEDGEMENT

This study was carried out as part of the work in OMAHA, a research project supported by the Federal Ministry of Economic Affairs and Energy in the national LuFo V program. Any opinions, findings and conclusions expressed in this document are those of the authors and do not necessarily reflect the views of the other project partners.

Supported by:

on the basis of a decision
by the German Bundestag

NOMENCLATURE

AIRTOBS Aircraft Technology and Operations Benchmark System
CBA cost-benefit analysis
CBM condition-based maintenance
DMC direct maintenance cost
DOC direct operating cost
FC flight cycle
FH flight hour
LCC life cycle cost
LRU line replaceable unit
MEL minimum equipment list
MH man-hours
MPD maintenance planning document
MRO maintenance, repair, and overhaul
MTTR mean time to repair
NFF no fault found
NPV net present value
PDF probability density function
PHM Prognostics and Health Management
ROI return on investment
RUL remaining useful life
XML Extensible Markup Language

REFERENCES


**Biographies**

**Nico B. Hözel** is a research engineer at the German Aerospace Center (DLR) within the Institute of Air Transportation Systems located at the Hamburg University of Technology, Germany. Since February 2009, he is a member of the system analysis group and works on the operational and economic assessment of PHM technologies and aircraft maintenance concepts. From 2006 to 2009, he was a Senior Consultant in the automobile industry in Germany. He received his Diploma degree in Industrial Engineering and Management from Hamburg University of Technology in 2006.

**Volker Gollnick** holds the chair and directorate of the Institute for Air Transportation Systems of the German Aerospace Center (DLR) at the Hamburg University of Technology (TUHH). He holds a Diploma degree in Mechanical and Aerospace Engineering from the Technical University of Braunschweig and a Ph.D. degree from the Technical University of Munich. In his research he is in particular interested in operational improvements and changes of sustainable but efficient air transportation systems. Integration and collaboration of the various stakeholders is his special focus, when he is researching on new aircraft, airports and aircraft operations. He represents more than 20 years of experience in aerospace industry and government. He held responsible roles in particular in flight testing, overall aircraft design, cockpit and avionics systems at EADS and its subsidiaries and the German Forces Flight Test Center.
Cost-Wise Readiness Enabled Through Condition Based Maintenance Plus (CBM+)

Josh Kennedy1, Jim Carter2, and Matt Carter3

1Associate Director, U.S. Army Aviation & Missile Command (AMCOM) Logistics Center (ALC), Supportability and Sustainment Directorate (SSD), Sustainment Optimization & Analysis (SOA), Redstone Arsenal 35808 AL, USA

2Aviation & Missile Research, Development & Engineering Center (AMRDEC); Reliability, Availability and Maintainability (RAM) Division RSA 35808 AL, USA

3ALC SSD SOA Contractor Support Staff and Employee Owner at Modern Technology Solutions, Inc.; 360 Quality Circle Huntsville, AL 35806

matthew.c.carter2@us.army.mil; matt.carter@mtsi-va.com

ABSTRACT

As Department of Defense (DoD) budgets continue to decrease through automatic spending cuts, Army Commands are pressured to develop, implement and manage new ways to reduce spending. The high cost of operation and sustainment (O&S) associated with the helicopters required to support the US Army’s global presence significantly increases this pressure. Reducing costs within O&S activities, while managing operational readiness is achieved through Cost Wise Readiness (CWR) initiatives. Goals and objectives are to increase efficiencies, thereby increasing the value of each budgeted dollar. Even as the budgetary environment becomes more challenging, the purpose of Army maintenance remains unchanged—to generate combat power. In support of continuing this capability, Army Aviation is leading the way with ongoing efforts to implement, measure and communicate efficiencies leading to benefits. The AMCOM Logistics Center (ALC) functions as the logistics component of the US Army’s Aviation and Missile Life Cycle Management Command (AMCOM) headquartered at Redstone Arsenal, Alabama. The ALC develops, acquires, fields and sustains logistics support for Army Aviation and Missile systems and associated support equipment to ensure weapon system readiness in any operation worldwide. The ALC, in support of Program Executive Offices, Project Managers, Army Depots, and partnering with industry are dedicated to provide real-time logistics support to the Soldier, Airman and Marine in training and combat. The ALC is dedicated to the development and implementation of CWR initiatives through the identification and pursuit of opportunities and investment in projects focused on reducing cost. Multiple Army offices have been instrumental in the development of technological capabilities in support of the CWR mission. One such high-tech capability includes the integration of systems which incorporate Condition Based Maintenance Plus (CBM+) into the management of logistics and airworthiness aspects of the Army’s helicopter fleets. Managing costs within O&S activities is achievable through the remediation of maintenance enabled through CBM+ initiatives.

1. INTRODUCTION

The challenge of declining defense budgets require the development of exceptional techniques to cut O&S costs. Overseas Contingency Operations (OCO) funding continues to be reduced from prior enacted levels. CBM+ supports the automation of monitoring the condition of certain components, therefore allowing for noteworthy remediation of oftentimes very conservative time based parts replacement practices. The technology enables significant cost benefits by facilitating time between overhaul and retirement change life limit extensions authorized for certain components. This supports reduced component replacement frequency, thereby cutting material costs, while enhancing mission readiness and efficiency by decreasing the Warfighter’s maintenance burden by automating routine inspections. This paper details how results of implementing CBM+ initiatives are calculated. The Army’s use of CBM+ technology, chiefly by the Apache PMO and its AH-64 aviation units, has supported the goal of CWR as a top objective of the ALC. One goal of the Supportability and Sustainment Directorate (SSD), Sustainment Optimization & Analysis (SOA) - Assessment Division is to support this mission by substantiating CWR benefits.

2. BACKGROUND - ISSUES BEHIND COST DRIVERS

In the interest of airworthiness, and to keep baseline risk well beneath accepted levels, maintenance procedures require multiple and frequent condition inspections, Maximum Allowable Operating Time (MAOT) and Time between Overhaul (TBO) intervals. The maintenance interval requirements are oftentimes extraordinarily conservative, as they are based upon preserving safety margins which were defined long before CBM monitoring systems were developed or installed on legacy aircraft. Therefore, in order to act according to established requirements, it has long been compulsory to take maintenance actions on equipment prior to MAOT and/or TBO before any evidence of need exists.
Regardless of the actual remaining useful life, for decades, extraordinarily conservative safety margins have imposed millions in added costs to O&S activities throughout the Army’s helicopter fleets.

One cost driver was the prior, ultraconservative MAOT of the AH-64D Main Transmission Accessory Sprag Clutch. It remained particularly high for well over a decade after trouble was caused by the dual failure of both the primary and the secondary Accessory Gearbox Sprag Clutches. The clutches experienced unanticipated early wear each triggering critical failure modes.

Figure 1. Transmission Accessory Gearbox Sprag Clutch

AH-64D Main Transmission Accessory Sprag Clutches provide the mechanical power input to critical accessory systems. The TBO of the transmission, to include its clutches was originally greater than 2000 hours. However the clutches were knocked down to a much more conservative 1000 hour MAOT limit following an airworthiness determination made in the late 1990s. This constricted life limit significantly increased O&S costs over the years since. In accordance with Army Regulation 700-82, the clutch is only replaceable at the depot maintenance level. Therefore, the main transmission had to be removed and replaced every 1000 hours, elevating maintenance burden fleetwide. Figure 1 above illustrates the affected areas that have historically caused limited life.

In July 2011, an additional the 1000 hour TBO was increased to 1250 hours, and again from 1250 to 1500 hours in April 2013. Substantial benefits have resulted from the increases. Additional AH-64 Apache components granted similar TBO and life extensions are also included in this manuscript. Each extension has been measured similarly and were each implemented through the outstanding efforts of all US Army Aviation offices which collaborated to execute the efforts to reduce burden and cost.

3. O&S Cost Mitigating Remediation Projects

Achieving authorization to extend MAOT and TBO intervals of components requires well-structured collaboration between the several Army organizations. Included in these efforts were the AH-64D Apache Helicopter, MSPU, and components’ original equipment manufacturers (OEM), the Aviation and Missile Research Development and Engineering Center (AMRDEC) Aviation Engineering Directorate (AED) and Engineering Directorate (ED), the Apache Attack Helicopter (AAH) Program Management Office (PMO), the Program Executive Office-Aviation (PEO-A), along with other Team Redstone offices, including Redstone Test Center (RTC) and the AMCOM G3 CBM Office. Each office was responsible for key elements which have led to several highly valuable fleet wide extension authorizations. The University of S. Carolina (USC) CBM Test Center and the S. Carolina Army National Guard (SCARNG) provided fundamental support for successes generated. Algorithms were developed through meticulous refinement efforts in accordance with the criteria defined in the ADS-79 Handbook. Multiple teardown inspections were accomplished. This ensures that the quantity of false positives, false negatives and other indications were reduced to a minimum. Teardown inspections were conducted through the coordination of AED, RTC, and the OEM. Each was essential to ensure the fielding of reliable algorithms; the kind of algorithms that maintainers could depend on to provide correct indications regarding the condition of parts, and aircrews could depend on to save lives.

TBO and MAOT imitations were primarily set through the conventional reliability analysis processes, before development of condition monitoring capabilities. The successful fleet wide extension of safe and valuable extensions required iterative processes of data analysis, laboratory testing, followed by limited fielding on actual helicopters. Further analysis was executed to complete detailed test plans. Test fixtures were designed to accommodate the need to develop and validate highly reliable CI’s in accordance with ADS-79. RTC and the USC CBM Test Center were each utilized to execute laboratory and testing and to conduct collaborative tear down analysis (TDA) with AED. TDAs are essential to the development of and validation of CI’s. Teardown evaluations continue through the RIMFIRE process at the Corpus Christi Army Depot. RIMFIRE stands for Reliability Improvement through Failure Identification and Reporting.

4. Application of CBM & Associated Challenges

Army Aviation’s application of CBM onto its multiplatform fleets of helicopters has not been without its various challenges. For the last 10 years, many advocates have led the way and overcome challenges. Some of the early challenges included funding the installation program. With no specific requirement, coupled with limited time and funding, user training has been most challenging.

Proper data management involves very complex tasks. The data must not only be collected and transmitted, but stored and
analyzed in a useful manner so as to provide actionable information to the maintenance officer. As one of the most high tech platform users, Apache Helicopter units have led the way in terms of managing each one of these complicated tasks. The Apache also has the highest percentage of Digital Source Collectors (DSC) installed, and the most mature Condition Indicators (CI) of all Army helicopter platforms. The Aeronautical Design Standard Handbook Condition Based Maintenance System for US Army Aircraft (ADS-79D) defines a CI as “A measure of detectable phenomena, derived from sensors, that shows a change in physical properties related to a specific failure mode or fault.”

CIs are extremely challenging to mature to the required confidence level. The ADS-79D defines this as follows: The probability that a confidence interval contains the true value of a population parameter of interest. When not otherwise specified in this ADS, the confidence level should be assumed to equal 0.9 (or 90%).” Any less and conditions indicated by the CI could be highly questionable or disregarded.

5. MEASURING AND COMMUNICATING CBM+ BENEFITS

In coordination with the Apache Attack Helicopter (AAH) Project Management Office (PMO), the Army Aviation and Missile Command (AMCOM) Logistics Center (ALC) has established the Post Implementation Assessment (PIA) capability. The methodology provides the Army with a repeatable technique to measure implemented projects’ tangible and traceable benefits. The methodology measures how CBM technology has enabled significant increases in efficiency, supplementing improved operational readiness rates. The dedicated participation and practical employment of the technology by Army Aviation battalions, particularly by the Apache Attack Helicopter Project Management Office and AH-64 aviation units is clearly demonstrated, and has supported ALC in its every day mission and objective to achieve its goal of CWR initiatives. As an example, the team collaborated with other Army Aviation offices to substantiate the benefits from an ongoing CBM project, one which has successfully extended the AH-64D Main Transmission time between overhaul (TBO) and its internal Sprag Clutches’ MAOT limit. CBM benefit metrics from these flight hour extensions have been identified and calculated. The goal of ALC’s Post PIA methodology is to fulfill the requirement to capture and communicate the benefits of CBM+. The PIA methodology does this by identifying the primary known benefit contributors. As with any globally deployed complex vehicles with high cost and maintenance requirements, large fleets of Army Helicopters have multiple efforts working in parallel, each aimed at generating combat power while decreasing soldier burden, demand, and cost through increasing installed components Time on Wing (ToW).

Fleet wide performance metrics of each implemented project are measured regarding: 1) The sum total costs assessed since implementation; 2) The average cost since implementation; 3) Demand reduction per 10,000 flying hours (FH); 4) ROI. All performance metrics are updated quarterly basis to continue measuring performance metrics as extensions remain valid and continue producing statistically significant benefits. Calculations to measure benefits include:

6. DATA ELEMENT DEFINITIONS

6.1. Demands

Demands (D): Item’s quantity of demands at the retail level during a timeframe before and after implementation. Demand data is pulled from the ILAP (Integrated Logistics Analysis Program) database.

6.2. Flight Hours

Flight Hours (FH): The quantity of flight hours flown during the same timeframe before and after implementation. FH data is Department of the Army (DA) Form 1352 data pulled from the Logistics Information Warehouse (LIW) and utilizing the Readiness Integrated Database (RIDB) application.

6.3. Exchange Price

Exchange Price (EP): The item’s price at the unit level for each year in the calculation. The EP includes current pricing as well as prior archived FY pricing (when available) and is pulled from the Logistics Modernization Program (LMP).

6.4. Base Line Time Period

Base Line (bl - appears as lower case in the calculation): The base line time period is two years prior to implementation unless otherwise identified.

6.5. After Implementation Time Period

After Implementation (ai - appears as lower case in the calculation): The time period since implementation, when data becomes statistically significant.

6.6. Rate of Demands

Rate of Demands (RD): The quantity of unit Demands as normalized per 10,000 FH for the Time Period (tp - appears as lower case in the calculation i.e. Baseline Rate of Demands appears as “RDbl”).

6.7. Expected Demands

Expected Demands (ED): A representation of the quantity of unit Demands that would have occurred after implementation using RD_M as calculated through the multiplication of each fiscal year’s (FY) actual FH.

6.8. Average Cost per Flight Hour

Average Cost per Flight Hour (ACFH): The cost calculated by multiplying the sum of each applicable FY’s unit Demand times the EP, divided by the sum of all associated FHs across the same FYs.
6.9. Cost Assessment
Cost Assessment (CA): The value of the monetized benefits yielded in terms of material demand changes as calculated during the specified time period after implementation of a project.

6.10. Additional Time on Wing
Additional Time on Wing (Tow): The quantity of additional flight hours a component remains installed on a system in operation, i.e. through TBO and/or MAOT extensions. Computed using AMCOM Message Tracking System (AMTRACKS), DA 1352, DA 2410 (Component Removal and Repair/Overhaul Record) databases to calculate the sum of actual hours flown above the base line life limit using the criteria in the message extending the life limit.

6.11. Additional Demands Based on Additional ToW
Additional Demands (AdD): A representation of the quantity of additional demands that would occur after implementation using baseline RD\(_{bl}\).

6.12. Adjusted ACFH with Added Demands
Adjusted ACFH with Added Demands (AdACFH): The cost per FH after implementation, adjusted to include the ED.

6.13. Cost Benefit
Cost Benefit (CB): The value of the overall project implementation as it relates directly to the specific change fielded through the project undergoing assessment (e.g. TBO extension), in terms of the change in supply cost during the time period after implementation, as referenced against the baseline time period.

Direct Cost Benefit per Cost Assessment (CB/CA): The percentage of the overall Cost Assessment yielded during the time period after implementation which can be attributed directly to the change implemented by the project undergoing assessment.

6.15 Return on Investment
Return on Investment: A performance measure used to evaluate the efficiency of an investment or to compare the efficiency of a number of different investments. To calculate ROI, the benefit (return) of an investment is divided by the cost of the investment; the result is expressed as a ratio. The PIA process includes two levels or ROI, as explained below:

6.15.1 Overall Cost Assessment ROI
Overall Cost Assessment ROI: The ratio of Demand Cost Assessment minus the Project Investment divided by the Project Investment. Overall Cost Assessment ROI cannot be fully attributed to the project undergoing assessment. Further calculations are required to formulate the project’s Direct ROI, as noted below.

6.15.2 Direct ROI
Direct ROI: The ratio of Cost Benefit minus the Project Investment divided by the Project Investment. Direct ROI is intended to formulate the project’s Direct ROI.

7. Computations with Demonstrative Examples
It should be noted that each example demonstrated below include fictitious figures in terms of demand, flight hours, prices and cost. These fictitious quantities are included below in Figure 2, with the objective is to avoid issues relative to applying actual figures, and the goal is to show how data elements are applied to advanced formulas.

Figure 2. Post Implementation Assessment Example

7.1. Rate of Demands: Base line & After Implementation

7.1.1. Base line Rate of Demands
As an example, using fictional figures, the Baseline (bl) Rate of Demand (RD\(_{bl}\)) before implementation is 13.16 Demands per 10K FH, calculated as follows: D = 325 units from 1 Oct 2010 through 30 Sep 2011 (FY11), and FH = 250,000 for the same period; and for from 1 Oct 2011 through 30 Sep 2012 (FY12): D = 300 units from 1 Oct 2011 through 30 Sep 2012, and FH = 225,000 for the same period.

\[
RD_{bl} = \frac{\sum_{m=1}^{n} D_m}{\sum_{m=1}^{n} FH_m \times .0001}
\]

7.1.2. Rate of Demands After Implementation
As an example, using conceptual figures, the After Implementation (ai) RD (FY13 & FY14) is 10.84 demands per 10K FH, calculated as follows: FY13 D = 250 units and FH = 250,000 for the same period; and for from 1 Oct 2011 through 30 Sep 2012 (FY12): D = 300 units from 1 Oct 2011 through 30 Sep 2012, and FH = 225,000 for the same period.

\[
RD_{ai} = \frac{\sum_{m=1}^{n} D_m}{\sum_{m=1}^{n} FH_m \times .0001}
\]
RD_{ai} = \frac{(250+200)}{(215,000+200,000)} \cdot 0.001 = 10.84 \text{ demands per 10K FH}

7.2. Avg Cost per FH Baseline & After Implementation

7.2.1 Average Cost per FH Average Cost per FH

As an example, using fictional figures, the baseline Average Cost per Flight Hour (ACFH) is $404.61; calculated as follows: FY11 D = 325 units, FH = 250,000 and Exchange Price (EP) = $150,000; FY12 D = 300 units, FH = 225,000 and EP = $157,500:

\[
ACFH_{bl} = \frac{\sum_{y=1}^{n} D_y \cdot EP_y}{\sum_{y=1}^{n} FH_y} \quad (3)
\]

\[
ACFH_{bl} = \frac{(325 \cdot 150,000) + (300 \cdot 157,500)}{(250,000+225,000)} = 404.61 \text{ per FH}
\]

7.2.2 ACFH After Implementation

As an example, using fictional figures, the ai Average Cost per Flight Hour (ACFH) is $292.04; calculated as follows: FY13 D = 250 units, FH = 215,000 and EP = $141,750; FY12 D = 200 units, FH = 200,000 and EP = $127,575:

\[
ACFH_{ai} = \frac{\sum_{y=1}^{n} D_y \cdot EP_y}{\sum_{y=1}^{n} FH_y} \quad (4)
\]

\[
ACFH_{ai} = \frac{(250 \cdot 215,000) + (200 \cdot 127,575)}{(215,000+200,000)} = 146.87 \text{ per FH}
\]

7.3. Expected Demands

As an example, using fictional figures, Expected Demands (ED) for FY13 is 282.9, calculated as follows, when RD_{bl} = 13.16, FY13 FH = 215,000 and FY14 FH = 200,000.

\[
ED_{tp} = \sum_{m=1}^{n} FH_m \cdot (RD_{bl} \cdot .0001) \quad (5)
\]

\[
ED_{ai} = (13.16 \left( \frac{215,000}{10,000} \right)) = 282.89 \text{ (FY13)}
\]

\[
ED_{ai} = (13.16 \left( \frac{200,000}{10,000} \right)) = 260.00 \text{ (FY14)}
\]

7.4. Cost Assessment

As an example, using fictional figures, the Cost Assessment (CA) is calculated as follows, when RD_{bl} = 13.16, FY13 D = 250 units, FH = 215,000 and EP = $141,750; FY12 D = 200 units, FH = 200,000 and EP = $127,575:

\[
CA_{ai} = \sum_{y=1}^{n} \left( RD_{bl} \left( \frac{FH_y}{10,000} \right) - D_y \right) \cdot EP_y \quad (6)
\]

\[
CA_{ai} = (13.16 \left( \frac{215,000}{10,000} \right) - 250) \cdot 141,750 + (13.16 \left( \frac{200,000}{10,000} \right) - 200) \cdot 127,575
\]

\[
CA_{ai} = (282.29 - 250) \cdot 141,750 + (260.00 - 200) \cdot 195,887 = 12,317,329
\]

7.5. Additional Demands Based on Time on Wing (ToW)

As an example, using fictional figures, Additional Demands (AdD) is based on Additional Time on Wing (ToW), calculated as follows, when FY13 Additional ToW is 20,000 FH and RD = 11.63, and when FY14 Additional ToW is 35,000 FH and RD = 10.84. Therefore, 58.3 (23.3+35.0) fewer units were required in FY13 and FY14 as a direct result of the 55,000 Additional ToW FH since TBO was extended.

\[
AdD_{tp} = \frac{ToW_{tp} \cdot RD_{tp}}{10,000} \quad (7)
\]

\[
AdD_{fy13} = \frac{20,000 \cdot 11.63}{10,000} = 23.26
\]

\[
AdD_{fy14} = \frac{35,000 \cdot 10.84}{10,000} = 35.00
\]

7.6. Adjusted ACFH per ToW

\[
AdACFH_{tp} = \frac{(D_{tp} + AdD_{tp}) \cdot EP_{tp}}{FH_{tp}} \quad (8)
\]

\[
AdACFH_{fy13} = \left( \frac{(250 + \left( \frac{20,000 \cdot 23.26}{10,000} \right)) \cdot 141,750}{215,000} \right) = 180.16
\]

\[
AdACFH_{fy14} = \left( \frac{(200 + \left( \frac{35,000 \cdot 35.00}{10,000} \right)) \cdot 127,575}{200,000} \right) = 149.90
\]

7.7. Cost Benefit

As an example, using fictional figures, the Cost Benefit (CB) is calculated as follows:

\[
CB_{tp} = \sum_{y=1}^{n} (ACFH_{bl}) \cdot ToW_{fy} \quad (9)
\]

\[
CB_{fy13} = 180.16 \cdot 20,000 = 3,603,164
\]

\[
CB_{fy14} = 149.90 \cdot 35,000 = 5,246,522
\]
7.8. Cost Benefit per Cost Assessment

\[ CB_{TP} = \frac{CB_{TP}}{CA_{TP}} \]  

\[ CB_{fy13} = \frac{\$3,603,164}{\$4,662,829} = 77.3\% \]

\[ CB_{fy14} = \frac{\$5,246,522}{\$7,654,500} = 68.5\% \]

7.9. Return on Investment

\[ ROI_{TP} = \frac{CB - PI}{PI} \]

\[ ROI_{fy13} = \frac{\$3,603,164 - \$2,000,000}{\$2,000,000} = 0.8 : 1 \]

\[ ROI_{fy14} = \frac{\$5,246,522 - \$2,000,000}{\$2,000,000} = 1.62 : 1 \]

\[ ROI_{15} = \frac{\$9,088,015 - \$2,000,000}{\$2,000,000} = 3.5 : 1 \]

8. ADDITIONAL AH-64 CBM+ ENABLED COMPONENTS

Figure 3. AH-64D Presenting Parts Remediated with CBM+

9. CONCLUSIONS

Maintenance improvements enabled through implementation of CBM+ projects featured in this report have resulted in substantial benefits. However, it is important to note that the results require time and hundreds of thousands of FHs to materialize. While implemented maintenance changes featured in this report include only those applied to the Apache airframe, similar work is being pursued across the Black Hawk and Chinook platforms to support CWR through the reduction of materiel costs.

ACKNOWLEDGEMENT

The authors acknowledge Army Civilians and other fellow defense industry support contractors who provided the tremendously helpful support and guidance to develop and repeat the PIA methodology. Capturing and communicating how CBM+ has enabled significant increases in efficiency is extremely important. This is particularly true as the Army seeks ways to execute CWR with increasing funding constraints. The PIA methodology demonstrates dedicated participation and practical employment of the technology by Army Aviation battalions. The research is supported by AMCOM, AMRDEC AED and ED offices, and the Apache Attack Helicopter Product Support Manager (PSM).

REFERENCES


day2_sessionviii_01_nrel_keller.pdf

Army Regulation, 700-82, Subject: Joint Regulation Governing the Use and Application of Uniform Source Maintenance and Recoverability Codes, 10 October 2007.

https://www.peoavn.army.mil.SitePages/Apache.aspx

BIographies

Josh Kennedy is the Associate Director, ALC SSD SOA with over 24 years of experience in Army operations and unit leadership. Education includes a West Point Engineering B.S. degree, and multiple M.S. degrees, that include Aeronautical Science earned at the Embry-Riddle Aeronautical University (ERAU) and in Operations Research/Human Systems Integration earned at the U. S. Naval Postgraduate School. Recent studies include an Intermediate Level Education from the Army Command and General Staff College. As a results-oriented, Soldier-focused program manager, systems engineer, operations research analyst, and military officer he has a proven record of accomplishment in the fields of DoD weapons systems acquisition, Operations Research/Systems Analysis (ORSA), Human Systems Integration (HSI), strategic planning, analysis, and Test and Evaluation (T&E) efforts for new, multi-billion dollar Army Aviation systems. Josh is also recognized as a subject matter expert in project management and systems analysis regarding its connections to the DoD Systems Acquisition framework.

Jim Carter works as a Senior Reliability Engineer in the AMRDEC RAM Division, specializing in component survivability analysis. He received a Bachelor of Science Degree in Industrial Engineering from the University of Missouri, Columbia.

Matt Carter provides Systems Engineering Technical Assistance (SETA) contractor support to the ALC SSD SOA Assessment Division, and is an employee-owner at Modern Technology Solutions, Inc. (MTSI). For 21 years, experience has included wide ranging engineering support of multiple Army and Naval Aviation programs, including roles such as Sr. Electrical Engineer at the Sikorsky Aircraft Corporation, L-3 Communications, Raytheon, and Sr. Systems Engineer at
Camber, in support of the AED Cargo Helicopter Division. Education includes an M.S. Degree in Aeronautical Science with a specialty in Management, earned at ERAU in Daytona, FL, USA.
Cyber Defense of Rotating Machinery Using an Integrated ‘Fuse’ Bearing

Evyatar Cohen¹, Jacob Bortman² and Renata Klein³

¹² Pearlstone Center for Aeronautical Engineering Studies and Laboratory for Mechanical Health Monitoring, Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer Sheva 84105, Israel. 
cohenavel@gmail.com 
jacobt@gmail.com

³R.K. Diagnostics, Gilon, P.O.B. 101, D.N. Misgav 20103, Israel 
Renata.Klein@RKDiagnostics.co.il

ABSTRACT

A new concept is proposed for protection against cyber-attacks aiming to create excessive loads that will eventually result in irreversible damage to critical rotating machines. This novel approach is used as an additional defense layer of cyber protection to prevent hostile entities from breaking into the control system of the critical machines.

A relatively small bearing is used as the "weak link", or the "fuse" mechanism, in the critical system. This mechanism allows rapid life degradation under harmful regimes, which leads to early detection of attack and finally to prevention of catastrophic damage to a critical machine. The detection of the fuse degradation process is based on techniques of machine health monitoring via vibrations signatures.

An analytical model was developed, allowing to design the 'fuse bearing'. The model examines the response of the fuse bearing, through its life degradation rate, by simulating a wide range of attack scenarios. The model integrates sub models: bearing life estimation models and a dynamic response of mechanical rotating machines model. In addition, a set of fatigue life experiments were conducted on a designated experimental test facility with the purpose of proving the early damage detection ability of the fuse bearing using vibration analysis.

1. INTRODUCTION

Many mechanical facilities in industry are defined as "critical", such as turbines in power plants, centrifuges for uranium enrichment, big pumps in water companies, etc. These machines work many hours continuously, and therefore require careful maintenance. These facilities are designed to work at limited regimes of speeds and loads. However, exceeding these regimes can lead to failures created by fatigue and rubbing, and finally to catastrophic damage and destruction.

Industries tend to operate these machines in safe conditions and try to extend the machine lives by frequent maintenance. Usually, these machines are controlled by a control system that help provide better supervision. These "critical" machines are strategic targets of different kinds of cyber-attacks. Most of the attacks try to damage the mechanical parts and cause irreversible damage. So far, most protection mechanisms are based on software and firewalls trying to prevent the penetration of harmful elements into control systems. But in some cases, these protections are not enough.

In this research, a new concept is proposed for mechanical protection against cyber-attacks on critical rotating machines (Figure 1). The proposed approach integrates a fuse mechanism into the critical rotating machine. This fuse mechanism is simply a relatively small ball bearing, designed to be the weakest component in the machine. It is expected that under attack or abuse, the fuse mechanism will be damaged first, ahead of the other critical components of the machine. In this case, the fast life degradation and early failure will lead to early detection of attack and finally to the prevention of catastrophic damage to a critical machine.

The detection process of fuse bearing failure caused by cyber-attack is based on techniques of machine health monitoring via vibrations signatures. Advanced signal processing and feature extraction methodologies are applied for initial failure detection and degradation tracking.

As a part of this research, an analytical model simulating the wear and failure of a fuse bearing under a wide range of attack scenarios was developed. Statistical models for bearing life estimation were used to build this model. The proposed
model helped us to design and to analyze the fuse system. Actually, life prediction and degradation simulations of the fuse bearing and the machine critical bearings at each attack scenario were performed, in order to define the proper requirements for an effective fuse system. As described in the conceptual diagram (Figure 1), an external load can be applied onto a fuse bearing with load level control set by a function (defined as a "Transfer Function", TF) of some attack parameters. This external load helps to accelerate life degradation and failure of the fuse bearing in case of attack. To reduce the possibility of taking over the fuse system by hostile elements, it is designed to operate autonomously and it is not controlled by a computer. Three optional TFs are examined and discussed in order to find the optimal TF that brings the fastest fuse life degradation rate for specific attack simulations.

Carried out, so the results and discussion on the applied signal analysis techniques will be presented separately.

2. Model description
In this chapter, a model that allows the running of multiple simulations of attack on a critical rotating machine and simulates its dynamic behavior and its effect on bearing fatigue, is presented. In addition, the model helps to design the fuse system and to analyze its efficiency at multiple attack scenarios. Please note that high efficacy of a fuse system is expressed as fast mechanical life degradation of the fuse bearing during an attack (which is estimated by life models), relative to the degradation rate at normal operation. A block diagram showing the structure of the overall model is depicted in Figure 2.

Figure 2. Model block diagram.
Mathematical expressions and models that describe each block in Figure 2 are listed and detailed in the next sections.

2.1. Critical rotating machine dynamics modeling
A dynamic model of a rotating machine with a single mass rotor (Meirovitch, 2001), which appears as the "roto dynamics model" block in Figure 2, is described in this section. A critical rotating machine is assumed to involve a heavy rotating disk, known as a rotor, attached to a flexible shaft mounted on rigid bearings. Typical examples are turbines, compressors, centrifuges, etc. It is assumed that the rotor has some eccentricity and, as a result, the rotation produces a centrifugal force causing the shaft to bend. For certain rotational velocities, the system experiences violent vibrations. Figure 3 shows a shaft rotating with angular velocity \( \omega \). The shaft carries a disk of total mass \( m \) and is assumed to be massless. Hence, the motion of the system can be described by the displacements \( x \) and \( y \) of geometric center \( S \) of the disk.

Figure 1. System defense scheme concept.
The regime of a critical rotating machine is defined by a set of parameters that are usually under supervision of a control system (for example, rotating speed). In terms of a cyber-attack, these parameters are called "parameters of attack". It is assumed that the attacker controls these parameters maliciously to bring the critical system to harmful regimes. Also, the parameters of attack may vary with time. Hence, the chosen bearing life models are applicable for variable operating conditions.

Endurance experiments were conducted on a fuse bearing, in a test facility that was designed for this research in the lab. These experiments were designed to test the feasibility of the protection by the fuse mechanism concept, i.e. to prove the ability to cause the fuse bearing failure and to detect early the damage through vibration analysis (to define the appropriate condition indicators). This paper focus mainly on the model development and the presentation of simulation results. The experiments of the fuse bearing fatigue life tests are still...
The equations of motion in x and y directions, as depicted in Meirovitch's book, can be described as follows:

\[
\begin{align*}
\ddot{x} + 2\zeta_x \omega_n \dot{x} + \omega_n^2 x &= e \omega^2 \cos(\omega t) \\
\ddot{y} + 2\zeta_y \omega_n \dot{y} + \omega_n^2 y &= e \omega^2 \sin(\omega t)
\end{align*}
\]

(1)

The most common case is that the shaft stiffness is the same in both directions, \(k_x = k_y = k\). In this case, the two natural frequencies coincide and so do the viscous damping factors:

\[
\zeta_x = \zeta_y = \zeta = \frac{c}{2m\omega_n}; \; \omega_{nx} = \omega_{ny} = \omega_n = \frac{k}{m}
\]

(2)

Since the equations of motion are linear and not coupled, the solution in both directions is:

\[
\begin{align*}
x(t) &= |X(\omega)| \cos(\omega t - \phi) \\
y(t) &= |Y(\omega)| \sin(\omega t - \phi)
\end{align*}
\]

(3)

\(|X(\omega)|, |Y(\omega)|\) and \(\phi\) are detailed in Appendix A. As a result of \(x(t)\) and \(y(t)\) motions, the bearing reactions, according to Newton's second law, are described as:

\[
F_x = m\ddot{x}/2 = m\omega^2 |X(\omega)| \cos(\omega t - \phi) + e \cdot \cos(\omega t)/2
\]

(4)

\[
F_y = m\ddot{y}/2 = m\omega^2 |Y(\omega)| \sin(\omega t - \phi) + e \cdot \sin(\omega t)/2
\]

The equivalent load amplitude can be defined as:

\[
F_{OS} = \sqrt{F_x^2 + F_y^2}
\]

(5)

A displacement amplitude \(|G(i\omega)|\) (Appendix A) is plotted in Figure 4 over the frequency ratio \(\omega/\omega_n\) for different values of damping ratio. For very low frequencies, the amplitude ratio is nearly zero since the unbalance forces are small. As the shaft speed increases, the amplitude shows a large peak near \(\omega/\omega_n = 1\), when \(\omega\) is near the resonance frequency of the system. As discussed above, there is a direct relationship between bearing loads and rotor disk displacements; therefore, the load amplitude \(F_{OS}\) over the frequency behaves similarly to the amplitude displacement \([G(i\omega)]\).

Figure 4. Amplitude response over frequency. (Meirovitch, 2001).

Under the assumption that the rotor disk does not affect the stiffness of the massless shaft, the lateral bending stiffness at the axial center of a simply supported uniform beam is given by:

\[
K_s = \frac{48EI}{L^4}
\]

(6)

where \(E\) is the elastic modulus of the shaft, \(L\) is the length between the bearings, and \(I\) is the shaft area moment of inertia, which can be expressed as follows:

\[
I = \frac{\pi D^4}{64}
\]

(7)

where \(D\) is the shaft diameter.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M) [kg]</td>
<td>25</td>
<td>rotor disk mass</td>
</tr>
<tr>
<td>(D) [mm]</td>
<td>20</td>
<td>shaft diameter</td>
</tr>
<tr>
<td>(L) [mm]</td>
<td>470</td>
<td>shaft length</td>
</tr>
<tr>
<td>(E) [Mpa]</td>
<td>245000</td>
<td>elasticity modulus</td>
</tr>
<tr>
<td>(e) [mm]</td>
<td>10</td>
<td>eccentricity</td>
</tr>
<tr>
<td>(\zeta)</td>
<td>0.15</td>
<td>damping ratio</td>
</tr>
</tbody>
</table>

Table 1. Dimensions and mechanical data parameters of the experimental test facility

The mentioned model is adapted to the experimental test facility in the lab, which was built especially for this study. Therefore, geometric dimensions and other mechanical data (shown in Table 1) were inserted as inputs to the model. By integrating these parameters to the expressions mentioned above, the lateral bending stiffness was found to be \(K_s \approx 9\).
10^5 [N/m], and the natural frequency of the critical machine is \( \omega_n \approx 30 \) [Hz].

### 2.2. A modified L10 bearing life model with ISO factor for variable operating conditions

In this section, a modified L10 life statistical model with \( a_{ISO} \) factor, which appears as the "life model select" block in Figure 2, is described and developed. An expression for the bearing life is presented (ISO 281, 2007; Harris & Kotzalas, 2006):

\[
L_{ISO} = A_1 A_{ISO} \left( \frac{C}{F_e} \right)^p \tag{8}
\]

where \( L_{ISO} \) is the bearing life in millions of inner ring revolutions, \( C \) is the dynamic load capacity, \( F_e \) is the external equivalent load (combination of axial and radial load), \( p \) is the constant exponent (\( p = 3 \) for ball bearings and \( p = 10/3 \) for cylindrical bearings), \( A_1 \) is the reliability factor and \( A_{ISO} \) is the life ISO factor. If the speed of the inner ring (\( \omega \)) is known, the \( L_{ISO} \) can be expressed in hours:

\[
L_{ISO} = A_1 A_{ISO} \frac{10^6 C}{60 \omega} \tag{9}
\]

The ISO 281 standard (2007) determined that \( A_1 \) factor can be calculated according to:

\[
A_1 = 0.95 \left( \frac{\ln \left( \frac{100}{s} \right)}{\ln \left( \frac{100}{90} \right)} \right) \tag{10}
\]

where \( s \) is the reliability. In most cases, the accepted reliability is \( s = 0.9 \) (90% reliability), where in that case \( A_1 = 1 \). The expression for the modified ISO factor \( A_{ISO} \) is:

\[
A_{ISO} = 0.1 \left[ 1 - \left( x_1 - \frac{x_3}{k^{x_7}} \right)^{e_2} \left( \frac{C_F L_{lim}}{F_e} \right)^{e_3} \right]^{e_4} \tag{11}
\]

where \( x_1, x_2, e_1, e_2, e_3, e_4 \) are constants that can be found in Harris and Kotzalas (2006), \( k \) is the viscosity ratio factor, \( C_F \) is the contamination factor and \( F_{lim} \) is the load limit that can be found in bearing catalogues (expressions for \( k, C_F \) are presented in Appendix B).

In the case that the load and speed profiles are not constant during the bearing operation, the life calculation with Equation 8 or Equation 9 is no longer valid, since it calculates bearing life only with fixed-in-time parameters. However, with the Miner method (1945), an equivalent life-time with variable operating conditions can be calculated. The Miner rule states that where there are \( k \) different load magnitudes in a spectrum, each contributing \( n_i \) cycles, then if \( L_i \) is the number of revolutions to failure of the \( i^{th} \) constant load, a failure will occur when:

\[
\sum_{i=1}^{k} \frac{n_i}{L_i} = 1 \tag{12}
\]

In other words, when the damage accumulation reaches 1, there is a fault in the bearing. In the case that load and speed of bearing are periodic in time, a ‘Duty Cycle’ can be defined as load or speed profiles in one cycle with length \( T \) (in time) or length \( N_{DC} \) (in revolutions). An expression for equivalent life can be written (Budynas & Nisbet, 2008):

\[
L_{eq} = \frac{1}{U_1 + U_2 + \ldots + U_i} \tag{13}
\]

The methodology assumes the loading history to be composed of thin slices of loading condition, and within each loading slice the rotation speed may be different from others. The ratio between the number of revolutions required for the \( i^{th} \) load \( (F_i) \) and the number of revolutions in one complete Duty Cycle is defined as a non-dimensional life fraction, which is calculated as follows:

\[
U_i = \frac{n_i}{N_{DC}} = \frac{\omega_i \Delta t_i}{\sum_{i=1}^{k} (\omega_i \Delta t_i)} \tag{14}
\]

### 2.3. Life model by Tallian

A rolling bearing spalling fatigue life model (Tallian, 1999), which appears as the "life model" block in Figure 2, is described and developed in this section. This model covers rolling contacts under operating conditions varying along the rolling track and time-variable conditions.

The model recognizes three families of defects in the stressed volume: (a) subsurface defects, (b) local surface defects and (c) the "surface distress" micro-spalls. Each defect population is incorporated into a hazard factor describing that population. Hazard factors designated \( \Phi_{21} \) measure the severity of these defect families. The overall stress level under which the contact operates is measured by hazard factor \( \Phi_k \). The specific characteristics of the stress field are characterized by hazard factors designated \( \Phi_31 \). Factors \( \Phi_{11} \) are model constants. The model-predicted 10% life quantile is:

\[
N_{10} = 12 A M^\frac{1}{\beta} \left[ N_{LP}^{-\frac{1}{\beta}} + \Psi_{PP} \tau_0^{\frac{1}{\beta}} + \Phi_{0LP}^{\frac{1}{\beta}} \right]^{-\frac{1}{\beta}} \tag{15}
\]

Where \( M \) is the matrix susceptibility factor of material, \( A \) is the scaling multiplier, \( \Phi_{0LP} \) is the baseline matrix susceptibility factor, \( \tau_0 \) is a fatigue limit stress and the exponents \( \zeta \) and \( \beta \) are constants. \( N_{LP} \) is the Lundberg-Palmgren rating life, which can be obtained by any of the classical methods applicable to space- and time-variable operating conditions, as described, for example, in Harris and Kotzalas (2007). The equivalent hazard factor ratio, \( \Psi \), calculated over one Duty Cycle, \( N \), is defined as:
\[ \Psi = \frac{\Phi_T}{\Phi_{TLP}} \]  

With the equivalent hazard factor product \( \Phi_T \):
\[ \Phi_T = \beta N^{-\beta} \]
\[ \sum_{j=1}^{q} \left( \sum_{k=1}^{m} \left[ \Phi_{4k,j} \sum_{i=a,b,f} (\Phi_{1i}\Phi_{2i}\Phi_{3i}) \right] \Delta l N_j^{\beta-1} \right) \Delta N \]  

and the Lundberg-Palmgren hazard factor \( \Phi_{TLP} \):
\[ \Phi_{TLP} = \beta N^{-\beta} \Phi_{2bLP} \Phi_{3bLP} \]
\[ \sum_{j=1}^{q} \left( \sum_{k=1}^{m} (\Phi_{4LP,k,j}) \Delta l N_j^{\beta-1} \right) \Delta N \]

where \( q \) is the number of increments \( \Delta N \) in the duty cycle \( \bar{N} \), \( m \) is the number of intervals \( \Delta l \) along the rolling track \( l \), \( N \) is the revolution count, \( \Phi_0 \) is the matrix susceptibility factor, \( \Phi_{2bLP} \) is the baseline inclusion defect factor and \( \Phi_{3bLP} \) is the baseline subsurface stress field factor.

As shown above, the model covers life prediction for conditions in which operating parameters may vary along the rolling tracks as well as in time, with the limitation that the time variability can be arbitrary within a relatively short duty cycle but repeats periodically from one duty cycle to the next. Under the variable operating conditions, hazard factors are calculated for short finite intervals along the rolling tracks and for increments of time over which conditions are considered constant. These factors are combined at each interval and then summed over the duty cycle to produce equivalent values \( \Phi_T \).

All parameter calculations, constants and default values that are mentioned above can be found in Tallian (1999).

### 2.4. Transfer functions selection for analysis

In this section, different transfer functions (the "Transfer Function" block in Figure 2) are proposed and discussed. The role of the transfer function is to translate the parameters of attack acting on the critical system (for example, loads, speeds or torque parameters) to an effective function of external load acting on the fuse bearing.

An effective transfer function is defined as a function which increase significantly the life degradation rate of the fuse bearing in case of harmful operating conditions in the critical system relative to normal operation. The bearing type and the transfer function of the fuse system, are examined. The fuse bearing acts as a weak mechanism in the critical system, and hence need to be a small bearing relative to the critical bearings, that fails within a relatively reasonable time. The bearing chosen to be used as the fuse is an SKF 61906 ball bearing, while the bearings of the 'critical' system are SKF 6208 (Appendix B).

Figure 5. Transfer functions as a function of \( \omega/\omega_n \) (a) TF1-linear from the origin, (b) TF2-linear, (c) TF3-proportional to critical machine load.

Three transfer functions are examined and shown in Figure 5. At the current analysis, only the rotating speed profile was chosen to be considered as parameter of the transfer function. The transfer functions get the speed as an input, and translate it to an external load that is applied on the fuse bearing. The first transfer function (a) is a linear function starting from the origin, with a slope of \( m = 2.5 \). The second transfer function (b) is also linear with the same slope, but not starting from the origin, and the third transfer function tested (c) is similar.
to the behavior of the load of the critical machine, multiplied by a factor \( A = 3.5 \).

The load is expressed as a function of the frequency ratio \( \omega/\omega_n \) in all transfer functions, but only TF3 function is sensitive to the natural frequency, \( \omega_n \), of the critical machine.

### 3. Simulation Results and Discussion

This chapter presents the results for a number of examples of attack simulations on a critical machine together with the fuse system. An analysis of the results is then performed in order to decide which of the transfer functions provides the best "defense".

#### 3.1. Attack Profiles Assumptions

To define the nature and form of the profiles of the attack scenarios, some assumptions are made:

1. The attack is defined as an arbitrary speed profile \( \omega(t) \) and acts on the critical system and on the fuse system simultaneously.
2. The speed profile, presented as the "profile of attack", repeats periodically from one duty cycle to the next.
3. Proper system operating is defined as an operation speed that is low enough compared to the natural frequency of the 'critical system' \( \omega_n \approx 30 \text{ [Hz]} \). For convenience, in a regular operation situation, the speed of the system was selected to be \( \omega_0 = 1100 \text{ [rpm]} = 18.3 \text{ [Hz]} \).
4. The loads acting on critical system bearings are applied only as a result of dynamic responses of the rotating system (section 2.1).
5. The load acting on the fuse bearing is applied as a result of the external load that is created by the applied (selected) transfer function (section 2.4).
6. It is true that cyber-attacks can damage many mechanical parts, but the assumption is that bearings are the weakest part in most of mechanical rotating machines, and as a result, this analysis focuses on bearing failures.

#### 3.2. Defining Characteristics of the Attack Profile

The attack can appear as short and high rotation speed peaks. Each peak is periodic in time (Figure 6.a), therefore by looking at the profile of one peak cycle (Figure 6.b), three characteristics of the peak profile can be defined. The first characteristic of the peak profile is its amplitude, the second is the width of the peak and the last is the cycle length of one peak.

The analysis carried out allows us to examine how these characteristics of the attack profile and their forms affect the bearing life of a critical system. In addition, it allows us to examine how each transfer function (Figure 5) reacts to each attack scenario and, as a result, on the fuse bearing life.

#### 3.3. The Effect of Peak Amplitude

The first characteristic of attack profile is analyzed with a set of different sized peak amplitudes (Figure 7). All peaks in the set are with the same size of peak width and same length of duty cycle (Table 2).

<table>
<thead>
<tr>
<th>Peak amplitude set [rpm]</th>
<th>Peak width [sec]</th>
<th>Duty cycle length [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1500</td>
<td>20</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 2. Parameters of peak profile.

It should be noted that the peak amplitude values are defined as the difference between the maximum peak value and the constant rotating speed \( \omega_0 \). Each peak profile in the set is used as input to the model described in Figure 2, which calculates the life time of the bearings according to two life models (modified L10 model and the Tallian life model).

Figure 8.a and Figure 8.b represent the bearing life results obtained using a modified L10 life model (section 2.2), and the results obtained using the Tallian’s life model (section 2.3) respectively. Each bearing life result is normalized by the bearing life at "regular operation" \( \frac{N_t^{\text{reg}}}{N_t^{\text{max}}} \) and represented as a function of speed profiles with different peak amplitudes. The results show, in both graphs, that the...
normalized bearing life curves of the critical machine (full line) decreases with an increase in peak amplitude; but, after a certain point of peak amplitude, the normalized bearing life curve increases moderately. The minimum point of the curve indicates the critical situation in which the system worked under a state of resonance. The logical explanation for the increase after the minimum point is due to the fact that the peak amplitude is moving away from the system's natural frequency, resulting in lower load amplitudes.

Three additional curves (dotted lines) are presented. Each curve belongs to another transfer function mentioned in section 2.4. As with the results of the normalized critical bearing life, these curves describe the normalized bearing life of the fuse bearing. The simulation results show that in most cases, the normalized fuse bearing life decreases with an increase in peak amplitude, for all transfer functions tested. Also noticed was that the biggest degradation occurs with TF3.

3.4. The effect of peak width

The second characteristic of attack profile is analyzed with a set of different sized peak widths (Figure 9). All peaks in the set have the same peak amplitude and same length of duty cycle (Table 3).

<table>
<thead>
<tr>
<th>Peak amplitude [rpm]</th>
<th>Peak width set [sec]</th>
<th>Duty cycle length [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0-60</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 3. Parameters of peak profile.

Figure 10.a and Figure 10.b represent the results, using a modified L10 life model, and the Tallian's model for a simulated set of peak widths at the speed profile, respectively. As with the analysis in section 3.3, each result was normalized by bearing life at regular operation. The results show, in both graphs, that the normalized bearing life of the critical machine curve (full line) decreases with an increase in peak width. Also noticed that with all transfer functions, the normalized life of the fuse bearing (dotted lines) decreases with an increase in peak width.

Compared to the other transfer functions, the third transfer function (TF3) gives the best fuse life degradation response due to the specific attack scenario. The transfer functions TF2 and TF3 have greater life degradation effects relative to the...
critical bearing life degradation with an increase in peak width.

![Figure 10](image1.png)

**Figure 10.** Simulation results under set profiles with different peak widths (a) using the L10 life model, (b) using the Tallian life model.

### 3.5. The effect of peak duty cycle length

The third characteristic of attack profile is analyzed with a set of different duty cycle lengths. All peaks in the set have the same size of peak amplitude and same size of peak length (Table 4).

<table>
<thead>
<tr>
<th>Peak amplitude [rpm]</th>
<th>Peak width [sec]</th>
<th>Duty cycle length set [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>20</td>
<td>100-10000</td>
</tr>
</tbody>
</table>

Table 4. Parameters of peak profile.

Figure 11.a and Figure 11.b represent the normalized bearings life results ($N_i/N_{reg}$) as a function of speed profile with different duty cycle lengths, for both life models (modified L10 and the Tallian, respectively). It should be noted that the shorter duty cycle length means that the peaks occur more frequently. The results show, in both graphs, that the normalized bearing life of a critical machine curve (full line) decreases with a shorter duty cycle length. Also noted is that with all transfer functions, the normalized life of the fuse bearing (dotted lines) decreases with the shorter duty cycle length. The simulation results are presented in a different form than the other analysis results, since the x-axis in the plots of the normalized degradation trends is in opposite direction. The figures below show a zoom into the interesting part of the curves. Far beyond these scales, all the curves coincide. Compared to the other transfer functions, the third transfer function (TF3) gives the best fuse life degradation response. In this profile character analysis, the transfer functions TF2 and TF3 have greater life degradation effects relative to the critical bearing life degradation.

![Figure 11](image2.png)

**Figure 11.** Simulation results under set profiles with different duty cycle lengths (a) using the L10 life model, (b) using the Tallian life model.

### 3.6. Optimal Transfer Function

All simulation results show that the fuse bearing life degradation is, to some extent, accelerated with all transfer functions that were considered. In all cases, the third transfer function (TF3) is the most effective when the peak of speed or any speed profile is close to the first natural frequency zone of the critical system. It is assumed that the most dangerous situation is when the system works near the natural frequency, TF3 is the best solution. In all simulations, TF2 and TF3 have a greater effect on fuse bearing life degradation relative to the critical bearings degradation, which is important for early attack detection, before serious damage is caused to the critical machine bearings. Many mechanical rotating machines also work with standard speeds that are far beyond the natural frequency (areas that do not place the machine in danger), where the vibrations are much lower than in a resonance situation. In that case, using TF3 results in much lower loads on the fuse bearing prevents its early deterioration.
failure and false alarms. No analysis was done for cases in which the system works at extreme speeds.

4. EXPERIMENTS

An experimental test facility was designed and built especially for this research (Figure 12), including a bearing kit directly connected to an electric motor, simulating the 'critical system'. Another kit representing the 'fuse bearing' kit is located in parallel to the first kit. A fuse bearing is located in the center of the shaft, while the external radial load is applied by a pneumatic piston.

![Figure 12. Experimental test facility.](image)

The pneumatic piston is connected to an electric pressure regulator that controls the pressure according to the voltage. The first transfer function (TF1), presented in section 2.4, has been implemented in the fuse system unit of the experimental test facility. The implementation of the transfer function is explained in Figure 13. This functionality creates a linear relationship between the external load and the rotational speed. In addition, it operates autonomously, without computer intervention. The other transfer functions have not yet been implemented.

![Figure 13. Actual implementation of a linear transfer function in the experimental test facility.](image)

Several fatigue life tests were performed on the fuse bearing with constant rotation speed and external load. These experiments allow the determination of the failure criteria for the fuse bearing selected, which helps to detect early fuse bearing fatigue failures in the future, and also helps to determine the construction of an early faults detection algorithm which will be integrated to the fuse system. The fuse degradation and failure tracking is applied using advanced signal processing and features extraction methodologies. An example of the failure tracking that has been implemented is presented in Figure 14. It shows trend of energy representing the pattern of a damage in the inner race of the fuse bearing as it is manifested in the order spectrum of the vibration envelope. Similar trends are calculated for different patterns representing damages at other locations, i.e. outer race, rotating elements, cage, etc. These trends help to determine the failure criteria of the specific defects. It is noticed that an early inner race defect started to appear after approximately 2300 minutes. A developed inner race defect appeared after 3500 minutes.

In order to demonstrate the model results that have been presented in chapter 3, further experiments should be performed. The experiments will include several attack scenarios, similar to those presented in chapter 3. Speed profile, as a parameter of attack, can be set up through a motor controller or a computer program, to have the desired shape and be periodic in time.

![Figure 14. Trend of energy representing a defect on the inner race based on the envelope spectrum.](image)

5. SUMMARY AND CONCLUSIONS

A new concept to cyber protection of critical mechanical rotating machines by a mechanical fuse was presented.

The model objective is to allow the design of the fuse mechanism and to define the requirements for the fuse bearing. The model includes a description of the dynamic behavior of the rotating machine, the transfer function of the loads on the critical component to the fuse bearing, and bearing life estimation models. This model facilitated the analysis of the effect of different transfer functions and attack scenarios.

Three transfer functions were considered, and their capability to accelerate the fuse failure for different attack scenarios was explained. Finally, the simulation results of bearing life-time attacked by several speed attack profiles were presented. The
results show the effects of these profiles on the critical rotating machine and on the fuse system including the transfer function and the fuse bearing. It was concluded that the transfer function that is proportional to the load on the critical system, TF3, was the most sensitive to the set of attacks simulated and, as a result, is the optimal transfer function. It was also shown that both life models have similar behavior under the simulated attack scenarios.

The experimental test facility design, the experiment results and the actual implementation of the transfer function (TF1) demonstrates the concept. It was also shown that early faults in the fuse bearing are detectable early enough.

In order to consolidate the concept, further research and model extension taking into account other parameters of attack should be explored. More scenarios of attack profiles should be simulated in the model, and endurance experiments with several attack profiles should be performed.

ACKNOWLEDGEMENTS

We would like to thank Perlston Foundation for the support of the research.

REFERENCES


M. A. Miner, Cumulative damage in fatigue. J. appl. Mech. 12, Al.59Al64 (1945).


Biographies

Evyatar Cohen was born in 1986 and currently lives in Beer Sheva city, Israel. He has obtained his first degree in Mechanical Engineering (2013) at the Ben Gurion University of the Negev, where he is currently doing his Master degree. The focus of his research is on promotion of the new concept of cyber defense that is the subject of the current paper, and also on characterizing of bearings life degradation through vibration analysis.

Prof. Jacob Bortman joined the academic faculty of BGU in September 2010 as a full Professor. He spent thirty years in the Israel Air Force (IAF), retiring with rank of Brigadier General. In the IAF he held the following positions: Head of the Fatigue and Damage Tolerance Lab; Head of the UAV (unmanned aerial vehicle) and Space Department; Head of the IAF’s Engineering Laboratories; Head of the Aircraft Department, and finally Head of the Material Directorate, a senior position which reports directly to the chief commander of the IAF. He received all three of his academic degrees in Mechanical Engineering. He received his D.Sc. degree from Washington University in 1991. His areas of research in the Dept. of Mechanical Engineering include: Health usage monitoring systems (HUMS), Conditioned based maintenance (CBM); Usage and fatigue damage survey; Finite Element Method; and Composite materials.

Renata Klein received her B.Sc. in Physics and Ph.D. in the field of Signal Processing from the Technion, Israel Institute of Technology. In the first 17 years of her professional career, she worked in ADA-Rafael, the Israeli Armament Development Authority, where she managed the Vibration Analysis department. In the next 14 years, she focused on development of automatic health management systems for machinery. She invented and managed the development of vibration based diagnostics and prognostics systems that are used successfully by the Israeli Air Force in its combat helicopters and UAV fleets, and by leading jet engine manufacturers. Renata is a lecturer at the faculty of Mechanical Engineering in Ben Gurion University of the Negev, where she supervises research in the area of vibration based Diagnostics and Prognostics. In the recent years, Renata is the CEO and owner of “R.K. Diagnostics”, providing R&D services and algorithms to companies who wish to integrate machinery health management and prognostics capabilities in their products.

Appendix A

The sizes $|X(\omega)| = |Y(\omega)|$ so

$$|X(\omega)| = e^{\left(\frac{\omega}{\omega_n}\right)^2} |G(i\omega)| \quad \text{(A1)}$$

While $|G(i\omega)|$ is

$$|G(i\omega)| = \frac{1}{\left\{1 - \left(\frac{\omega}{\omega_n}\right)^2\right\}^2 + \left(\frac{2\zeta\omega}{\omega_n}\right)^2} \quad \text{(A2)}$$

And the phase $\phi$

$$\phi = \tan^{-1}\left(\frac{2\zeta\omega/\omega_n}{1 - (\omega/\omega_n)^2}\right) \quad \text{(A3)}$$
APPENDIX B

The $k$ viscosity factor ratio is calculated by:

$$ k = \frac{v}{v_i} \quad \text{(B1)} $$

While $v$ is the actual viscosity factor that depends on grease temperature and $v_i$ is the viscosity factor desired for proper separation between contacts, which is calculated by:

$$ v_i = \begin{cases} 
45000 \cdot n^{-0.83} d_m^{-0.5} & \text{for } n < 1000 \text{ [rpm]} \\
4500 \cdot n^{-0.5} d_m^{0.5} & \text{for } n \geq 1000 \text{ [rpm]} 
\end{cases} \quad \text{(B2)} $$

The contamination factor $C_L$ is expressed by:

$$ C_L = \min(C_{L1} \cdot k^{0.68} \cdot d_m^{0.55}, 1) \cdot \left(1 - \frac{C_{L2}}{\sqrt{d_m}} \right) \quad \text{(B3)} $$

While $d_m$ is the mean diameter of the bearing. $C_{L1}$ and $C_{L2}$ are contamination constants that can be found in Harris’s book.

The properties of the bearings selected are listed below:

<table>
<thead>
<tr>
<th>Bearing</th>
<th>$C$</th>
<th>Inner diameter $d$ [mm]</th>
<th>Outer diameter $D$ [mm]</th>
<th>Width $B$ [mm]</th>
<th>$F_{lim}$ [kN]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKF 6208 'critical'</td>
<td>32.5</td>
<td>40</td>
<td>80</td>
<td>18</td>
<td>0.8</td>
</tr>
<tr>
<td>SKF 61906 'fuse'</td>
<td>7.28</td>
<td>30</td>
<td>47</td>
<td>9</td>
<td>0.212</td>
</tr>
</tbody>
</table>

Table 5. Model bearing properties.
Asynchronous Motor Test Bench for the Generation and Current Signal Diagnostics of Accelerated Bearing Damage

Christoph Anger¹, Christian Preusche², and Uwe Klingauf³

¹²³ Institute of Flight Systems and Automatic Control, Technische Universität Darmstadt, Darmstadt, Hessen, 64287, Germany
anger@fsr.tu-darmstadt.de
preusche@fsr.tu-darmstadt.de
klingauf@fsr.tu-darmstadt.de

ABSTRACT

The data acquisition of run to failure data by means of degrading components is one of the most delicate tasks in evaluating new diagnostic and prognostic approaches, since it is cost-intensive and time-consuming. Therefore, a test rig for a cost-efficient generation of artificial bearing damages is described below. The test rig is thereby based on an ordinary asynchronous motor.

This paper mainly concentrates on the description of the test rig’s setup and first diagnostic findings. One aim of the experiments is the investigation of several variations of the applied loads for the artificially accelerated bearing aging. Thus, radial force and fluting are examined. The latter causes a damage triggered by a current flow through the test bearing. Both load types reduce the overall lifespan of bearings to about few weeks.

The generated faults are a broken cage and chattermarks due to a radial force higher than the design point. The bearings are diagnosed by means of frequency analysis of the phase current signal, which is produced in the stator of the motor. Beside the current signal, also temperature, vibration and revolution of the shaft are measured, whereas the vibration signal is used only for the comparison to the current signal. The comprehensive measurement concept allows a performance evaluation of diagnostic and prognostic algorithms based on different physical indicators.

It can be shown that especially the current frequency spectrum of a faulty bearing differs significantly from a healthy one. In order to face the high amount of measurement data, the Principal Component Analysis is used for data reduction to generate features for the diagnosis and prognosis. Thus, a classification of different fault modes and loading conditions is possible.

1. INTRODUCTION

In recent researches three different approaches for the diagnosis of bearing faults are widely used for the validation of diagnosis methods. They can be classified as offline faults, when the tested bearings are removed from their in situ position, and online faults, which are generated by means of measures for the accelerated bearing aging.

The authors in (Blödt, Granjon, Raison, & Rostaing, 2008) describe the usage of bearings, which are issued from industrial maintenance, for the verification of their bearing fault modeling. Others like in (Bellini, Immovilli, Rubini, & Tassoni, 2008) examine new bearings which are artificially damaged. They introduce a mechanical load of 40 kN to simulate brinelling and roughen the surface of the outer ring to produce a single defect on the raceway. Another method is also the drilling of holes into the raceways of the bearings, which is used in the approach of (Blödt et al., 2008).

In (Stack, Habetler, & Harley, 2005) the problems of both offline approaches are discussed. They state, since the changes
to the probed machine during the removal and mounting of the bearings lead to differing test conditions, the experimental data is corrupted. Thus, to avoid these problems the third approach is to produce an artificially accelerated bearing damage by the application of loads which are higher than the approved design point of the bearing. In (Janjarasjitt, Ocak, & Loparo, 2008) it is described that bearing failure can be generated by a combination of axial load (about 154 kg) and a high operating temperature (about 260 °F). The overall lifespan of new bearings are thereby reduced to approximately one month.

Another approach to generate online bearing failures is the application of a current, which flows directly through the test bearing. This method is called fluting and the phenomenon is rarely discovered. The authors in (Boyanton & Hodges, 2002) discuss their observations concerning fluting in case of paper machines. Here, the control unit of the paper machines created a potential on the shaft, which led to electrical discharge machining (EDM) in the bearing. Beside the existence of a potential, whereat about 3 or 4 V are enough to produce first spark erosions, the appearance of fluting also depends on the lubrication’s state, since the oil film around the rolling elements has an insulating effect. When the oil film is thin enough (for example in areas of high load) first spark erosions arise, which generate small pits burned into the races.

Beside the decision between online or offline bearing faults also the selected load influences the diagnostic methods. In addition to the aforementioned fluting, many authors like in (Kim & Parlos, 2002) or in (Raison, Rostaing, Butscher, & Maroni, 2002) concentrate on an applied moment. Mostly, this load is generated by a second electric motor with adjustable torque and speed. Only few authors like in (McFadden & Smith, 1985) choose radial force for online bearing faults, although it is a realistic and common load of a bearing.

The implementation and evaluation of diagnostic and prognostic algorithms demand signal processing and data reduction techniques. (Jardine, Lin, & Banjevic, 2006) suggest different signal processing methods aiming at the extraction of features from the signal, which cover vital information about the motor. The authors distinguish between time domain, frequency domain and time-frequency signal processing methods. Each obtained feature can be sensitive to different fault modes.

One main problem of using multiple feature extraction methods is the resulting high amount of data, which is hard to deal with in case of fault diagnostics (Aye, Heyns, & Thiart, 2014). Thus, the need of data reduction methods arises. One way to identify pattern and transform the data into fewer principal components is the Principal Component Analysis (PCA).

The method is widely used with a broad application field like face recognition and data compression, but also for fault detection and classification. For instance (Chirico, Kolodziej, & Hall, 2012) use the PCA for detection and isolation of electro-mechanical actuator faults and (Malhi & Gao, 2004) for a feature selection scheme for a bearing test bed.

The test bench described in the next section provides the application of the two loads for the generation of online bearing faults: Fluting and radial force. After the explanation of the test bed setup, the data reduction and processing part is shortly introduced in section 3. First experimental results of the radial force and fluting concerning the frequency spectrum and the PCA are presented in sections 4 and 5, respectively. The paper ends with a conclusion and a short outlook on upcoming steps.

2. TEST BED SETUP

One main topic of this paper is the description of the test bench, which is used for the generation of run to failure data of bearings. Therefore, the platform with its mechanical parts and the components for the different types of applied loads are presented in the first part. The measurement concept for the data acquisition with the plugged sensors is explained afterwards and the last subsection will be a compilation of the different types of test bearings.

2.1. Mechanical parts and types of applied load

A cross section of the CAD sketch in Figure 1 is presented in Figure 2. Core piece of the test bench is a simple three-phase a.c. motor of type 80S/2 of the manufacturer *Emod Motoren*. The standard power of this motor is 0.75 kW at a line to line voltage of 115 V. The supply frequency is \( f_s = 50 \) Hz, it has two terminal pairs and for most of the experiments a slip of \( s \cdot f_s \approx 0.3 \) Hz. The motor is depicted on the left of Figure 2. The power supply is provided by three type 1001SL of the manufacturer *Elgar*, which enable a variable supply fre-
quency and voltage.
One main change in comparison to the delivery status of the motor is the displacement of the loose bearing (which is also the test bearing) out of the casing into a separate bearing bracket (see right side of Figure 2) for a more time-efficient disassembly. Therefore, the motor shaft is extended by a second shaft, whereat both are connected by a stiff coupling. This ensures that vibrations of the test bearing are directly transmitted to the motor shaft and, thus, are detected by the analysis of the stator current.

Three different types of load are chosen for an artificially accelerated bearing aging: Radial force, fluting of the test bearing and the contamination of the lubrication. The radial force is introduced by a ball joint bearing, which ensures a free angular movement of the shafts. By turning the crank lever (see Figure 1) in combination with a threaded rod and a bearing sleeve, the joint ball bearing is deflected. A spring in the force flow ensures a linear increase of the applied force, which is proportional to the number of revolutions (each revolution corresponds to 140 N). The maximum force that can be provided is 3320 N until the spring runs onto block. Since the distance between test bearing and joint ball bearing is small ($\approx 32$ mm), the effective load of the test bearing is approximately 3 kN.

Another change to the motor is caused by the application of fluting. As a current is supposed to flow through the test bearing, the motor shaft has to be insulated. Thus, the original fixed bearing of the motor is exchanged by a hybrid bearing to circumvent a current flow through the motor casing. Also the ball joint bearing is in an insulated hull to prevent a flow through this bearing instead of the tested one. As depicted in Figure 3 the current is introduced by a carbon brush, which is pressed on the shaft near the test bearing. To imitate the influences of high frequency switching motor drives, the current is rotary with a supply frequency of 50 Hz and a voltage of up to 9 V. A DSPACE system provides the signal for the voltage, whereat it is amplified by an op-amp (OPA541 by Burr-Brown). The op-amp and the voltage is designed to generate a resulting current through the bearing with a maximum amplitude of about 3 A; a cutout of this signal is plotted in Figure 4. It can be seen that the current shows a hysteresis behavior especially in the voltage area near null. It is assumed that this behavior is caused by the lubrication which varies the overall resistance between the outer and inner ring. Especially in areas of low voltage, the lubrication increases the overall resistance so that the current flow is blocked. Finally, the circuit is closed by connecting the bearing bracket first with a load resistor which prevents a real short-circuit. The load resistor is then connected to the ground of the motor.

![Figure 3. Carbon brush for fluting](image)

On the basis of this test bed the generation of run to failure data in a time duration of few days until several weeks with respect to the applied amount and type of load is possible.

2.2. Data acquisition
The data acquisition system is based on a DSPACE 1103 controller board and a comprehensive sensor suite, which is listed in Table 1. The phase currents of the motor are measured by three closed loop sensors. Another current sensor determines the applied load of the test bearing during a fluting experiment by recording the output of the op-amp. In order to compare and especially assess the performance of current based diagnostic methods, an accelerometer is mounted on the bearing bracket next to the test bearing in radial direction. The vibration and current signals are sampled with a frequency of 25 kHz, before the signals are filtered by means of a $10$ kHz low-pass filter. The ambient and bearing temperature as well as the rotation speed of the shaft are captured once per second. The recording of measurement data is performed every two minutes and lasts 10 seconds.
Table 1. Sensor suite of the test rig

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Type</th>
<th>Qty</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current sensor</td>
<td>LEM LA 25-NP</td>
<td>3</td>
<td>Motor phase</td>
</tr>
<tr>
<td>Current sensor</td>
<td>LEM LA 25-NP</td>
<td>1</td>
<td>Fluting current</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>Kistler 8702</td>
<td>1</td>
<td>Radial vibration</td>
</tr>
<tr>
<td>Temperature sensor</td>
<td>National Semiconductor LM35</td>
<td>2</td>
<td>Bearing and ambient temperature</td>
</tr>
<tr>
<td>Position sensor</td>
<td>Honeywell SS495A1</td>
<td>1</td>
<td>Shaft rotation</td>
</tr>
</tbody>
</table>

2.3. Examined bearing types

The test bearing in the bearing bracket is mounted in a hull which can be exchanged. Depending on the diameter $D$, different types of bearings can be examined. Other differentiating factors are manufacturer, rolling element, dynamic and static load $C$ and $C_0$, the designation and the price per bearing (exclusive VAT). A small listing of the employed bearings is given in Table 2.

Table 2. Listing of the examined bearings

<table>
<thead>
<tr>
<th>OEM</th>
<th>$D$ [mm]</th>
<th>Rolling element</th>
<th>$C/C_0$ [kN]</th>
<th>Type No.</th>
<th>Price [€]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKF</td>
<td>32</td>
<td>Balls</td>
<td>4.03/2.32</td>
<td>61804</td>
<td>8,73</td>
</tr>
<tr>
<td>SKF</td>
<td>42</td>
<td>Balls</td>
<td>7.28/4.05</td>
<td>16004</td>
<td>5,81</td>
</tr>
<tr>
<td>ISB</td>
<td>32</td>
<td>Balls</td>
<td>3.95/2.30</td>
<td>61804</td>
<td>2,46</td>
</tr>
<tr>
<td>ISB</td>
<td>42</td>
<td>Balls</td>
<td>7.14/4.00</td>
<td>16004</td>
<td>2,35</td>
</tr>
<tr>
<td>ISB</td>
<td>47</td>
<td>Cylinder</td>
<td>25.00/22.00</td>
<td>NU204</td>
<td>8,30</td>
</tr>
</tbody>
</table>

3. DATA PROCESSING AND REDUCTION METHOD

For the evaluation and interpretation of the measured current data a signal processing and reduction method is applied. The applied approach is based on the current signal of one motor phase. The performance of the motor diagnostic depends highly on the extraction of an appropriate feature set. In order to make sure that important information of the motor condition is covered a comprehensive set of features is prepared, which is discussed in section 3.1. As a result each measurement record is transformed into a high dimensional feature vector.

Though the resulting feature vector includes much vital information of the motor, a drawback is that the generated high dimensional vector is not suitable for many data driven diagnostic methods. Hence, the principal component technique is executed in order to map the condition information of the motor in a fewer dimensional vector. Section 3.2 gives a short introduction of this method.

3.1. Signal processing

Given the total amount of $N$ measurements, which are obtained from one experiment, each record $n$ is transferred to a feature vector. The resulting vector is defined as $f_n = \langle f_{n1}, f_{n2}, f_{nj}, \ldots, f_{nI} \rangle$, where $f_{nj}$ is the $j^{th}$ feature of overall $I$ characteristic values. In our case all in all 46 features are generated and summarized in the vector. The condition of the motor is described by means of a set of statistical time domain and frequency domain values. The features were taken from the reference (Delgado, Garcia, & Ortega, 2011), where a detailed explanation and the equations of the features can be found.

The statistical information of the temporal signal is determined by six features: Root mean square, peak to peak, standard deviation, crest factor, skewness and kurtosis. For the frequency domain features the amplitudes of each record’s Power Spectral Density (PSD) up to a frequency of 400 Hz is investigated. Higher frequencies are neglected, since fault phenomenons, e.g. characteristic bearing fault frequencies, are expected below this limit (see section 4.1). The relevant frequency range is divided into ten equal bands. The amplitudes within each band are the basis for the calculation of four features: Mean value of the band, standard deviation of the band, skewness and kurtosis of the band. Alltogether 40 features are obtained from the frequency domain.

Finally, all vectors $f_n$ of an experiment are combined into a $N \times J$ feature matrix $F$. Since the PCA is very sensitive to outliers, a moving average filter is applied, which smoothes the trend of each feature value over the time.

3.2. Principal Component Analysis

The PCA is a widely accepted technique for data compression and feature extraction. A detailed description of the method including the mathematical equations can be found in (Jolliffe, 2002) or (Alpaydin, 2014).

In general, the PCA is used to transform the feature matrix $F$ in a new $N \times J$ matrix $G$. One aim of the method is that $J < I$ is valid without much loss of information. For this purpose correlated variables of the data set $F$ are combined by the PCA into a set of linearly uncorrelated variables. The PCA identifies the so-called principal components of the feature matrix, which emphasize variation and patterns in the data set. Each feature vector $f_n$ is mapped to a new vector $g_n = (g_{n1}, g_{n2}, g_{nj}, \ldots, g_{nJ})$, where $g_{nj}$ is the $j^{th}$ of a total of $J$ principal components. The transformation is done by the
following equation:
\[ g_{nj} = e_{j1}f_{n1} + e_{j2}f_{n2} + e_{j3}f_{n3} + \ldots + e_{jI}f_{nI} \]  
(1)

The coefficients \( e_{ji} \) are eigenvector components, which are obtained from the covariance matrix of the feature matrix \( F \). Performing the transformation for \( j = 1 \ldots J \) by using equation 1 gives the principal components in order of significance (highest to lowest). In other words, the first component \((j = 1)\) describes the most variance of the training data set, whereas the second component explains the second most variance, and so forth. Hence, the former components give the most insight into a change of the motor condition. On the other hand, discarding the last components in order to reduce the dimension is possible, since the loss of information is insignificant. However, the rate of reduction depends on the data set \( F \) and the level of variance, which is covered from the remaining components, see section 4.2.

4. EXPERIMENTAL RESULTS WITH RADIAL FORCE

The application of radial force as a load of the tested bearing provides several experimental possibilities, which will be discussed in this section. Most of the trials focus on the generation of run to failure data by means of a variable applied force. The data set which is the basis for the investigations in this section has e.g. five different load states: first state is nearly with no force at the beginning of the trial (corresponds to 16 revolutions of the crank lever; at this point, the spring is still unstressed). The second state is an increased load of about 1680 N or 28 revolutions. In case of the third, forth and fifth state the crank lever is turned once at each time (every revolution introduces an increase of 140 N) so that the fifth state corresponds to a load of 2100 N and 31 revolutions, respectively.

The results in the next sections are based on the analysis of the stator current signal and its variations due to the radial load only. The mechanism behind these variations can be explained as follows: an increase in the load of the shaft results in a higher displacement of the shaft’s rotation axis. Thus, a varying load also leads to a changing air gap in the asynchronous motor which can be detected in the stator current. The influences of this varying radial force concerning the PSD and the resulting damage cases are presented in a first part. Afterwards, the data reduction to distinguish the single load states by means of the PCA is shown.

4.1. Influences and damage cases

In Figure 5 the PSD spectrum of the current signal during two load changes is depicted over about 1200 measurement cycles, whereas one cycle corresponds to 2 min. For the reduction of noise, the PSD values over the time are low-pass filtered. The first load change is in cycle 420 from about 1680 N to 1820 N (corresponds to one revolution) and the second change is another turn in cycle 1180 to about 1960 N. The corresponding deflection of the shaft inside the motor is about 0.13 mm (about 0.067 mm/kN). Although the differences in the deflections are small, they can be detected in the PSD spectrum by comparing the amplitudes of frequency 155 Hz (\( \equiv 1680 \) N), 162 Hz (\( \equiv 1820 \) N) and 191 Hz (\( \equiv 1960 \) N) over the time. Although these frequencies result from a graphical investigation only, it can be seen in Figure 5 that e.g. the amplitudes of frequency 162 Hz and 191 Hz in the left part of the plot are comparatively small in contrast to those of frequency 155 Hz and vice versa. Thus, the amount of load can be spotted by means of the stator current signal.

The cases of damage which are produced by a radial force higher than the approved load are mainly broken cages and chattermarks, which are located in the upper half of the outer ring (area of highest load). The corresponding characteristic fault frequencies in the outer raceway, which are presented e.g. in (Blödt et al., 2008), could not be verified during our experiments, since there were no changes in comparison to a healthy bearing.

Although the diagnosis of faulty bearings are in the focus of the investigations, also the failure of the shaft can be detected by analyzing the stator current. One consequence of the high radial load is also a high stress in the shaft which led to a crack in the feather key groove during one of the experiments. Since this fault was not recognized, the crack grew over several hours, which is depicted by means of the PSD spectrum in Figure 6. The exponential behavior of the amplitudes is clearly visible which is characteristic for a crack growth concerning assumptions like the Paris law. However, it must be mentioned that the data base for a certain statement about the cause of damage is too small so that the depicted rise of the amplitudes can also refer to a bearing fault.
4.2. Determination of load conditions using PCA

In this section the focus is on the distinction between the load conditions applying the data processing and reduction method, described in section 3. Therefore 100 records of the current signal for each five load conditions are used to generate a $46 \times 500$ feature matrix $F$. By means of the reduction method the feature matrix is transformed into a $3 \times 500$ matrix $G$. The remaining components cover about 94 percent of the variance of the original data set. This means, instead of 46 features only the first three principal components are used for the load determination. The data of the resulting matrix is illustrated in Figure 7.

5. EXPERIMENTAL RESULTS WITH FLUTING

The influences of fluting on the degradation of a bearing is rarely discovered as mentioned in section 1. Especially a convenient magnitude of the applied current through the bearing is hardly discussed or the recommended values vary in recent researches.

Similarly to section 4 the influences of fluting on the frequency spectrum of the current signal and the generated cases of damage are presented first. An approach for the extraction of a health indicator representing the current state of the bearing by means of the PCA is described afterwards. Since the application of fluting in this test bench is new, only three run to failure data sets are available for the comparison.

5.1. Influences and damage cases

One of the main challenges for the application of fluting is to provide a voltage so that a current flow through the bearing is possible. This aim is complicated, since the resistance between the outer and the inner ring of the bearing varies strongly, as discussed in section 2.1. Especially at the beginning of a new run to failure test this phenomenon can lead to a delayed start of the current flow, since the lubrication of a new bearing insulated the rolling elements from the outer and inner ring. Consequently, the current is too small for the existence of EDM. After a certain period of time (between hours and approximately one day) it is assumed that the thickness of lubrication decreases so that EDM is possible and the current begins to flow through the test bearing.

The beginning of a new test is depicted in Figure 8. Beside the measured bearing current given as the effective value also the PSD values at a frequency of 28 Hz is plotted, since the beginning of the current flow is also visible in the frequency range of the stator current in measure cycle 40. One explanation for this might be that a first EDM produces a pitting in the outer or inner ring of the bearing which leads to a displacement of the shaft’s rotation axis. Thus, this movement of the shaft can be detected by the analysis of the stator current signal. Another reason could be that a small current flow through the shaft and the hybrid bearing directly corrupts the stator current.

The PSD spectra of three different instants of time during a life cycle of a bearing are pictured in Figure 9. It is obvious that especially the amplitudes in the frequency domain
between 20 and 50 Hz increase with proceeding degradation of the bearing. It is important to mention that the supply frequency of 50 Hz is notch filtered and the frequency domain between 50 and 80 Hz is the result of mirroring the aforementioned frequency domain at the supply frequency.

A cross section of all PSD spectra over the time at a frequency of 29 Hz is depicted in Figure 10. The failure mode during this trial was a broken cage, which led to a nearly instant breakdown of the bearing at the end of its life cycle, although the slightly increased amplitudes in the middle of the lifetime is also visible. The overall lifetime was about two days.

The damage which is produced by fluting is material removal. Thus, a broken cage and an increased clearance of the bearing are consequences, whereat the latter leads to a direct contact between the cage and the inner raceway of the bearing. A cross section of all PSD spectra over the time at a frequency of 29 Hz is depicted in Figure 10. The failure mode during this trial was a broken cage, which led to a nearly instant breakdown of the bearing at the end of its life cycle, although the slightly increased amplitudes in the middle of the lifetime is also visible. The overall lifetime was about two days.

The damage which is produced by fluting is material removal.
of the rotor and the stator. In case of a broken cage, two different failure modes could be observed as depicted in Figure 11a and Figure 11b. In Figure 11a the cage is completely deformed and quarried out. In Figure 11b only one link (on the right of the picture) is bent, which sticks the bearing. Both led to a nearly abrupt failure of the bearing.

5.2. Extraction of a health indicator using PCA

Since the PCA is suitable to highlight variance of the determined features over the time (see section 4.2), the described method is applied to extract a health indicator from the current signal of the first three measured run to failure data sets. For this purpose, the first data set is selected as training data, which is used to determine the eigenvalue coefficients of the transformation Equation 1. According to the resulting coefficients, the first three principal components are determined of the three run to failure data sets.

The first principal components of the three experiments are shown in Figure 12. The last 22 hours are displayed before the failure of the bearing occurred. It can be ascertained that all three courses differ strongly and no clear tendency of the bearing’s degradation is visible. Furthermore, the final failure limit is located on different values. Thus, the first principal component is not suitable as a health indicator for the motor. Since the other components reveal a similar behavior a presentation is neglected at this point.

Another approach to generate a health indicator is to determine the euclidean distance of the first three principal components:

$$G_n = \sqrt{g_{n1}^2 + g_{n2}^2 + g_{n3}^2}. \quad (2)$$

The result is illustrated in Figure 13 which reveals the identical phenomenon of an oscillating trend. A benefit of the generated trends can be seen in a similar failure limit. However, two features cross the limit several times without a failure of the bearing. Thus, a prediction of a failure would be highly unlikely using the combined feature as a health indicator.

For the purpose of comparing vibration and the current data Figure 14 shows the root mean square values of the corresponding vibration signals. A roughly exponential degradation trend is visible. However, the variation of the final failure limit also hinders a precise prediction of the remaining useful lifetime of the motor.

6. CONCLUSION

The first diagnostic experimental results of a new test bench were presented in this paper. The focus of this paper was the
illustration of the test bed setup and here especially the components for the application of fluting and radial force. The influences of these loads concerning the frequency spectrum and the achieved bearing failures were discussed, whereas both can be roughly detected by analyzing the PSD of the stator current.

First examinations of these current signals were done by means of common methods for the post-processing step. The applied data reduction and processing method was tested both for the classification of the applied radial force and for the identification of a health indicator in case of fluting. The PCA was used to find only those features which correspond to the current radial force or the effective health indicator. A data-based classifier trained with these features could distinguish between different states in case of new test data. The classification of the different load levels were successful and especially in comparison to the graphical analysis of the PSD spectra more distinct. The extraction of a health indicator by means of the PCA was more challenging and needs more data sets for better results. The large variance concerning the causes of damage in particular complicates this method.

In the future, it is planned to extend the platform for the radial force by a step motor for an automatic application of predefined loads so that the crank lever will be replaced. Another challenge is the investigation of both combined loads, i.e. fluting and radial force, concerning the lifespan of a bearing. It is expected that this combination reduces the life cycle dramatically, since the insulating effect of the lubrication will decrease especially in the areas of high radial load. Another main issue of the current setting is the high number of broken cages. Since this case of damage occurs abruptly, the extraction of degradation courses is challenging. Thus, the generation of pittings in the raceways or on the rolling elements, which then can be analyzed by means of the familiar characteristic fault frequencies, is one main topic.

When the test bench is completed with all the planned changes, the superior goal will be to generate a data basis for the evaluation of prognosis and diagnosis algorithms to estimate the remaining useful lifetime of bearings by using the current data only.

**NOMENCLATURE**

- $C$: Dynamic bearing load
- $C_0$: Static bearing load
- $D$: Diameter of the outer ring
- $F$: Feature matrix of an experiment
- $f_n$: Feature vector of record $n$
- $f_s$: Supply frequency
- $G$: Reduced feature matrix of an experiment
- $g_n$: Reduced feature vector of record $n$
- $I$: Total number of features
- $J$: Total number of principal components
- $N$: Total number of measurement records
- $n$: Number of a record
- $s$: Slip
- EDM: Electrical Discharge Machining
- PCA: Principal Component Analysis
- PSD: Power Spectral Density

**REFERENCES**


Experimental Approach for Estimating Mesh Stiffness in Faulty States of Rotating Gear

Jungho Park¹, Jong moon Ha², Hyunseok Oh³, Byeng D. Youn⁴, Sungho Park⁵, and Joo-Ho Choi⁶

¹, ², ³, ⁴Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, 151-919, Republic of Korea
billyhjm@snu.ac.kr
hihijung@snu.ac.kr
hsoh52@hotmail.com
bdyoun@snu.ac.kr

⁵, ⁶Department of Aerospace & Mechanical Engineering, Korea Aerospace University, GoyangCity, Gyeonggido, 412-791, Republic of Korea
shpark6386@nate.com
jhchoi@kau.ac.kr

ABSTRACT

Gear mesh stiffness (GMS) is a principal factor in understanding a dynamic behavior and estimating a health condition of the gear system. Lots of methodologies have been proposed to estimate GMS in normal and abnormal states. However, most of them are performed in an analytical way, therefore experimental studies are limited. Moreover, previous experimental studies have limitations that they were only performed either in a static state or for a normal gear. In this study, we develop a methodology to estimate GMS of a rotating gear in faulty states, root crack and spalling. In the procedures, we employ transmission error (TE) which is defined as the difference between rotation of input and output gear. The methodology proposes the concepts of relative stiffness to remove the effect of low frequency component from shaft motion and variability of individual teeth, and corrected stiffness to exactly estimate GMS of cracked gear. Meanwhile, the study proposes a differentiating algorithm of gear faults between root crack and spalling considering the failure mechanisms of each fault. The developed algorithm is validated measuring the TE from a test-bed of a spur gear. Consequently, the algorithm has differentiated the gear in root crack and surface failure, and estimated the GMS of the gear in faulty states.

1. INTRODUCTION

Gear systems are widely used in many engineering applications like wind turbines, industrial robots, helicopters, etc. In gear systems, gear mesh stiffness (GMS) is a key parameter to understand a dynamic behaviors as it is one of the main sources of excitation for the systems. Therefore, the GMS has been widely studied, especially when the gear is in faulty states. Chaari et al. (2008) investigated the effect of spalling and tooth breakage on the stiffness and vibratory motions by an analytic method. The effects of tooth root crack on the stiffness were also studied by Chaari et al. (2009). In the study, after the time-varying profiles of GMS are analytically evaluated, they are demonstrated using a finite element method. Chen and Shao (2013) studied the effect of tooth root crack under the tooth profile modification. Liang et al. (2014) calculate the mesh stiffness of a planetary gear with a crack using the potential energy method.

On the other hand, experimental studies for estimating GMS are limited. The GMS is estimated measuring transmission error (TE) over the range of path of contact in gear teeth (Munro, Palmer & Morrish. 2001). In the proposed methodology, however, gear teeth near the measured tooth are artificially modified to minimize the effect of the teeth on TE. Yesilyurt, Fengshou, and Andrew (2003) developed the modal testing apparatus to estimate reduction ratio of GMS in wear conditions. The severity of the faulty states was assessed by calculating the peaks of frequency response functions. However, due to the characteristics of modal testing, the method cannot be applied to rotating gears. The GMS was also evaluated for a spline coupling with teeth (Curà & Andrea. 2013). A hexapod specially designed for measuring angular deformation of the tooth pairs is developed. In the study, the effects of angular misalignment on the stiffness were also inspected. However, the device cannot be adopted for a spline coupling in operation.
The objective of this study is, therefore, to develop experimental procedures to evaluate GMS of faulty gears in rotating condition, which have not been performed in previous studies. To this aim, TE, which is defined as the difference between rotation of input and output gear, is adopted to estimate GMS in faulty states, root crack and spalling.

In the procedures, relative stiffness and corrective stiffness are proposed. TE, which we employed for estimation, is subject to shaft motion and variability arising from gear teeth. Therefore, the relative stiffness is proposed to remove the effect of shaft and variability of gear teeth. Then, relative reduction ratio of GMS at each tooth for each fault can be quantified from relative stiffness. Next, corrective stiffness is proposed to exactly estimate the GMS for a root crack fault. Due to peculiar fault mechanism of rotating gear in path of contact, calculated relative stiffness could be underestimated for root crack of gear tooth. Therefore, the underestimated values of GMS in root crack are compensated using corrective stiffness. The proposed two stiffness lead to experimentally estimate GMS of a fault gear in operating conditions.

The rest of this paper is organized as follows. Section 2 reviews the characteristics of TE which we adopted as an estimation signal. Then, the proposed methodologies are introduced in Section 3. After the methodologies are demonstrated using a case study in Section 4, Section 5 concludes this paper with recommendations for future work.

2. REVIEW OF TRANSMISSION ERROR

In this study, we use TE to estimate GMS of rotating gears in faults. Although TE is a physically meaningful signal in relation with GMS, it could show some biased behaviors in measuring the signal. To fully utilize physical properties of the signal for GMS estimation, characteristics of the signals are explained in this section.

2.1. General Behaviors of TE

TE is usually measured by encoder, which calculates the rotational displacement. As mentioned above, TE is defined as the difference between rotation of input and output gear, and can be formulated as below:

\[ TE = \theta_h - R \times \theta_l \]  

where \( \theta_h \) and \( \theta_l \) are the rotational displacements of high speed and low speed shaft; \( R \) is gear ratio. When TE is measured from an encoder, it is usually composed of shaft motion and tooth motion like Figure 1 which shows simulated TE signals. Shaft motion happens due to pitch line run-out error while showing large fluctuations in the motions (Inalpolat, Handschuh & Kahraman, 2015), and tooth motions happens due to GMS while showing small fluctuations in the motions. To closely observe the effect of gear meshes on TE, the TE from shaft motion should be removed by filtering techniques like in Figure 1 (b). In the tooth motion of an actual gear, however, the behaviors of TE are not consistent unlike Figure 1 (b). Lots of unpredictable factors like machining errors, tip relief errors, and indexing errors affects the behaviors of TE in an actual condition (Inalpolat et al. 2015, Kahraman & Blankenship. 1999, Li. 2007). Therefore, the TE behavior of each tooth from the individual gear is not consistent.

2.2. TE Behaviors in Tooth Root Crack and Spalling

Faults in gear teeth degrade the GMS. Then, due to the degraded GMS, TE shows distinct behavior when faults arise in the gear teeth. However, each fault has its own fault mechanism, and degraded GMS also shows distinct behaviors at each fault. GMS is composed of three components; (i) stiffness of the tooth, (ii) stiffness of the gear body, (iii) contact stiffness between gear teeth. Among the components, the tooth root crack mainly degrades the stiffness of the tooth (Wu, Zuo & Parey. 2008). Therefore, the effect of the fault lasts all over the contact regions. On the other hand, the localized faults like spalling would alter the contact stiffness (Tan, Irving & Mba. 2007). Then, the spalling would affect the only single portion of the whole contact regions.

Figure 2 and 3 show the change of GMS and TE from gear path of contacts with tooth root crack and spalling. In these figures, TE arising from shaft motion is not included which is shown in Figure 1 (a). Since root crack affect overall region of contact path as mentioned above, GMS and TE are modified at whole region of contact path, which is shown in Figure 2. For the case of spalling, the fault modified GMS and TE only at limited region of contact path, at single contact region, which is shown in Figure 3. Although these fault mechanisms can be different according to a contact ratio of gear (Pandya & Parey. 2013), the behaviors are common in usual gears (Endo, Randall & Gosselin 2009).
3. RELATIVE STIFFNESS AND CORRECTIVE STIFFNESS

Due to the characteristics of TE mentioned in Section 2, direct estimation of GMS using TE is impossible. Therefore, this section proposes the ideas of relative stiffness and corrective stiffness which enable correct estimation of GMS in faults.

3.1. Relative Stiffness

As mentioned in Section 2.1, TE is composed of two components, shaft motion and tooth motion. To estimate a health condition of each tooth using TE, first of all, shaft motion should be removed from the measured TE. Moreover, TE of each tooth is affected by many factors like machining errors, tip relief errors, and indexing errors. This cause different magnitude of TE at each normal tooth according to its machining state from manufacturing errors even in a same gear. Therefore, the health condition of each tooth is estimated by its relative magnitude in waveforms of current TE to that of previously measured TE. The overall procedures of estimating relative stiffness are described in Figure 4.

First of all, raw TE signals are calculated using Equation (1). The detailed procedures for calculating TE is explained in Remond and Mahfoudh (2005). The raw signals pass through filters to remove shaft motions. One wave motion of the filtered signals represents an effect of TE from one tooth in gear path of contact indicated in Figure 1 (in Section 2.2). Then, the effect of TE from one tooth is quantified using peak-to-peak (P2P) values of waveforms. Higher magnitudes of P2P imply degraded conditions of tooth. However, the health states of teeth cannot be evaluate from magnitudes of P2P in different teeth, but from those of P2P in same tooth due to different machining states. Hence, the reference signals for estimating health condition are calculated by a mean of accumulated magnitudes of P2P from each tooth. Then, the relative stiffness is calculated from the ratio between currently measured signals and reference signals like below:

\[
GMS_{rel}^{i} = \frac{1}{n-1} \sum_{k=1}^{n-1} \frac{A(TE_k^i)}{A(TE_n^i)}
\]

where \(GMS_{rel}^{i}\) is relative stiffness of the \(i^{th}\) tooth and \(A(TE_k^i)\) is P2P values of the \(i^{th}\) tooth at the \(n^{th}\) test set. Using this equation, health state of each gear can be quantified with normalized value, where one means perfectly normal state and zero means perfectly faulty state. In Figure 4, the P2P value at each tooth from reference signals is different, which does not imply different health conditions. As the reference signals are accumulated in normal states, different P2P values at each tooth represent individual normal states. However, when the faults happens, the P2P value gets higher than reference signals, which induces the reduction of relative stiffness.

3.2. Corrective Stiffness for tooth root crack

Using the relative stiffness, the health condition of gear teeth could be normalized with values from zero to one. However, due to the fault mechanisms explained in Section 2.2, the relative stiffness of the root crack could show biased behaviors. The reason of biased behaviors for root crack is explained in Figure 5.

As described in section 2.2, TE at root crack increased at overall regions of contact path. Therefore, when calculating
Figure 5. The change of P2P values due to crack: Solid and dotted arrows means P2P values with and without a crack.

P2P values of each waveform, the P2P value before crack shows a smaller value than reference signals due to increased P2P of a prior tooth like Figure 5. Moreover, the P2P value at crack is relatively underestimated as much as reduced magnitude of a P2P value before a root crack as indicated in Figure 5 with dotted arrows. Therefore, corrective stiffness is calculated to compensate the lost P2P values as below:

\[
GMS^i_{cor} = \frac{1}{n-1} \sum_{k=1}^{n-1} A(TE_k^i) - A(TE_{n-1}^i)
\]

Where \(GMS^i_{cor}\) is corrective stiffness of the \(i^{th}\) tooth. The difference between Equation (2) and (3) is the compensation term in the denominator in Equation (3). The term means the elevated values of P2P in the \((i-1)^{th}\) tooth which cause the underestimation of P2P of the \(i^{th}\) tooth. This corrective term could compensate the underestimated values of P2P for the tooth root crack while not affecting the values of P2P for normal and other fault cases.

This section explains the concept of relative stiffness and corrective stiffness. The relative stiffness could quantify the health state of individual gear tooth while considering the variability of the individual tooth and shaft motion. And the corrective stiffness could improve the accuracy of reduction ratio of stiffness for root crack as it considers the effect of adjacent tooth to the measured tooth. The following section demonstrates the described procedures of calculating GMS using a test-bed of a spur gear.

4. CASE STUDY

The concepts of relative and corrective stiffness are verified using a case study in this section. After introducing the test set-up and specimens used in this study, the TE measured from the test-bed are transformed into forms of relative stiffness and corrective stiffness. Then, we discussed the results comparing with results from other studies.

4.1. Test Set-up and specimens

As mentioned above, the GMS is estimated using TE. Therefore, we constructed the test-bed that can measure the rotational displacements using encoders while applying the inverse torque using a brake system. An overview of the test-bed is shown in Figure 6. The parameters of the gears used in this study are specified in Table 1.

Table 1. Parameters of the gears

<table>
<thead>
<tr>
<th>Gear data</th>
<th>Wheel</th>
<th>Pinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of teeth</td>
<td>70</td>
<td>35</td>
</tr>
<tr>
<td>Pressure angle (deg)</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Module (mm)</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Face width (mm)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Pitch circle diameter (mm)</td>
<td>280</td>
<td>140</td>
</tr>
</tbody>
</table>

And the faults are seeded in a pinion gear like Figure 7. The types of seeded faults are tooth root crack and spalling. The length of the crack is 5 mm and the width of the spalling is 2.6mm. The faults are seeded by a wire-cutting method not to deteriorate the gear teeth shapes. And the crack and spalling were seeded in the 34th and 12th tooth, respectively.

Figure 7. (a) Root crack of 5mm and (b) Spalling of 2.6mm seeded in a specimen gear

A 2.9kW servo motor drives the wheel gear with 30rpm and the brake implies 450Nm of the inverse torque. And the rotational displacements of the input and output shaft are achieved from encoders of 8192 pulse per revolution (PPR). After calculating TE from the rotational displacements, the TE data obtained are manipulated to calculate relative and corrective stiffness.

In this step, we performed multiple normal tests by re-assemblies processes. The re-assembly process of test-bed is inevitable during seeding the faults in the normal gear. Therefore, we could isolate the effect of fault-seeding on TE from re-assembly of test-bed by accumulating normal data by re-assemblies. In this study, we performed five times of re-assemblies, which lead to six data sets of normal data including the first data set. After accumulating the normal test
data, root crack and spalling are seeded into the normal gears. And, single set of a faulty data set is compared with the accumulated normal data sets.

4.2. Relative Stiffness and Corrective Stiffness
Calculated from Measured TE

TE data are calculated using Equation (1) after obtaining the rotational displacement from the encoders. Then, the re-assemblies tests are performed for six times with a normal gear. Then, P2P of TE of each waveform from one tooth are calculated as shown in Figure 4. As mentioned in Section 3.1, the P2P values of TE are not consistent in each tooth due to the variability of teeth. Despite the uncertainty from re-assemblies, P2P values show consistent values in the same teeth except for the 6th test data. And the data are transformed into the relative stiffness using the Equation (2). The six accumulated normal data are used for calculating relative stiffness and calculated values of relative stiffness are shown in Figure 8. Most of the values shows values of one which means normal states. The abnormal changes of the 6th normal case might come from vibration of the test-bed.

Figure 8. (a) P2P values of TE and (b) relative stiffness at each tooth of the spur gear for the six normal re-assembly tests

Then, relative stiffness with root crack and spalling are calculated using TE values measured from the test-bed. The calculated values of relative stiffness are shown in Figure 9. At the 34th tooth with the crack seeded, the relative stiffness is 0.746. And, at the 12th tooth where the spalling was seeded, the relative stiffness is 0.4319. Also, as noted in Section 3.2, the relative stiffness of the tooth before crack is increased due

Figure 9. The relative stiffness at (a) root crack (12th tooth), (b) spalling (34th tooth) and (c) corrective stiffness for the six normal re-assembly tests and one faulty test (marked as red circles)
to the decreased P2P value of TE, which causes the underestimation of the relative stiffness of the root crack. Therefore, the underestimated value is compensated to the relative stiffness of the cracked tooth using the equation (3). Then, the corrective stiffness value is 0.683.

4.3. Discussion

From the relative stiffness, we could estimate the reduction ratio of GMS in the root crack and spalling. The reduction ratios are about 0.25 and 0.57 for root crack and spalling which can be obtained from relative stiffness in Figure 8. After calculating the corrective stiffness for root crack, the reduction ratio is 0.32. We could observe that spalling induced the more reduction in GMS than root crack. The results are also consistent with other studies that investigate the GMS and TE for root crack and spalling (Chaari et al. 2008, Chaari et al. 2009, Endo et al. 2009). As mentioned in Section 2.1, spalling deteriorated contact stiffness in GMS. Therefore, we could conclude that the contact stiffness which was degraded by spalling takes up the largest portion in GMS at the given operating conditions. And the results is consistent with the previous study by Kiekbusch, Sappok, Sauer & Howard (2011). The study showed that the proportions of contact stiffness get larger as magnitudes of torque become larger, which is similar with our operating conditions. And the calculated results implies the discriminating algorithms between root crack and spalling. A root crack is known to be a more serious fault of gear than surface damages like spalling and pitting (Fan & Zuo. 2006) as propagation of the crack could result in loss of the tooth. Therefore, the concept of corrective stiffness could make it possible to exactly estimate the reduction ratio of GMS by compensating the underestimated values of GMS. Moreover, the existence of the higher relative stiffness than one (or lower P2P values of TE than reference signals) before fault signals could be a feature that discriminates the faults between crack and spalling when the relative stiffness shows decreased values due to faults

5. Conclusion

This paper presents the experimental approach to estimate reduction ratio of GMS in faulty states. In the procedures, we employed the TE to develop relative stiffness and corrective stiffness. The relative stiffness could remove the biased behavior of TE from shaft motion and variability of gear teeth. And the corrective stiffness could compensate the underestimated value of reduction ratio in GMS from the fault mechanism of a root crack. The 2.6mm of a spalling reduces the 57% of GMS, and the 5mm of a crack reduces the GMS 32% after calculating corrective stiffness.

The main advantages of the proposed methodology are as follows. First, it provides experimental procedures to estimate reduction ratio of GMS in gear faults. Second, it differentiates the different types of gear faults; root crack and spalling. The described experimental procedures could be used to gears in a rotating state and health condition of a measured gear could be estimated using the calculated reduction ratio of GMS. Also, the differentiation method could provide maintenance strategies based on a fault type along with reduced ratio of measured signal.

This study was demonstrated using the test-bed measuring the TE signals of a spur gear in normal and faulty states. However, the operating conditions (e.g., rotating speed of a spur gear, inverse torque) and the sizes of faults are limited. In the future studies, the proposed methodology would be applied to various operating conditions and various sizes of faults. In the study, the methodology will be demonstrated in various operating conditions. Also, relationship between reduced ratio of GMS and faults sizes will be studied.

Acknowledgement

This work was partially supported by the two research projects of the International Collaborative Energy Technology R&D Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20118520020010) and Mid-career Researcher Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science ICT and Future Planning (MSIP) (2013R1A2A2A01068627).

References


Processing and Interpretation of Crack-Propagation Sensors

Nenad G. Nenadic¹, Paul A. Ardis², Adrian Hood³, Michael G. Thurston⁴, and David G. Lewicki⁵

¹,⁴ Rochester Institute of Technology, Rochester, NY, 14526, USA
nxnasp@rit.edu
mgtasp@rit.edu

² GE Global Research Center, 1 Research Circle, Niskayuna, NY, 12309
ardis.p@ge.com

³ US Army Research Laboratory, 4603 Flare Loop, Aberdeen Proving Ground, MD 21085
adrian.a.hood.civ@mail.mil

⁵ NASA Glenn Research Center, 21000 Brookpark Rd. - MS 23-3, Cleveland, OH 44135
david.g.lewicki@nasa.gov

ABSTRACT

The goal of gearbox prognostics, once a failure has been detected, is to estimate the degree of damage. Development of probabilistic damage assessment requires high quality ground truth. The present report describes practical analysis, processing, and interpretation of signals from crack-propagation sensors and damage estimation during gear teeth crack propagation. The study considers two types of crack propagations: one for a fatigue-tester-based crack propagation, and the other for a propagation in a gearbox. Crack closing occurs in both types of crack propagation and must be accounted for in assessing the damage. The analysis is conducted for sensors connected as a voltage divider. Signal conditioning and wire breakage inference are examined in detail. In addition, we discuss how equipping each gear tooth with two crack-propagation sensors, one on each gear face, can provide a better damage estimation.

1. INTRODUCTION

Of the four dominant modes of gear tooth failure (breakage, wear, pitting, and scoring), breakage is the most catastrophic and occurs precipitously, often with no advanced warning. From the fatigue viewpoint a life time of a gear can has two phases: crack initiation and crack propagation (Kramberger, Šraml, Potrč, & Flašker, 2004; Kramberger, Šraml, Glodež, Flašker, & Potrč, 2004; Glodež, Šraml, & Kramberger, 2002). Gear research community has developed many vibration-based, condition indicators (CIs), to detect these features and assess damage, as summarized in (Lebold, McClintic, Campbell, Byington, & Maynard, 2000; Samuel & Pines, 2005). To validate the performance of these CIs, researchers has been employing crack propagation (CP) sensors (see e.g. (Zakrajsek & Lewicki, 1998; N. Nenadic et al., 2013)). The physical principle of CP sensors is the change in resistance as a function of crack length. These sensors consist of thin, parallel strands of known resistance which snap as the crack propagate through and the total resistance increases.

Crack propagation sensors are typically used to measure crack lengths on the surface of mechanical structures. This study is focused on the analysis of signals from CP sensors implemented on spur gears. Two types of tests are considered: crack propagation in a single-tooth fatigue-based tester and crack propagation inside a gearbox. The initial cracks were not notched but imparted using fatigue overload (N. G. Nenadic, Wodenscheck, Thurston, & Lewicki, 2011; Stringer, LaBerge, Burdick, & Fields, 2012). In both cases the CP sensors provide measurable ground truth for the level of damage.

After the signals are conditioned and interpreted, they can be used to assess the damage on the face and predict the internal crack. In the first approximation the damage is expressed as the crack length. For a conservative approach, the selected length would be the longer of the two crack lengths, visible on the faces of the gear. However, the cracks often do not propagate uniformly across the face of a gear tooth. Lewicki et al. (Lewicki, Sane, Drago, & Wawrzynek, 1998) studied crack surfaces computationally, using boundary element methods. The information of crack lengths on both gear faces can be used to estimate these wavefronts and, more generally, to es-
Estimate the overall crack damage.

The importance of damage estimation becomes immediately clear when considering a large number of individual crack propagations. Because crack propagation on a fatigue tester is a less expensive and less complex experiment than a crack propagation in a gearbox, a statistically large collection of cracks was obtained on the single-tooth, fatigue tester. Figure 1a shows a histogram of the of cycles to induce a crack under fixed conditions for 166 gear teeth. Fatigue cracks were initiated on 93% of gear teeth before 50,000 cycles and 85% before 30,000 cycles. With an excitation frequency of 5 Hz, 85% of cracks were initiated within 100 minutes.

While the large uncertainty in crack initiation is to be expected, crack propagation is considered much better understood. For example, Paris Law (Paris, Gomez, & Anderson, 1961) has a long history of successful practical use; it is given by

\[ \frac{da}{dN} = C(\Delta K(a))^m, \]

where \( a \) is the crack size, \( N \) is the number of cycles, and \( m, C, \) and \( K(a) \) are material properties. However, the experiments show that evolution of small cracks can also have considerable uncertainty, especially initially, when fatigue-induced cracks are small and asymmetric.

After crack initiation, the tooth was subjected to a cycling excitation until the crack propagated until all the strands on both crack propagation sensors were broken. Figure 1b shows the remaining useful life (RUL) vs damage size for 11 of the 166 tested gear teeth, with the statistical information superimposed. Individual propagations are indicated by markers, with normal distributions estimated for each number of broken strands. Two different colors signify the crack lengths on two different sides of a gear. Note the square markers at the bottom: they denote a propagation where the crack lengths were about equal on both faces. In our experiments, these symmetric cracks propagate considerably faster than their asymmetric counterparts. The shaded area signifies two standard deviations about the mean. For small cracks (low number of broken strands), the life remaining is not normally distributed and the uncertainty is very large (note that the y-axis scale is logarithmic). After the 4th strand broke (which corresponds to crack length of about 1.3 mm), the propagation accelerated for most of the gear teeth. Thus, crack length of 1.3 mm denotes prognostics horizon and in this region, Paris’ Law given by Eq. (1) can be reasonably fitted. Note, however, that for this gear and this loading, there is only 12,000 cycles left after the horizon is reach. An angular speed is needed to translate the prognostic horizon in time. For example, for an angular speed at \( \omega = 1200 \text{ rpm} = 20 \text{ rad/s} \), the prognostic horizon is only 600 seconds, or 10 minutes.

2. Sensor Placement and Sensing Circuit

While our results are quite general, further elements of the particular implementation are described in detail. This section provides specific aspects of sensing during the test of this study. Each sensor is placed so that it strands are approximately perpendicular to a typical path of crack propagation (Lewicki et al., 1998). We designed a mask, shown in Figure 2, to ensure close agreement in sensor placements among samples.

Figure 3 shows the circuit diagram for the CP sensor, based on Kelvin (four-wire) connection, where \( R_s \) is the pull-up source resistor, \( V_{ss} \) is the power supply voltage, \( R_{cp} \) signifies the resistance of the crack-propagation sensor, and \( I \) is the current. Four-wire measurement method is employed to minimize the effect of the lead wires indicated by \( R_W \) in Fig-
Figure 2. Design mask overlaid on top of a gear.

Figure 3. The circuit diagram for 4-wire connection of a CP sensor. The inset shows an image of a real, instrumented tooth.

Figure 4. Computed voltage levels of the CP sensor for \( R_s = 100 \, \Omega \) and \( V_{ss} = 5 \, \text{V} \).

Table 1. Theoretical voltage levels for three different values of the pull-up resistor \( R_s \) and supply voltage \( V_{ss} = 5 \, \text{V} \).

<table>
<thead>
<tr>
<th>Broken wires ( k )</th>
<th>( v_{out} ) [mV] ( (R_s = 100 , \text{k}\Omega) )</th>
<th>( v_{out} ) [mV] ( (R_s = 1 , \text{k}\Omega) )</th>
<th>( v_{out} ) [mV] ( (R_s = 100 , \text{\Omega}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.25</td>
<td>24.9</td>
<td>238.10</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>27.6</td>
<td>263.16</td>
</tr>
<tr>
<td>2</td>
<td>0.31</td>
<td>31.1</td>
<td>294.12</td>
</tr>
<tr>
<td>3</td>
<td>0.36</td>
<td>35.5</td>
<td>333.33</td>
</tr>
<tr>
<td>4</td>
<td>0.42</td>
<td>41.3</td>
<td>384.62</td>
</tr>
<tr>
<td>5</td>
<td>0.50</td>
<td>49.5</td>
<td>454.55</td>
</tr>
<tr>
<td>6</td>
<td>0.62</td>
<td>61.7</td>
<td>55.56</td>
</tr>
<tr>
<td>7</td>
<td>0.83</td>
<td>82.0</td>
<td>714.29</td>
</tr>
<tr>
<td>8</td>
<td>1.25</td>
<td>122.0</td>
<td>1000.00</td>
</tr>
<tr>
<td>9</td>
<td>2.50</td>
<td>238.1</td>
<td>1666.67</td>
</tr>
<tr>
<td>10</td>
<td>5000.00</td>
<td>5.000</td>
<td>5000.00</td>
</tr>
</tbody>
</table>

The manufacturer of CP sensors does not recommend their utilization with pull-up resistor \( R_s \) smaller than 100 \( \text{k}\Omega \) to limit the current through the CP sensor. However, to increase the sensitivity we reduced the value of the pull-up resistor down to \( R_s = 100 \, \text{\Omega} \). We have not observed any damage to the sensor during our testing. Figure 4 shows the output voltage \( v_{out} \) as a function of the number of the broken strands for the supply voltage \( V_{ss} = 5 \, \text{V} \). The voltage level for each state of the CP sensor are indicated in the plot. Table 1 shows the computed voltage levels for three different pull-up resistors (recommended 100 \( \text{k}\Omega \), and employed 1 \( \text{k}\Omega /100 \, \text{\Omega} \)).

### 3. Interpretation of CP Signals in Single-Tooth Crack Propagation

The first type of tests propagated a crack on a single-tooth test fixture. After the crack is initiated on a single tooth and verified using magnetic particle inspection (N. G. Nenadic et al., 2011), the crack is propagated by applying a sinusoidal load with an offset \( F(t) = F_o + F_{max} \sin(2\pi f t) \), with \( F_o = 750 \, \text{lb (3,336 N)} \), \( F_{max} = 650 \, \text{lb (2,891 N)} \), and \( f = 20 \, \text{Hz} \) (see the right axes of the inset of Figure 6).
of the fixture is shown in Figure 5a. The anvil pushes a single tooth at the prescribed load, while the reaction is shared among three reaction teeth. The photograph of the assembled fixture is shown in Figure 5b. The anvil applies the force at the highest point of single tooth contact (HPSTC). Figure 5c shows a close-up diagram of the contact between the tooth under test and anvil.

Figure 5. Single-tooth crack-initiation and crack-propagation fixture. (a) Drawing. (b) Photograph. (c) Sketch depicting point of contact.

Figure 6 shows an example raw voltage signal obtained from a CP sensor during a test on the single-tooth tester. The information of the crack propagation is contained in the envelope of the signal. The raw signal is not strictly non-decreasing because of the crack closing. The effect of crack closing is further illustrated in the inset of Figure 6, which shows the voltage signal over much shorter time scale. The waveform of the force applied via the anvil is overlaid (with the scale provided on the right-hand side). The step levels of the envelope of the measured signal do not perfectly line up with the computed levels, indicated by the grid lines. The imperfect match between measured and computed levels is hypothesized as due to the resistance of individual wires of the CP sensor are not precisely 50 $\Omega$. 

An example of a processed data is shown in Figure 7. The blue traces correspond to the left $y$-axis and indicate the processed crack propagation voltage; the solid trace corresponding to one of the crack-propagation sensors, denoted as front, and the other to the other sensor, denoted as back. The red traces correspond the right $y$-axis and indicate the length of the crack estimated based on the number of broken strands of the crack-propagation sensor in inches. The estimated length of the crack is simply as the normal distance between the strands $d_n$ (see Figure 3) of a CP sensor multiplied by the number of broken strands $k$: $\hat{l} = kd_n$. The normal distance between the strands is $d_n = 0.01'' = 0.254$ mm. For simplicity, the plot assumes linear growth between cycles.

4. INTERPRETATION OF CP SIGNALS IN A GEARBOX

The second type of test propagated a gear tooth crack inside a gearbox loaded using dynamometer fixture. This proved to be a much electrically noisier environment due to a slip-ring requirement. The gear was rotated at $\omega = 1500$ rpm producing a toothmesh period of $2\pi/(28\omega) \sim 9$ ms (a gear has 28
Figure 7. Processed cracks on tooth

Figure 8. Gearbox fixture built on a dynamometer (a) open gearbox (b) slip ring

This tooth loading duration is a fraction of that experienced on the fatigue tester. Thus, the tooth loading duration is $\sim 4.5$ ms. Relative duration of the applied load to the cracked tooth is much shorter 3.6% (100/28) of the cycle. In addition, to connect to the rotating instrumented gear teeth the signals are transmitted via slip-rings which add more noise to these feeble electrical signals. Figure 8a shows a part of a disassembled fixture with two gears meshed directly. Figure 8b shows the slip ring.

Figure 9 shows example waveforms midway through a propagation (compare the time line of Figure 9 to that of Figure 10). The subplot on top shows the waveform of the tachometer. The middle subplot shows a waveform of a less developed crack and the bottom shows the waveform of a more developed crack, where the effects of crack opening and closing can be readily observed.

Post-processing of the CP sensor signals was performed in order to remove noise artifacts prior to use as ground truth in detection and assessment tasks. As mentioned earlier, these artifacts took two forms: false “open” readings caused by intermittent slip ring failure, and voltage “jitter” caused by inconstant power supply. Correction for these artifacts was performed using global and local outlier rejection, respectively, as summarized in Algorithm 1.

**Algorithm 1 CP Postprocessing Algorithm**

```plaintext
1: procedure POST-PROCESS(s, n, ϵ)
2: // s is raw CP signal
3: // n is length of signal
4: // ϵ is Δ voltage threshold (%) for step detection
5: Global(s, n)
6: Local(s, Steps(s, n, ϵ))
7: end procedure
```

False openings are characterized by short, disconnected periods of approximately-full voltage observation, rapidly returning to baseline behavior at the current level of propagation. These observations arise from periods of slip ring decoupling, where the rotating brushes become disconnected by sudden impulses. In our experiments, these impulses were most likely to occur periodically in conjunction with particular tooth compression and release, although not all impulses strictly matched these mesh periods. In order to remove false openings, we performed outlier suppression at twice the standard deviation of the raw signal values, replacing spurious readings with the global mean, as written in Algorithm 2.

Once false openings are corrected, the remaining artifacts are modeled as the result of a multiplicative noise factor resulting from aperiodic variation in input power supply and an additive base factor resulting from regular systemic loss. Given our interest in using the CP sensor signals for discrete baseline classification of propagation (used in conjunction with accelerometer-based vibration features), we chose to ignore
Algorithm 2 Suppress Global Outliers

1: procedure GLOBAL($s$, $n$)
2: \hspace{1em} // $s$ is raw CP signal
3: \hspace{1em} // $n$ is length of signal
4: \hspace{1em} FIR($s$, 0.0001)
5: for $i$ ← 1, $n$ do
6: \hspace{2em} if abs(mean($s$) − $s[i]$) ≥ 2 * std($s$) then
7: \hspace{3em} $s[i]$ ← mean($s$);
8: \hspace{2em} end if
9: end for
10: end procedure

the additive factor as it would not affect computational ability to distinguish between propagation levels. For the multiplicative factor, we hypothesized that it was well approximated by a Gaussian process with mean of 1 (zero impact) and magnitude no greater than the minimum factor between two adjacent propagation levels, confirmed empirically. In order to remove outliers that could lie within the output range of other propagation levels, we first identified all voltage changes in excess of this process, indicative of steps from one propagation level to another. This is shown in Algorithm 3, where we explicitly identify and lock in observable steps as new voltage baselines.

Algorithm 3 Identify Voltage Steps

1: procedure $m$ ← STEPS($s$, $n$, $\epsilon$)
2: \hspace{1em} // $s$ is global-filtered CP signal
3: \hspace{1em} // $n$ is length of signal
4: \hspace{1em} // $\epsilon$ is $\Delta$ voltage threshold (%) for step detection
5: $c$ ← 1
6: $k$ ← 0
7: $i$ ← 1
8: while $i$ < $n$ do
9: \hspace{2em} while abs($s[c]$ − $s[i]$)/$s[c]$ ≤ $\epsilon$ do
10: \hspace{3em} $i$ ← $i$ + 1
11: \hspace{2em} end while
12: $k$ ← $k$ + 1
13: $m[k]$ ← $c$
14: $c$ ← $i$
15: end while
16: end procedure

Once voltage steps were identified, we then identified and suppressed all local outliers that could result in level confusion (i.e., those with multiplicative factors in excess of half of the minimum factor between adjacent propagation levels); suppression was done via mean replacement, as shown in Algorithm 4. The result of this process was a CP signal with minimized potential for label confusion when used in classification, allowing us to progress with further experiments in crack detection and assessment.

The final result of a damage extraction is shown in Figure 10.

Algorithm 4 Suppress Local Outliers

1: procedure LOCAL($s$, $m$)
2: \hspace{1em} // $s$ is global-filtered CP signal
3: \hspace{1em} // $m$ is vector of voltage step indices
4: \hspace{1em} $m[0]$ ← 1
5: for $i$ ← 1, $k$ do
6: \hspace{2em} for $j$ ← $m[i] - 1$, $m[i]$ do
7: \hspace{3em} if abs(mean($s[m[i] - 1]$ : $m[i]$)) − $s[j]$) ≥ std($s[m[i] - 1]$ : $m[i]$)) then
8: \hspace{4em} $s[j]$ ← mean($s[m[i] - 1]$ : $m[i]$))
9: \hspace{2em} end if
10: \hspace{2em} end for
11: \hspace{2em} end for
12: \hspace{2em} end procedure

Figure 10. An example voltage levels parsed after a crack propagated in a gearbox.

5. DAMAGE ESTIMATION AND FUTURE WORK

While cracks are typically thought of as lengths, in reality, they are surfaces. This quickly becomes apparent when one attempts to reconcile two different lengths, one on each gear face. Considerable information on damage and its likely future states may be contained in the relationship between the two lengths. For example, one can use breakage of CP strands to visualize a crack propagation, as illustrated in Figure 11. Here the data from CP sensors was used to plot two 3D stem plots (black traces), where the $x$-coordinates signify their physical location (the stem plots are separated by gear width $w$), $y$-coordinates signify propagation, expressed in terms of cycles $N$, and $z$-coordinates crack lengths (CP voltage levels mapped onto physical locations of particular strands and crack length estimated as the normal distance between strands). The blue traces indicate plausible wavefronts of the crack surface.
propagation. These are not accurately estimated but can be nonetheless useful in assessing actual damage.

Figure 11. Estimated propagation wavefronts based on the CP data.

To obtain a better estimate of these damaged surfaces, it is necessary to perform a post-mortem analysis of the fractured surface. For example, Figure 12 shows a cracked surface of a gear tooth. Crack initiation phase, with larger forces and intergranular fracture, is readily distinguished from the crack propagation, with smaller forces and transgranular fracture (wavefront 1). Wavefronts 2-4 show different stages of crack propagation and wavefront 5 marks the boundary of the crack propagation and full fracture (in our experiments, propagation was stopped once the crack had propagated through both CPs and a large force was applied to fully fracture the tooth).

Future work will further investigate the opportunity of improving damage assessment of cracked teeth on spur gears by establishing correlations (in a broad sense) between the time-domain condition indicators extracted from non-destructive measurements and analysis of images of cracked surfaces during the post-mortem analysis. After applying material science expert knowledge to identify important damage features in the images, the correspondence between image features and temporal features will be sought. The ultimate objective is to enable damage assessment in the noisier, gearbox environments, which, in turn, would allow better damage assessment for diagnostic and prognostics, and thereby condition based maintenance.

6. CONCLUSIONS

Crack propagation sensors can provide valuable ground truth information on crack damage, which is essential for development of practical prognostics metrics. It may be important to capture the ground truth in more than one location because empirical results that show that crack lengths on two gear faces can differ considerably during a crack propagation and that the asymmetry may significantly affect the rate of the propagation. This paper describes extraction of this ground truth information on damage from noisy measurements. Two types of measurement settings were discussed: crack propagation on a fatigue tester and crack propagation in a gearbox. In both cases cracks close when the force is either removed or reduced. Information extraction from CPs is easier in the case of fatigue-tester propagation because gearbox instrumentation requires that slip rings, with their additional noise sources, be employed and because the relative opening of the crack in a gearbox is considerably shorter than the crack opening in the fatigue fixture. Finally, visualization of crack propagation, based on information from CPs is illustrated and the next steps of relating the temporal data and image features from the post-mortem analysis are

ACKNOWLEDGEMENTS

This work was made possible by the Office of Naval Research under Award No. N0004-07-1-0823 and Army Research Laboratory under Award No. W911NF-14-2-0029. The authors gratefully acknowledge the help of our colleagues from Rochester Institute of Technology: Joseph Wodenscheck and Art Dee for skilful LabVIEW implementation of the control for both fixtures; Joseph Wodenscheck, Allen Luccitti, Michael Bradly, and Robert Kosty for instrumenting gears with CP sensors, and Mark Walluk and Domenic Maiola for the mechanical design of the fixtures. This work would not have been possible without their contributions.
REFERENCES


BIographies

Nenad Nenadić received his B.S. in Electrical Engineering from University of Novi Sad (Novi Sad, Serbia) in 1996 and his MS and Ph.D. in Electrical and Computer Engineering from University of Rochester (Rochester, NY, USA) in 1998 and 2001, respectively. He joined Kionix Inc. in 2001, where he worked on development of micro-electromechanical inertial sensors. Since 2005, he has been with Center for Integrated Manufacturing Studies (CIMS) at Rochester Institute of Technology, where he is currently a Research Associate Professor. His research interest include design, analysis, and monitoring of electromechanical devices and systems. He has two patents in electromechanical design and eight publications. He co-authored a textbook “Electromechanics and MEMS” and is a member of IEEE.

Paul Ardis received his B.S. in Computer Science and Mathematics from Purdue University (West Lafayette, IN, USA) in 2005 and his M.S. and Ph.D. in Computer Science from the University of Rochester (Rochester, NY, USA) in 2007 and 2009 respectively. He is Lead Scientist in the Machine Learning Lab at GE’s Global Research Center, formerly Research Associate Professor at Rochester Institute of Technology and Research Scientist under contract to the Air Force Research Laboratory. He holds two patents in signal processing and has published thirteen scholarly articles in machine learning, computer and human vision, biometrics, and machine diagnostics and prognostics. His research interests include data-driven predictive modeling, decision theory, surveillance, natural language processing, and machine sensing.

Adrian Hood Adrian works in the Propulsion Division of the Army Research Lab’s Vehicle Technology Directorate at Aberdeen Proving Ground, MD. His background is in helicopter transmission diagnostics. He earned his PhD in Aerospace Engineering at the University of Maryland, College Park.

Michael Thurston received his B.S. and M.S. in Mechanical Engineering from Rochester Institute of Technology (Rochester, NY, USA) in 1988, and his Ph.D. in Mechanical and Aerospace Engineering from the University of Buffalo (Buffalo, NY, USA) in 1998. He is the Technical Director and Research Associated Professor at the Center of Integrated Manufacturing Studies at Rochester Institute of Technology. He formerly held positions in air conditioning system development at General Motor and Delphi, and as a Researcher at the Applied Research Laboratory at Penn State University. He holds 7 patents in the areas of air conditioning and asset health monitoring. His research interests include: sustainable design and production, condition...
based maintenance and prognostics, and asset health management. He is a member of the Society of Automotive Engineers, and was awarded the Boss Kettering Award for product innovation by Delphi.

David Lewicki is an employee of the NASA Glenn Research Center, Cleveland, Ohio. He started with NASA in August 2009. Prior to that, Dr. Lewicki was an employee of the U.S. Army Research Laboratory’s Vehicle Technology Directorate at the NASA Glenn Research Center from 1982 to 2009. In his career, Dr. Lewicki has been performing analytical and experimental research in transmission, gearing, and bearing areas for rotorcraft and turboprop drive train applications. Specifically, he has been involved in research for the following areas: face gears for helicopter transmissions, low noise gears, gear crack propagation, gear diagnostics, engine disk crack detection, lubrication, transmission life and reliability predictions, gear dynamics, and variable-speed transmissions. Dr. Lewicki has earned a Ph.D. from Case Western Reserve University, a M.S.M.E. degree from the University of Toledo, and a B.S.M.E. from Cleveland State University. He has authored/co-authored 119 technical publications to date in the drive system area, and has been the Government manager for 37 contractor reports. He is also a Fellow of the American Society of Mechanical Engineers, previous Associate Technical Editor for power transmission and gearing for the Journal of Mechanical Design, and past Chairman of the ASME Power Transmission and Gearing Committee.
Advances in Ultrasonic Imaging for Internal Flaws in Structures
Thompson Vu Nguyen1, Simone Sternini2, and Francesco Lanza di Scalea3
1,2 PhD Candidate, Department of Structural Engineering, La Jolla, California, 92093, USA
tvn007@eng.ucsd.edu
ssternin@eng.ucsd.edu

3 Professor, Department of Structural Engineering, La Jolla, California, 92093, USA
flanzadiscalea@ucsd.edu

ABSTRACT
In the field of non-destructive testing of structures, 3D imaging of internal flaws is a critical task. Defect imaging allows the engineer to make informed follow-up decisions based on the morphology of the flaw.

This paper will present advances in ultrasonic tomography for the 3D visualization of internal flaws in solids. In particular, improvements to the conventional tomographic imaging algorithms have been made by utilizing a mode-selective image reconstruction scheme that exploits the specific displacement field, respectively, of the longitudinal wave modes and the shear wave modes, both propagating simultaneously in the test volume. The specific mode structure is exploited by an adaptive weight assignment to the ultrasonic tomographic array. Such adaptive weighting forces the imaging array to look at a specific scan direction and better focus the imaging onto the actual flaw (ultrasound reflector).

This study shows that the adaptive weighting based on mode structure improves image contrast and resolution compared to a conventional ultrasonic imaging technique based on delay-and-sum. Results will be shown from simulations and imaging experiments of simulated flaws in an aluminum block.

1. INTRODUCTION
Non-invasive 3D imaging techniques are essential to providing a comprehensive diagnosis and prognosis of potentially damaged structures and materials. Ultrasonic and thermal waves can be used in such non-destructive evaluation (NDE) techniques in a variety of applications such as structural health monitoring (SHM), material characterization, flaw detection and characterization, etc. As such, applications of ultrasonic arrays have widely been used in the NDE / SHM field and the medical diagnostic imaging field.

In the literature, advances in the ultrasonic imaging field have been made possible by synthetic apertures and beamforming frameworks. In one application, by manipulation of transmission sequences from the transducer elements, the array can transmit a global wave front such as plane waves (Garcia et al., 2013),(Ekroll et al., 2013),(Salles et al., 2014) and circularly-crested waves from virtual sources (Frazier & O’Brien, 1998),(Nikolov & Jensen, 2000). These advances provide a greater advantage by minimizing the transmissions while producing more energy in the wavefront. In reception, the full matrix capture (FMC) waveforms are collected and processed through various beamforming algorithms. There are several frameworks in the ultrasound imaging community that have been explored.

Both the conventional and proposed beamforming algorithms utilize a FMC in the processing. Conventional delay-and-sum (DAS) algorithms are typically used for fast and robust image reconstruction. DAS combines the summation of the responses from all the elements for a given pixel point by applying calculated delays, or time-of-flight (TOF), across the elements in the array. A different approach using the minimum-variance distortionless algorithm (MVD), also known as the Capon method, has been used in various array processing applications. This method improves the quality and accuracy of the images by applying different weights in addition to the summation of the waveforms to the DAS algorithm to minimize noise, thereby, improving the clarity of the image. (Hall & Michaels, 2010),(Austeng et al., 2011).

In this paper, a new adaptive beamforming framework is presented. A set of global matched coefficient weights (GMC) is introduced to be used in conjunction with the mentioned beamforming frameworks. By exploiting the correlation of the expected response and the recorded response from the elements in the array, this additional information can improve
the overall quality and accuracy of the image. The GMC can be applied to existing algorithms such as the DAS and MVD. Because these GMC rely on an expected response, specific replica vectors are introduced. By extracting the out-of-plane component from the surface of the transducer, the replica vectors are here calculated using two modes of propagation, specifically the longitudinal wavemode transmission to longitudinal wavemode reception (L-L) and the shear wavemode transmission to longitudinal wavemode reception (L-S). Other combinations of wavemodes can also be exploited, but in this paper, only the two mentioned will be discussed.

The paper is structured in the following portions: (1) modes of transmissions, (2) adaptive beamforming techniques (3) correlation coefficient weights, (4) simulation and experimental configurations, and (5) results and conclusions.

2. Modes of Transmissions

There are several conventional ways to excite the elements of an array. First, in the synthetic aperture approach, a sequence encompassing all elements, or a selected number of sparse elements can input a pulse, one element at a time, into the test specimen (Lockwood et al., 1998). The objective is to illuminate the scatterers from multiple directions. This produces a better resolution image. This is a very common method of transmission.

In another mode, multiple transducer elements can be fired simultaneously to produce a plane wave, or the elements can have a linear time-delay applied across the elements to produce an inclined plane wave (Garcia et al., 2013),(Ekroll et al., 2013),(Salles et al., 2014). This aim of this method is to transfer the maximum amount of energy from multiple angles in minimum numbers of transmissions as compared to the first method.

The third method is to use a virtual source. Usually a virtual element is placed behind the array. The distance between the elements of the array and the virtual element can be used to calculate the delayed time in transmission. This set of time delays are applied to the elements in the array (Frazier & O’Brien, 1998), (Nikolov & Jensen, 2000). This virtual source method allows more energy in the immediate wavefront with fewer transmissions as compared to the first method. Similar to the first method, by using several virtual sources, scatterers can be illuminated from various angles enhancing the clarity of the image.

3. Beamforming Techniques

With these several methods of excitations, the data collected in reception from the elements in the array can be processed in several ways. The different algorithms have one common fundamental: synthetic aperture beamforming. By shift-
ing the waveform in different time-delays, the scatterer location or pixel location can synthetically be focused. This section will review two conventional algorithms: the delay-and-sum (DAS) and the minimum variance distortionless method (MVD). Following, two additional algorithms are presented with the proposed adaptive weights. In all waveforms used to process, the differential signals of the baseline signals and test signals are used. Baseline signals are defined as signals collected from a non-defect specimen, and the test signals are from the unknown specimen.

\[ A_{ij} = A_{ij}^{test} - A_{ij}^{base} \]  

(1)

3.1. Delay-and-Sum

The conventional delay-and-sum (DAS) shifts and sums the waveforms from each element in accordance to the desired scatterer. The total distance of the waveform from the transmitter to the scatterer location and the return path from the scatterer location to a receiver is calculated. Using the proper velocity, a TOF can be found. This is processed for each receiver for each transmitter. More explicitly, for transmitter \( i \)-th transmitter and \( j \)-th receiver, with a scatterer located at \((x,y)\), the delay time is calculated as:

\[ \tau_{ij,xy} = \frac{d_{ij,xy}}{v} \]  

(2)

where
\( d_{ij,xy} \) is the distance of the wave propagation from the \( i \)-th transmitter to the scatterer at \((x,y)\) to the \( j \)-th receiver
\( v \) is the medium velocity
Translated into pixel values, for \( N \) transmissions and \( M \) receivers, the intensity is defined as:

\[ P_{xy} = \left| \sum_{i}^{N} \sum_{j}^{M} w_{ij,xy} A_{ij}(\tau_{ij,xy}) \right|^2 \]  

(3)

where
\( w_{ij,xy} \) is the unique weighting coefficient for the \( i \)-th transmission, the \( j \)-th receiver, and the pixel \((x,y)\)
\( A_{ij} \) is the amplitude value of the waveform for the \( i \)-th transmission to the \( j \)-th receiver at the delayed time \( \tau_{ij,xy} \)
In matrix form, it can be expressed as:

\[ P_{xy} = w_{xy}^T R_{xy} w_{xy} \]  

(4)

where
\( w_{xy} \) is the vector of weighting coefficients, \( w_{ij,xy} \) for the pixel \((x,y)\)
\( T \) denotes the Hermitian transpose

\( R_{xy} \) is the auto-correlation matrix: \( r_{xy} r_{xy}^T \)
\( r_{xy} \) is the vector of all \( A_{ij}(\tau_{ij,xy}) \) values for the pixel \((x,y)\)
For the conventional DAS imaging algorithm, the weighting coefficients, \( w_{ij,xy} \) are uniform. However, a window function can be used here to minimize sidelobes. For this paper, the weighting coefficients will be uniform (i.e., equal to 1), and thus will be neglected.

3.2. Minimum Variance Distortionless Method

Although DAS will produce fair quality images, the minimum variance distortionless method can potentially produce more accurate and lower noise images by applying adaptive weights for \( w_{xy} \) in equation (4). By minimizing unwanted scattering reflections, the clarity of the image increases. This framework is formed by using a replica vector, \( e_{xy} \), and by satisfying the two conditions (Kuperman & Turek, 1997):

\[ P_{xy} = \min_{w} w_{xy}^T R_{xy} w_{xy} \]  

(5)

\[ w_{xy}^T e_{xy} = 1 \]  

(6)

By using Lagrange multipliers, \( w_{xy} \) can be expressed as:

\[ w_{xy}^{MV} = \frac{R_{xy}^{-1} e_{xy}}{e_{xy}^T R_{xy}^{-1} e_{xy}} \]  

(7)

where \(-1\) denotes the inverse. The final MVD form is:

\[ P_{xy} = (w_{xy}^{MV})^T R_{xy} (w_{xy}^{MV}) \]  

(8)

or, by substitution:

\[ P_{xy} = \frac{1}{(e_{xy})^T R_{xy}^{-1} (e_{xy})} \]  

(9)

In summation form:

\[ P_{ij,xy} = \left| \sum_{i}^{M} \sum_{j}^{N} w_{ij,xy}^{MV} A_{ij}(\tau_{ij,xy}) \right|^2 \]  

(10)

3.2.1. Replica Vectors

In most applications of the MVD method, replica vectors are features typically extracted from analytical or numerical modelling for the specific scatterer location. For example, some replica vectors are derived from the calculated expected time of arrival for various scattering sources coupled with the expected attenuation in a medium (Chen et al., 2014). In ocean acoustics and geophysics, the vectors are extracted from the expected frequency spectrum across the array from the numerical simulated waveforms for each noise source (Baggeroer et al., 1993) (Corciulo et al., 2012). In structural vibrations, the vector of expected deflections for the nodes of a rigid bar for all possible loading locations are
the replica vectors (Kuperman & Turek, 1997). Essentially, across the domains of utilizing replica vectors in the MVD or Capon’s method, a modeled expected response is used.

In the application of ultrasound imaging, the replica vectors can be described as the expected signals of the array, from all scattering sources in the imaging domain. In this paper, the replica vectors will be utilizing the expected amplitudes of the longitudinal wavemodes. More specifically, the out-of-plane component of the received waves are considered, to model the expected response from longitudinal transducers. For simplification, a 2D medium model was used to extract these amplitudes.

Although only one wavemode is of interest, two combinations of the overall wavemodes are considered to use with the replica vectors. First, the longitudinal wavemode in transmission coupled with the longitudinal wave mode in reception (L-L). Second is the shear wavemode in transmission with the longitudinal wavemode in reception (L-S). Essentially, the perpendicular component to the surface was extracted from the location of the sensors to best model the expected response of a longitudinal type transducer as shown in Figure 3. The return path to the sensors are the same for both cases, the replica vectors, $e_{ij,xy}$, are the same for both combinations. Therefore, the replica vectors, $e_{ij,xy}$, depend on the wavemode arrivals in $A_{ij}(\tau_{ij,xy})$, and the locations of the transducer receiver with respect to the scatterer. These are derived from trigonometric considerations of the particle displacement field specific to the wavemodes as shown:

$$e^{(L-L),(S-L)}_{ij,xy} = \frac{y - y_j}{\sqrt{(x - x_j)^2 + (y - y_j)^2}}$$ (11)

For completion, using trigonometric relationships of the displacement field of a shear reflection response, the shear wavemode combinations from the scatterer to the transducer receivers are calculated as:

$$e^{(S-S),(L-S)}_{ij,xy} = \frac{x - x_j}{\sqrt{(x - x_j)^2 + (y - y_j)^2}}$$ (12)

In addition to these beamforming algorithms, there is another level of weights to explore. Given that the expected values are known for each transmission - receivers combinations for all scatterer locatins, a correlation factor, $\alpha_{i,xy}$ can be calculated to determine the significance of the data received in a more global perspective. For the conventional DAS algorithm, the weight is applied as:

$$P_{ij,xy} = \left| \sum_{i}^{N}{\sum_{j}^{M}{\alpha_{ij,xy}A_{ij}(\tau_{ij,xy})}} \right|^2$$ (13)

where $\alpha_{ij,xy}$ is the cross-correlation at zero-lag, or the inner product, of the expected values vector and the observed values super-vector for all transmissions.

$$\alpha_{ij,xy} = \sum_{i}^{N}{\sum_{j}^{M}{e_{ij,xy}A_{ij}(\tau_{ij,xy})}}$$ (14)
After substitution, this can be expressed as:

\[ P_{ij,xy} = \left| \sum_{i=1}^{N} \sum_{j=1}^{M} e_{ij,xy} A_{ij}(\tau_{ij,xy})^2 \right|^2 \]  \hspace{1cm} (15)

or in matrix form:

\[ P_{xy} = \alpha_{xy}^2 w_{xy}^T R_{xy} w_{xy} \]  \hspace{1cm} (16)

For the MVD framework considering the \( \alpha_{i,xy} \) terms, the image operator can be expressed as:

\[ P_{ij,xy} = \left| \sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij,xy} A_{ij}(\tau_{ij,xy})^2 \right|^2 \]  \hspace{1cm} (17)

and in matrix form:

\[ P_{xy} = \alpha_{xy}^2 (w_{xy}^{MV})^T R_{xy} (w_{xy}^{MV}) \]  \hspace{1cm} (18)

setting \( w_{xy}^{MV} \) as in equation (7) gives:

\[ P_{xy} = \frac{\alpha_{xy}^2}{(e_{xy})^T R_{xy}^{-1} (e_{xy})} \]  \hspace{1cm} (19)

5. SIMULATIONS

To determine how these different methods compare using the GMC, an aluminum block with several defects (top drilled holes) was the test subject. Numerical simulations were conducted to extract waveforms and experiments were carried out. The waveforms were compared to verify the simulation results. Imaging algorithms were then applied to both simulation waveforms and experimental waveforms.

5.1. Numerical simulation

Numerical simulations were carried out using k-Wave, an open source MATLAB toolbox used to model elastic waves (MATLAB, 2010) (Treeby et al., 2014). From the simulation, a FMC data structure can be captured. The FMC provided sets of waveform data to be processed in the prior discussed conventional beamforming frameworks with and without the global matched coefficients.

Several numerical models were conducted. First, numerical models were conducted with a defected aluminum block. Typical synthetic aperture excitations produced waveforms to be used in conventional beamforming frameworks and global adaptive beamforming frameworks. These results will be validated with experimental results, presented in the next section. Numerical simulations with a defective rail were also conducted.

Table 1. Simulated aluminum block

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2710 kg/m³</td>
</tr>
<tr>
<td>( v_L )</td>
<td>6150 m/s</td>
</tr>
<tr>
<td>( v_S )</td>
<td>3050 m/s</td>
</tr>
<tr>
<td>Width</td>
<td>57.7 mm</td>
</tr>
<tr>
<td>Length</td>
<td>57.7 mm</td>
</tr>
<tr>
<td>Height</td>
<td>57.7 mm</td>
</tr>
</tbody>
</table>

5.2. Simulation configuration

The simulation considered a aluminum block modeled in 2D with a defect (hole) towards the center of the plate (Figure 5). Impulse excitations centered at 2.25 MHz were generated with linear array. A 32-element linear array was simulated at the side of the block. The arrays followed a synthetic focusing aperture scheme, exciting element by element with a FMC. The TOF can be extracted using trigonometry relationships.

Simulations also included cases when the aluminum block had no defects to serve as a baseline.

5.3. Experimental configuration

The experiment involves an aluminum block with identical cross section dimensions as in table 1. However, only a portion of the block is used to minimize boundary reflections. The block with the defect is shown below. The excitation mode is identical to the simulation, with an element-by-element excitation, each with a FMC.
Figure 6. Image of defective block with a top drilled hole considered in experiment.

The array used has the properties below in table 2. The acquisition system is made by Advanced OEM Systems (Advanced OEM Systems, n.d.). This digitizer is a FPGA system that is a fully data acquisition system. Data is extracted and loaded into MATLAB for analysis.

Table 2. Linear array

<table>
<thead>
<tr>
<th>elements</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_0 )</td>
<td>2.11 MHz</td>
</tr>
<tr>
<td>frequency band</td>
<td>0.1 MHz</td>
</tr>
<tr>
<td>pitch (p)</td>
<td>0.6 mm</td>
</tr>
<tr>
<td>connector</td>
<td>Omniscan</td>
</tr>
</tbody>
</table>

Figure 7. Image of the linear array on the side of the aluminum block.

5.4. Waveform comparison

The simulation provided a FMC, consisting of all the waveforms for each transmitter’s excitation. To validate the results of the simulation, waveforms were also experimentally gathered later from an aluminum block of the same dimensions and defect size. Some waveform results are shown below. These waveforms are very similar and therefore, validate the simulation model.

Figure 8. Waveforms extracted for element 28 firing an excitation and element 8 receiving. Top: Simulation results. Bottom: Experimental results.

Figure 9. Waveforms extracted for element 12 firing an excitation and element 2 receiving. Top: Simulation results. Bottom: Experimental results.

5.5. Results

Images generated from the frameworks of: DAS, DAS with GMC, MVD, and MVD with GMC are produced from the simulated waveforms. The images are shown below. The last set of images, are an incoherent sum of the \((L - L)\) and \((L - S)\) wave mode images.

5.5.1. Simulation results

For a simulated defect at \((28\, mm, 30\, mm)\), the simulations prove to be effective. In figures 10, 11, 13, 14, and 16, the images produced with the addition of the GMC show significant improvement. Furthermore, in figures 12 and 15, the resolution in both axial and lateral directions have a higher dynamic range when the images are produced with the GMC.
Figure 10. Simulation: Images for conventional DAS (first), DAS with GMC (second) using the \((L - L)\) wavemode combination from simulation. (Dimensions are in [mm].)

Figure 11. Simulation: Images for conventional MVD (first), MVD with GMC (second) using the \((L - L)\) wavemode combination from simulation. (Dimensions are in [mm].)

Figure 12. Simulation: (L-L) Point spread function for the four methods mentioned in Figures 10 and 11. Top: Lateral resolution. Bottom: Axial resolution.

Figure 13. Simulation: Images for conventional DAS (first), DAS with GMC (second) using the \((L - S)\) wavemode combination from simulation. (Dimensions are in [mm].)
5.5.2. Experimental results

For a drilled hole at about (29mm, 29mm), the images prove to be very clear. In figures 17, 18, 20, 21, and 23, the images produced with the addition of the GMC show significant improvement. Furthermore, in figures 19 and 22, the resolution in both axial and lateral directions have a higher dynamic range when the images are produced with the GMC. The experimental results are very comparable to the simulation results.

6. Conclusion

Over all, with the addition of the global matched coefficients, the images have higher dynamic resolution in both lateral and axial directions. Accuracies are similar for all beamforming frameworks, however, the SNR improves with the addition of GMC. For all cases, the lateral resolution is superb. In the (L-S) modes, the resolution is limited to identifying scatterers that are at least 6mm apart. Case 2 study shows this. However, it is effective in identifying the existence of a scatterer at that location. The images from the (L-L) wave mode are excellent, and show very clear promise in the ability to characterize and image defects.

From this, the imaging frameworks benefit from the addition of GMC, and with defects less than 6mm apart, only the (L-L) mode is effective in distinguishing the defects from one another.

A new adaptive beamforming framework with the GMC was presented and formulated using a unique set of replica vec-
Figure 17. Experimental: Images for conventional DAS (first), DAS with GMC (second) using the \((L - L)\) wave-mode combination from experimental data. (Dimensions are in [mm].)

Figure 18. Experimental: Images for conventional MVD (first), MVD with GMC (second) using the \((L - L)\) wave-mode combination from experimental data. (Dimensions are in [mm].)

Figure 19. Experimental: With (L-L) wave mode: Point spread function for the four methods mentioned in Figures 10 and 11. Top: Lateral resolution. Bottom: Axial resolution. (Dimensions are in [mm].)

Figure 20. Experimental: Images for conventional DAS (first), DAS with GMC (second) using the \((L - S)\) wave-mode combination from experimental data. (Dimensions are in [mm].)
Figure 21. Experimental: Images for conventional MVD (first), MVD with GMC (second) using the \((L - S)\) wave-mode combination from experimental data. (Dimensions are in [mm].)

Figure 22. Experimental: With (L-S) wave mode: Point spread function for the four methods mentioned in Figures 20 and 21. Top: Lateral resolution. Bottom: Axial resolution. (Dimensions are in [mm].)

Figure 23. Experimental: Images for conventional DAS (first), DAS with GMC (second) using both the \((L - L)\) and \((L - S)\) wavemode combination from experimental data. (Dimensions are in [mm].)

These vectors are modeled from specific wavemodes, specifically the longitudinal and shear in transmission, and the longitudinal wavemode in reception. These replica vectors exploit the out-of-plane particle displacement field with respect to the transducers, effectively modeling the expected relative amplitudes of the received signals.

The GMC framework was applied to the conventional DAS and MVD frameworks. The reconstruction of images using ultrasound are presented in both of DAS and MVD frameworks of a defected aluminum block. Additionally, images constructed utilizing the GMC weights are also presented.

The figures show a substantial improvement in the clarity and precision of the defect in the steel. When comparing the point spread functions, the GMC weights using the adaptive wavemode-based replica vectors result in the higher ratio of defect scattering to noise scattering and minimize artificial scatterings.

This paper introduced new adaptive weights based on replica vectors modeled from the displacement structure of specific propagating wavemodes. The new approach shows excellent results in a simulation of a aluminum block with an artificial hole defect and experimental results validate these simulations. Further studies will exploit other expected response features to be used in the GMC formulations.

REFERENCES


**Biographies**

**Thompson Vu Nguyen** was born in Fountain Valley, California in 1989. He is a graduate student researcher pursuing his Ph.D. at the Structural Engineering Department at the University of California, San Diego. Also at the University of California, San Diego is where he obtained his M.S. in Structural Engineering (2013), his B.A. in Mathematics (2011) and B.S. in Structural Engineering (2011).
Analyzing high-dimensional thresholds for fault detection and diagnosis using active learning and Bayesian statistical modeling

Yuning He

1 UARC, NASA Ames Research Center, Moffett Field, CA, 94040, USA
yuning.he@nasa.gov

ABSTRACT

Many Fault Detection and Diagnosis (FDD) systems use discrete models for fault detection and analysis. Complex industrial systems generally have hundreds of sensors, which are used to provide data to the FDD system. Usually, the FDD wrapper code discretizes each sensor value individually and ignores any non-linearities as well as correlations between different sensor signals. This can easily lead to overly conservative threshold settings potentially resulting in many false alarms.

In this paper, we describe an advanced statistical framework that uses Bayesian dynamic modeling and active learning techniques to detect and characterize a threshold surface and shape in a high-dimensional space. The use of active learning techniques can drastically reduce the effort to study threshold surfaces. Automated Bayesian modeling of complex threshold surfaces has the potential to improve quality and performance of traditional wrapper code, which often uses hypercube thresholds.

1. INTRODUCTION

Many Fault Detection and Diagnosis (FDD) systems use discrete models for detection and reasoning. To obtain categorical values like "oil pressure too high", analog sensor values need to be discretized using a suitable threshold. Time series of analog and discrete sensor readings are discretized as they come in before processed by the diagnosis engine. This task is usually performed by the “wrapper code” of the FDD system, together with signal preprocessing and filtering.

In practice, selecting the right threshold is very difficult, because it heavily influences the quality of diagnosis. If a threshold causes the alarm to trigger in nominal situations, false alarms will be the consequence. On the other hand, if threshold setting does not trigger in case of an off-nominal condition, important alarms might be missed, potentially causing hazardous situations.

Usually, each sensor is handled individually and different threshold values might exist for different modes of the plant. For example, the threshold for the oil pressure for a cold engine (mode: cold) might be different from that for a hot engine (mode: hot). For complex industrial systems with hundreds of sensors and dozens of modes, a large number of thresholds must be selected and validated.

The use of a threshold for the discretization of a sensor signal, however, ignores any dependencies and correlations between different signals. Therefore, discretization with individual thresholds can only form a coarse approximation. Essentially, the thresholds form a hypercube in the high-dimensional space of sensor signals. This approach can easily lead to over-conservative settings. In those cases, proper thresholding would need non-linear high-dimensional threshold surfaces to accommodate dependencies between system components and different sensors (Figure 1).

Figure 1. Mode-specific thresholds (blue) for two modes for value $x$ over a parameter $p$. The curve for actual threshold is shown in red separates the safe region (top) from the unsafe region. Areas, where false alarms occur are shaded.

Often, dependencies between system components and different sensors are complicated and often not fully under-
stood. Therefore, experiments need to be carried out to determine the threshold surface. Because of high dimensionality and lack of analytical solutions, straight-forward grid-based methods are not applicable in general.

In this paper, we describe an advanced statistical method that uses Bayesian dynamic modeling and on-line learning techniques to estimate threshold surfaces in a high-dimensional space. Once a representation of the threshold surface has been obtained, techniques for fitting its shape and estimate shape parameters (He, 2015, 2012) can be applied. Our approach can incorporate domain knowledge about these surfaces. This approach goes way beyond traditional algorithms, which obtain thresholds in the form of hyper surfaces. By selecting the most likely shape of a surface from a domain-specific “library” and estimating it’s parameters, the domain expert can immediately recognize and understand that shape—a very important prerequisite for Verification and Validation (V&V) of FDD systems. This is in stark contrast to other well-known techniques like neural networks, where this information is hidden in a representation that is not suitable for human understanding. Here, however we will focus on statistical modeling and active learning for the detection of threshold surfaces.

The rest of this paper is structured as follows: Section 2 gives background on fault detection and diagnosis. In Section 3, we present an overview of our statistical modeling method. Section 4 focuses on the use of active learning for finding threshold surfaces. Active learning is discussed and a novel metric for the threshold-aware selection of new data points is presented. In Section 5, we discuss how Bayesian analysis of the threshold surfaces can result in a compact representation that is easy to understand by the domain expert. Section 6 illustrates our approach using artificial data sets and data from aerospace applications. Section 7 presents future work and concludes.

2. FAULT DETECTION AND DIAGNOSIS

Typically, Fault Detection and Diagnosis (FDD) systems are used to continuously monitor complex systems, e.g., an aircraft or spacecraft. Observable information obtained by sensors is used to detect any off-nominal situation and to perform root cause analysis. A number of different approaches for FDD or vehicle health management exist, but for this paper we focus on a very generic architecture as shown in Figure 2. The plant is observed using a number of analog sensors (e.g., pressure, temperature, battery voltage). Each signal is discretized by the wrapper code using thresholds $\theta$ in order to obtain discrete values comprising the outcome of a test. For example, for measurements of oil pressure $p$, $(p < \theta_p) \equiv true$ might indicate a dangerously low pressure. Often, one analog signal is discretized into various discrete ranges like “too low”, “nominal”, and “too high” using thresholds $\theta_{low}$ and $\theta_{high}$. The discrete outcomes of the tests are then fed into the diagnosis engine where hypotheses about the most likely set of failure modes (e.g., pump faulty, fuse open) is produced. That information can then be used to initiate mitigation and recovery actions. Diagnostic engines could be, for example, TEAMS/RT, TFPG (Abdelwahed, Dubey, Karsai, & Mahadevan, 2011; Mahadevan & Karsai, 2000–2014), or a Bayesian Network (Pearl, 1988), just to mention a few. In practice, discretization thresholds are, in most cases defined during design time. There might be different thresholds for different modes or configurations of the plant.

![Figure 2. High-level architecture of an FDD system](http://www.teamqsi.com)

3. METHODOLOGY OVERVIEW

We propose a sequential method for the estimation of parameterized threshold surfaces in high dimensional spaces. We represent this problem as learning the response surface for the function $f$, where $f(x) = 1 - \epsilon$ for some small $\epsilon > 0$ if the experiment succeeds and $f(x) = 0 + \epsilon$ otherwise. In this representation a classification threshold surface is determined by points $x$ with $f(x) = 0.5$.

Given an initial set of labeled data $D_0$, our approach builds a hierarchical Bayesian representation. Using active learning and computer experimental design, the number of required experiments and simulation runs can be kept small. The hierarchical representation provides information and confidence intervals for subsequent estimation of shape parameters $\Theta$ for the threshold surface.

The overall process is depicted in Figure 3. The active learning algorithm builds an initial classifier based upon $D_0$. Then, candidate points (i.e., sets of input parameters) are selected by the algorithm and handed over to the computer experiment, which executes the system under consideration at the candidate point and returns a categorical result (success or failure). Since each run of the simulator can require substantial computational resources, the overall number of new data points should be kept as small as possible.

1http://www.teamqsi.com
Our algorithm is based upon the sequential classification and regression framework as given by DynaTree (DT) (Taddy, Gramacy, & Polson, 2011; R. B. Gramacy, 2007). It features dynamic regression trees and a sequential tree model. Particle learning for posterior simulation makes Dynatrees a good candidate for applications, where new data points are processed sequentially. At any given point in time, the classifier is represented by a DynaTree. Figure 4 shows the individual steps of our overall algorithm. In the initial phase, a classifier using the data set \( D_0 \) is constructed. It provides an initial partitioning of the space and provides the information to estimate posteriors over given sets of data points. The main body is an iterative loop where, by adding new data points, the classifier will be extended and improved with the main goal of identifying and characterizing the threshold surfaces. In the first step, the current classifier is used to estimate a set of data points, which are close to the current prediction of the threshold. These comprise a subset of data points from a regular grid or a Latin hyper square, for which their entropy measure is high or the estimated response value is close to 0.5. The location of these points do not only depend on the actual threshold surface, but also on the shape of the dynamic tree and the size of the partitions, because points in the same partition have the same values. This set of data points is then used to estimate the best parameters \( \Theta \) for each of the threshold surfaces, together with a confidence interval for each of the parameters.

The candidate point selection in this active learning algorithm can use as much information as is available at the current stage, for example, information and entropy of the current data set. It then selects a new point (i.e., set of input parameters), for which the label is obtained by running the system simulator. Next we present the individual steps in detail.

4. **ACTIVE LEARNING AND EXPECTED IMPROVEMENT**

4.1. **Threshold surfaces**

Each data point describing one simulation run (experiment) is defined as \( x = (P_1, \ldots, P_p) \), where \( P_i \) are the input parameter settings and the outcome \( o(x) \in \{ \text{success}, \text{failure} \} \). Thus these data define a classification problem with \( C = 2 \) classes. We can view a threshold surface as a classification boundary between regions, where all experiments yield success \( p(x = \text{success}) = 1 \) and those, where the experiments do not meet the success criterion \( p(x = \text{failure}) = 1 \). Therefore, we can define a point \( x \) to be on the threshold surface if \( p(x = \text{success}) = p(x = \text{failure}) = 0.5 \). Because of the strong relationship between threshold surface and boundary, we will be using these terms interchangeably in the following sections.

A common metric to characterize points on the boundary is based upon the entropy. The entropy \( \text{entr} = -\sum_{c \in \{1, \ldots, C\}} p(x = c) \log p(x = c) \) becomes maximal at the boundary. In cases of more than two classes, R. Gramacy and Polson (2011) uses a BVSBS (Best vs. Second Best) strategy. Wickham (2008) defines a metric advantage as essentially \( \text{adv}(x) = [p(x = \text{success}) - p(x = \text{failure})] \). Then he considers points with minimal advantage to be close to the boundary.

In general, there are two basic methods to approach this classification boundary problem: explicitly from knowledge of the classification function, or by treating the classifier as a black box and finding the boundaries numerically. For some classifiers it is possible to find a simple parametric formula that describes the boundaries between groups, for example,
LDA or SVM. Most classification functions can output the posterior probability of an observation belonging to a group. Much of the time we do not look at these, and just classify the point to the group with the highest probability.

Points that are uncertain, i.e., have similar classification probabilities for two or more groups, suggest that the points are near the threshold surface between the two groups. For example, if a point is in Group 1 with probability 0.45, and in Group 2 with probability 0.55, then that point will be close to the boundary between the two groups. We can use this idea to find the threshold surfaces. If we sample points throughout the design space we can then select only those uncertain points near the threshold. The thickness of the threshold surface can be controlled by changing the value, which determines whether two probabilities are similar to each other or not. Ideally, we would like this to be as small as possible so that our boundaries are informative. Some classification functions do not generate posterior probabilities. In this case, we can use a k-nearest neighbors approach. Here we look at each point, and if all its neighbors are of the same class, then the point is not on the boundary and can be discarded. The advantage of this method is that it is completely general and can be applied to any classification function. The disadvantage is that it is slow \(O(n^2)\), because it computes distances between all pairs of points to find the nearest neighbors. In general, finding of the boundaries faces the “curse of dimensionality”: as the dimensionality of the design space increases, the number of points required to make a perceivable boundary (for fitting or visualization purposes) increases substantially. This problem can be attacked in two ways, by increasing the number of points used to fill the design space (uniform grid or random sample), or by increasing the thickness of the boundary.

### 4.2. Active Learning

Computer simulation of a complex system like those discussed above, is frequently used as a cost-effective means to study complex physical and engineering processes. It typically replaces a traditional mathematical model in cases where such models do not exist or cannot be solved analytically.

Active learning, or sequential design of experiments (DOE), in the context of estimating response surfaces (in our case boundaries), is called adaptive sampling. Adaptive sampling starts with a relatively small space-filling input data, and then proceeds by fitting a model, estimating predictive uncertainty, and choosing future samples with the aim of minimizing some measure of uncertainty, or trying to maximize information. We perform active learning with new data until the threshold surface is characterized with sufficient accuracy and confidence, and the whole space has been sufficiently explored to not miss any boundaries in the space.

Consider an approach which maximizes the information gained about model parameters by selecting the location \(x\), which has the greatest standard deviation in predicted output. This approach has been called ALM for Active Learning-Mackay, and has been shown to approximate maximum expected information designs (MacKay, 1992). An alternative algorithm is to select the variance minimizing the expected reduction in the squared error averaged over the input space (Cohn, 1996). This method is called ALC for Active Learning-Cohn. Rather than focusing on design points which have large predictive variance, ALC selects configurations that would lead to a global reduction in predictive variance.

The ALM/ALC algorithms are suitable for classification but not primarily for boundary detection (R. B. Gramacy, 2005). These heuristics are in general not suited for modeling the boundary because they do not take the specifics of the boundaries into account and they tend to also explore sparsely populated regions far away from current boundaries.

### 4.3. Boundary Expected Improvement

Finding a threshold surface corresponds to finding a boundary between two classes and can be considered as finding a contour with \(a = 0.5\) in the response surface of the system response. Inspired by (Jones, Schonlau, & Welch, 1998) and (Ranjan et al., 2008) and work on contour finding algorithms, we loosely follow (Ranjan, Bingham, & Michailidis, 2008), and define our heuristics by using an improvement function. In order to use the available resources as efficiently as possible for our contour/boundary finding task, one would ideally select candidate points which lie directly on the boundary, but that is unknown. Therefore, new trial points \(x\) are selected, which belong to an \(\epsilon\)-environment around the current estimated boundary. This means that \(0.5 - \epsilon \leq \hat{y}(x) \leq 0.5 + \epsilon\). New data points should maximize the information in the vicinity of the boundary. Following (Jones et al., 1998) and (Ranjan et al., 2008), we define an improvement function for \(x\) as

\[
I(x) = \epsilon^2(x) - \min\{(y(x) - 0.5)^2, \epsilon^2(x)\}
\]

Here, \(y(x) \sim N(\hat{y}(x), \sigma^2(x))\), and \(\epsilon(x) = \alpha \sigma(x)\) for a constant \(\alpha \geq 0\). This term defines an \(\epsilon\)-neighborhood around the boundary as a function of \(\sigma(x)\). For boundary sample points, \(I(X)\) will be large when the predicted \(\sigma(x)\) is largest.

The expected improvement \(E[I(x)]\) can be calculated easily following (Ranjan et al., 2008) as

\[
E[I(x)] = -\int_{0.5-\alpha\sigma(x)}^{0.5+\alpha\sigma(x)} (y - \hat{y}(x))^2 \phi\left(\frac{y - \hat{y}(x)}{\sigma(x)}\right) dy
\]

\begin{align*}
&+ 2(\hat{y}(x) - 0.5)\sigma^2(x) \left[\phi(z_+(x)) - \phi(z_-(x))\right] \\
&+ (\alpha^2\sigma^2(x) - (\hat{y}(x) - 0.5)^2) \left[\Phi(z_+(x)) - \Phi(z_-(x))\right],
\end{align*}

583
where \( z_\pm(x) = \frac{0.5 - \phi(x)}{\alpha(\sigma(x))} \pm \alpha \), and \( \phi \) and \( \Phi \) are the standard normal density and cumulative distribution, respectively. Each of these three terms are instrumental in different areas of the space. The first term summarizes information from regions of high variability within the \( \epsilon \)-band. The integration is performed over the \( \epsilon \)-band as \( \epsilon(x) = \alpha(\sigma(x)) \). The second term is concerned with areas of high variance farther away from the estimated boundary. Finally, the third term is active close to the estimated boundary. After the expected improvement has been calculated, the candidate point is selected as the point, which maximizes the expected improvement.

### 5. Shape Estimation of Threshold Surfaces

Given a classifier \( P_n \) based on a data set \( D_n \) consisting of \( n \) data points, we want to fit simple, parameterized shapes (from a dictionary provided by experts) to areas of high entropy that approximate the boundaries between the two classes.

#### 5.1. Notation

Suppose there are \( m \) shape classes \( M_1, \ldots, M_m \) with \( m \geq 1 \), which are parameterized by \( \Theta_1, \ldots, \Theta_m \). The task is to fit \( l \) shapes \( S_1, \ldots, S_l \), \( l \geq 1 \), where \( S_i = (i_1, \Theta_i), \ldots, S_l = (i_1, \Theta_i) \) and \( i_j \) denotes the shape class for the \( j \)th shape with \( i_j \in M = \{ M_1, \ldots, M_m \} \). Several of the \( i_j \) can be the same to accommodate more than one shape belonging to the same class. The \( \Theta_i \) should be different since we do not want to represent the same boundary shape twice. We also seek to determine the correct number of shapes \( l \) that represents the input point set \( X_n \).

For example, we may consider the \( m = 2 \) shape classes \( M_1 = \) hyperplane and \( M_2 = \) sphere in \( \mathbb{R}^d \). Hyperplanes are represented as \( a_1x_1 + \cdots + a_dx_d + a_{d+1} = 0 \) with parameter vector \( \Theta_1 = (a_1, \ldots, a_d, a_{d+1}) \in \mathbb{R}^{d+1} \). In the same \( d \)-dimensional space, a sphere of radius \( r \) with center \( c = (c_1, \ldots, c_d) \) is described by \( (x_1 - c_1)^2 + \cdots + (x_d - c_d)^2 = r^2 \) with parameter vector \( \Theta_2 = (c, r) \in \mathbb{R}^{d+1} \).

#### 5.2. What is a Good Shape Set \( S \)?

There are three conditions that specify when a shape set \( S \) provides a good fit to the data \( X_n \):

(i) **Summary:** each point on a shape \( S \in S \) is close to some classifier boundary point in \( X_n \).

(ii) **Completeness:** each classifier boundary point in \( X_n \) is close to some shape point on some shape \( S \in S \), and

(iii) **Minimality:** the shapes in \( S \) are as different from one another as possible.

Let us now explain why the above properties are desirable in a fitted shape set. These properties are illustrated in Figure 5

Condition (i) encourages each shape \( S \in S \) to be a good summary of one of the parts of the boundary of classifier \( P_n \).

That is, the points of a shape should lie along high entropy areas of \( P_n \).

The shape in Figure 5A is not a good summary of any part of the input point set. The shape in Figure 5B is a good summary of the points on the left. But this shape set is not a complete summary of the point set. C: This shape set is a complete, minimal summary of the point set. D: This shape set is a complete summary, but it is not minimal.

Condition (iii) encourages that shape set \( S \) to be minimal. This means that \( S \) will not use any extra shapes to form a complete summary of the boundaries of classifier \( P_n \). A complete summary \( \hat{S} \) (i.e., one satisfying (i) and (ii)) remains a complete summary if one of its shapes \( S \in S \) is added to \( S \) either exactly or after a small perturbation. In fact, adding a small perturbation \( \hat{S} \) of \( S \) may actually improve completeness slightly since \( \hat{S} \) can be even closer to some high entropy points than \( S \). And if \( S \) were a good summary, then so too would \( \hat{S} \). We need the minimality condition (iii) to be able to obtain the simplest (i.e., smallest) shape set that is a complete summary of the classifier boundaries.

The shape set in Figure 5D is minimal, but the shape set in the bottom-right is not minimal.

![Figure 5](image-url)

**Figure 5:** A: The shape is a poor summary of any part of the input point set. B: The shape is a good summary of the points on the left. But this shape set is not a complete summary of the point set. C: This shape set is a complete, minimal summary of the point set. D: This shape set is a complete summary, but it is not minimal.
### 5.3. Statistical Modeling

The shape set posterior is

\[ P(S | X_n) = \frac{P(X_n | S) P(S)}{P(X_n)} \propto P(X_n | S) P(S). \]

We build the posterior model \( P(S | X_n) \) by modeling the likelihood \( P(X_n | S) \) and the shape set prior \( P(S) \). In the posterior \( P(S | X_n) \propto P(X_n | S) P(S) \), we will model the likelihood \( P(X_n | S) \) to encourage completeness and the prior \( P(S) \) to encourage distance between shapes and therefore minimality. It makes sense that the data likelihood accounts for completeness because completeness requires observed points to be close to a shape and the observed points arise from the ground truth shapes with the addition of noise. We will encourage good summary using a Bayesian loss function that grows with increasing distance of the shapes to the point set. Finally, we determine the number of shapes \( l \) by minimizing the expected posterior loss.

A complete shape set with an extra shape near the observed points will have a low posterior probability because apriori we encourage separation between the shapes. An example of such a shape set is shown in the left column of Figure 6. If we remove the extra shape near the data from the previous example as in the right column of Figure 6, then the prior is high because the shapes are separated and the likelihood is high because the shape set is complete and therefore the posterior probability is high as well.

While our posterior decreases when an extra shape is added near the data points to a complete shape set, it will not decrease if an extra shape is added far from the data points. This case is shown in the middle column of Figure 6. In this example, the posterior is still high because the shape set is complete and the shapes are separated from another. In order to make this configuration unattractive, we need to use the summary property and encourage all shapes to be close to data points. Combining the shape set prior with summary will ensure that good shape sets do not have any extra shapes, as the prior prevents extra shape sets near the data and the summary prevents extra shapes far from the data.

We will encourage good summary using a Bayesian loss function that increases with increasing distance of the shapes to the point set. The Bayesian loss is low for the left column of Figure 6 because this configuration has good summary—all the shapes are close to data points. The loss is obviously high for the middle column because the shape on the left is far from the data points. The right column shows the desired case of the correct number of shapes with high posterior and low loss. Thus we will determine the number of shapes \( l \) by minimizing the expected posterior loss.

#### Likelihood

Our likelihood will encourage completeness. For the completeness condition (ii), we are interested in making the average squared distance \( D_{X_n,S}^2 \) of boundary points in \( X_n = \{x_1, \ldots, x_n\} \) to shapes in \( S \) small:

\[ D_{X_n,S}^2 = \frac{\sum_{x \in X_n} d_{X_n,S}^2 (x)}{|X_n|} = \frac{\sum_{j=1}^n d_{X_n,S}^2 (x_j)}{|X_n|}, \tag{1} \]

where

\[ d_{X_n,S}^2 (x) = \min_{s \in S} ||x - s||^2 \tag{2} \]

is the minimum squared distance of a high entropy point \( x \) to a point on any shape in the collection \( S = \{S_1, \ldots, S_l\} \).

An observed point \( x_j \in X_n \) is assumed to have been generated from a shape \( S_{z_j} \), where \( z_j \) gives the shape number that explains \( x_j \). Given \( z_j \), we model the likelihood of \( x_j \) as a decreasing function of the minimum distance from \( x_j \) to \( S_{z_j} \). The closer \( x_j \) is to shape \( S_{z_j} \), the higher the likelihood of \( x_j \). The observations \( x_j \) are assumed to be independent and modeled as

\[ x_j = s_j + \varepsilon_j = s_j + r_j n_j, \quad r_j \sim N(0, \sigma^2_r), \]

where \( n_j \) is a unit normal to \( S_{z_j} \) at \( s_j \) and \( r_j = (x_j - s_j) \cdot n_j \). Here the noise vector \( \varepsilon_j = r_j n_j \) is along a unit normal \( n_j \) to the shape \( S_{z_j} \) at the closest shape point \( s_j \) to \( x_j \). The scalar residual \( r_j \) is the signed distance along \( n_j \) from the shape \( S_{z_j} \) to \( x_j \). We model the observation error \( \varepsilon_j \) by modeling the signed residual as a \( N(0, \sigma^2_r) \) random variable.

Note that the squared residual \( r_j^2 \) is just the minimum distance squared from \( x_j \) to the closest point \( s_j \) on shape \( S_{z_j} \):

\[ r_j^2 = \min_{s \in S_{z_j}} ||x_j - s||^2, \]

where the minimum occurs at \( s = s_j \). Let \( Z = (z_1, \ldots, z_n) \). Assuming independence of points and that \( x_j \) depends only on shape \( S_{z_j} \), then

\[ P(X_n | Z, S) = \prod_{j=1}^n P(x_j | z_j, S_{z_j}) = \prod_{j=1}^n N(r_j | 0, \sigma^2_r). \]

Since \( r_j \sim N(0, \sigma^2_r) \), it follows that

\[ P(X_n | Z, S) = K \sigma^{-n} \exp\left( -\frac{1}{2\sigma^2_r} \sum_{j=1}^n \min_{s \in S_{z_j}} ||x_j - s||^2 \right), \tag{3} \]

for a constant \( K \). Note that if the observed point set \( X_n \) is close to the shapes in \( S \), then \( P(X_n | Z, S) \) is high. This state-
ment assumes, of course, that the correct shape $S_{z_j}$ explaining each point $x_j$ has also been identified.

We can obtain the likelihood $P(X_n|S)$ by modeling $Z|S$ and integrating out $Z$ as in $P(X_n|S) = \int_Z P(X_n|Z,S)P(Z|S)dZ$. We could, for example, model $Z|S$ by modeling a count vector $C = (c_1, \ldots, c_l)$ which holds the number of observations $c_i$ explained by shape $S_i$. Here $c_i = \sum_{j=1}^n 1_{x_j \in s_i}$. We can encourage good summary by modeling $C \sim \text{multinomial}(n, (1/l, 1/l, \ldots, 1/l))$ where each of the $l$ shapes in $S$ has the same probability $1/l$ of generating an observed point. This would make shape sets with any shapes that are from the data quite unlikely because we would expect to see points around each shape according to the given multinomial distribution.

It is difficult, however, to optimize over shape sets with the hidden random variables $Z$ in our models. Instead, we make a simple but accurate and effective approximation in our models and assume that the shape $S_{z_j}$ that explains observation $x_j$ is the shape in $S$ which is closest to $x_j$. Thus we replace the minimization in equation (3) over $s_j \in S_{z_j}$ with a minimization $s_j \in S$ over the entire shape set to obtain the approximation

$$P(X_n|S) = K\sigma^{−n} \exp\left(−\frac{1}{2\sigma^2} \sum_{j=1}^n \min_{s_j \in S} ||x_j−s_j||^2\right). \quad (4)$$

From equations (1),(2), we can see that the inner sum in equation (4) is just a scaled version $|X_n|D^2_{X_n,S}$ of our completeness measure. We can easily write our likelihood in terms of the completeness measure $D^2_{X_n,S}$. To do so cleanly, define

$$\sigma^2_{\text{complete}} = \sigma^2/|X_n|.$$

Then

$$P(X_n|S) = K\sigma^{-n} \exp\left(−\frac{1}{2\sigma^2_{\text{complete}}} D^2_{X_n,S}\right),$$

where another constant factor has been absorbed into $K$.

**Shape Set Prior** We build the shape set prior $P(S)$ based on the distances of points on each shape $S_i$ to the rest of the shape set $S_{−i} = S \setminus \{S_i\}$. To keep shapes apart from another, we want a large average squared distance from points on each shape to the rest of the shapes. Let $d^2_{S_i,S_j}(s_i)$ be the minimum squared distance of a point $s_i \in S_i$ to another shape $S_j$:

$$d^2_{S_i,S_j}(s_i) = \min_{s_j \in S_j} ||s_i−s_j||^2.$$

Then the squared distance of $s_i \in S_i$ to the shape set $S_{−i}$ is

$$d^2_{S_i,S_{−i}}(s_i) = \min_{s_j \in S_{−i}} d^2_{S_i,S_j}(s_i),$$

which finds the closest point in the rest of the shapes $S_{−i}$ to $s_i \in S_i$. Finally we average the inter-shape squared distances over all points on all shapes to get

$$\bar{D}^2_S = \frac{\sum_{S_i \in S} \sum_{s_i \in S_i} d^2_{S_i,S_{−i}}(s_i)}{\sum_{S_i \in S} |S_i|}.$$

To keep the shapes apart a priori, we want $\bar{D}^2_S$ to be large, indicating that on average the inter-shape distance is large. Equivalently, $1/\bar{D}^2_S$ should be small. Therefore we model the prior for $S$ using the normal distribution

$$S \sim N(\bar{D}^{-1}_S; 0, \sigma^2_{\text{shapesim}}).$$

**Bayesian Loss** Next we define a Bayesian loss function that encourages good summary. We can think of the summary condition (i) as requiring a small distance from each shape $S \in S$ to the set of classifier boundary points $X_n$. Let $d^2_{S,X_n}(s)$ denote the squared distance from a shape point $s \in S$ to the point set $X_n$:

$$d^2_{S,X_n}(s) = \min_{x \in X_n} ||s−x||^2.$$

We capture the average squared distance $D^2_{S,X_n}$ from the shape set $S$ to the input points $X_n$ by averaging over all points on all shapes in $S$ ($S = \{S_1, \ldots, S_l\}$):

$$D^2_{S,X_n} = \frac{\sum_{a=1}^l \sum_{s_a \in S_a} d^2_{S_a,X_n}(s)}{\sum_{a=1}^l |S_a|}.$$

We define our Bayesian loss function as

$$\text{loss}(S, X_n) = \lambda_{\text{summary}} D^2_{S,X_n}.$$

The smaller the distance from each shape in $S$ to the point set $X_n$, the smaller the loss. Thus minimizing the loss will encourage good summary.

**5.4. Shape Fitting Method**

Our shape fitting method has two main steps:

**Step 1** Minimize the expected posterior loss

$$g(l) = E[\text{loss}(S, X_n)], \quad |S| = l$$

over $l$ to obtain the number of shapes $l^*$

**Step 2** Compute the MAP shape set $S^{*,l^*}$ for sets of size $l^*$

As we shall see, our method in Step 1 for choosing the number of shapes $l^*$ to fit requires sampling from the shape set posterior. While drawing shape set samples, we can keep track of the maximum posterior probability shape set for each $l$ to obtain the MAP shape set output in Step 2. Another option in Step 2 is to return an entire posterior shape set summary with confidence intervals around posterior mean shape sets of size $l^*$. During Step 1 processing, we can save all the
posterior shape set samples for an $l$ that gives a new minimum expected loss. Then we will have the shape set samples for the chosen number of shapes $l^*$ and we simply compute a summary of those samples to output.

**Determining the Number of Shapes** We assume that we can apriori limit the number of shapes $l$ to some set $L$. For example, if we know that there will not be more than five boundaries then we can set $L = \{1, 2, 3, 4, 5\}$.

For each $l \in L$, we compute the expected posterior loss

$$g(l) = E[\text{loss}(S, X)] = \int_{\{S:|S|=l\}} \text{loss}(S, X_n) \hat{P}(S|X_n) dS.$$

Here we denote the shape set posterior probability distribution for shape sets with a fixed number of shapes as $\hat{P}(S|X_n)$. Then we choose the number of shapes to minimize the expected posterior loss:

$$l^* = \arg \min_{l \in L} g(l).$$

The integral in equation (5.4) is difficult to compute analytically. Therefore we approximate the integral for $g(l)$ by drawing $K$ shape set samples $S^{(k)}$ of size $l$ from the posterior $S|X_n$:

$$g(l) = E[\text{loss}(S, X)] \approx \sum_{k=1}^{K} \text{loss}(S^{(k)}, X_n) \hat{P}(S^{(k)}, X_n).$$

Thus our method for determining the number of shapes to fit requires the ability to draw posterior shape set samples of a fixed number of shapes $l$.

For a fixed shape set size $|S| = l$, we will draw samples from the posterior $P(S|X_n) \propto P(X_n|S)P(S)$ using an iterative procedure. Shape set samples $S$ with a small value for

$$- \log(P(X_n|S)P(S)) = - \log(P(X_n|S)) - \log(P(S))$$

should be more likely to occur.

**6. Experiments**

**6.1. Active Learning**

We illustrate the behavior of our approach using 2D artificial data sets and a quadratic threshold curve normalized to the unit square. Starting with a low number of $N_{init} = 126$ randomly selected initial data points, the active learning procedure selects $N = 500$ data points according to different candidate selection rules (random, ALC, ALM, EI, and boundary EI). $N$ has been selected this large for visualization purposes. Figure 7 shows, how the different selection algorithms behave. Our goal is to find many data points near the threshold curve in order to enable accurate representation and to facilitate subsequent shape estimation. Thus random selection (Figure 7A) and the classical ALC (Cohn, 1996) (Figure 7B) are not suitable for our purpose, because they require prohibitively large $N$ for reasonable results. On the other hand, the entire area should be considered as well in order not to miss any other boundary. Other algorithms are too localized and do not even explore the entire threshold curve (Figure 7C, D). Our approach (Figure 7E) tries to find a suitable balance between both requirements.

![Figure 7](image7.png)

**Figure 7. Candidate points during active learning:** (A) random selection, (B) ALC, (C) ALM, (D) EI, and (E) boundary-EI. Circles: initial data points. Solid: points added during active learning (colored according to experiment outcome).

Our boundary metric is parameterized by the parameter $\alpha$ (see Section 4.3). This parameter influences the width of the "band" around the threshold surface that is considered for the selection of the candidate point. Figure 8 shows runs with several values of $\alpha$. It seems that values around $\alpha = 0.8$ produce the best results; values of $\alpha$ that are too small or too large tend to lead to a situation, where the new points are located too far from the threshold surface.

![Figure 8](image8.png)

**Figure 8. Boundary-EI for $\alpha = 0.2, 0.5, 0.8, 1$**
6.2. Shape Selection

6.2.1. Artificial Data

For illustration of our shape selection algorithm, we generated artificial data sets in 2 and 5 dimensions. The ground-truth data, normalized to a unit hypercube contain one or two threshold surfaces in the shape of hyperspheres.

Our active learning algorithm starts with an initial 126 data points. Then, 700 data points were selected by the algorithm. The resulting model was used for shape estimation. Table 1A shows the results for the two-dimensional case. 25 runs with different randomly generated initial data points were executed. In all 25 runs, sphere $S_1$ was correctly recognized; the parameters for $S_2$ were only correctly estimated in 5 of the runs. The table shows the ground-truth values, means and variances for those runs where the shapes were detected.

<table>
<thead>
<tr>
<th>A</th>
<th>true value</th>
<th>$\hat{\mu}(\sigma^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>0.3</td>
<td>0.295(1.4e-5)</td>
</tr>
<tr>
<td>$y_0$</td>
<td>0.3</td>
<td>0.289(3e-5)</td>
</tr>
<tr>
<td>$r_0$</td>
<td>0.2</td>
<td>0.20(6.6e-6)</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0.7</td>
<td>0.715(8.5e-5)</td>
</tr>
<tr>
<td>$y_1$</td>
<td>0.7</td>
<td>0.720(8.6e-5)</td>
</tr>
<tr>
<td>$r_0$</td>
<td>0.2</td>
<td>0.20(5.2e-5)</td>
</tr>
</tbody>
</table>

Table 1B shows the situation in a 5-dimensional space. The centers of the hypersphere are located at $\vec{c}_1 = (0.3, 0.3, 0.3, 0.3, 0.3)^T$, and $\vec{c}_2 = (0.7, 0.7, 0.7, 0.7, 0.7)^T$, respectively and the radius is $r = 0.3$. Active learning selected 1000 data points. Here, the results are worse. E.g., the second sphere was not recognized in any of the 10 runs, indicating that for a 5-dimensional space, the number of data points must be considerable larger.

6.2.2. Intelligent Flight Control

The Intelligent Flight Control System (IFCS) is a damage-adaptive Neural Networks (NN) based flight control system developed by NASA and test flown on a manned F-15 aircraft (Rysdyk & Calise, 1998). An on-line trained NN provides control augmentation to dynamically counteract damage to the aircraft. For our experiments, we considered this system as a black box, controlled by numerous parameters (e.g., NN weights, controller gains, or learning rate). A simulation run was considered to be successful, if, after an injected damage, the aircraft remained stable for at least 20 seconds. After an initial parameter sensitivity analysis, we selected the parameters $w_p$, $w_q$, $w_r$, $K_{lat}$, and $\zeta$ for further analysis, where the $w_i$ are proportional gains of the controllers, $K_{lat}$ the lateral stick gain, and $\zeta$ a damping coefficient. We generated a combinatorial data set of 32,768 data points, out of which 7,992 runs were successful.

A boundary over these parameters exist in a shape of a hypersphere. This spherical shape is a consequence of the IFCS design, and the shape can be described by $(\frac{w_p-x_0}{\phi_1})^2 + (\frac{w_q-y_0}{\phi_2})^2 + (\frac{w_r-z_0}{\phi_3})^2 = \zeta \times \phi_0 - K_{lat}$. This stability boundary is parameterized by unknown $\phi_i$, $x_0$, $y_0$, $z_0$ are design-time constants.

Figure 10A shows the actual and estimated boundary in a projection into $w_p$, $w_q$, and $K_{lat}$. For our shape fitting and estimating experiment, we used 1000 initial data points. 5000 data points were selected by active learning. The shape parameters for the boundary in Figure 10B were estimated based upon 485 points near the boundary within an $\epsilon$-band of width 0.2.

7. Conclusions

In this paper, we addressed the discretization of sensor values for Fault Detection and Diagnosis systems. Traditionally, the wrapper code uses hypercube thresholds, which ignores non-linear threshold surfaces and dependencies between different system components and sensors. We described an advanced statistical methodology that uses Bayesian dynamic modeling and active learning techniques to detect and characterize threshold surfaces and shapes in a high-dimensional space. We presented an active learning algorithm, which can dras-
tically reduce the effort for determination and modeling of threshold surfaces.

Our Bayesian modeling approach for shape characterization and parameter estimation for the threshold surfaces incorporates domain knowledge in the form of a dictionary of suitable shape candidates. This enables insights for the domain expert and provides a way to implement compact and efficient wrapper codes for real-time diagnostic systems.

Future work will address metrics for the assessment of discretization quality with respect to false and missed alarms and will thus provide a statistical method to decide when hypercube-based thresholding is not sufficient. Of major importance will be the synergistic combination our active learning approach with shape estimation. Here, intermediate results from shape estimation could improve the candidate selection during active learning. Finally, we aim to evaluate our approach with a realistic FDD application.

REFERENCES


Biographies

Yuning He is a research scientist with the Robust Software Engineering Group in the Intelligent Systems Division (Code TI), at NASA Ames Research Center, employed by the University of California Santa Cruz. She obtained her PhD in Applied Mathematics and Statistics from UC Santa Cruz, and M.S. in Statistics from Stanford University. Her research interests include applied Bayesian statistics with emphasis on computer experiment, statistical learning, using statistical models and tools in combination with traditional formal methods for the analysis and verification and validation (V&V) of complex safety critical and online software systems.
Improved Fault Detection by Appropriate Control of Signal Bandwidth of the TSA

Eric Bechhoefer¹, and Xinghui Zhang²

¹GPMS Inc., President, Cornwall, VT, 05753, USA
eric@gpms-vt.com

²Mechanical Engineering College, Shijiazhuang, 050003, China
dynamicbt@aim.com

Abstract

Vibration analysis is perhaps the longest serving technology used in condition monitoring. It is usually assumed that higher sampling rates improve fault detection due to the increased bandwidth of the acquisition. That said, increased bandwidth may decrease the signal to noise, impairing fault detection. Alternatively, if the fault features’ bandwidth is greater than the system bandwidth, the fault cannot be observed. One tool of vibration analysis is the Time Synchronous Averaging (TSA) analysis. Statistics of the TSA itself can be used as a fault feature, or statistics based on analysis performed on the TSA (energy operator, residual analysis, amplitude/frequency modulation analysis) are used as fault features. Additionally, the computation of the TSA requires a tachometer signal for zero crossing, which has its own bandwidth effect on the TSA analysis. This paper discusses bandwidth control techniques to improve fault detection using the TSA. The techniques are validated using real world pinion data. These techniques have other advantages for embedded condition monitoring systems.

1. The Time Synchronous Average

Time Synchronous Averaging (TSA) was developed by (McFadden 1987), revolutionizing vibration analysis of rotating equipment. The TSA is integral for helicopter Health and Usage Monitoring System (HUMS) design (McInerny et al., 2003). Analyses, such as FM0, NA4, NB4, etc., based on the TSA, are now commonly used for vibration analysis in a number of different condition monitoring applications (Lebold et al 2000; Samuel and Pines 2005).

For a given shaft, the vibration model is defined as:

\[ x(t) = \sum_{k=1}^{K} X_k \left(1 + a_k \cos(2\pi k f_m(t) + \phi_k) + b(t) \right) \]

Where:

- \( X_k \) is the amplitude of the \( k \)th mesh harmonic,
- \( f_m(t) \) is the average mesh frequency,
- \( a_k(t) \) is the amplitude modulation function of the \( k \)th mesh harmonic,
- \( \phi_k(t) \) is the phase modulation function of the \( k \)th mesh harmonic,
- \( \Phi_k \) is the initial phase of harmonic k,
- While \( b(t) \) is additive background noise.

The mesh frequency is a function of the shaft rotational speed: \( f_m = Nf \), where \( N \) is the number of teeth on the gear and \( f \) is the shaft speed. Due to the finite bandwidth of the feedback control of the gearbox under analysis, or variation in power (as in the case of a wind turbine, where torque is a function of the wind speed) there will be variation in the shaft speed. This change in speed will result in smearing of amplitude energy in the frequency domain. The smearing effect, and non synchronous noise, is reduced by resampling the time domain signal into the angular domain:

\[ m_x(\theta) = E[x(\theta)] = m_x(\theta + \Theta) \]

The variable \( \Theta \) is the period of the cycle to which the gearbox operation is periodic, and \( E[x(\theta)] \) is the expectation (e.g. ensemble mean). The transformation from time \( t \) to \( (\theta) \) angle is done by resampling. For example, in one revolution, the time domain data is resampled (e.g. interpolated) such that the angular displacement over the shaft is constant. For example, if in a given revolution one sampled 827 data points, this would be interpolated to 1024 samples over the shaft revolution.

One important assumption is that \( m_x(\theta) \) is stationary and ergodic. This results in non-synchronous noise reduced by
1/rev, where rev is the number of cycles measured for the TSA. A number of interpolation techniques are available: linear, comb filter, polynomial or cubic spline. In (Decker, 1999), a comparison of these techniques was conducted: effectively no difference in TSA fault detection performance was found. However, there was a measurable difference found in computational loading. This researcher has found that spline and polynomial interpolation had an order of operation 6x of linear interpolation.

In (Bechhoefer, 2012), an enhancement for the TSA was developed, which allowed correction for changes in shaft speed within one revolution of the shaft under analysis. For the TSA using linear interpolation, the TSA can resample data that is linearly increasing or decreasing in shaft rate over the shaft revolution. Spline interpolation can resample for one change in shaft rate (e.g. sign change in dθ/dRev). Unfortunately, there are many cases where within one revolution, the shaft changes speed multiple times, such as

- the 2/revolution change in main shaft associated with blade flapping motion in the helicopter, or
- the 3/revolution change in main shaft associated with wind turbine tower shadow.

The inter-revolution resampling allowed accurate resampling of the shaft, which improved the performance of the TSA. This was seen in the Fourier domain with superior resolution of gear mesh frequencies and side band, especially at higher harmonics. Since side band modulation is one indication of gear fault, this technique resulted in improved gear fault detection.

2. Bandwidth Reduction of the TSA

Generally speaking, because the radix-2 Fast Fourier Transform (FFT) is easy to code and has less computation burden than an arbitrary length discrete Fourier transform, the TSA is implemented based on the radix-2 length. That is, the number of measured data points in one revolution is resampled to the next larger power of 2. The average number of points in one revolution is then a function of the acquisition system sample rate (sr) divided by the shaft rate (Hz):

\[ N\text{Points} = 2^{\text{ceil}(\log_2(sr/\text{Hz}))} \]  

(3)

For example, given a shaft rate of 6.57 Hz, and a sample rate of 48,828 samples/sec, the number of points in the TSA is 8192. On average, there are 48828 / 6.57 Hz = 7432 data points per revolution. These 7432 are interpolated to 8192 data points.

In the Fourier transform of the TSA, each bin represents 1 shaft order, e.g. 6.57 Hz. In the example, the pinion on this shaft has 28 teeth, so that in the spectrum of the TSA, bin 29 would (DC + 28) be the gear mesh energy. Nyquist for the sample data is sr/2 or 24,414 Hz. The Nyquist for the TSA would be 4096, but bins greater than 24,414/6.57 Hz = 3714 will have only noise.

More importantly, because of system jitter (due to the rising edge of the tachometer signal, limitations of the TSA algorithm, etc.), the practical limits of signals associated with the 28 tooth gear is no more than 10 or 20 harmonics. This means that in the Fourier domain, bins 560 to 4096 (in the example of a 8192 point TSA) are effectively noise.

In general, for a given white noise signal, the relationship between RMS noise and bandwidth is:

\[ \text{RMS} \propto \sqrt{\text{bandwidth}} \]

Thus, by reducing the bandwidth by a fourth (e.g. decimating the 8192 TSA to 2048), the RMS noise would be cut in half, or alternatively, this will improve the signal to noise ratio by 3 dB.

Because of the radix-2 length of the TSA, it is a simple matter to use Fourier domain decimation to reduce bandwidth of the TSA.

2.1. Mechanization Issues

Because of the relationship between sample rate, shaft rate and TSA length, one can ask the question as to why not use a lower sample rate? In the given example, the sample rate of 48,828 for the 6.57 Hz shaft resulted in an 8192 length TSA. The 1024 length TSA could be calculated directly by sampling at 6104 sampling per second. There are a number of reasons why this is not a good solution.

- For a given FFT bin, the noise is proportional to \(1 / \sqrt{n/2}\), where \(n\) is the FFT length. Since the bandwidth reduction occurs in the Fourier domain, the 8192 TSA reduces the noise in the FFT bin 0 through 511 by 4.5 dB over the 1024 point FFT.

- Further, Shaft and gear analysis is not performed alone, but with bearing analysis. Envelope analysis is the demodulation of high frequency resonance associated with bearing damage. The frequency is dependent of the bearing material stiffness and mass. Typically, one sees resonance for these bearing between 4 and 15 KHz. The Nyquist theorem requires the sample rate to be twice the highest frequency of interest. Sampling at 6104 would likely be too low to measure bearing faults.

- As an experiment, the raw time domain data was resampled from 48828 to 6104 sps. The resulting TSA displayed no fault indications. Evidently, features associated with the fault have a frequency greater than 3052 Hz.

The mechanization of efficient bandwidth reduction is performed by:

591
• Taking the real FFT (positive frequency only) of the TSA (Press et al., 1992)
• Zeroing the real and imaginary values from bin 512 to original TSA/2 length
• Taking the inverse FFT
• Reordering the TSA by mapping every, \( \rho^k = \text{TSA Length}/1024 \) (e.g., \( 8196/1024 = 8^k \)) data point into the new TSA.

Other advantages to this method are a large reduction overall order of operations. This is achieved because the TSA is the waveform that is used for all other gear analysis. Thus, reducing the TSA length to 1024, of 8196 (in this example) reduces the overall computation burden by a significant amount, since this all of the subsequent analysis are reduced by a factor of 8.

2.2. Tachometer Signal Jitter

A tachometer signal is used to synchronize the vibration data with the shaft position over time. If there is one target on the shaft under analysis (i.e., a key phasor), then the phase of a vibration signal can be calculated. However, it is usually the case that the tachometer signal is not taken from the shaft under analysis, and that there is more than 1 target. For many applications, the speed sensor target is a pinion within the gearbox (e.g., 18 tooth pinion), or on an external shaft coupling.

It is assumed that the spacing of the targets is uniformly spaced. Additionally, it is assumed that the sensor itself will trigger identically as target passes it. If either of these assumptions is poor, the jitter this adds to the zero crossing time will degrade high frequency components of the TSA. In Figure 1, the tachometer target is taken for a 3 point shaft coupling.

2.3. Effect of Bandwidth on the Narrowband Analysis

For gear tooth faults, a powerful analysis is the Amplitude and Phase Demodulation (McFadden, 1986). The analysis is based on an ideal band pass filter is applied to the TSA (defined as the Narrowband, (NB) signal). The amplitude-demodulated (AM) signal is derived by taking the absolute value of the Hilbert transform of the NB signal. The phase demodulated (FM) signal is calculated from the pseudo derivative of the argument of the Narrowband signal. The argument is calculated by taking the arctangent of the ratio of the imaginary to real component of the Hilbert transform of the NB signal.

The selection of the bandwidth of the NB is a somewhat ad-hock method, based more on experience than theory. Note that the feature of the AM/FM analysis is different than that of TSA statistics or energy operator/residual. These analyses are sensitive to 1/Rev impact events, such as seen with a soft/cracked tooth (Figure 1).

Consider the phenomenology of a soft tooth. When the tooth comes into the load zone (e.g., engaging mesh), because the tooth has less stiffness, the load is transferred to the other teeth in mesh contact. This results in two phenomena. The periodic displacement of the gears (which results in vibration) is dissimilar in the area surrounding the soft tooth (AM feature). The angular velocity of the gear will be dissimilar in the area surrounding the soft tooth (speeding up then slowing down, FM feature).

Thus, the bandwidth of the NB signal needs to be wide enough to capture this phenomenology, but narrow enough to filter noise not associated with the gear. This researcher typically used a bandwidth of 2. That is, when Fourier domain, all real/imaginary values greater than +/- 2 bins around the gear tooth, are set to zero, and then the inverse FFT is calculated.

For example, the following analysis can be performed on the TSA for gear fault detection:

• Gear Residual Analysis (1 FFT, 1 IFFT)
• Gear Narrowband Analysis (1 FFT, 1 IFFT)
• Gear AM Analysis (1 Hilbert Transform)
• Gear FM Analysis (1 Hilbert Transform)
• Each Hilbert transform calls a FFT, and the IFFT.

3. Case Studies

These examples are taken from operational wind turbines, where the site ID is identified as A, B or C. Site A has gear fault in low speed shaft, whereas site B and C are nominal. The low speed shaft has a frequency 6.57 Hz and the pinion on it has 28 teeth. The sampling frequency is 48,828 Hz.
There are 3 pulses per revolution for the tachometer signal. From Eq. 3, the TSA length is 8192. Clearly, the fault is more visible as the TSA is decimated to 4096, 2048 and finally to 1024 points (Figure 2).

Figure 2. TSA at 8192, 4096 (1.5 dB), 2048 (3 dB) and 1024 (4.5 dB)

The 8x reduction in the TSA length (e.g. 4.5 dB improvement to SNR) at approximately 0.7 Rev. Note that the kurtosis went from 3.3975 in the 8196 point TSA to 4.1567 for the 1024 point TSA. Similar improvements to the Energy Operator signal were observed. Note that with the reduction in bandwidth, the fault is clearly seen in the energy operator. The kurtosis for the energy operator (8196 point) went from 7.8 to 30.4. The large increase in kurtosis improves the probability of fault detection.

Figure 3. Energy Operator of the TSA at 8192, 4096, 2048 and 1024 points

The improved probability of detection can be demonstrated by comparing the condition indicator population statistics for residual kurtosis. Here the bad pinion (A) is compared to platforms with nominal gear (B and C). Note that large difference in the population of the residual kurtosis in going 8196 to 1024 points. At 8196, there is effectively no difference between the CI. This would result in a missed detection. However, 4.5 dB increase in SNR significantly improves the fault detectability. Without bandwidth reduction, the fault cannot be detected from the population of the bad pinion.
3.1. Effect of Jitter on the TSA

In this example, the tachometer target is 3 pulses per rev, where it is assumed that the targets are evenly spaced at 120 degrees. The decimated tachometer (i.e., using every 3rd pulse, so that there is in effect 1 pulse per revolution) TSA has more energy than the full bandwidth TSA (Figure 5).

3.2. AM/FM Analysis Verification using High Speed Gear Fault Data

A recent high-speed pinion fault was detected on a 3 MW wind turbine. Raw data was collected once per day on the damage pinion, and nominal data was taken from two other machines. This allows an investigation of determining the bandwidth of the NB signal, which maximized the separation between the bad pinion from the good ones. The number of teeth on the pinion is 32. The sampling frequency is 97,656 Hz. The tachometer takeoff had 8 pulses per revolution. Figure 7 compares a bad pinion to the nominal pinion for NB bandwidth from 1 to 10 bins (i.e., changing the bandwidth from 1 to 10 of the Narrowband Signal) for AM RMS analysis. Figure 8 is the comparison of FM RMS analysis. With the increase in the bandwidth, the separation between the bad pinion and the nominal pinions increases. While the fault can be detected with a bandwidth of 2, the separability is much greater at higher bandwidth. This means that the fault is more reliable detected with AM/FM analysis with a higher bandwidth, such at 8 or 10.

From this analysis, it is suggested the bandwidth of floor (number of teeth/4) should be used. For this example bandwidth of 8 is used given the 32 teeth pinion. The AM and FM analysis of the damage pinion is compared to a nominal pinion in Figure 9.

Figure 9 is instructive that the relationship of the damage tooth (seen at 0.7 revolution in the TSA) and the distribution of load (AM analysis) and frequency modulation (FM analysis) are clearly observable. Note the phase change of the pinion, indicating the pinion slowed then increase in speed as the gear mesh passes over the damage area of the pinion. Figure 10 is an image of the damage pinion.
4. Conclusion

The control of bandwidth on synchronous analysis, such as the TSA, can greatly improve the ability to detect fault. Noise, and therefore signal to noise, is proportional to the square root of bandwidth. However, it must be noted that reducing bandwidth too much may in fact filter the fault signature out of the analysis (case in point bandwidth of 2...
vs. 8 on the AM/FM analysis). For the TSA and associated analysis (energy operator, residual) the bandwidth of the TSA should not be less than 10 points per tooth. Thus, for a shaft with a 32 tooth pinion, the TSA bandwidth should not be reduced below 512 points. For ring gear, 111 tooth (for example) the TSA should not be less than 1024 points. For AM/FM analysis, the narrowband bandwidth should be approximately 25% the number of teeth. The bandwidth here should be great enough to capture the feature (change in tooth loading and phase) in order to maximize fault detectability.

REFERENCES


BIographies

Eric Bechhoefer is the president of GPMS, Inc., a company focused on the development of low cost condition monitoring systems. Dr. Bechhoefer is the author of over 100+ juried papers on condition monitoring and prognostics health management, and holds 23 patents in the field of CBM.

Xinghui Zhang received the BS and MS degree in Mechanical Engineering College of Shijiazhuang, China, in 2005 and 2010, respectively. He is a full time PhD student of Mechanical Engineering College, Shijiazhuang, China. He is also a worker of communication training center of Beijing, China. His current research interests include mechanical fault diagnosis, fault prognosis and performance based contracts.
Abort Trigger False Positive and False Negative Analysis Methodology for Threshold-based Abort Detection

Kevin J. Melcher¹, José A. Cruz², Stephen B. Johnson³, and Yunnhon Lo⁴

¹,²NASA Glenn Research Center, Cleveland, OH, 44135, U.S.A.
kevin.j.melcher@nasa.gov
jose.a.cruz@nasa.gov

³Jacobs ESSSA Group / Dependable System Technologies, LLC, Larkspur, CO, 80118, U.S.A.
stephen.b.johnson@nasa.gov

⁴Jacobs ESSSA Group / Ducommun Miltec, Huntsville, AL, 35806, U.S.A.
yohon.lo@nasa.gov

ABSTRACT

This paper describes a quantitative methodology for bounding the false positive (FP) and false negative (FN) probabilities associated with a human-rated launch vehicle abort trigger (AT) that includes sensor data qualification (SDQ). In this context, an AT is a hardware and software mechanism designed to detect the existence of a specific abort condition. Also, SDQ is an algorithmic approach used to identify sensor data suspected of being corrupt so that suspect data does not adversely affect an AT’s detection capability. The FP and FN methodologies presented here were developed to support estimation of the probabilities of loss of crew and loss of mission for the Space Launch System (SLS) which is being developed by the National Aeronautics and Space Administration (NASA). The paper provides a brief overview of system health management as being an extension of control theory; and describes how ATs and the calculation of FP and FN probabilities relate to this theory. The discussion leads to a detailed presentation of the FP and FN methodology and an example showing how the FP and FN calculations are performed. This detailed presentation includes a methodology for calculating the change in FP and FN probabilities that result from including SDQ in the AT architecture. To avoid proprietary and sensitive data issues, the example incorporates a mixture of open literature and fictitious reliability data. Results presented in the paper demonstrate the effectiveness of the approach in providing quantitative estimates that bound the probability of a FP or FN abort determination.

Kevin J. Melcher, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

This paper describes a quantitative methodology for bounding the false positive (FP) and false negative (FN) probabilities associated with abort triggers (ATs) that include sensor data qualification and constant abort detection thresholds during a given phase of flight. The methodology was developed to support the verification of design requirements for NASA’s Space Launch System (SLS).

Flight systems may have thousands of failure modes. These failure modes – typically identified via a failure modes and effects analysis (FMEA) – can be broadly classified as “hard” failures and “soft” failures. Hard failures occur rapidly and typically result in sustained large changes in the measured system states, e.g., a sensor failing to zero or to full-scale. Soft failures occur more slowly and typically result in gradual changes in the measured system states, e.g., a sensor drift that results in an intermediate value between zero and full-scale. Hard failures are fairly easy to detect while soft failures can be difficult to detect without significant FP and FN results. However, many failure modes are defined broadly enough that they are difficult to classify as either hard or soft failures. For example, a power supply failure may result in no power or reduced power depending on the exact nature of the failure.

The uncertainty associated with understanding the impact of failures on the abort detection system provides motivation for bounding the FP and FN probabilities. A Monte-Carlo simulation and physics-based model of the system are typically used to estimate FP and FN rates. However, significant time and effort are required to develop the simulation and conduct this kind of analysis. A more cost-effective and sufficiently accurate approach for SLS purposes is to use a simpler bounding estimate. In the approach proposed here, the failure rates and probabilities associated
with both soft failures and broadly-defined failure modes are first classified as failure-to-intermediate value (F2IV). F2IV values are then allocated to both failure to zero (F2Z) and failure to full-scale (F2FS). The aggregate F2Z and F2FS data then become the basis for calculating bounds on the FP and FN probabilities for a given AT.

To facilitate the discussion embodied in this paper, a clear understanding of the following terms is necessary.

Abort condition (AC): The state or behavior of a launch vehicle which indicates that a threat to the crew exists and that an abort response is required to mitigate the threat. A successful abort response during ascent enables the crew to escape from a failed or failing vehicle and return safely to Earth.

Abort trigger (AT): A mechanism that is used to detect an AC. Each AT includes all of the hardware and software components required to detect a specific AC. The success or failure of an AT is ultimately measured by the probability that the crew returns safely to Earth when vehicle system failures threaten their safety.

Defined below, two key attributes of an AT’s performance are the probability of a FP detection and the probability of a FN detection. Ideally, these probabilities will be zero or an acceptably low value.

False positive (FP): Occurs when, despite the fact that an AC does not exist, the associated AT indicates that it has detected the AC and sends an abort recommendation.

False negative (FN): Occurs when an AC exists and the associated AT does not detect the AC.

Sensor data qualification (SDQ): This is software that monitors the sensor data at the Flight Computer (FC). It classifies data suspected of being corrupt as disqualified. Disqualified data are not used by ATs and, consequently, do not adversely affect an AT’s detection of its associated AC. SDQ is intended to reduce the probability of FPs and FNs.

Abort Condition Detection Logic (ACDL): The ACDL is part of the AT software. On each FC, it compares the consolidated value to an AC detection threshold. If the consolidated value exceeds the threshold on a given FC for a pre-specified persistence period, that FC makes an immediate abort recommendation.

Sensor data consolidation (SDC): These are algorithms that combine multiple time-synchronous measurements into a single data value that is typically used by higher-level operations and control algorithms, e.g. ADCL.

This paper assumes that the threshold values used in abort detection result from analyses not discussed in this paper. From an academic point of view, the selection of abort detection thresholds has previously been addressed by a number of authors including Yachtsevanos, Lewis, Roemer, Hess, & Wu (2006).

This paper is organized as follows. Section 2 provides a brief overview of the theory behind fault management as an extension of control theory. It describes how this theory applies to ATs and the calculation of FP and FN probabilities. In Section 3, a methodology for calculating FP and FN probabilities for threshold-based ATs is described. Section 4 presents an example showing how the FP and FN calculations are performed in practice using the methodology described in Sec. 3. Because actual SLS data cannot be disclosed for general publication, a combination of fictitious and open-literature reliability data provide the basis for this example. In Sec. 5 Discussion, observations about the methodology and modifications toward improving the approach are discussed. Concluding remarks are presented in Section 6 which gives a summary of the paper and briefly describes plans for applying the methodology to the SLS.

2. BACKGROUND

The term System Health Management (SHM) addresses activities that are described under several names, including: Prognostics and Health Management; Fault Protection; Vehicle Health Monitoring and Management; Fault Detection, Isolation, and Response; Diagnostics; Maintainability; Reliability; Availability; aspects of Safety; as well as others. SHM has historically been a relatively ad hoc set of processes and technologies focused on predicting, detecting, diagnosing, and responding to failures. The core idea that the operational aspects of SHM are related to control theory goes back 20 years (Albert, Alyea, Cooper, Johnson, & Ulrich, 1995). More recently, a unifying theory of SHM was developed and published (Johnson & Day, 2010) (Johnson & Day, 2011) (Johnson, 2011) (Day & Johnson, 2014). This theory provides a conceptual framework for the field and for its operational subset, Fault Management (FM) theory. The unifying theory is based on the idea that FM theory and practice is essentially an extension of control theory and practice.

The purpose of SHM is to provide capabilities to preserve a system’s ability to function as intended. SHM can be divided into passive capabilities such as design margins, and operational capabilities such as failure detection, isolation, and response (FDIR). These latter operational capabilities, termed Fault Management, are implemented as control loops, known as FM control loops (FMCLs). The FMCL detects system degradation or failure, and then determines which part of the system has failed or will fail (prognosis). Here, failure implies that all or part of the system cannot be controlled within acceptable limits to achieve its objectives. Having detected or predicted a failure, FMCLs then decide what control action (response) to take. The objective being to return the system to a controllable state or take an action to prevent or mitigate the predicted failure (Johnson, 2011).

This extension to control theory is used in this paper to assess the failure detection portion of FMCLs in a human-rated
launch vehicle application. In control theory, state space control loops can be separated into two major portions: state estimation and state control. Calculation of overall control loop performance is also divided into two parts, with separate metrics to determine the performance of state estimation and state control. For FMCLs, state estimation can be measured and assessed using "confusion matrix" parameters: false positive (FP), false negative (FN), true positive (TP), and true negative (TN). State control success is based on the ability of the system to correctly determine the correct response action to take, and then assess the performance or effectiveness of that action. Effectiveness of the FM response typically estimated by comparing the speed of the FM response and the time available to correct for a current or impending failure. If the response completes before the failure effects compromises relevant systems goals, the response is effective; else, it is considered to be either less or not effective. This aspect of the use of ideas that extend control theory will not be pursued further in this paper.

For human-rated launch vehicles (LV), the effectiveness of the FM mechanisms called ATs are measured in terms of their ability to protect the crew, which is estimated by determining the change in loss of crew (LOC) probability that occurs if an AT or suite of ATs is implemented. Taking classical control theory concepts of state estimation and state control, the metric of this change in LOC probability, the LOC Benefit, is calculated by subdividing it into state estimation and state control elements. These are calculated separately and used to calculate the LOC Benefit numbers associated with proposed AT implementations. The LOC Benefit value provides a quantitative basis for deciding which ATs will be provided on the human-rated LV and for measuring their effectiveness in particular scenarios and across all relevant scenarios.

A human-rated LV can have many ATs with varying types of failure detection approaches. Calculation of the LOC Benefit contribution for each AT is key to an accurate accounting of the total LOC probability. The sum of the LOC Benefit of each AT across all relevant scenarios provides the LOC Benefit of the entire suite of ATs, which is the measure of their benefit to the system.

3. FP AND FN METHODOLOGY

In this section, a general methodology is briefly described for (a) quantitatively determining the performance of threshold-based ATs used to detect abort conditions and (b) estimating the improvement or degradation of that performance due to the inclusion of SDQ (Maul, Melcher, Chicatelli, & Sowers, 2006) as a component of the AT.

Quantitatively estimating the probabilities of FP and FN abort detections is crucial. High FP and FN probabilities indicate that the AT has high loss of mission (LOM) costs, or is ineffective (i.e., fails to decrease Loss of Crew probability), and, hence, should not be incorporated into the design at all.

A methodology for calculating FP and FN probabilities for threshold-based ATs is composed of the following five steps.

1. Define the AT – Construct a functional block diagram integrating all of the hardware and software components included in the AT architecture. The diagram is useful for understanding the data flow from each sensor to the AC Detection Logic (ACDL).

2. Analyze the Physics of Failure – Analyze how failures upstream of the ACDL can result in FP and FN detections. This analysis is helpful in understanding the effects of redundancy on the FP and FN probabilities of ATs.

3. Determine Bounds on Component Failure Probabilities – Calculate the probability of failure for each component included in the AT architecture. Here, “component” is a general term used to describe the individual functional blocks that comprise the AT architecture, which in the case of SLS is composed of both hardware and software. A list of failure modes and their probability of occurrence are required to complete Step 4.

4. Conduct Analysis of FP and FN Probabilities for the Baseline System – In this step, a fault tree (FT) is created to estimate the probability of a FP or FN abort detection based on the probability that known failure modes may occur. The FT is typically created using an available Probabilistic Risk Assessment (PRA) software tool which is programmed to analyze the AT failure space.

5. Determine the Benefit Provided by SDQ – This step is similar to Step 4, however, the analysis is focused on an AT architecture that includes the SDQ function. Resulting FP and FN abort probabilities are subtracted from those for the baseline AT calculations (step 4) to calculate the FP and FN benefit of the SDQ function.

The novelty of this methodology is as follows. First, PRA and reliability block diagrams are applied to the failure detection problem of FP and FN. Second, the benefit of SDQ is estimated as part of a failure detection process. Third, these methods are developed and applied to the failure detection portion of FMCLs within the overall theory of SHM and FM. To see an example of performance calculations for entire FMCLs for the human-rated launch vehicle application; and thus, how the FP and FN calculations fit into the overall LOC benefit calculation, see (Lo, Johnson, & Breckenridge, 2014).

In general, the calculations described in this paper are a key part of the LOC benefit analysis used to estimate the value of ATs. These calculations help to quantitatively determine whether or not SDQ algorithms are beneficial to ATs. This allows the AT design to be optimized and reduces unnecessary design complexity.

4. APPLICATION OF FP AND FN METHODOLOGY

In this section, the methodology described in Sec. 3 is applied to a generic AT designed to detect a low-pressure AC. The
description is provided as a practical example of how the FP and FN methodology may be used to detect and respond to an AC resulting, for instance, from a propellant leak. To avoid data proprietary and sensitivity issues associated with the SLS, actual SLS data are not used. Instead, a mixture of open literature and fictitious data are utilized.

Human-rated flight hardware typically contains significant redundancy to protect the crew. The example is intended to illustrate that redundancy without duplicating it. Further, although the example presented here is intended to be simple for illustrative purposes, it should be fairly easy to see how the complexity of an AT in a real system can escalate.

4.1. Step 1: Define the AT

As a first step, it is necessary to identify all of the hardware and software components required to detect a specific AC. These components comprise the AT – both collectively and through the manner in which they are connected (i.e., the architecture). As part of the subsequent methodology for estimating SDQ benefit, a baseline AT that does not include SDQ is required. The baseline AT is used to determine the reduction in the FP and FN probabilities provided by SDQ. This is accomplished by comparing results for an AT that includes SDQ against results for a baseline AT.

The main function of the AT presented here is to monitor a pressure and provide actionable knowledge to the crew so they can initiate an abort action if necessary. Low pressure conditions are a well-known issue for liquid-propellant-based LVs. Inordinately low propellant tank pressures during flight are indicative of a course of events that may result in catastrophic explosions with loss of the vehicle and/or crew.

Figure 1 presents the baseline functional block diagram of the architecture for the threshold-based AT used in this paper. To facilitate the calculation of the SDQ benefit, the baseline architecture does not include the SDQ function. Figure 2 presents a block diagram for the same AT, but with the addition of SDQ. In the following discussion, previously undefined elements of the AT diagrams are described and the specific SDC implementation is detailed.

Pressure Sensors (PSs): The AT architecture contains four (4) redundant pressure sensors – PS1, PS2, PS3, and PS4. Each pressure sensor transducer generates analog voltage signals proportional to the sensed pressure. Said signals are inputs to the Sensor Electronics (SE). PS1 and PS2 are connected to SE1, while PS3 and PS4 are similarly connected to SE2.

Sensor Electronics (SEs): There are two sets of SEs which include (a) signal conditioning equipment required to power the pressure sensors, (b) hardware and firmware required to digitize and discretize the sensor’s analog signal, and (c) hardware and firmware required to interface to a digital data bus. The SE outputs for each sensor are cross-strapped to each of the FCs via the data buses.

Flight Computers (FCs): There are three (3) FCs – FC1, FC2, and FC3. The FC functional block represents both hardware and software implemented to support operation of the launch vehicle.

Sensor Data Consolidation (SDC): For the baseline system shown in Fig. 1, sensor measurements PS1, PS2, PS3, and PS4 are averaged on each FC to obtain a single consolidated measurement that is used by the ACDL. Averaging was selected as the SDC algorithm to simplify the example. Other approaches (e.g., mid value select) could be used in place of averaging.

Some broad assumptions and ground rules that are used to analyze this example AT follow.

- The mission time is 10 minutes or 0.166 hrs.
During the mission, components are considered to be in either an operational or failed state. In other words, an AT with a degraded response is not considered.

The AT is single fault tolerant with respect to the SE and FC components:
- At least one (1) properly functioning SE component is needed to complete the mission.
- At least two (2) properly functioning FCs – includes both hardware and software components – are needed to complete the mission.

Additionally, the AT is two fault tolerant with respect to PS components. At least two (2) properly functioning PSs are needed to complete the mission.

Random part failure and the common cause failure (CCF) of redundant components are considered.

Single-point estimates, rather than distributions, are used to represent component failure rates. This simplifies the analysis and discussion of the results.

The limit of resolution of the analysis is at the component level. Analysis is not performed below that level.

### 4.2. Step 2: Analyze the Physics of Failure

Given a complete description of the components and architecture of the AT, the next step is to develop a clear understanding of the failure modes associated with those components and how the physics of failure may modify data used by the ACDL.

Here, an approach based on Receiver Operating Characteristic (ROC) theory (Vachtsevanos, et al, 2006) is used. The analysis was first simplified by defining three classifications for the effect of failures. Then, the impact of each of those classifications on the probability of a FP or FN abort detection was explored.

The probability of FP and FN aborts may be bounded by considering the following three common classifications for the effect of failures: Failure to Zero (F2Z), Failure to Intermediate Value (F2IV), and Failure to Full Scale (F2FS).

A discussion of each of these classifications follows and addresses the potential for the failure class to generate a FP or FN abort detection.

**Failure to Zero (F2Z)** – Occurs when data associated with one or more of the PSs fails to a value at or near zero. Small variations about zero may result from improper sensor calibration or from ambient noise. Further, when averaging is the consolidation algorithm – see Eqs. (1) and (2) – this failure classification has the effect of driving both the consolidated measurement value, \( u_{j,c} \), and the standard deviation of the consolidated measurement value, \( \sigma_{j,c} \), toward zero.

\[
    u_{j,c} = \frac{1}{4} \sum_{i=1}^{4} u_{j,i}
\]

\[
    \sigma_{j,c} = \frac{1}{4} \sum_{i=1}^{4} \sigma_{j,i}
\]

Here, \( u \) represents a measured or calculated system state; \( \sigma \) indicates the standard deviation of \( u \); \( i \) is an index associated with the individual data buses that deliver sensor data to the FCs; \( j \) is an index that indicates a specific FC; and “c” indicates that the associated value is the result of the SDC calculation. For the example presented in this paper, \( u_{\mu} \) and \( \sigma_{\mu} \) respectively represent the individual pressure measurements and their standard deviations.

**Failure to Full Scale (F2FS)** – Occurs when data associated with one or more of the PSs fails to a value at or near full-scale, F2FS will not contribute to a FP abort detection of low pressure. It may however, contribute to a FN abort detection if the following three conditions exist.

- a system failure has occurred, and
- the system failure results in a low pressure condition, and
- the value of the pressure data, \( u_{j,i} \), resulting from that failure are less than, yet sufficiently close to, the low pressure detection threshold, \( u_{TH} \).

If these three conditions exist, then an F2FS of one or more pressure sensor data signals will result in a FN abort detection.

**Failure to Intermediate Value (F2IV)** – This more complicated and often more likely case occurs when occurs when data associated with one or more of the PSs fails to values greater than zero but less than full-scale. This situation could be caused, for example, by a partially blocked sensing port or by intermittent short or open circuits. In reality, failures associated with this failure effect classification may or may not result in an AC. As a result, quantification of the FP and FN probabilities associated with these failures typically requires significant Monte Carlo analysis. Since tools and resources are not currently available to conduct the required analysis, other approaches are needed to estimate and bound the probabilities. One means of providing a conservative bound for assessing the FP rate is to attribute all of the F2IV failure rate to the F2Z classification. That is:

\[
    FR(F2Z)_{Upper\, Bound} = FR(F2Z) + FR(F2IV),
\]

where FR is the failure rate. Reasoning in a similar (but “opposite” in terms of using the data) manner for FN, one means of providing a conservative bound for assessing the FN rate is to attribute all of the F2IV failure rate to the F2FS classification, so that:

\[
    FR(F2FS)_{Upper\, Bound} = FR(F2FS) + FR(F2IV).
\]

For the purposes of the analyses described in this report, all F2IV probabilities are estimated conservatively as F2Z for FP calculations and F2FS for FN calculations. This logic follows from the observations that assigning F2IV cases to F2Z will always create a FP, and assigning F2IV to F2FS for FN calculations will always create a FN. As F2IV cases will in
reality only sometimes create these conditions, but at rates difficult to predict, we deliberately create overestimates of FP and FN cases to ensure a conservative estimate.

4.2.1. Failure to Zero and FP Analysis

To show the impact of an F2Z on the ACDL, cases for the F2Z of 0, 1, and 2 sensors were examined. Relevant parameters are identified in Table 1. A nominal value of \( u_{\text{nom}} = 40 \) pounds per square inch (psi) was selected for the pressure sensor and a value of \( \sigma_{\text{nom}} = 0.75 \) psi for the standard deviation. For F2Z sensor signals, both the signal value and the signal standard deviation are assumed to be zero. To calculate the probability that the consolidated pressure is less than the AC detection threshold, a Gaussian probability distribution is assumed and an AT threshold, \( u_{\text{TH}} = 25 \) psi, is used. The results for each of the three cases examined are given in Table 2. Of primary interest are the consolidated values, \( u_{jc} \) and \( \sigma_{jc} \), and \( P(u_{jc} < u_{\text{TH}}) \) which is the probability that an F2Z of the signals will result in a FP abort detection.

Table 1. Parameters used in example AT for FP analysis.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_{LFS} )</td>
<td>60 psi</td>
<td>Nominal pressure</td>
<td></td>
</tr>
<tr>
<td>( u_{Lnom} )</td>
<td>40 psi</td>
<td>Full-scale pressure</td>
<td></td>
</tr>
<tr>
<td>( \sigma_{Lnom} )</td>
<td>0.75 psi</td>
<td>Standard deviation of nominal pressure</td>
<td></td>
</tr>
<tr>
<td>( u_{TH} )</td>
<td>25 psi</td>
<td>Detection threshold for low pressure AC</td>
<td></td>
</tr>
</tbody>
</table>

Note here the effect of failures on the value of the consolidated signal. As more signals F2Z, both the consolidated signal value and the consolidated standard deviation move closer to zero. Also, for this example, an F2Z does not result in an overlap between the nominal and failed probability distributions as would be typical for F2IV.

Further, Fig. 3 shows the probability distribution of the consolidated signal, \( u_{jc} \), for no F2Z signals, for one F2Z signal, and for two F2Z signals. An important observation from both Table 2 and Fig. 3 is that the AT is single fault tolerant. The F2Z of a single sensor data signal is not sufficient to cause a FP abort detection. The F2Z of two or more sensor data signals on the same FC are required to generate a FP abort detection.

4.2.2. Failure to Full-scale and FN Analysis

The impact of F2FS on the ACDL is illustrated by looking at cases for 0, 1, and 2 sensors failing to full-scale. As shown in Table 3, a nominal value of \( u_{Lnom} = 20 \) psi was selected for the pressure sensor and a value of \( \sigma_{Lnom} = 0.75 \) psi for the standard deviation. The nominal value is assumed to be the result of an AC, as an AC must exist for a FN to occur. For F2FS sensor signals, the signal value and standard deviation are assumed to be 60 psi and 0.75 psi, respectively. To calculate the probability that the consolidated pressure is greater than the AC detection threshold, a Gaussian

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Data Values:</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>No. Nominal Data Values:</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>No. F2Z Data Values:</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>( u_{jc} )</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>( u_{jc} )</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>( u_{jc} )</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>( u_{jc} )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>0.75</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma_{jc} )</td>
<td>0.56</td>
<td>0.38</td>
</tr>
<tr>
<td>( P(u_{jc} &lt; u_{TH}) )</td>
<td>3.08E-19</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

![Figure 3. Probability distribution vs. pressure for \( u_{jc} \) given 0, 1, and 2 pressure signals failing to zero without the application of SDQ.](image-url)
probability distribution and an AT threshold value of $u_{TH} = 25$ psi were also assumed. Results for each of the three cases examined are given in Table 4. Of primary interest are the consolidated values, $u_{j,c}$ and $\sigma_{j,c}$, and $P(u_{j,c} > u_{TH})$ which is the probability that an F2FS of the signals will result in a FN abort detection.

Further, Fig. 4 shows the probability distribution of the consolidated signal, $u_{j,c}$, for the three cases. An important observation from both Table 4 and Fig. 4 is that, if the low pressure AC exists and the pressure is sufficiently close to the detection threshold, the F2FS of a single sensor is enough to generate a FN abort detection.

### 4.3. Step 3: Determine Bounds on Component Failure Probabilities

The overall goal of this step is to calculate the probability of failure for each component that is part of the AT architecture. The process for accomplishing this goal is described below.

**Step 3.1** Identify the failure modes and associated failure rates for each component in the AT architecture. Failure modes and failure rates (i.e., reliability data) are typically determined by referencing a system-specific failure modes and effects analysis or similar documents. Reliability data used in this paper are presumed not to include the flight application software.

**Step 3.2** Classify the effect of each failure mode identified in Step 3.1 as F2Z, F2IV, or F2FS. This is typically accomplished through discussions with one or more subject matter experts who understand the failure modes and the impact of those failures on the data used to detect a given AC.

**Step 3.3** Conservative (i.e., upper) bounds for the component’s F2Z and F2FS probabilities are determined. To do this, failure rates classified as F2Z are only allocated to the F2Z rate (i.e., F2Z per hour). Those classified as F2FS are only allocated to the F2FS rate. And, those classified as F2IV are allocated to both the F2Z and F2FS rates. An example showing how this is done for the F2Z case is given in Table 5 where, for the Electrical Short failure, the failure rate (col. 2) is allocated to F2Z rate (col. 6), while a failure rate of zero is allocated to the F2FS rate (col. 7). Similarly, an example of how this is done for the F2FS case is shown in Table 6 where, for the High Voltage failure, the failure rate (col. 2) is allocated to F2FS rate (col. 7), while a failure rate of zero is allocated to the F2Z rate (col. 6). Finally, an example showing allocation for the F2IV case is given in Table 5 where, for the degraded failure, the failure rate is allocated to both the F2Z rate and the F2FS rate.

**Step 3.4** Calculate the F2Z and F2FS total failure rates for each component by summing the rates for each failure mode in cols. 6 and 7, respectively. In Table 5, the total F2Z rate is 4.2E-05 failures per hour and the F2FS rate is 3.16E-05 failures per hour.

### Table 3. Impact of F2FS on example ADCL.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Data Values:</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>No. Nominal Data Values:</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>No. F2FS Data Values:</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$u_{j,1}$</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$u_{j,2}$</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$u_{j,3}$</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>$u_{j,4}$</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>$u_{j,c}$</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>$\sigma_{j,c}$</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>$P(u_{j,c} &gt; u_{TH})$</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

### Table 4. Parameters used in example AT for FN analysis.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_{FS}$</td>
<td>60</td>
<td>psi</td>
<td>Full-scale pressure</td>
</tr>
<tr>
<td>$u_{nom}$</td>
<td>20</td>
<td>psi</td>
<td>Nominal pressure</td>
</tr>
<tr>
<td>$\sigma_{nom}$</td>
<td>0.75</td>
<td>psi</td>
<td>Standard deviation of nominal pressure</td>
</tr>
<tr>
<td>$u_{TH}$</td>
<td>25</td>
<td>psi</td>
<td>Detection threshold for low pressure AC</td>
</tr>
</tbody>
</table>

![Figure 4. Probability distribution vs. pressure for $u_{j,c}$ given 0, 1, and 2 pressure signals failing to full-scale without the application of SDQ.](image-url)
Table 5. Failure modes for the example PS showing the contribution of each failure mode to conservative bounds for F2Z and F2FS classifications.

<table>
<thead>
<tr>
<th>Failure Mode</th>
<th>Failures per hour</th>
<th>Quantitative impact of failure mode on the component output signal</th>
<th>Conservative Upper Bound</th>
<th>F2Z per hour</th>
<th>F2FS per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Short</td>
<td>3.500E-06</td>
<td>X</td>
<td></td>
<td>3.500E-06</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>No Output</td>
<td>6.900E-06</td>
<td>X</td>
<td></td>
<td>6.900E-06</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>Cracked or Fractured</td>
<td>3.500E-06</td>
<td>X</td>
<td></td>
<td>3.500E-06</td>
<td>3.500E-06</td>
</tr>
<tr>
<td>Degraded</td>
<td>2.810E-05</td>
<td>X</td>
<td></td>
<td>2.810E-05</td>
<td>2.810E-05</td>
</tr>
<tr>
<td>Totals: 4.200E-05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating Hours: 0.1667</th>
<th>Totals: 4.200E-05</th>
<th>3.160E-05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of Failure: 7.000E-06</td>
<td>5.267E-06</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Failure modes for the example SE showing the contribution of each failure mode to conservative bounds for F2Z and F2FS classifications.

<table>
<thead>
<tr>
<th>Failure Mode</th>
<th>Failures per hour</th>
<th>Quantitative impact of failure mode on the component output signal</th>
<th>Conservative Upper Bound</th>
<th>F2Z per hour</th>
<th>F2FS per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defective Component</td>
<td>4.290E-07</td>
<td>X</td>
<td></td>
<td>4.290E-07</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>Fails During Operation</td>
<td>1.430E-07</td>
<td>X</td>
<td></td>
<td>1.430E-07</td>
<td>1.430E-07</td>
</tr>
<tr>
<td>Connection Failure</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>Failed to Operate</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>7.133E-08</td>
</tr>
<tr>
<td>High Voltage</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>7.133E-08</td>
</tr>
<tr>
<td>Improper Output</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>7.133E-08</td>
</tr>
<tr>
<td>Inoperative</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>7.133E-08</td>
</tr>
<tr>
<td>Logic Fault</td>
<td>7.133E-08</td>
<td>X</td>
<td></td>
<td>7.133E-08</td>
<td>7.133E-08</td>
</tr>
<tr>
<td>Totals: 1.000E-06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating Hours: 0.1667</th>
<th>Totals: 9.287E-07</th>
<th>3.570E-07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of Failure: 1.548E-07</td>
<td>5.950E-08</td>
<td></td>
</tr>
</tbody>
</table>

failures per hour. These are shown in the totals row in cols. 6 and 7, respectively.

Step 3.5 Multiply the conservative total failure rate for the F2Z and F2FS classifications by the time in the mission phase to obtain the probability of failure for each classification. The goal of this step is to calculate the probability of F2Z and F2FS for each of the components represented in Tables 5 through 8. To do that, the failure rates calculated in the previous step are multiplied by the operating time—for this example 10 minutes or 0.166 hours is assumed. Results for probability of failure calculations are given in the last row of cols. 6 and 7 in each of the failure mode tables.

When appropriate, Steps 3.1 through 3.5 may be repeated for each mission phase.

4.4. Step 4: Conduct Analysis of FP and FN Probabilities for the Baseline System

In this Section, the methodologies and modeling approaches used to derive the probability of a FP and a FN abort recommendation are discussed. A FT analysis methodology was used to provide a systematic means of identifying system component failure events that lead to these undesired recommendations.

4.4.1. Fault Tree Development

FT models were constructed using the NASA fault tree analysis guidelines (Stamatelatos and Homayoon, 2011). The Systems Analysis Programs for Hands-on Integrated
Reliability Evaluations (SAPHIRE) software was used to generate the failure combination of events that lead to a FP or FN abort recommendation, quantify probability of those recommendations, and identify the major failure contributors or risk drivers to those recommendations. SAPHIRE is a publically-available, government-developed software tool that is useful for performing Probabilistic Risk Assessment (PRA). SAPHIRE is documented in a number of reports including a summary manual by the NRC (Wood, Smith, Kvarfordt, & Beck, 2008). The SAPHIRE FT is not shown due to complexity and space limitations.

### 4.4.2. Common Cause Event Modeling

Common Cause Failure (CCF) events are accounted for in the SAPHIRE FT model. CCFs have been shown by many reliability studies to contribute significantly to the overall unreliability of complex systems. A CCF event is defined as the failure of multiple redundant components due to shared causes. The incorporation of CCF events into the FT model results in more realistic estimates of system unreliability. In this work, CCF events are modeled in the FT to account for the possible failure of AT components due to external causes. For example, multiple FCs might fail simultaneously or generate erroneous signal output indicating the occurrence of an abnormal system state. This type of failure event can be caused by loose connections of interface cables. Cable connection errors can be attributed to installation or assembly errors (human error), high levels of vibration during launch vehicle ascent, or by design faults in FC hardware, firmware or software. To reduce the underestimation of probabilities for FP and FN abort recommendations, combinations of multiple CCF events were considered for each AT.
component. The CCF probability equations (Mosleh, Rasmussen, & Marshall, 1998) and associated alpha factor values (Atwood, Kelly, Marshall, Prawdzik, & Stetkar 1996) used in this study are given in Table 9.

4.4.3. Estimation Approach for FP and FN Abort Detection

The methodology in this section is developed by first considering the FN case. The occurrence of a FN detection depends on the occurrence of two events.

1. A system failure of sufficient magnitude to exceed prescribed detection thresholds and
2. A failure of the AT to detect that system failure.

The occurrence of a FN event may then be represented using the following Boolean algebraic expression:

\[ \text{FN} = \text{AC} \cap \text{AT}\mid \text{AC}. \]  

Here, FN is true if an abnormality event occurred and an abort trigger occurred given that an abort condition is true.

The probability of a FN event is given by,

\[ P(\text{FN}) = P(\text{AC} \cap \text{AT}\mid \text{AC}) = P(\text{AC}) \times P(\text{AT}\mid \text{AC}). \]  

Here \( P(\text{AC}) \) denotes the probability of an AC and \( P(\text{AT}\mid \text{AC}) \) denotes the conditional probability of failure of the AT given that an AC event has occurred.

If various ACs are considered, a general expression for the overall system probability of a FN detection can be obtained by applying the additive rule of probability as shown below. This expression assumes the occurrences of FN scenarios are mutually exclusive.

\[ P(\text{FN}) = \sum_{k=1}^{n} P(\text{AC}_k) \times P(\text{AT}\mid \text{AC}_k) \]  

For the remainder of this paper, \( \text{AC}_k = 1 \) implies that the probability a given AC will occur was accounted for as part of a separate analysis. This approach has the added benefit that the structure and failure logic of FP and FN events become identical. As a consequence, the SAPHIRE model and results used to analyze a FN abort recommendation may also be used to estimate the probability of a FP abort recommendation.

### Table 9. CCF probability equations and \( \alpha \) values for CCF alpha factor model (non-staggered testing scheme).

<table>
<thead>
<tr>
<th>Success Configuration (k-out of- n)</th>
<th>Common Cause Failure Probability Equations</th>
<th>( \alpha ) factor Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 out 2 (CCSE)</td>
<td>( P(\text{CCF}_2) = \frac{a_2}{1.0257} \times P_t )</td>
<td>( \alpha_1 = 0.97430 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha_2 = 0.02570 )</td>
</tr>
<tr>
<td>2 out 3 (FC Hardware/FC Software)</td>
<td>( P(\text{CCF}_2) = \frac{1}{2} a_2 \times \frac{1.0303}{P_t} )</td>
<td>( \alpha_1 = 0.97550 )</td>
</tr>
<tr>
<td></td>
<td>( P(\text{CCF}_3) = \frac{a_3}{1.0303} \times P_t )</td>
<td>( \alpha_2 = 0.01870 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha_3 = 0.00579 )</td>
</tr>
<tr>
<td>2 of 4 (PS)</td>
<td>( P(\text{CCF}_2) = \frac{1}{3} a_2 \times \frac{1.0376}{P_t} )</td>
<td>( \alpha_1 = 0.97410 )</td>
</tr>
<tr>
<td></td>
<td>( P(\text{CCF}_3) = \frac{1}{3} a_3 \times \frac{1.0376}{P_t} )</td>
<td>( \alpha_2 = 0.01700 )</td>
</tr>
<tr>
<td></td>
<td>( P(\text{CCF}_4) = \frac{a_4}{1.0376} \times P_t )</td>
<td>( \alpha_3 = 0.00589 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \alpha_4 = 0.00298 )</td>
</tr>
</tbody>
</table>

For each AT component, Table 10 lists the success configuration (i.e., the minimum redundancy required) and the single-point failure rates to be used in conducting reliability analyses for both F2Z and F2FS. Success configurations are based on the assumptions stated at the end of Sec. 4.1. For components other than SDQ, single-point failure rates were obtained from the bounded F2Z and F2FS failure rates listed in Tables 5 through 8. Failure rates were not available for SDQ, so a failure rate equivalent to the FC software failure rate was assumed. Although this is believed to be a very conservative estimate, it is useful for explaining the FP and FN methodology.

### Table 10. Individual component failure rates for F2Z and F2FS

<table>
<thead>
<tr>
<th>AT Component</th>
<th>FC Hardware</th>
<th>FC Software</th>
<th>SDQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success Configuration</td>
<td>2 out 4</td>
<td>1 out 2</td>
<td>2 out 3</td>
</tr>
<tr>
<td><strong>F2Z</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure Rate (failures/hour)</td>
<td>4.2E-05</td>
<td>8.57E-07</td>
<td>7.70E-05</td>
</tr>
<tr>
<td><strong>F2FS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure Rate (failures/hour)</td>
<td>3.16E-06</td>
<td>3.57E-07</td>
<td>4.62E-05</td>
</tr>
</tbody>
</table>
that, should they occur, lead to the top event occurring. In this paper, members of this set of events are called Minimal Cut Sets (MCS) or Risk Drivers (RDs).

The FP and FN probabilities of occurrence obtained from SAPHIRE are shown in Tables 11 and 12, respectively. To consolidate the data for presentation, results for both the AT baseline and the AT+SDQ architectures are given in the same table. All of the risk drivers in these tables are CCF events associated with the AT’s redundant components. For this example, the probability of random component failure is negligible. The data, which represents the top risk drivers for each classification, will be discussed in more detail in Sec. 4.5.

4.5. Step 5: Determine the Benefit Provided by SDQ Algorithms

The overall goal of this step is to determine the benefit provided by SDQ. For this example, the SDQ algorithm would be composed of two thresholds. One threshold near zero to detect F2Z and one near full-scale to detect F2FS.

The goal of this step is accomplished by calculating probability of a FP or FN abort for the AT+SDQ architecture and comparing the results to those for the baseline AT architecture. In a process similar to that used to analyze baseline AT, calculation of the SDQ FP and FN probabilities and the SDQ benefit may be achieved as follows:

Step 5.1 Revise the baseline AT architecture to include SDQ. The revised architecture is shown in Fig. 2.

Step 5.2 Analyze the physics of failure for the AT+SDQ architecture. This can be accomplished with a cursory review of Figs. 3 and 4 and Tables 2 and 4. If SDQ successfully identifies and disqualifies the failed signal, the shift in the consolidated signal value

<table>
<thead>
<tr>
<th>Set No.</th>
<th>AT Baseline FP Probability</th>
<th>AT+SDQ FP Probability</th>
<th>Basic Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.9E-07</td>
<td>1.9E-07</td>
<td>FC1 &amp; FC2 hardware CCF</td>
</tr>
<tr>
<td>2</td>
<td>1.9E-07</td>
<td>1.9E-07</td>
<td>FC1 &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>3</td>
<td>1.9E-07</td>
<td>1.9E-07</td>
<td>FC2 &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>4</td>
<td>7.39E-08</td>
<td>7.39E-08</td>
<td>FC1, FC2, &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>5</td>
<td>2.06E-08</td>
<td>2.06E-08</td>
<td>PS1, PS2, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>6</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC1 &amp; FC2 software CCF</td>
</tr>
<tr>
<td>7</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC1 &amp; FC3 software CCF</td>
</tr>
<tr>
<td>8</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC2 &amp; FC3 software CCF</td>
</tr>
<tr>
<td>9</td>
<td>1.36E-08</td>
<td>1.36E-08</td>
<td>PS1, PS2, &amp; PS3 CCF</td>
</tr>
<tr>
<td>10</td>
<td>1.36E-08</td>
<td>1.36E-08</td>
<td>PS1, PS2, &amp; PS4 CCF</td>
</tr>
<tr>
<td>11</td>
<td>1.36E-08</td>
<td>1.36E-08</td>
<td>PS1, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>12</td>
<td>1.36E-08</td>
<td>1.36E-08</td>
<td>PS2, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>13</td>
<td>9.60E-09</td>
<td>9.60E-09</td>
<td>FC1, F2, &amp; FC3 software CCF</td>
</tr>
<tr>
<td>14</td>
<td>N/A</td>
<td>4.92E-09</td>
<td>SDQ1, SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>15</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>16</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>17</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>18</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>19</td>
<td>1.53E-09</td>
<td>1.53E-09</td>
<td>SE1 &amp; SE2 CCF</td>
</tr>
</tbody>
</table>

Table 11. Top risk drivers for an AT FP detection due to F2Z.

<table>
<thead>
<tr>
<th>Set No.</th>
<th>AT Baseline FN Probability</th>
<th>AT+SDQ FN Probability</th>
<th>Basic Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.16E-08</td>
<td>7.16E-08</td>
<td>FC1 &amp; FC2 hardware CCF</td>
</tr>
<tr>
<td>2</td>
<td>7.16E-08</td>
<td>7.16E-08</td>
<td>FC1 &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>3</td>
<td>7.16E-08</td>
<td>7.16E-08</td>
<td>FC2 &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>4</td>
<td>4.44E-08</td>
<td>4.44E-08</td>
<td>FC1, FC2, &amp; FC3 hardware CCF</td>
</tr>
<tr>
<td>5</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>PS1, PS2, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>6</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC1 &amp; FC2 software CCF</td>
</tr>
<tr>
<td>7</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC1 &amp; FC3 software CCF</td>
</tr>
<tr>
<td>8</td>
<td>1.55E-08</td>
<td>1.55E-08</td>
<td>FC2 &amp; FC3 software CCF</td>
</tr>
<tr>
<td>9</td>
<td>1.02E-08</td>
<td>1.02E-08</td>
<td>PS1, PS2, &amp; PS3 CCF</td>
</tr>
<tr>
<td>10</td>
<td>1.02E-08</td>
<td>1.02E-08</td>
<td>PS1, PS2, &amp; PS4 CCF</td>
</tr>
<tr>
<td>11</td>
<td>1.02E-08</td>
<td>1.02E-08</td>
<td>PS1, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>12</td>
<td>1.02E-08</td>
<td>1.02E-08</td>
<td>PS2, PS3, &amp; PS4 CCF</td>
</tr>
<tr>
<td>13</td>
<td>9.60E-09</td>
<td>9.60E-09</td>
<td>FC1, F2, &amp; FC3 software CCF</td>
</tr>
<tr>
<td>14</td>
<td>N/A</td>
<td>4.92E-09</td>
<td>SDQ1, SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>15</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>16</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ2, SDQ4 CCF</td>
</tr>
<tr>
<td>17</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ1, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>18</td>
<td>N/A</td>
<td>3.24E-09</td>
<td>SDQ2, SDQ3, &amp; SDQ4 CCF</td>
</tr>
<tr>
<td>19</td>
<td>1.53E-09</td>
<td>1.53E-09</td>
<td>SE1 &amp; SE2 CCF</td>
</tr>
</tbody>
</table>

Table 12. Top risk drivers for an AT FN detection by the AT due to F2FS.
required to generate a FP or FN abort detection will not exist. As a result, the probability of a FP abort detection due to a double failure or a FN abort detection due to a single failure then changes from a certainty to zero.

**Step 5.3** Create and analyze an FT for the AT+SDQ architecture. This can be accomplished by revising the baseline FT in SAPHIRE to include SDQ components and related failure data. Then perform the SAPHIRE analysis to identify cut sets that are the top risk drivers for this architecture.

As noted previously, the FP and FN probabilities of occurrence obtained from SAPHIRE are shown in Tables 11 and 12, respectively. These data represent the top risk drivers for the FP and FN classifications. In both of these tables, risk drivers are numbered as shown in column 1. For each of these cut sets, FP or FN probabilities for the AT baseline architecture is given in column 2; while probabilities for the AT+SDQ architecture are given in column 3. Column 4 lists the basic failure events that are the cause of the FP or FN abort detection.

**Step 5.4** Determine the net SDQ benefit – the reduction in FP and FN probabilities that results from including SDQ in the AT architecture.

First, calculate the FP and FN probabilities for the AT Baseline. For the example used in this paper, this is accomplished by summing the values in column 2 of Tables 11 and 12. Results of these calculations are given in row 2 of Table 13.

Second, determine the SDQ benefit by identifying risk drivers that will be mitigated by SDQ and separately summing the FP and FN probabilities associated with those risk drivers. Risk drivers mitigated by SDQ are typically associated with components downstream – in terms of information flow – of the SDQ component. For the example used in this paper, SDQ mitigated risk drivers are identified in Tables 11 and 12 by cells with a gray background. The SDQ FP benefit is obtained from Table 11 by summing the values in column 3 (or column 2 since the values are the same) for only the gray cells. A similar calculation is applied to Table 12 to obtain the SDQ FN benefit. Results of these calculations are given as SDQ benefits in row 3 of Table 13.

Third, the addition of SDQ to the AT architecture comes at the cost of increasing the FP and FN probabilities. The SDQ cost is determined by identifying the risk drivers added by SDQ and summing the probabilities of those risk drivers. SDQ risk drivers are identified by italicized text in Tables 11 and 12. The FP SDQ cost is then determined by summing the probabilities (Table 11, column 3) for the identified SDQ risk drivers. A similar calculation is applied to Table 12 to obtain the FN SDQ cost. The FP and FN SDQ costs are given in row 4 of Table 13.

Finally, metrics for the SDQ benefit can be calculated as shown in Eqs. 8 and 9.

\[
\text{Net SDQ Benefit} = \text{SDQ Benefit} - \text{SDQ Cost} \quad (8)
\]

\[
\text{Net SDQ Benefit \%} = 100 \times \frac{\text{Net SDQ Benefit}}{\text{AT Baseline}} \quad (9)
\]

Using Eq. 8, the net SDQ benefit to the FP probability may be calculated by subtracting the FP SDQ cost from the FP SDQ benefit. Similarly, the net SDQ benefit to the FN probability may be calculated by subtracting the FN SDQ cost from the FN SDQ benefit. The percent improvement in the FP and FN net SDQ benefit over the baseline AT may then be calculated using Eq. 9 in conjunction with the previously calculated values in Table 13. The FP and FN results of for Eqs. 8 and 9 are given in the next to last row and last row of Table 13, respectively.

### 5. DISCUSSION

Some observations based on data resulting from application of the FP and FN methodology to the example application are given in this Section.

First, because this methodology uses a conservative upper bound for the component failure rates, the FP and FN probabilities for the AT baseline and AT+SDQ architecture are also upper bounds. This means that the actual FP and FN rates and probabilities will likely be less than those presented in the first two rows of Table 13. The practical significance of the estimate is that if the upper bound values meet requirements for FP and FN probabilities, then more detailed FP and FN analyses are not needed.

Second, the methodology presented used single-point probability estimates for the reliability analysis. The analysis could be made more rigorous by performing the analysis with probability distributions instead of the single-point estimates.

Another observation is that, a significant amount of uncertainty in the failure rates results from the classification

| Table 13. Summary of SDQ benefit calculations for AT FP and FN detections. |
|-------------|-------|-------|
|             | FP    | FN    |
| AT Baseline | 5.66E-07 | 3.73E-07 |
| SDQ Benefit | 7.87E-08 | 5.63E-08 |
| SDQ Cost    | 1.79E-08 | 1.79E-08 |
| Probability | 6.08E-08 | 3.84E-08 |
| %           | 10.7% | 10.3% |


process that was applied. The sum of the F2IV for each component is essentially the failure rate uncertainty for that component. For example, in Table 8, all of the FC software failure modes are characterized as F2IV resulting in a failure rate uncertainty of 100% for that component. Proper classification of the failure modes is necessary to ensure that the uncertainty in the FP and FN probabilities is minimized and the accuracy maximized. Another option might be to consider a different classification approach.

The impact of the SDQ failure rate used in the example application is another important consideration. Given that the failure rate for SDQ is likely to be lower than that for the flight software, one might consider the bounding case where the SDQ failure rate and resulting cost are both zero. In that case, the SDQ Benefit given in row 4 of Table 13 becomes the upper bound for the net SDQ Benefit.

The methodology could also be expanded to examine the uncertainty in the net SDQ Benefit by considering the case where failure rates associated with F2IV are allocated to neither F2Z nor F2FS. Results for FTs associated with these cases could be compared to those already presented to arrive at an uncertainty bound for the net SDQ FP and FN benefits.

6. CONCLUDING REMARKS

This paper presented a methodology that was developed to calculate quantitative bounded estimates of the false positive (FP) and false negative (FN) detection probabilities for an abort trigger (AT) with sensor data qualification (SDQ) and a constant abort threshold during a given flight phase. To illustrate the methodology, an example application was given that included the type of redundancy typically found in human space flight hardware and software. The example starts with the definition of the AT architecture. It then analyzes the AT’s physics of failure to arrive at three failure classifications: failure to zero, failure to intermediate value, and failure to full scale. These classifications are used to bound the component failure rates. Using the Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) software, a fault tree is created that captures the component failure modes. The SAPHIRE fault tree is used in concert with the single-point estimates for the failure rates, and parametric common cause failure models to conduct a risk and reliability analysis as a means to identify the probabilities of and top risk drivers for FP and FN abort detections. Finally, reliability analysis results for a baseline AT without SDQ are compared to an AT that includes SDQ components. This provides a means of determining the net SDQ benefit in terms of reduced FP and FN probabilities of abort detection.

Observations resulting from the example application and ways to improve the methodology are also discussed. Two key means of improving the methodology are: (1) replacing single-point probability estimates with probability distributions and (2) by a more detailed investigation of the impact on the methodology of uncertainty in the component failure rates.

Current plans are to apply a version of this methodology to all SLS threshold-based ATs with the intent of refining calculations for loss of mission and loss of crew probabilities. Further, these calculations are and will be used to select the appropriate ATs for the vehicle, the SDQ algorithms for the ATs, and for verification and validation of the AT designs.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support for this effort provided by the SLS project and the Mission and Fault Management team at the NASA Marshall Space Flight Center, Huntsville, AL.

NOMENCLATURE (ACRONYMS)

AC  abort condition
AT  abort trigger
ACDL abort condition detection logic
CCF common cause failure
F2FS failure to full-scale
F2IV failure to intermediate value
F2Z failure to zero
FC flight computer
FDIR fault detection, isolation, and response
FM fault management
FMCL fault management control loops
FN false negative
FP false positive
FT fault tree
LOC loss of crew
LOM loss of mission
LV launch vehicle
NRC nuclear regulatory commission
PRA probabilistic risk assessment
PS pressure sensor
ROC receiver operator characteristic
SAPHIRE systems analysis programs for hands-on integrated reliability evaluations
SDC sensor data consolidation
SDQ sensor data qualification
SE sensor electronics
SHM systems health management
SLS Space Launch System
TN true negative
TP true positive
psi pounds per square inch

REFERENCES


**Biographies**

Kevin J. Melcher is the technical team lead for exploration systems health management activities in the Intelligent Control and Autonomy Branch at the NASA John H. Glenn Research Center (GRC), Cleveland, OH. In that role, he is responsible for coordinating GRC support of the Space Launch System (SLS) Mission and Fault Management (M&FM) project. Mr. Melcher and his team are responsible for developing algorithms for nominal and off-nominal operation of the electrical power system (EPS) and the thrust vector control (TVC) systems, for developing algorithms to qualify and consolidate sensor data, for developing fault propagation models of the EPS and TVC systems, and for developing and conducting M&FM supporting analyses. In 1983, he received a Bachelor of Science degree in Applied Mechanics from the University of Cincinnati, Cincinnati, OH. And in 1996, he received a Master of Science degree in Mechanical Engineering from Cleveland State University. He is a member of the IEEE and a senior member of the AIAA. He currently participates as co-chair of the Awards Subcommittee for the AIAA Intelligent Systems Technical Committee. He also serves on the AIAA Northern Ohio Section Council as Past Chair having served as Section Chair from June 2012 to May 2014.

Mr. José Cruz is currently a reliability engineer in the NASA Glenn Safety and Mission Assurance Division. During the past 15 years, he has provided reliability engineering and risk analysis support to various space flight projects and agency-wide efforts including the International Space Station Fluid Combustion Facility design project, the Ares 1 upper stage power system, the Ares V payload shroud, the Crew Exploration Vehicle (CEV) Smart Buyer, and the Constellation Altair lunar lander. Mr. Cruz holds a Bachelor of Science degree in Industrial Engineering from the University of Puerto Rico (1983), a Master of Science degree in Reliability Engineering from the University of Maryland (1995), and a Master of Business Administration (MBA) degree from Cleveland State University (2010). Additionally, Mr. Cruz holds a Professional Engineer (PE) license from the State of Ohio.

Dr. Stephen Johnson is the analysis lead for Mission and Fault Management on NASA’s Space Launch System program, led by NASA’s Marshall Space Flight Center. He is also an associate research professor with the Department of Mechanical and Aerospace Engineering at the University of Colorado, and the President of Dependable System Technologies, LLC. Among many publications, Dr. Johnson is the general editor for System Health Management: with Aerospace Applications (2011), the author of *The Secret of Apollo: Systems Management in American and European Space Programs* (2002), and many other articles and books.
in system health management, systems engineering, space 
history, and space economics.

Dr. Yunnhon (Yohon) Lo is a senior systems engineer with 
Ducommun Miltec supporting the Mission and Fault 
Management (M&FM) team on the NASA space launch 
system (SLS) program, led by NASA’s Marshall Space 
Flight Center where he is the M&FM Abort and Safing lead. 
During the past 15 years, he has provided critical support for 
the Space Shuttle, Ares I, and Ares I-X probabilistic risk 
assessment, SLS fault management and flight software 
development for NASA. Dr. Lo holds Bachelor of Science, 
Master of Science, and Ph.D. degrees in nuclear engineering 
from the University of Tennessee, Knoxville. He is also an 
avid space advocate who is actively involved in the National 
Space Society and is a member of the Board of Directors for 
the Tennessee Valley Interstellar Workshop.
Model-based Reasoning Approach for Automated Failure Analysis: An Industrial Gas Turbine Application

Gulnar Mehdi1, Davood Naderi2, Giuseppe Ceschini3, Mikhail Roshchin4

1,4 Siemens AG, 81739, Germany
gulnar.mehdi.ext@siemens.com
mikhail.roshchin@siemens.com

2 Siemens Industrial Turbomachinery AB, 61241, Sweden
davood.naderi@siemens.com

3 Siemens AG, 90001, Germany
giuseppe.ceschini@siemens.com

ABSTRACT
Keeping up with the technological advances, turbo-machinery industry aspires to integrate manufacturing, servicing and maintenance of their plants. Typically, these objectives may be accomplished by adoption of condition monitoring services and diagnostic solutions, resulting in improved plant operations, lower maintenance cost, and impart safety and reliability. Specifically, failure analysis, within systematic diagnostics, is a fundamental feature of design and maintenance phase, as it allows fault identification, and its causes and effects that propagate at different system levels. With the large number of subsystems and process flows, failure analysis for industrial gas turbines is non-trivial, and requires expertise of system mechanics, aerodynamics, thermodynamics, etc. Consequently, in order to realize an efficient system analysis, we device an automated model-based approach to failure analysis for industrial gas turbine applications. This paper presents context-independent qualitative models of key turbine components, which are most error-prone, together with their potential failure mode descriptions, and their impact at different system levels. Using an existing reasoning engine, we present behavior models and results for two most vulnerable turbine subsystems i.e. Lubrication Oil System and the Core Gas Turbine Engine. Finally, we evaluate the practical use-cases of this model-based solution implemented for diagnostic services at Siemens AG.

1. INTRODUCTION
Over the decades, the turbo-machinery industry has been operating complex and expensive machines, with a long history of providing quality products and services to their customers. In addition, this industry has been successful in utilization and implementation of various degrees of diagnostics, prognostics and health management capabilities, which has helped the entire turbo-machinery industry to manage and keep up with the desired efficiency of their massive systems, and most importantly, gain customer loyalties. Nevertheless, the automation curve is still pretty steep, and the plant performance is highly dependent on diverse and time-variant technical, operational, environmental and financial conditions (Siemens AG, 2014).

From the customer perspective, large process units produce daily revenue in excess of 5 million US dollars. In this context, component availability, reliability assessment and optimization are an important part of plant revenue and profits, as stated by Forsthoffer (2011). Consequently, industrial communities remain concerned to maximize reliability and product throughput, and at the same time minimize the maintenance and operating cost. This can be achieved by adopting a new business model that integrates manufacturing, service and maintenance, and furthermore, employs intelligent diagnostics and prognostics technologies. This integration would aim to boost the industrial plant operations, its longevity, impart safety, reliability, asset integrity, mitigation of risks, and help to continuously improve plant’s compatibility to variable conditions. According to the current trend, following are the key areas being explored as a part of automation and reliability improvement programs (Forsthoffer, 2011):

Gulnar Mehdi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
Site reliability audits
- Assessment methods
- Availability improvement plans
- Condition monitoring techniques
- Diagnostics Solutions
- Prognostics and maintenance plans

Focusing on the practical implementation of aforementioned solutions, numerous industry leaders emphasize on failure analysis as an input to diagnostic framework. Failure analysis serves as a baseline to identify and analyze the most error-prone units, and strengthen the tangible problem solving capabilities. This approach serves to determine system behavior and component failure, and its impact across the subsystems. Therefore, failure analysis is a key enabler and attributes to “enlighten” the diagnostic capabilities. It also improves the design, service and maintenance decisions by anticipating required actions, and provide unprecedented insight into the system’s health. Thus, it is widely adopted as a successful approach.

From stakeholder’s perspective (including senior managers, end-users, service engineers, design engineers etc.), failure analysis is a way to quantify reliability, and improve the quality of the plant. These stakeholders are a part of design and maintenance cycle and contribute in their own capacity. The motivation and role of the computer scientist is to provide next-generation technology tools, in order to match the diverse requirements set by the stakeholders (see Fig. 1).

Considering the different perspectives described above i.e.; i) the reliability management and improvement process, ii) adoption of failure analysis approach, iii) requirements from the stakeholders, and iv) opportunities within the existing infrastructure (software services), the experts themselves have to decide a methodology and tools that best fits for them for failure analysis and which also align with the standards. This is indeed a task because it requires them to understand the properties of the system failures, the standard requirements and how to achieve it.

Figure 2. Environment and Infrastructure for Turbo-industry

Another constraint to the practical implementation of failure analysis for turbines is a diverse set of configurations of every unit. Every power generating plant has different operating and process requirements and thus, often differs in its design. In addition to this, it is non-trivial to capture the behavioral properties and dependencies of critical units in the rotating equipment because it requires greater expertise of mechanics, thermodynamics and aerodynamics as discussed by Ceschin and Saccardi (2002). Currently, several off-the-shelf approaches are available that conduct failure analysis manually and/or take support from semi-automated tools. The results from these approaches are high in efforts and costs, while still compromise the quality with respect to the completeness and accuracy of results. This identifies the demand for a systematic and innovative solution as an addition to the available software services. The solution should be reliable, easy to use and cost effective.

To address the above mentioned challenges, this paper presents a model-based solution to automate the task of failure analysis for diagnostic purposes. The high demand of a sophisticated tool would justify to the design and service recommendations, especially when the changes in the system design happen on yearly basis. We adopt a software engineering approach at a very abstract level in order to make a context-free solution that is independent of a fixed structure or architecture design.
The core of our solution task is devising a software system to identify faults and their impact. Elements for this task, are:

- specifications of the faults modes of the components, turbine situations and ambient conditions;
- the failure modes, which are violations of system functions;
- and impacts such as turbine trip or low shaft speed. The impact can be monitored at different system-levels such as component-level, sub-system level and in the entire system.

Our software solution follows the knowledge-based systems and software engineering principles for problem solving and is based on so-called qualitative deviation models (Werthne, 1994) to capture the domain application. These models can capture how significant deviations from nominal behavior are generated and propagated by components models. By using an automated model-based reasoned along with an existing constraint-based predictive algorithm (Raz’r OCC’M, 2014), we provide a model-based generation of failure analysis results (which has been developed for the physical components of the system, and also extended to include electrical control units) along with its effects. Our solution has been successfully introduced at Siemens AG and this paper presents some industrial use-cases of our implementation.

The paper follows with Section 2 describing the application task at hand with an overview of industrial gas turbines. Section 3 presents the proposed model-based solution architecture and its foundations. Section 4 presents the case study and two use-case scenarios for industrial gas turbines along with the results. Finally, we conclude in Section 5 with discussion and future outlook.

2. THE APPLICATION BACKGROUND

2.1. Reliability Perspective of Industrial Gas Turbines

In the context of rotating equipment engineering, gas turbines moves product i.e. gases; either for power generation or mechanical drive applications. In general, every unit of a plant consists of a driven machine, driver, transmission device and is supported by auxiliary equipment as discussed by Forsthoffer (2011). Fig. 3 shows topology of an industrial plant.

Each of the equipment mentioned above can be classified further and have different configurations. For example: drivers can be classified as steam turbines, gas turbines, motor (Induction, synchronous or vari-speed) or engines (Internal combustion, Diesel or Gas). The key is to understand the functionality of its critical components in order to effectively monitor and maximize plant safety and reliability. Reliability is commonly defined as the amount of time equipment operates in one year. It is an ability of the equipment unit to perform its specified function without a forced (unscheduled) outage in a given period of time (Forsthoffer, 2011). In case of an outage, the loss of revenue can exceed a million U.S. Dollars a day as shown by Forsthoffer (2011). The cause of an outage is usually the shutdown of a critical component. Many leading companies including our industrial partner recognize the reliability management of the critical component and adopt following strategies (Ceschini and Saccardi, 2002):

- Involve the end-user in the specification, design and installation phase of the plant.
- Determine the life span of the plant and its component which is extremely long compared to development phases.
- Analyze the instrumentation and location of the plant that directly impact the equipment’s reliability.
- Focus the design and installation because it has a substantial influence on the maintenance requirements, its cost and availability of particular piece of machinery.

2.2. Failure Analysis

Failure analysis fulfills the reliability requirement by predicting what could go wrong in the system. It determines the severity and probability of a component’s failure mode that can occur in a given system and is considered to be a bottom-up inductive technique which starts from faults/failure modes and ends at the resultant effects (Dobi, Gleirscher, Spichkova, & Struss, 2013).

2.2.1. Requirements on fault identification and impact determination

The pre-requisite of performing model-based failure analysis is to check the system design for completeness and consistency of the models. The analysis could go wrong when the rules and component models are wrongly designed. Furthermore it can be applied to many different levels in a hierarchy of an industrial plant shown in Fig. 4. The effects are produced at the boundaries of the systems and subsystems, and for this reason it is necessary that the intermediate effects keep track of fault/failure modes at each level (Dobi et al, 2013). The analysis can be performed by using parameters such as severity, probability of occurrence,
and detect ability. In our system, we consider the effects of (single) faults on the system behavior.

![Diagram of an Industrial Plant](image)

**Figure 4. Decomposition of an Industrial Plant**

In Fig. 5 we draw an example of an axial bearing using a traditional failure analysis template as presented by Abilla (2011). The process starts by defining failure modes as a first step, the functional aspect, type, failure impact, causes and detection mode. The criteria i.e. severity, occurrence and detection levels are calculated to quantify the decisions, setup priorities and corrective measures.

![Axial Bearing Failure Mode Analysis](image)

**Figure 5. Axial Bearing Failure Mode Analysis**

The tool is acquired to address the reliability and quality aspects of the system. Though, manual adoption of failure analysis can be very expensive. But with automation, it can be cost effective in terms of design changes and can increase satisfactory level of manufacturers and customers. In our presented solution, we automate by using behavior models and check for implication or entailment to the functions.

### 3.1. High-level System Design

Fig. 6 shows the high level system design of our solution. Using remote monitoring service database (i.e. MS SQL database in our case), first we formulate interesting turbine scenarios by adopting sensor signal processing techniques. These scenarios are presented as set of complex event processing rules using physical parameters that define different states of the turbine. In the next step, we instantiate complex event processing for each unit under consideration. Few examples of these events are “modelX01 startup condition”, “modelX02 turbine operating high ambient conditions”, “modelX03 operating low ambient conditions” etc. In parallel to this, we develop a component library to model various critical components of the turbine system. The nominal (OK mode) and faulty behaviors (failure mode) of each component is captured as qualitative constraints along with its impact on the system level as presented by Struss (2004). Structure for a given physical system is defined separately as interface variables that will connect components together. Once the component library and structural description is made available, we construct a system model for analysis. System model comprises of different configurations supported by connecting the components in various styles. Once the system model is ready, we run an existing automated model-based reasoner for failure analysis task. Part of this algorithm is presented by Struss and Fraracci (2014) which solves for the finite constraint satisfaction problem. The reasoner considers the complex events as scenarios and a specific system model description for each plant to check if the propagation of a failure would entail the local or system level impact or not. Finally the model-based failure analysis results are presented as recommendation and alert messages. The solution determines the impact that may occur under a particular failure mode and predicts whether it can lead to a critical situation or violate any reliability requirement. These results serve as input to the diagnostic framework. They are useful together with other methodologies to strategize and follow for root-cause analysis and other diagnostic tasks. The qualitative model-based system development and its foundation concepts are described in the following sections.

### 3. A Systematic Model-based Approach to Failure Analysis for Industrial Gas Turbines

The proposed approach has been applied to a specific product line of Siemens industrial gas turbines. The following section describes the application details and solution implementation.
3.2. Model-based Solution Foundations

The key feature of model-based approach is the re-use of models and easy adaption to new structures/topologies and variants (physical system, software architecture). Some key features described by Struss (2008) of the models used in our work to produce successful results are:

- The models are context-free and compositional.
- The models should realize how the faults in one component of the system propagate to the rest of the system.
- The model’s qualitative deviations from their nominal behavior serve as a basis for detecting faulty components.

The failure analysis formalization as shown by Struss and Fraracci (2014) considers a set of scenarios and a set of relevant component failure and checks whether these can lead to an unwanted effect (violations of the system functionality). If we consider one component and one faulty mode \( \text{MODEL}_f \), for a given input scenario \( \text{SCEN} \), we need to check whether it entails the specified effect \( \text{EFFECT} \), or that they are all consistent with each other. The check can be performed by a constraint satisfaction algorithm (Dobi et al, 2013).

\[
\text{SCEN} \cup \text{MODEL}_f \models \text{EFFECT} \\
\text{SCEN} \cup \text{MODEL}_f \cup \text{EFFECT} \not\models
\]

E.g. consider the scenario where the gas turbine is in operating phase, compressor is on and the rotor shaft is active. If the compressor falls into the high pressure faulty mode, then it is possible that the rotor shaft speed will turn less than it should which is also the effect. Formally the inference of the system is:

\[
\text{(GasTurbine\_Demand} = \text{“Operating”) } \cup \text{ (Compressor FaultMode} = \text{“High\_Inlet\_Pressure”) } \models \text{Low\_ShaftSpeed}
\]

\[
\text{(GasTurbine\_Demand} = \text{“Operating”) } \cup \text{ (Compressor FaultMode} = \text{“High\_Inlet\_Pressure”) } \cup \text{ Low\_ShaftSpeed} \\
\not\models
\]

3.3. Tasks

The task is to make the failure analysis of the turbine system, the causal relationships between faults which occur in the system and their effects which are unintended behavior of the different components such as bleed valve stuck opened etc. These effects are part of series of situations such as turbine is in operating condition, coasting-down, and stopping, with high ambient conditions or normal ambient and so on. The overall impact is either the automatic trip or effects on exhaust pressure, temperature and mass flow. The purpose of this work is to identify possible faulty components that can lead to trips of the turbine or high exhaust conditions that can cause high CO\textsubscript{2} emission, with the objective to reduce these risks by maintenance, redesigning the existing components, or adding others in some cases e.g. more sensors. Information provided from the industrial partner includes information about the modes of operation of the turbine, system functions, list of faults, turbine situations, and impacts.

Our task is achieved by modeling two sub-systems components that is of the core engine and lube oil system, identifying the failure modes and their effects along with the overall impact on the turbine system.

4. CASE STUDY: AUTOMATED FAILURE ANALYSIS OF AN INDUSTRIAL GAS TURBINE

The case study was conducted to demonstrate the feasibility of the approach described in section 3 for one of the product-line of Siemens industrial gas turbines. The turbine system in general has a number of sub-systems that work together to perform a specific task such as power generation or mechanical drive. Fig. 7 gives an overview of turbine model at sub-system level. These subsystems are functional and can be configured differently for every model/design of the industrial plants. In this paper, we present our solution for two of the most vulnerable sub-systems i.e. lubricating oil system and the core gas turbine engine. It is important to note that the main system which is evaluated is the gas turbine at system level, which has: turbine driver, the physical subsystems containing both electrical and mechanical subsystems; the software controlling part and the electrical subsystem. The turbine driver gives commands to the turbine like start up, coast-down and stop. The physical system contains all the necessary mechanisms to allow the physical phenomena like gas flow path, combustion, etc., to occur. The electrical system offers the platform to send commands initiated from the software package of the turbine, which in the end are needed for the mechanical system and its components.

In the following sections, we present the component models and results for Siemens use-cases respectively, including fault modes, effects, the turbine conditions (e.g. turbine driving situations), and the impacts.
4.1. Use-case #1: The Core Gas Turbine Engine

The core gas turbine engine is the heart of any industrial gas turbine. Its purpose is to generate a flow of pressurized hot gas which is converted into mechanical energy, which drives the load (e.g. an electric generator) via a gear box.

The specified model under consideration operates in an open cycle with straight air and gas flow through the turbine. The core engine can be divided into three major sections: namely the compressor, the combustor and the turbine section.

4.1.1. The Physical Model

The main mechanical, thermo-dynamical, hydro-dynamical and software (control unit) components considered in the study of the core engine are presented in the Fig. 8. The ambient air is captured and either cool down or heated up by the heat exchanger component. Later the compressor draws this air and compresses it by using an adiabatic process of thermo-dynamics. The compressor is dependent on startup motor in the turbine startup phase and uses variable guided vanes and bleed valves to control the pressure ratio and prevent surge. The compressed air enters the combustor where it is heated up. The burner mixes the gas fuel coming from the fuel system with the compressed air in the combustor and maintains stability of the main and the pilot flame. Finally, the hot gas from the combustor enters the turbine section. The turbine section expands the air and drives the compressor and the generator. The gearbox transmits power from turbine to the generator. Ultimately the generator is being operated to generate electricity for the power grid and the hot gas is exhausted by the diffuser to the air exhaust system. The rotor assembly is associated with the rotor shaft speed and considers the rotor welded on the shaft. It consists of casing, blades, discs and bearings. Here we only consider the radial and thrust bearing that affects the shaft speed when faulty. The cooling system maintains the temperature of the bearings. These mechanical components are controlled from specialized Electronic Control Units (ECUs) which controls the heat exchanger, start-up motor, variable guided vanes, bleed valves, rotor assembly and the gas fuel system.

4.1.2. Component Models

In this section, we show the basic examples of component models, their physical quantities, domain types, connecting terminals and conventions that we have modeled so far for the physical system with the intention to connect them as in Fig. 8. The components exchange variables which represent physical quantities through the interfaces (terminals). The physical quantities exchanged between them are: temperature (T), pressure (P), flow rate (F), position (pos), Speed (V), Active power (A), signal/commands etc. and their deviations from nominal values expressed as ∆"Physical Quantity", e.g. for pressure it would be ∆Pressure. Most models variables and all deviations have values from the domain Sign = { -, 0, +}, whereas he commands and states have Boolean values {0, 1}.

The core purpose of the core engine model is to determine if the pressure ratio in the compressor is sufficient enough, temperature in combustor is nominal; rotor speed is up to the setting point and power output of the turbine can synchronize with the load (e.g. generator).

Example Component: Variable Guided Vanes

The purpose of variable guided vanes, VGV, is to control the compressor inlet mass flow. It is controlled by ECU, which send signals (command = {0, 1}) to position the VGV depending on the inlet temperature and rotor shaft speed. The qualitative behavior model of VGV is described as:

\[ Position = function(Command) \]

The faulty modes:

- "Stuck_at_PositiveSwirl" is when Position is greater than the function(Command)
Vibration-based Fault Diagnostics of Planetary Gearbox Using Time Synchronous Averaging with Multiple Window Functions

Jong M. Ha¹, Jungho Park², Hyunseok Oh³, and Byeng D. Youn⁴

¹,², ³, ⁴ Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, 151-919, Korea
billyhjm@snu.ac.kr
hihijung@snu.ac.kr
hsoh52@hotmail.com
bdyoun@snu.ac.kr

ABSTRACT

Recently, numerous number of studies have been made to advance TSA for vibration-based diagnostics of planet gears of a planetary gearbox. To increase signal-to-noise ratio of the vibration signals, various narrow-range window functions centered on sensor’s position have been developed to capture the instances when the planet gears are adjacent to the sensor. However, TSA with such narrow-range window functions is unable to detect the abnormal signal if it is amplified at outside of the sensor’s position due to an unexpected vibration modulation characteristics of the gearbox. This paper proposes a TSA which is robust toward the unexpected vibration modulation characteristics of the gearbox. Multiple narrow-range window functions were employed to perform multiple TSAs rather than employing the sole window function centered on the sensor’s position. Condition indicators with regard to every ring gear’s teeth were derived, and accumulated for the purpose of condition monitoring. Then, optimal position of the window function was determined to maximize capability to detect signals from the faulty gear. For demonstration of the proposed TSA, test was performed with a 2kW testbed having one-stage planetary gearbox within which a planet gear with artificial fault was assembled.

1. INTRODUCTION

Planetary gearbox is widely used for various large-scale engineering systems such as wind turbines, helicopters and mining machines. However, planetary gearbox frequently suffers from unexpected failures which causes huge amount of operation and maintenance (O&M) cost of the system. Thus, it is necessary to early detect fault of the gearbox to prevent catastrophic failure of the entire system, which also results in reducing the O&M cost. For this purpose, vibration-based diagnostics techniques have been widely used.

Time synchronous averaging (TSA) enables to minimize noise of vibration signals measured from a gearbox, thus resulting in an effective vibration-based diagnostics of the system with low computational cost (Bechhoefer et al., 2009). However, conventional TSA cannot be used for fault diagnostics of the planetary gearbox because of the following reasons; 1) vibration signal is made by multiple vibration sources including a ring gear, a sun gear, and several planet gears, 2) vibration signal is modulated by the planet gears revolving around the sun gear. For the use of TSA for the planetary gearbox, narrow-range window functions have been employed (McFadden & Howard, 1990, McFadden, 1991, Samuel, Conroy & Pines, 2004, Lewicki, Ehinger & Fetty, 2011). The narrow-range window function is located at the position of the sensor to capture the instances when the planet gears are adjacent to the sensor. This approach enables to focus on the certain meshing condition of interest and to minimize the undesirable effect of revolution of the planet gears.

However, it was reported that the loads carried by the planet gears can vary with several causes such as manufacturing tolerance and excessive radial load (Inalpolat & Kaharman, 2010). In the worst-case scenario, contact with high load between planet gears and the ring gear can be made at the opposite position of the sensor. In this case, narrow-range window function may not detect the abnormal signal that could be amplified at the opposite position of the sensor.

This paper newly proposes a TSA which is robust toward the unexpected vibration modulation characteristics of the planetary gearbox. The proposed method suggests to employ multiple TSAs with multiple window functions rather than a sole window function centered on the sensor’s position. The number of window functions was set to be same as the number of teeth of the ring gear. Afterward, condition indicators with regard to every ring gear’s teeth were derived.
and accumulated. When a typical tooth of the planet gear is failed, condition indicator at certain ring gear’s tooth will become out of range of normal condition. Then, optimal position of the window function can be determined to maximize the capability to detect faulty signals in tooth domain. For demonstration of the proposed TSA, tests were performed with a 2kW testbed having one-stage planetary gearbox in which a planet gear with an artificial fault was assembled. Although the proposed diagnostics method requires relatively long computational time compared to the conventional diagnostics method, the demonstration results showed that the proposed method performed quietly well. The remaining parts of the paper are organized as follows. In Section 2, TSA is briefly reviewed. Section 3 introduces a possible vibration modulation characteristics of the planetary gearbox when non-ideal operating condition is made. In addition, TSA with multiple window functions is proposed. And then, demonstration study is presented in Section 4.

2. Review of Time Synchronous Averaging

2.1. Conventional time synchronous averaging (TSA)
A synthesized signal (v(θ)) from a sensor attached to a gearbox can be expressed as (Hochmam & Sadok, 2004):

\[ v(t) = S(t) + N(t) + R(t) \]  

where \( S(t) \) is the synchronous coherent signal, \( N(t) \) is the non-synchronous coherent signal, and \( R(t) \) is the non-coherent random signal. In the case of the gearbox, synchronous coherent signal \( S(t) \) is produced by meshing of the gears of interest, and non-synchronous coherent signal \( N(t) \) is generated from other vibration sources which are out of interest (e.g., bearings). Non-coherent random signal is Gaussian noise in general. For condition monitoring of a particular gear of interest, it is essential to isolate the synchronous coherent signal from the measured synthesized signal. TSA enables to isolate the synchronous coherent signal by suppressing the non-synchronous coherent signal, and de-noising the non-coherent random signal. Figure 1 illustrates the general procedures of TSA which are made up of two steps. First, the synthesized signal is divided into \( N \) segments (e.g., \( v_i \) in Figure 1) based on the rotational frequency of the gear of interest. For this purpose, resampling of vibration signals should be performed with the help of encoder system. Second, the divided segments are ensemble averaged to make a TSA signal.

Because the non-synchronous coherent signal and the non-coherent random signal follow a Gaussian distribution with a mean of zero, they theoretically converge to zero as the number of segments accumulate. On the other hand, every divided segments have almost identical deterministic vibration signals with almost same phase and magnitude. Thus, a synchronous coherent signal aren’t significantly affected by the ensemble average process. The TSA for isolation of the synchronous coherent signal is defined as:

\[ S(t) = \frac{1}{N} \sum_{i=1}^{N} v_i(t) \]  

where \( v_i(\theta) \) is the \( i^{th} \) segment corresponding to \( i^{th} \) rotation of the gear of interest, and \( N \) is the number of segments averaged for TSA.

2.2. Time Synchronous Averaging for Planet Gears
Planetary gearboxes have four main components including a carrier, a ring gear, a sun gear, and multiple planet gears. This paper employed a 2kW testbed with a planetary gearbox which has a sun gear (31 teeth), a ring gear (95 teeth), and three planet gears (31 teeth) as shown in Figure 2. The ring gear is fixed on the gearbox housing and every planet gear revolves around the sun gear. Because vibration sensors are fixed on the gearbox housing, relative distance of the planet gears to the sensors varies. Sensor was attached to the top of the gearbox housing under which tooth number one was assigned to the tooth of the ring gear. At the initial condition of test, tooth number one of the planet gear of interest was set to contact with tooth number one of the ring gear. In addition, position of the inner gears are tracked in real time with the help of a high-resolution encoder system.

Narrow-range window functions have been employed to extract the vibration signals when the planet gear of interest

![Figure 1. Procedures of time synchronous averaging](image1.png)

![Figure 2. Planetary gearbox; (a): outside of the gearbox, (b): Inner side of the gearbox](image2.png)
Figure 3. Window function for extraction of vibration signals is adjacent to the sensor (McFadden et al., 1990). Figure 3 illustrates the procedure of TSA with the narrow-range window function. The narrow-range window function helps remove the noisy vibration signals generated from the other gears which are out of interest. Consequently, the extracted signals with reduced noise can effectively serve as a qualified source for TSA.

For TSA with the narrow range window function, tooth sequence of the planetary gearbox should be considered. Tooth sequence for a planet gear is defined as tooth number of the planet gear that mesh with the ring gear under the sensor. Tooth sequence arises from a feature that \( N_p \) (the number of teeth of the ring gear) is non-integer multiple of \( N_p \) (the number of teeth of the planet gear). Tooth sequence of the planet gear \( (T_p) \) which meshes with the ring gear under the sensor at \( n^{th} \) rotation of the carrier can be identified as:

\[
T_p(n) = \text{mod}(n, N_c, N_p) + 1
\]  

where \( n \) is the number of carrier rotation, \( N_c \) is the number of ring gear’s teeth, \( N_p \) is the number of planet gear’s teeth.

Figure 3 illustrates an example where it is assumed that \( v_{p1} \) was generated by tooth number one of the planet gear whereas \( v_{p3} \) was generated by tooth number three of the planet gear. In this case, every extracted signal should be transported to the appropriate position of the planet gear’s tooth domain as shown in upper side of Figure 3. After transforming all of the extracted vibration signals, TSA can be performed to isolate the synchronous coherent signal which is generated by the planet gear of interest.

2.3. Extraction of Condition Indicator

For quantitative fault diagnostics of the gearbox, various condition indicators (CIs) can be defined using the signals processed with the TSA. Although several attempts were made to find a best CI for condition monitoring of a specific engineering system, consideration of various CIs is valuable if best CI is not identified for the given system. This paper employed three kinds of CIs (i.e., \( SER \) (Sheng, 2012), \( FM4 \) (Stewart, 1977) and \( M6A \) (Martin, 1989)) which can be defined as:

\[
SER = \frac{\sum_{i=1}^{N} (A(f_{1i})/A(f_i))}{N}
\]  

\[
FM4 = \left[ \frac{1}{N} \sum_{i=1}^{N} (DIF(i) - \mu_{DIF})^4 \right]^{1/4}
\]  

\[
M6A = \left[ \frac{1}{N} \sum_{i=1}^{N} (DIF(i) - \mu_{DIF})^6 \right]^{1/6}
\]

where \( A(f_i) \) and \( A(f_{10}) \) denote amplitude of vibration signal at fundamental frequency and \( i^{th} \) sideband of the fundamental frequency respectively, and \( \mu_{DIF} \) stands for mean of \( DIF \). \( DIF \) denotes a difference signal which can be defined by excluding fundamental gear mesh frequency and their harmonics along with their sidebands (Samuel & Pines, 2005). Because resampling was performed before TSA processing, fundamental frequency and their harmonics are integer numbers represented in order domain.

3. TSA WITH MULTIPLE WINDOW FUNCTIONS

Section 2 described TSA which was especially developed for the diagnostics of the planet gears. Overall procedures for the advanced TSA was based on an assumption that the vibration signal is modulated to have an enlarged magnitude when the planet gears are passing the vibration sensor. Consequently, it was believed the faulty signal can be effectively captured when the faulty tooth of the planet gear meshes with the ring gear around the sensor. This section investigates a planetary gearbox having an unexpected vibration modulation characteristics. And then, TSA with multiple window functions is proposed to use TSA effectively even when the unexpected vibration modulation occurs.

3.1. Vibration Modulation Characteristics

For identification of the vibration modulation characteristics of the given planetary gearbox, this paper performed envelop analysis for the band-pass filtered signal around gear mesh frequency (GMF). Figure 4 represent identified vibration modulation characteristics of the planetary gearbox (normal condition) in 2kW testbed by investigating vibration level with regard to the ring gear’s teeth. Vibration level was obtained by calculating RMS (Root Mean Square) value of the vibration signal for each gear tooth. Band-pass filter was designed to include only fundamental GMF as shown in Figure 4 (a). In addition, the range of the band-pass filter were...
Figure 4. Vibration level of the planetary gearbox with regard to ring gear’s tooth; (a) 1st gear mesh frequency is included for band-pass filter, and (b) 1st, 2nd, and 3rd gear mesh frequency is included for band-pass filter.

(Sensor Position)

Figure 5. TSA with narrow-range window function; (a) Sole window function is employed (b) Multiple window functions are employed.

Figure 6. Condition indicators in the domain of ring gear’s tooth.

3.2. Multiple Window Functions

From Section 3.1, it was found that vibration modulation characteristics is not solely dependent to the revolution of the planet gears. In some cases, vibration level can be enlarged when the planet gears are located at particular tooth of the ring gear by several reasons. In these cases, the narrow-range window function centered on the sensor position may lose the opportunity to capture the abnormal signals generated by faulty tooth of the planet gear far from the sensor. Therefore, window function should be centered on the particular tooth of the ring gear where the vibration signal is amplified. Best solution to solve this challenge is to identify vibration modulation characteristics to define optimal position of the narrow-range window function. However, it is difficult to define deterministic solution of vibration modulation characteristics because results of the envelop analysis are sensitive to detailed design of the band-pass filtering.

As an alternative way, this paper employed multiple number of narrow-range window functions located to every teeth of the ring gear. The comparison of the proposed approach with TSA employing a sole window function is illustrated in Figure 5. As shown in Figure 5 (a), the conventional TSA employs a sole window function centered on the sensor’s position (TR=1). Corresponding condition indicator (CI) is represented as a square marker in Figure 6 where TR=1. However, this CI possibly fail to represent current health state of the gearbox where abnormal signal is generated far from the sensor due to the unexpected vibration modulation characteristics. Proposed method performs additional TSAs at each ring gear’s tooth with the multiple window functions as shown in Figure 5 (b). Corresponding CIs are independently accumulated and monitored in ring gear’s tooth domain as represented by circle markers in Figure 6. As a fault occurs in a planet gear, CI corresponding to one or more teeth of the ring gear will vary.

4. Demonstration

For the demonstration study, two kinds of artificial faults were manufactured to tooth number one of planet gears (TR=1) as shown in Figure 7. First one is a tooth wear which was machined by peeling the surface of a gear tooth. Second one is a spall which was simulated by machining two thin lines.
on the surface of a gear tooth. For comparison study, two gears were used so that a gear has only one type of fault. Two independent tests were ran for 60 seconds with the faulty gears (i.e. a gear with wear and a gear with spall). Rotational speed and torque were set to be 1500 rpm and 4Nm at the high-speed shaft. For TSA processing at each ring gear’s tooth, this paper employed five-teeth Tukey window as a narrow-range window function, which was proposed by Samuel et al. (2004). From the test with the gear having a tooth wear, condition indicators (CIs) for every ring gear’s tooth were obtained using the TSA with multiple window functions as presented in Figure 8. Figure 8 (a) compares three CIs (i.e. SER, FM4 and M6A), and Figure 8 (b) shows sum of the three CIs in the ring gear’s tooth domain to present general behavior of the CIs. In this study, trend of SER differed from the trend of FM4 and M6A. SER had its maximum value near the sensor’s position ($T_R=3$), whereas FM4 and M6A had their maximum values around tooth number 42 of the ring gear ($T_R=42$).

Although faulty signature was visually found in the ring gear’s tooth domain as shown in Figure 8, identification of the faulty signature in planet gear’s tooth domain is valuable to understand physical meaning of increasing value of CIs. For example, Figure 9 (a) and (b) represent difference signals (DIF) in the planet gear’s tooth domain which were obtained by locating Tukey window on tooth number 3 ($T_R=3$) and 42 ($T_R=42$) of the ring gear respectively. With the window function centered on near the sensor’s position ($T_R=3$), TSA couldn’t detect the abnormal signals generated from the faulty tooth of the gear (Figure 9 (a)). Whereas, TSA with the window function centered on the tooth number 42 of the ring gear worked well in finding abnormal signals in planet gear’s tooth domain. Interestingly, abnormal signal was found when tooth number 18 of the planet gear mesh with the tooth number 42 of the ring gear. In this moment, faulty tooth of the planet gear ($T_p=1$) simultaneously contacts with the sun gear. This is because of a fact that the abnormal planet gear was assembled to the gearbox so that the faults of the gear (left side of the gear surface in Figure 7) meshes with the sun gear. When the abnormal planet gear meshes with the ring gear, contact plane of the faulty tooth is positioned to opposite side of the faults of the gear (right side of the gear surface in Figure 7).

Figure 10 and Figure 11 represent condition indicators and difference signals which were obtained from the test using the gear with a spall. Processing procedures were set to be same as those employed for analyzing the previous case (a gear with wear). FM4 and M6A in Figure 10 (a) represented similar trend with those in Figure 8 (a). In this case, they had their maximum values around tooth number 44 of the ring gear. On the other hand, SER had two local maximum values. First maximum value was found near the sensor’s position which is same phenomenon as wear case in Figure 8 (a). Second maximum value was found around tooth number 50 of the ring gear. From this result, it can be said that SER partially represent similar trend to the FM4 and M6A. Even
if the magnitude of abnormal signal was reduced, the results of Figure 10 and Figure 11 reached the same conclusions.

5. Conclusion

TSA with a narrow-range window functions have been effectively used for diagnostics of the planetary gearbox. However, it was found in this paper that such narrow-range window function is possibly unable to detect the abnormal signal when the window function is fixed on the sensor’s position despite the unexpected vibration modulation characteristics of the gearbox. To solve this challenge, this paper proposed to employ multiple number of narrow-range window functions which are located to every teeth of the ring gear. The proposed method was demonstrated by the tests using a 2 kW testbed with two kinds of artificial fault of the planet gears. By the proposed approach, the abnormal signal generated far from the sensor was effectively detected.

Among three CIs employed in this study, FM4 and M6A performed well compared to SER. Although FM4 and M6A successfully detect the fault of the gearbox in this case, they could fail to capture abnormal signal in different operating condition (e.g. varying rotational speed). For more general validation of the proposed fault diagnostics method, different kinds of condition indicators should be investigated to find best candidate in the future works. For even further study, causes of the unexpected vibration modulation characteristics should be identified. Simulation models of the planetary gearbox such as a finite element model or a lumped parameter model can be used for this purpose. In addition, a testbed that facilitates control of various operating conditions such as alignment condition and radial load condition will also be needed for the further demonstration of the proposed method.

Although the proposed method can advance the conventional TSA in the aspect of diagnostics accuracy, computational cost should be additionally considered. The proposed method performs multiple TSA processing as much as the number of ring gear’s teeth, which means that computational cost significantly increases. In this case, conventional TSA required 1 second to process 60 seconds of vibration signal. On the other hand, TSA with 95 window functions necessitated 95 seconds of the processing time for the identical vibration signal. Because the processing time is longer than the running time of the gearbox, multiple TSA cannot be used as an on-line condition monitoring tool of the system. Computationally efficient signal processing technique, thus, is needed to be developed for the use of the multiple window functions as a real-time condition monitoring tool.

Acknowledgement

This work was partially supported by the two research projects of the International Collaborative Energy Technology R&D Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20118520020010) and Mid-career Researcher Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science ICT and Future Planning (MSIP) (2013R1A2A01068627).

References


A New Generic Approach to Convert FMEA in Causal Trees for the Purpose of Hydro-Generator Rotor Failure Mechanisms Identification

Olivier Blancke¹, Normand Amyot², Claude Hudon², Mélanie Lévesque², Antoine Tahan¹

¹École de technologie supérieure, Montréal, Québec, QC H3C 1K3, Canada
olivier.blancke.1@ens.etsmtl.ca
antoine.tahan@etsmtl.ca

²Institut de recherche d’Hydro-Québec, Varennes, Québec, J3X 1S1, Canada
hudon.claude@ireq.ca
levesque.melanie@ireq.ca
amyot.normand@ireq.ca

ABSTRACT

At Hydro-Québec (HQ), an integrated diagnostic system (MIDA) is currently used to assess hydro-generators health index. This system gives the global health index but does not propose any understanding of active failure mechanisms. At this point, this work needs to be done by experts after analysis of the diagnostic data in MIDA.

To relieve the expert from part of this work, a prognostic tool, that uses a Failure Mechanisms and Symptoms Analysis (FMSA), is under development. The approach is based on the understanding of the evolution of degradation processes for each failure mechanism. Failure mechanisms are structured as causal trees and defined as a sequence of physical states starting from a root cause and ending with a failure mode. A physical state corresponds to characteristic degradation condition of a component of the generator. Each physical state being defined by a unique combination of symptoms as measured with diagnostic tools. After consigning all possible mechanisms occurring in both the rotor and the stator, the symptoms logged into a database can be read to automatically identify all active physical state and active failure mechanisms. This approach has been under development in HQ for the stator for a number of years and is now extended to the rotors of hydro-generators.

The purpose of this paper is to present the structured method used to build the failure mechanisms from bits and pieces of information (sub-mechanisms) found in the literature and from discussions with experts. This new methodology is based on a two steps process. First, sub-mechanisms were extracted from FMEA in the literature. Then, an algorithm was used to generate a set of causal trees from these sub-mechanisms. The generated results then had to be validated by experts to make sure that automatically generated mechanisms were logical and plausible. The resulting extended failure mechanisms trees built can then be used for the purpose of Root Cause Analysis (RCA), model-based diagnostics and prognosis. This method was developed to be as generic as possible so it could be applied to any complex system.

1. INTRODUCTION

Significant improvement has been done on fault detection and diagnosis during the last decade (Schwabacher and Goebel 2007). Many tools such as condition monitoring systems have been developed for this purpose in various industries to improve health management of complex and critical systems. However, the large amounts of data generated by those systems are rarely used for prognosis.

One of the key issues of condition-based prognostic approaches is to develop a model providing an identification of degradation processes and their future evolution based on historical data such as symptoms obtained from measurements and observations. Several examples of prognostic approaches based on historical data and symptoms resulting from diagnostic tools can be found in the literature; namely an approach based on Bayesian networks (Medina-Olivier et al. 2012), approaches based on Fault Trees (Junjie et al. 2011; Sun et al. 2012) and an approach based on Decision Trees structure (Lee et al. 2005). In this paper, the proposed approach uses another model to analyze symptoms and historical data.
At Hydro-Québec, different diagnostics tools have been developed and implemented to evaluate the condition of hydro-generators using a web-based application called MIDA (Vouligny et al. 2009), which gives the global health index of generators. In order to structure and take advantage of diagnostic data, a damage propagation model has been proposed. It is the backbone of a prognostic model based on a Failure Mechanism and Symptoms Analysis (FMSA). This approach has been detailed elsewhere (Amyot et al. 2013).

The FMSA identifies and structures all possible failure mechanisms occurring in a system of causal trees using successions of physical state starting from root causes and ending with failure modes, as shown in Figure 1. A physical state corresponds to a characteristic degradation condition of a component of the generator. By defining such physical states it becomes possible to discretize each failure mechanism in order to track their progression. In the example in Figure 1, four failure mechanisms are represented and discretized using physical states (e_i). For example, the failure mechanism starting from the Root cause C_1 and ending on the failure mode F_2 is discretized by 4 successive physical states (e_1, e_2, e_3, and e_4).

Each physical state corresponds to a unique combination of symptoms with threshold values associated to the recorded diagnostic data and visual inspection observations. An example of threshold definition is given in Figure 2. The degradation symptoms of a generator are automatically retrieved from the diagnostic database, and compared with the individual thresholds defined by the experts. This makes it possible to identify active physical states and consequently all active mechanisms in the system.

It is then possible to monitor failure mechanisms progression using physical states activation time as shown in the Figure 3. Based on historical data and probabilistic approach a prognosis can be performed. Moreover, some targeted maintenance tasks will eventually be proposed by the system.
In complex systems such as hydro-generators, failure mechanisms evolve in all system components. To build a list of all possible failure mechanisms is a complex task; it requires many areas of expertise. After several experts’ interviews, it was noticed that they usually have a clear view of failure sub-mechanism but have more difficulty in building complete complex failure mechanisms longer than 5 successive physical states. This is especially true when multiple side branches split from a main common branch. Even in the literature, failure mechanisms are indirectly described as degradation progress and symptoms but report or papers seldom present them in a logical succession of physical state. A methodological approach is thus needed to build failure mechanisms in a causal tree structure.

In the literature, some methods and models are already used for Root Cause Analysis (RCA) which shows similarities with Failure Mechanism Analysis (FMA). A paper written by (Medina-Olivier et al. 2012) proposed a survey on reliability and risk analysis main approach for RCA. Five methods such as Causes and Effect analysis, Hazard and operability studies (HAZOP), Bayesian Networks, FMEA and Fault tree are compared according to different criteria (experience dependence, time and resource consumption, providing a path to root causes…). The author concludes that a single method does not allow characterizing all causal relationship of the degradation process of a system but they all give parts of information about it. For complex systems, those methods give essential information on degradation mechanism allowing fault detection and probable root causes categories. However, they do not allow defining the failure mechanism from a root cause to a failure mode. The authors suggest that the best way to perform a Failure Analysis is to combine information from different approaches such as FMEA and Fault Tree.

The approach we propose is doing exactly this by using information captured from literature such as FMEA and expert knowledge to build structured causal trees step by step. As information available only contains parts of failure mechanisms, we have built an algorithm to assemble those sub-mechanisms into complete failure mechanisms. Once a complete set of detailed causal trees have been built, it can be used to perform Condition Based Maintenance (CBM) and improve asset management. This is true for any industry. In our case, it serves as the premise for the implementation of a FMSA-based prognostic approach.

A statistical analysis of hydro-generator failures performed by CIGRE in 2003 (CIGRE 2003) revealed that stator failures represent 70% of generator failures and the rotor, bearings and the excitation system represent all together approximately 25% of all failures. As the FMSA of the stator has already been performed, this paper will focus on hydroelectric rotor failure mechanisms.

2. METHODOLOGY

The following section will describe step by step the methodology developed to generate a structure causal tree from FMEA and from sub-mechanisms found in the literature.

2.1. System definition & delimitation

The first step is to define and delimitate the scope of the work. Each component of the system has to be defined precisely by a unique appellation in order to avoid misunderstanding between experts. A universal terminology has to be defined for physical state naming such as System/Components/Sub-component/Part/ physical state. The system has to be delimitated and external components which will interact with the system as input or output on failure mechanism should be identified.

2.2. Sub-failure mechanisms analysis based on FMEA interpretation

Once the system is defined, a survey on its failure mechanisms has to be performed by gathering FMEA, technical papers, accident reports, RCA… The methodology proposed can be defined as a “Bottom Up” strategy. At first, failures modes and sub-mechanisms leading to failure have to be identified by interpreting the FMEA results. It creates a work base. The next step is to identify and build all possible logical successions of physical states called sub-mechanisms (i.e. sets of a few successive physical states) of the system by interpreting the information from scientific literature and expert knowledge. The analysis can be carried out on individual components or on the entire system.

All sub-mechanisms interpreted have to be listed in a database, like the one shown in Figure 4. For example, it was found that the rotor degradation state e_4 (ex: Rotor Interpol connection cracking) is caused by the degradation e_3 (ex: Rotor Interpol connection mechanical fatigue stress) and can induce e_2 (ex: Rotor Interpol connection failure). In this first rough analysis, all lines are independent and yet no links between sub-mechanisms exist.
2.3. Causal Trees structure assumptions

A physical state can be represented as a node which has inputs and outputs as shown in the Figure 5.

\[ e_n \in C \text{ if } \forall i E_{i,n} = \emptyset \]  \hspace{1cm} (1)

\[ e_n \in F \text{ if } \forall j E_{n,j} = \emptyset \]  \hspace{1cm} (2)

All outputs of a physical state are independent of its inputs:

\[ \forall i,j E_{i,n} \not\Rightarrow E_{n,j} \]  \hspace{1cm} (3)

All inputs of a physical state are independent of its outputs.

\[ \forall i,j E_{n,j} \not\Rightarrow E_{i,n} \]  \hspace{1cm} (4)

2.4. Causal tree generation algorithm

Based on those assumptions, an algorithm has been developed in order to assemble sub-mechanisms retrieved from the literature into a structured causal tree. The causal tree generation methodology is presented in the Figure 6.

Once the Sub-failure Mechanisms Analysis database has been finalized, the algorithm scans all sub-mechanisms and identifies inputs and outputs of each physical state in a matrix. For example, using the fictive sub-mechanisms in Figure 4, the resulting input/output matrix is presented in the Table 2.

<table>
<thead>
<tr>
<th>Input (i)</th>
<th>( E_{i,j} )</th>
<th>( e_1 )</th>
<th>( e_2 )</th>
<th>( e_3 )</th>
<th>( e_4 )</th>
<th>( e_5 )</th>
<th>( e_6 )</th>
<th>( e_7 )</th>
<th>( e_8 )</th>
<th>( e_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_2 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_4 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_5 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_6 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_7 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( e_8 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( e_9 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The physical state \( e_2 \) have one input \( e_1 \) and two outputs \( e_3 \) and \( e_7 \) as shown in Eqs. (1, 2):
\[ E_{i,2} = (e_1) \]  
\[ E_{2,j} = (e_3, e_7) \] 
(5)  
(6)

The algorithm will assemble physical states together independently of previous sub-mechanisms. Thus, generating a large amount of new combinations.

Based on the proposed assumptions, the algorithm is able to identify the root causes and failure modes of the system. In our example there is only one root cause \( e_1 \), and two failure modes \( e_3 \) and \( e_7 \). Once the root causes are all identified, they need to be classified into categories in order to structure the causal trees and to allow advanced root cause analysis. The categorization chosen has been taken from a report of the US. Nuclear Regulatory Commission (Nuclear Regulatory Commission 2003). This report proposed more than 20 sub-categories originating from 7 main categories such as Design/Construction/Manufacture or Operation/Human Error. In order to categorize identified root causes, each corresponding category should be added as input. The algorithm will take the categories into account during the tree generation. Our root cause \( e_1 \) has been affected to the fictive root cause category \( \text{Cat}_3 \) as shown in Figure 7.

To generate a structured causal tree, the algorithm calls each root cause and scans the input/output database. Then a “While” loop is initiated generating failure mechanisms by adding each physical state output by output from a root cause up to failure modes. Still based on the same fictive example shown in Figure 4 and Table 2, a structured causal tree has been generated in the Figure 7. An important remark is that this logic can be easily reversed for root cause analysis.

The main goal of the algorithm is to allow the transition from a simple structure composed of sub-mechanisms to a complex structure of all mechanisms.

2.5. Expert validation process
A validation process has to be done on each failure mechanism generated. This is one of the most important and complex part of the process. For this, two methods were used: a validation process based on sub-mechanisms and one on the complete causal tree generated as shown in the Figure 6.

3. HYDRO-GENERATOR ROTOR APPLICATION:

3.1. Hydro-generator Rotor definition & delimitation:

Hydro-generators are large machines which can measure over 12 meters in diameter. They are composed of three main components, the stator, the rotor and the excitation system. In the current case, the system studied has been defined as the rotor. As shown in the Figure 8, the rotor is composed of three main internal components which are themselves composed of sub-components and parts. Six external components related to our main component have been identified, such as the stator and the excitation system. They may interact with the rotor as input or output in some failure mechanisms.

![Hydro-generator Rotor definition & delimitation](image)

3.2. Hydro-generator Rotor Failure mechanisms analysis

A literature review has been done on hydro-generator rotor failure mechanisms. Using FMEA such as those from the Electric Power Research Institute (EPRI 1999) and other sources (Calleecharan and Aidanpa 2011; EEA 2013; EPRI 2009; Hydro-Québec 2007; Walker 1981), failure modes and sub-failures mechanisms leading to them have been identified. Then an interpretation of the literature review and expert knowledge lead to the creation of 108 rotor sub-mechanisms based on the combination of 110 physical
states. Those sub-mechanisms have an average of 3 successive physical states. They are listed in a structure similar to the one presented in the Figure 4.

3.3. Hydro-generator Rotor Causal Tree generation

Based on the 108 sub-mechanisms, an input/output analysis has been done and 24 root causes have been identified. These causes have been associated with 5 main categories. Then, by assembling the sub-mechanisms, a structured causal tree has been generated comprising a total of 294 rotor failure mechanisms with an average of 10 successive physical states for each failure mechanisms. An example is illustrated in the diagram in Figure 9. In this example, three complete rotor failure mechanisms all originating from the same root cause (“Over speed excursion”) lead to two different failure modes (“Half phase current unbalance” & “Rotor guide bearing excessive vibration”). The root cause has been categorized in a sub-category (“Incorrect procedure”) belonging to the “Operation/Human error” category. Since each failure mechanism is defined by a unique sequence from a root cause to a series of physical state then to the final failure; as soon as a mechanism differs from another by one single state, it is considered as a separate mechanism. This is the case in figure 9 where two of the mechanisms only differ by the failure mode. This will become important later when the severity (impact) of failures will be considered.

The algorithm developed has allowed us to generate systematically and reliably all possible failure mechanisms in a detailed causal tree structure based on sub-mechanisms interpreted from the literature. This automatic generation makes sure that no possible mechanisms are omitted.

3.4. Validation process

After the tree generation process a validation must be done by experts. All the mechanisms automatically generated by aggregation of sub-mechanisms do not necessarily bare any physical meaning. In some case the algorithm may have assembled sets of physical states into mechanism according to our rules and it is thus mandatory that experts validate all proposed mechanisms. Another aspect of the validation process is related to the scope of our work, aimed at improving preventive maintenance and mainly looks at slow degradation process. Thus any instantaneous failure mechanisms not giving any warning sign, should be discarded by the expert from our analysis. Although these modes will be extracted from the FMEA and are part of the original 294 rotor mechanisms, they will be discarded by the experts, not because they do not exist, but because they are not relevant for the sake of prognosis.

![Figure 9. Illustration of the Rotor Causal Tree generated](image)

4. DISCUSSION

Based on assumptions made on sub-mechanisms and physical states input/output analysis, an algorithm has allowed to generate all possible failure mechanisms in a coherent causal tree structure. We believe that it is much easier for expert to discard from all the mechanisms proposed by an algorithm, the ones that are not plausible, than to generate all possible mechanisms one by one without any omission. This constitutes the first step in the rotor FMSA process. The next step will be to combine the final failure tree with the results of diagnostic measurement from rotor using a symptom and threshold analysis. Once this is done, automatic identification of active failure mechanism will be possible. This will eventually lead to identification of personalized maintenance actions suited for each generators’ condition and probable timeframes to carry out the maintenance while reducing failure risk. This can be called a prognosis.

5. CONCLUSION

A methodological approach has been developed in order to identify and structure all possible failure mechanisms...
potentially occurring in a complex system as causal trees. Based on FMEA reports and expert knowledge, a Failure Mechanisms Analysis was built from identification of sub-mechanisms from short sequences of physical states. Then an algorithm has been used to assemble those sub-mechanisms and propose structured causal trees representing all possible failure mechanisms of the system. The proposed causal trees have to be validated to ensure that all failure mechanisms are coherent and real. This approach was used to build a Failure Mechanism Analysis of hydro-generator rotors and causal trees have been generated. The algorithm generated 108 sub-failure mechanisms with an average of 3 successive physical states and 294 complete failure mechanisms with an average of 10 successive physical states. Results have shown that in general generated failures mechanisms are coherent and well structured.

The main advantages of this approach are to help experts to move from a simple system such as sub-failure mechanisms to a complex system (causal trees) using an algorithm to give guidelines to experts by presenting them well-structured causal trees for further validation.

**NOMENCLATURE**

| $e_n$ | Physical state n |
| $E_{i,n}$ | Physical state input i of the physical state n |
| $E_{n,j}$ | Physical state output j of the physical state n |
| $C$ | Root Cause Domain |
| $Cat_i$ | Root Cause Category i |
| $F$ | Failure Mode Domain |

**REFERENCES**


Calleecharan, Y and Aidanpa, J-O (2011), 'Dynamics of a hydropower generator subjected to unbalanced magnetic pull', *Journal of Mechanical Engineering Science*, Part C.


EEA (2013), 'Asset Health Indicator Guideline for Generators', (Wellington, New Zealand: Electricity Engineers’ Association of New Zealand (Inc.)).

EPRI (1999), 'Advanced Condition Monitoring of Hydrogenerators', (Electric Power Research Institute).

--- (2009), 'Handbook to assess the insulation condition of large rotating machines', (Toronto, Canada: Electric Power Research Institute).

Hydro-Québec (2007), 'Guide d’analyse de la maintenance - Système groupe turbine alternateur'.
Normand Amyot was born in Sorengo, Switzerland in 1964. He received the M.Sc.A. degree in physics engineering from École Polytechnique de Montréal in 1990. He has worked one year at the Université des Sciences et Techniques (USTM) de Masuku in Gabon, after which he joined the Research Institute of Hydro-Québec (IREQ), where he is now employed as a senior research scientist. His field of expertise includes electrical insulation aging and characterization, diagnostic techniques, condition based maintenance and more recently prognostics. He is an active member of CIGRÉ working groups and member of IEEE DEIS. He is the author or co-author of more than 40 scientific papers.

Claude Hudon was born in Montreal, Québec, Canada on 21 December 1963. He received the Ph.D. degree in engineering physics from École Polytechnique de Montréal in 1993. He worked for two years at the Corporate Research and Development center of General Electric, after which he joined Hydro-Québec’s Research Institute, IREQ, where he is now employed as a senior researcher. His fields of interest are generator and motor diagnostics, the development of diagnostic tools and the measurement and analysis of partial discharges. He is the author of more than 45 scientific papers.

Mélanie Lévesque was born in Chicoutimi, Québec, Canada on 24 November 1980. She received the Ph.D degree in electrical engineering from the École de Technologie Supérieure (ÉTS) of Montréal in 2012. Her Ph.D’s degree was done in collaboration with the Research Institute of Hydro-Québec (IREQ). She is currently working as a researcher at IREQ on generator diagnostics.

Antoine Tahan was born in Aleppo, Syria in 1966. He received his Ph.D. in Electrical Engineering from Laval University (Québec), and his Master’s Degree in Mechanical Engineering from the same university. He worked for ten years as consulting engineer for the aerospace industry, after which he joined in 1994 the École de technologie supérieure (ETS) in Montreal (Québec, Canada) as professor in the Department of Mechanical Engineering. His research interests are in the area of industrial statistics, and include productivity improvement, quality control, metrology and probabilistic approach. He is a registered professional engineer in Quebec, Canada, and member of the ASME (American Society of Mechanical Engineers), the CMVA (Canadian Machinery Vibration Association), and the ASQ (American Society for Quality). He is the author or co-author of more than 40 scientific papers.
ABSTRACT

In recent years, the service business of the global turbo-machinery industry has undergone important changes. Many of these changes have been motivated by an increased demand for dedicated and systematic approaches to process safety, reliability, asset integrity and the overall health of the system. This has strengthened the role of key performance indicators (KPIs) as a means of providing guidance for the system’s health state and improve risk management. In order to provide trustable and accurate calculations of these performance indicators in an automated fashion, we argue for a model-based solution that deals with the complexity of diverse configurations and interdependences between system components. This paper presents a solution for calculating KPIs by a semi-automated process based on post-data processing from the site and specific system models. The models consist of a combination of system descriptions in terms of ontologies and complex event processing models. By virtue of our models, state indicator rules for KPI calculations can be formulated at different levels, identifying performance gaps and indicating precisely where action should be taken by the service engineers. With the adopted solution, we discuss the practical implementation and present results of our success story at Siemens AG for the Industrial Gas Turbines.

Finally, we provide an evaluation and future developments.

1. INTRODUCTION

In recent years, the turbo-machinery industry has provided a wide range of products and comprehensive services to their customers. The industry has evolved in terms of increasing product standardization and continues to adopted strategies to enhance their value-added services. As part of that industry, Siemens AG aims to expand their service business to mobilize the additional potential of sustainable growth. Keeping up with the technological advances, Siemens Corporate Technology (CT) and the turbo-machinery portfolio is laying the foundations for next-generation smart and efficient solutions in the energy sector. Their focus is to enable improved plant operations, lower maintenance costs, increased plant lifetime, safety, reliability, asset integrity, and mitigation of risks. In general, these objectives can be achieved by the adoption of appropriate monitoring, diagnosis and maintenance tools that support effective decision-making and customer service.

KPI-based approaches are among the most practical and popular ways to describe the state and efficiency of the plant. Ceschini (2002) states that KPIs also provide guidance for monitoring, availability, maintenance and review of the system’s health and help to derive sound statistics directly from the operational data. Recently automated calculations for machine performance indices have been reported by Ding et al (2013) and Odgaard et al (2013) which significantly focus on developing engineering models of the machine components and drive results using...
statistical methods. Whereas Márquez et al (2012) describes various state-of-the-art techniques including qualitative fault tree analysis for performance monitoring of established thermodynamic models. These reported techniques require greater engineering expertise to build a system model, is less transparent and lacks usability. Nevertheless, the application of KPIs for industrial turbines still has its challenges. Some of the prominent features that introduce substantial complexity to the computation of KPIs are as follows:

1. Forsthoffer (2011) shows that industrial turbines may have many different sets of configurations and topologies depending on design and applications. For example, twin-shaft turbines versus single-shaft and applications for mechanical drive versus power generation. As an example Fig. 1 shows a sample list of various configurations occurring in the industry.

![Figure 1. Samples of different designs and applications of industrial gas turbines.](image)

2. In addition to the complexity of diverse plant descriptions, there is also another dimension of the “level” in the plant model. The plant model gives an overview of the main components of the plant in a hierarchical fashion and comprises many levels. Each level consists of number of individual components and supports level-specific information. Within one level, each component contains its physical parameters relevant to computations. Fig. 2 gives an overview of a generic plant model at site level, plant level, system level and so on.

![Figure 2. Hierarchical structure of a plant model.](image)

3. It is important to note that the interaction and dependences of components within one level as well as between levels may be quite complex and hence creating the model requires greater expertise.

4. Available off-the-shelf statistical approaches as discussed by Ceschini (2002) and Márquez et al (2012) are based entirely on manual data gathering and manual assessment of scenarios for asset downtime. Such data is often contaminated by human factors and potentially by forced business incentives. Even today, service engineers still need to spend considerable time and effort calculating KPIs for a single site.

Considering all the challenges described above in terms of complexity, diverse configurations, interdependences of the plant model and data acquisition, the key idea is to simplify the computation of KPIs in two steps. Firstly, rather than addressing the KPIs of a plant at each level of its hierarchy in isolation, we introduce dedicated level-oriented rules that re-use KPIs already computed on one level for the computation of related KPIs on another. Secondly, in order to avoid re-phrasing KPI computation rules for each of the numerous different turbine configurations, we introduce an abstraction layer hiding the different configurations and define our KPI computation rules against the abstraction layer rather than the actual machine configurations. The abstraction layer will be based on a domain ontology describing turbines, their components and functions. The level-based KPI computation rules mentioned above will equally make use of the ontology providing the abstraction layer but will be encoded as Complex Event Processing (CEP) rules.

For a given specific plant the computation of actual KPIs does not utilize the abstract CEP rules expressed in terms of the ontology-based abstraction layer but rather depends on an instantiation step in which the abstract rule-base is instantiated for the specific plant and its configuration. This instantiation step is based on mappings between the concrete plants and the abstraction layer. The key observation here is that maintaining these mappings for a variety of different plants and configurations is a small task in comparison to maintaining the entire rule base for each plant and configuration. The paper follows with Section 2 describing the basic standards and KPI definitions used in the model for Industrial Turbines. Section 3 presents our case study and the proposed model-based solution architecture Section 4 introduces the basic concepts and application of ontology and complex event processing technology used for KPI computations. Section 5 presents results and serves for the evaluation and future developments. Finally, we conclude in Section 6.

2. Key Performance Indicator Standards

Performance measurement is important to the management of industrial turbines. It identifies performance gaps
between the desired and actual state and provides indications of the progress to meet those gaps. While KPIs are common tools for the measurement of system performance, the choice and definition of specific KPIs for a given system is not trivial.

For our KPI solution framework, we have revised and adopted definitions from the IEEE (2006) and ISO (1999) standards. Since we will rely on historic data in our computations, we introduce an additional parameter, “NoData”, that deals with possible data gaps. The following is a list of basic KPIs used in our solution.

**Period Hours (PH)** – Time, in hours, in the period under consideration.

**No Data Hours (NoData)** – Time, in hours, where not all required data is available, here we use the term PH* for Period hours excluding the no data hours.

**Available Hours (AH)** – Time, in hours, during which the unit was capable of providing service, whether or not it was actually in-service, regardless of the capacity level that it can provide.

**Service Hours (SH)** – Time, in hours, during which the unit was in-service, i.e., it is electrically connected to the system and performing generation function. For gas turbines, this covers from main flame ignition through to flame extinction.

**Reserve Shutdown/Service Hours (RSH)** – Time, in hours, during which the unit was available, but not in service (Number of hours when the gas turbine is available but there is no demand).

**Unavailable Hours (UH)** – Time, in hours, during which the unit was not capable of operation. The unavailable state persists until the unit is made available for operation, either by being synchronized to the system (in-service state) or by being placed in the reserve shutdown state.

**Planned Outage Hours (POH)** – Time, in hours, during which the unit (or a major item of equipment) was originally scheduled for a planned outage with a pre-determined duration plus the extension of planned work beyond this pre-determined duration. Note that the extension due to either a condition discovered during a planned outage or a startup failure would result in a forced (unplanned) outage.

**Forced Outage Hours (FOH)** – Time, in hours, during which the unit was unavailable due to a component failure or another condition that requires the unit to be removed from service immediately or before the next planned outage.

Fig. 3 shows a hierarchical overview of these definitions that forms the basis of the solution framework.

Using the four KPIs defined above, we can compute the following factor KPIs:

**Availability Factor (AF)** – Probability that the unit will be usable at a point in time based on past experience:

\[ AF = \left(\frac{AH}{PH^*}\right) \times 100\% \]

**Unavailability Factor (UF)** – Probability that the unit will be unusable at a point in time based on past experience:

\[ UF = \left(\frac{UH}{PH^*}\right) \times 100\% \]

**Reliability Factor (RF)** – Probability that the unit will not be in a forced outage condition based on past experience:

\[ RF = \left(\frac{PH^* – FOH}{PH^*}\right) \times 100\% \]

**Service Factor (SF)** - Probability that the unit will be in an
operating condition based on the past experience:

\[
SF = \left(\frac{SH}{PH^*}\right) \times 100\%
\]

**Forced Outage Factor (FOF)** – Probability that the unit will be in a forced outage condition based on past experience:

\[
FOF = \left(\frac{FOH}{PH^*}\right) \times 100\%
\]

**Mean Time Between Failures (MTBF)** – Average time between failures initiating a forced outage based on the past experience. Here, FON is the number of forced outages:

\[
MTBF = \frac{SH}{FON}
\]

For simplicity, we also use N/A for indicating the case where the KPI value cannot be correctly computed, e.g., PH == 0, PH == NoData or FON == 0.

### 3. A Novel Approach for KPI with Application to Industrial Gas Turbines

The proposed approach has been applied to a fleet of Siemens industrial gas turbines located at different sites around the globe.

#### 3.1. Case Description

At any given site, see Fig. 4, the plant system consist of two subsystems, namely drive train and balance of plant. Based on the configuration, each drive train subsystem comprises i) a *driver package* (for example, gas turbines or steam turbines), ii) *driven equipment* (for example, a compressor or pump), and iii) *gearbox*. Furthermore, within a driver package, a turbine component may consist of a *gas generator, power turbine and auxiliary system*. Each functional component includes physical parameters and threshold values, for example, speed, load, temperature etc. Each of this physical parameter needs to be configured. This configuration is a mapping of parameters to one or several sensors (for example, Two-out-of-three) and a setting threshold values.

The following section describes the solution architecture. Details on the models used for our approach will be introduced in Sections 4 and 5.

#### 3.2. Solution Architecture

Modeling a plant system is a critical step for constructing KPIs that accurately reflect the impact of actions taken to manage the plant. The proposed approach uses two well-established paradigms from AI, namely ontology and complex event processing, to be discussed separately in the following sections.

In Fig. 5 we present the overall solution architecture: a domain ontology is used to represent turbine configurations and the relationships between different physical components in the plant and their function and performance variables (such as speed, main flame, active power, etc.). The performance parameters describe the primary behavior of the plant at different levels. We store these configurations in a separate database (turbine configuration database) for easy access. In the next step, we model complex events and formulate abstract state indicators rules and update rules for each node. These rules are abstract in that they are defined w.r.t. the ontology-based vocabulary from the configuration database rather than data specific to any individual plant coming from the remote monitoring service database.

In order to actually compute KPIs for a specific plant however, i.e., apply the rules, we instantiate the abstract set of CEP rules with the concrete plant information using a semantic mapping mechanism. Once the instantiation is completed, we proceed with the KPI computation procedure.

![Figure 5. KPI System Architecture](image)

The computation framework uses operational sensor data as well as event streams for its state evaluation. Because of the large amount of data, we use a data cache for storing and post-processing.

### 4. Ontology

As described above, our approach to KPI computation rests on an abstraction layer by means of which a comparatively small set of abstract rules can be instantiated to match a very large set of turbines and their various concrete configurations. At the core of this abstraction layer lies a domain ontology that represents basic knowledge about the compositional structure of plants, types of its components, and their function.

Ontologies are logic-based knowledge representation (KR) formalisms that evolved from frame-systems, see Baader (2003). Chandrasekaran and others. (1999) characterize ontologies as “a formal, explicit specification of a shared
conceptualization”. They usually represent the core notions of a domain of discourse and the relations existing among them. A key advantage of ontologies over many other KR formalisms is their formally well-defined semantics. These enable so-called reasoners to derive implicit knowledge from the explicit ontology statements, detect redundancies and inconsistencies, and discover relationships that may not have been clear to the author of the ontology in the first place. Currently, the most commonly used ontology formalism is OWL\(^1\) and its sublanguages.

Some additional characteristics of Ontology, which address the key challenges in the turbo-machinery domain, are as follows as stated by Ming and Jie (2002):

- Ontologies clarify the structure of knowledge and devices for an effective KR system.
- They separate factual knowledge about the domain from problem-solving knowledge.
- They facilitate sharing and re-using knowledge as well as interoperability of information resources between humans and software agents.

For the purpose of computing KPIs we have developed a domain ontology of turbines, their components and functions. Note that multiple kinds of relationships different from ‘is-a’ can easily be expressed in OWL. Fig. 6 illustrates a basic example with Driver and Driven equipment as classes, Gas turbine as a subclass and SGT-800 as an object, called ‘individual’ in OWL. Object properties, such as ‘provides power’ in the example, establish links between classes or individuals. The combination of object properties and subclass relationships now give rise to additional implicit relationships, for example: “If SGT-800 provides power” then this implies “Generator requires power”. A key advantage of OWL is that all implicit knowledge is fully automatically taken into account by the reasoner. Hence, redundancies and inherent contradictions are detected automatically, leading ultimately to smaller and more easily maintainable models.

For the purpose of computing KPIs we have developed a domain ontology of turbines, their components and functions. Note that multiple kinds of relationships different from ‘is-a’ can easily be expressed in OWL. Fig. 6 illustrates a basic example with Driver and Driven equipment as classes, Gas turbine as a subclass and SGT-800 as an object, called ‘individual’ in OWL. Object properties, such as ‘provides power’ in the example, establish links between classes or individuals. The combination of object properties and subclass relationships now give rise to additional implicit relationships, for example: “If SGT-800 provides power” then this implies “Generator requires power”. A key advantage of OWL is that all implicit knowledge is fully automatically taken into account by the reasoner. Hence, redundancies and inherent contradictions are detected automatically, leading ultimately to smaller and more easily maintainable models.

### 5. Complex Event Processing

Complex Event Processing (CEP) is a paradigm of choice for many monitoring and reactive applications. It supports decentralized information sources by deploying tagging and sensing technology along with integration to real-world objects. CEP helps to build highly scalable and dynamic systems by decoupling the provider and receiver of the information and mediates in form of events. Temporal relations can also be specified by using correlation rules (often called Event Patterns) as mentioned by Robins (2010). CEP also benefits the scalability of the system by reducing the massive event load through stepwise correlation of events.

In general, CEP is used to generate new set of complex events by aggregation and composition. Its processing promotes detection of a plant-significant situation, which typically involves a collection of evaluation conditions and constraints over an event set as founded by Wasserkrug, S., Gal, A., Etzion, O., & Turchin, Y. (2008). Another

---

\(^1\) [www.w3.org/2004/OWL](http://www.w3.org/2004/OWL)
characteristic of CEP is event transformation, filtering, enrichment, pattern recognition, routing, validation etc. Figure 5. serves as an example of constructing new signal event processing rules for speed and load of turbines by using sensor data and events.

Figure 8. Alarms for high speed and low load using CEP Rules

For our solution, we have devised two set of abstract rules that are encoded as CEP rules to identify the state of a plant at different levels. The following sections briefly discuss the implementation and purpose of these rules.

5.1. State Indicator Rules

The state indicator rules define how a given state of a plant is acquired that is useful for our computation. These states are determined using physical parameters, such speed, load, temperature etc. Every plant has its own set of threshold values and specific events from the control system to indicate its performance. These features in our case study is encoded in the abstraction layer i.e. domain ontology. By using expert knowledge, here we formulate abstract set of state rules that can incorporate all configurations of turbines. The three important state indicators are; i) State of Service Hours (SH), ii) State of Outage Hours (OH), and iii) State of Start Attempt (SA) / Start Failure (SF) / Start Success (SS).

Figure 9. State Indicator Rules

In Figure 9, we have a KPI state machine with State indicator rules on edges, for example. a drive train can move from “start-success” state to “service hours” state if the rotor speed is greater than \#value1 RPM and generator load is greater than \#value2 MW. Here the tags $value1$ and $value2$ will be replaced upon instantiation.

Figure 10 gives an overview of the states required for computation. For outage hours (OH), we can define more specifiers. For example: reserve shut down (RSH), forced outage (FOH), and planned outage (POH). For our implementation, we do not go into the details of the outage hours at the moment. Though the solution is flexible enough to identify these states based on the manual entries by the service engineers.

Figure 10. Overview of State Specifiers

5.2. Level Update Rules

The level update rules are formulated to capture the state dependencies at one level of the plant to the other. For example, any entry of outage specifier interval on one system level will lead to the respective “updates” of the outage specifier intervals on the other system levels. One concrete case would be entering a forced outage interval in the gear box. This will lead to a forced outage interval in the drive train, but will be treated as a reserve shutdown interval in the driver package.

This indicates that as soon as an outage specifier, e.g., RSH or FOH, is added to one component, we have to perform so-called Level Update Rules. Figure 11 shows the update mechanism for a drive train at level 1 and gas turbine and generator at level 2. The rules can be:

- If Drive Train is in reserve shutdown state, then gas turbine and generator at level 2 are updated to reserve shutdown state as well.
- If Generator is in forced outage state, then drive train at level 1 is updated to the same state whereas the gas turbine at the same level is updated as reserve shutdown.

Figure 11. Example of level update rules (Part I)
and unit as identified by state CEP rules and its updated version as specified manually by the engineer.

![Figure 12. Example of level update rules (Part II)](image)

6. RESULTS

The first set of results using our KPI application provide an availability and reliability comparisons between three design model of gas turbine by year. These indicators play an important role in decision making and put a real challenge when the system model is complex and involves large set of engineering rules. In comparison to the manual calculations, our results are more reliable and accurate because of the adoption of ontology based configurations and reusable rule production system.

![Figure 13. Availability and Reliability Comparison by turbine type and year](image)

Another visualization of results is with respect to a specific drive train and its respective units within the hierarchy. Most of the recent methodologies do not consider the component and system level setup. Whereas our approach facilitates the engineers and managers to look up for indices at any given hierarchy and package level. Another highlight is the use of sensor data and events together to detect the state of the machine. Therefore, our results are more accurate, reliable and justifiable than any other traditional approaches.

![Figure 14. KPIs per drive train and its units.](image)

Here we incorporate the high level performance indices at the train level where we specifically visualize for the unavailability, availability and no data states for a specific unit. Such kind of visualization is readily available at the dashboard for high level managers and is also helpful to detect malfunctions of the data collectors on site.

![Figure 15. KPI per unit](image)

Similarly, using our approach and generated KPI result database, we can provide different views based on site region or country, customer, driver, driven unit, etc. We claim that our approach is unique and fits best for calculating KPIs in different fashions and provides customized visualization of results that could be integrated.
as a part of monitoring dashboard services. Fig. 16 shows another view of KPI results filtered for a service region.

Figure 16. KPI results based on ServiceRegion

7. CONCLUSION
We demonstrated a KPI systems approach using an abstraction layer based on a domain ontology and complex event processing technology. This allows us to adopt our KPI computations for different turbine types, different control system types and incorporate additional information available from the external systems. We extended the standard definitions from IEEE and ISO to be used for our case-study. The solution makes use of the sensor data and events from the control system to identify turbine states and perform the computations. The solution also provides different visualization of the results. The presented architecture is distributed, extensible and scalable. The computations are automated and have minimum dependency on user-interaction. Hence, they provide reliable and trustworthy results for decision-making. By including the maintenance calendar, we can also automate the computation for the reserve shutdown and planned outage hours. Also the inclusion of events from the control system that specify for the internal and external outage would add value to the application. For the future, the KPI application can be integrated with the remote diagnostic solution framework to evaluate its potential.

ACKNOWLEDGEMENT
We would like to acknowledge Mr. Michal Skubacz for the support and encouragement. Their valuable feedback sessions were important in development of rule-base and software system.

REFERENCES


Linear Temporal Logic (LTL) Based Monitoring of Smart Manufacturing Systems

Gerald Heddy1, Umer Huzaifa2, Peter Beling3, Yacov Haimes4, Jeremy Marvel5, Brian Weiss6, and Amy LaViers7

1,2,3,4,7 University of Virginia, Charlottesville, VA, 22904, USA
grh4aw@virginia.edu
muh5zc@virginia.edu
beling@virginia.edu
haimes@virginia.edu
alaviers@virginia.edu

5,6 National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA
jeremy.marvel@nist.gov
brian.weiss@nist.gov

ABSTRACT

The vision of Smart Manufacturing Systems (SMS) includes collaborative robots that can adapt to a range of scenarios. This vision requires a classification of multiple system behaviors, or sequences of movement, that can achieve the same high-level tasks. Likewise, this vision presents unique challenges regarding the management of environmental variables in concert with discrete, logic-based programming. Overcoming these challenges requires targeted performance and health monitoring of both the logical controller and the physical components of the robotic system. Prognostics and health management (PHM) defines a field of techniques and methods that enable condition-monitoring, diagnostics, and prognostics of physical elements, functional processes, overall systems, etc. PHM is warranted in this effort given that the controller is vulnerable to program changes, which propagate in unexpected ways, logical runtime exceptions, sensor failure, and even bit rot. The physical component’s health is affected by the wear and tear experienced by machines constantly in motion. The controller’s source of faults is inherently discrete, while the latter occurs in a manner that builds up continuously over time. Such a disconnect poses unique challenges for PHM. This paper presents a robotic monitoring system that captures and resolves this disconnect. This effort leverages supervisory robotic control and model checking with linear temporal logic (LTL), presenting them as a novel monitoring system for PHM. This methodology has been demonstrated in a MATLAB-based simulator for an industry inspired use-case in the context of PHM. Future work will use the methodology to develop adaptive, intelligent control strategies to evenly distribute wear on the joints of the robotic arms, maximizing the life of the system.

1. INTRODUCTION

Industries active in the manufacturing sector exist in a competitive landscape where profitability is heavily influenced by their operational directives. A manufacturer choosing to implement Smart Manufacturing Systems (SMS) would likely drive down their costs, improve their manufacturing goals, and meet continuous improvement objectives. Robotics and automation are often a logical and feasible ingredient to increasing productivity, while also maintaining or improving product quality and operational safety goals. A recent national report on advanced manufacturing showed that industry use of automation positively impacted profitability such that manufacturers were more likely to keep their internal operations vertically integrated (Anderson, 2011). This report also highlights the important role that next-generation robotics will play in the future of manufacturing such as realizing improvements in flexibility, time to market, cost, quality, and human safety.

Prognostics and Health Management (PHM) is a comprehensive field that attempts to create the systems and methods which manufacturers employ to enhance their asset maintenance programs. PHM standards are developed as a better alternative to traditional reactive maintenance programs primarily defined by initiating action only after a breakdown or some lost production time event has occurred. It is through the use of condition-monitoring, diagnostic, and prognostic
methods that PHM attempts to understand the states of the system and create a manufacturing environment where maintenance is carried out on a more preventative, predictive, and proactive basis as compared to being purely reactive. A PHM approach to maintenance proves beneficial by reducing manufacturer dependence on non-value added maintenance time and capital of parts replacement. PHM strives to increase asset lifespan while operating at lower cost.

The emergent contributions of robots to higher efficiency and product quality in smart manufacturing processes have also introduced new sources of risk thereto including: (i) safety risks resulting from the collaborative and proximal interface between humans and robots; (ii) maintenance schedule and operations; and (iii) sensitivity to irregularities associated with out-sourced parts and raw materials, among others. In this sense, the centrality of PHM in smart manufacturing has necessitated expansion to embrace systems-based risk modeling, assessment, management, and communication (Haines, 2012, 2005). In particular, the interdependencies between the robotics subsystems and the human operators necessitate an understanding of the epistemological human behavior and responses under extreme events originating from either the robotics or human subsystems.

It is then necessary to think about PHM in the context of robotics as both of these fields (PHM and robotics) enable development of SMS. As private and public investment rises to implement and develop next-generation robotics, we will also need to create the high-level control strategies which seek to attain condition-based PHM goals. This work introduces a novel robotic monitoring system as a step towards PHM with the motivation to display and predict both discrete system failures and continuous motion wear.

After further review of SMS, the paper introduces an industry-inspired use case. We will then apply a novel methodology from (Huzaifa, Umer and Marvel, Jeremy A. and LaViers, Amy E., 2015) that can incorporate a high-level description of the correct behavior for the robotic system to our use-case. This is accomplished with linear temporal logic (LTL) specifications and a labeled, discrete representation of the SMS. By generating a Büchi automaton representation of the high-level specification phrased in LTL, the system dynamic and correct behavior can be represented in the same product automaton. This resulting automaton encodes all system behavior that is within the specification and forms the basis of the monitoring system. This methodology has been implemented in a MATLAB-based simulator, which also tracks a continuous system variable.

Finally, the paper presents results of this methodology with respect to PHM. Correct control sequencing is represented at a high-level using task-level labels for the discrete system model. It is over these task-level labels that the specification will monitor the correct behavior of the system. Wear monitoring is achieved using a differential equation model of wear in both loaded and unloaded conditions. These discrete and continuous statuses are tracked and displayed and will be used to develop corrective control strategies to maximize the lifetime of the robotic system. This work is part of a larger effort to create a modular, adaptive multi-scale PHM scheme (AM-PHM) where we take operational demand profiles, generate performance and health assessments, then create operational objectives.

2. Prognostics and Health Management for Smart Manufacturing Systems

Prognostics and health management (PHM) technologies reduce time and costs for maintenance of products or processes through efficient and cost-effective diagnostic and prognostic activities. In 2010, a comprehensive review was conducted of prognostic and diagnostic methodologies for condition-based maintenance (CBM) that presented the existing strategies within four categories: physical models, knowledge-based models, data-driven models, and combination (hybrid) models (Peng, Dong, & Zuo, 2010). This review highlighted many specific methods across these four categories (e.g., Hidden Markov Models, Bayesian network-related methods, Fuzzy Logic, Principal Components Analysis) along with their successes and limitations. No one method stood out as being sufficient to provide both diagnostic and prognostic intelligence at multiple levels. This review demonstrated that for every method’s strength, there was at least a single weakness. Similarly, another review of existing methods was conducted in 2012 that focused on comparing time-based maintenance (TBM) and condition-based maintenance (CBM) (Ahmad & Kamaruddin, 2012). TBM, commonly referred to as preventative maintenance, is typically simpler to implement (in that maintenance is scheduled based upon a specific unit of time; e.g., cycle time) while CBM, sometimes termed predictive maintenance, may ultimately be more cost effective if a process’s or equipment’s health data accurately reflects its current state and allows a machine to run longer until maintenance (as compared to a TBM schedule). The challenge in CBM is gathering sufficient data to make a reasonably accurate prediction. Both of these studies revealed that PHM is applicable to both products and processes; this makes PHM a tremendous, and necessary, asset to SMS.

Product PHM (providing health monitoring, diagnostics, and/or prognostics for a finished system; e.g., automobile, aircraft, power generation station) is more widespread as compared to process PHM (providing health monitoring, diagnostics, and/or prognostics to a system that integrates one or more pieces of equipment to complete a task; e.g., assembly process, welding process, machining process). (Batzel & Swanson, 2009) (Holland, Barajas, Salman, & Zhang, 2010) (Hu & Koren, 1997) (Shen, Wan, Cui, & Song, 2010). Likewise, PHM techniques have been developed and applied more
widely at the component/equipment level, yet some work has occurred at the higher/system levels. For example, innovative methods have been developed for various machining operations (Al-Habaibeh & Gindy, 2000) (Altuntas, Verl, Brecher, Uriarte, & Pritschow, 2011) (Biehl, Staufenbiel, Recknagel, Denkena, & Bertram, n.d.) (Borisyov, Fletcher, Longstaff, & Myers, 2013). System level PHM methods have also been developed, yet seem to be very focused in their applicability and/or limited in capability (Barajas & Srinivasa, 2008) (Datta, Jize, Maclise, & Goggin, 2004) (Hofmeister, Wagner, & Goodman, 2013).

The paper (Vogl, Weiss, & Donmez, 2014) conducted a detailed review of existing standards that were designed to help guide implementation of PHM in manufacturing. Specifically, many of the current PHM standards were developed within the International Organization for Standardization (ISO) and focus primarily on condition monitoring and diagnostics (ISO, 2002) (ISO, 2003) (ISO, 2012). Few standards include discussion of prognostics (ISO, 2004). The standards review highlighted that only very specific processes benefited from these standards; they are not considered broadly applicable. This study highlights a gap in that no standards are currently available that are both robust and flexible to address the diverse and dynamic environments presented by Smart Manufacturing.

Smart Manufacturing presents a paradigm shift in that manufacturers are thinking differently about how they implement their production technologies, tools, and teams. The field of robotics has already released and is actively working towards a next generation of new products, bolstered by developments in low-level controllers such as proximity detection, image processing, and precise human-safe actuators. In addition, collaborative robotics systems are emerging, enabling robots to work side-by-side with humans and other robots without requiring physical safety barriers. Collaborative robotics are characterized by:

1. Lower total implementation costs
2. Reduced barrier-to-entry in the form of operational technical skill
3. Improved efficiencies and overall equipment effectiveness (OEE) as discussed in (Jeong & Phillips, 2001)
4. Flexible spatial feasibility and responsive configurations
5. Increased safety features allowing humans to work alongside them

For many small and medium-sized manufacturers, the cost of integrating a robot into a historically manual process is the most prohibitive barrier to automation. While the purchase price of a robot is sometimes significant, it is dwarfed by the cost of process integration, programming, and support. Many collaborative robot technologies effectively reduce the overhead associated with safety, programming, and factory floor real estate. As such, the promise of reduced cost and ease of use are seen as a means by which even small and medium-sized enterprises may access and adopt automation technologies (Marvel, 2014).

However, with safety being the principal focus for the current development of collaborative technologies, system performance and reliability have yet to be verified. As such, these systems require the means by which end users can guarantee the application performance, and ultimately establish confidence in the systems on which they will rely. Proper health monitoring and prognostics modeling of system and process performance, in particular, will provide end users with the necessary insights into the reliability of such emerging smart manufacturing technologies.

With this profound interest for installing robotic and other automated platforms, it is increasingly important to create the high-level control strategies necessary for operating them. The competitive landscape has changed the way corporations manage their supply chain solutions. A plant manager cannot lead his or her world class facility with only reactive maintenance systems in place. Rather, PHM based techniques could be seen as a corollary to the cultural principles established in Total Productive Maintenance (TPM) (Nakajima, 1988) and Lean Manufacturing (Shah & Ward, 2003).

3. THE INDUSTRY-INSPIRED USE-CASE

For our use-case, we have created a scenario with two robots collaborating together to accomplish a task in a work cell that is assumed to be a part of an entire production line. The task to be completed can be subdivided into a pick and place operation combined with a drilling operation, as seen later in Figure 3. The pick and place will be performed by a robot which we will name “Ben”. The drilling operation is performed by a robot named “Mike”.

Boxes are generated according to a predetermined cycle time, arriving from an upstream work cell and appearing on a conveyor in front of Ben. Ben picks up a single box after it has been detected, rotates his torso actuators ninety degrees, and places it on a second conveyor that is elevated off the factory floor. Boxes then continue their conveyance route, already facing the correct orientation to receive the drilling operation. When a box is detected in front of Mike, the end effector is extended, grabs the box, drills a hole, and retracts the arm.

We will engage the use case to show the many motion trajectories that could be employed to accomplish this specific work cell’s task. It is an exciting contribution of the work to introduce the notion that we can generate redundant motion sequences to be leveraged for PHM. These will later be identified by the novel monitoring methodology achieved by a formalized separation between the overall system task and the single strategy employed at any one point in time.
It should be noted the use-case assumes a dynamic model of wear that shows increases in wear over time as the number of movements increase in the robot. We are also using a discrete transition system model of each robot’s behavior and capabilities.

4. An LTL-Based Monitoring Simulator For The Industrial Use-Case

We will now review the individual components of the software simulator framework as implemented on the industry inspired use case. This includes the representation of the involved robot subsystems as discrete transition systems. Further, we explain the linear temporal logic based high level objective description and monitoring.

4.1. Transition System Representation

The two robots in our use case are represented in the form of discrete transition systems. A discrete transition system is a well known concept in Computer Science where it is extensively used in formal proofs for different algorithms and software. For our case, we have also incorporated a continuous state variable in the respective transition systems for representing the wear in the robots. The transition systems of the robots for the industry inspired use-case are given in the Fig. 1. Using notation described in (LaViers, Chen, Belta, & Egerstedt, 2011), for the two robots this representation is given as:

\[ T_1 = (Q_1, q_0, \rightarrow_1, \Pi_1, h_1, C_1, w_1), \]
\[ T_2 = (Q_2, q_0, \rightarrow_2, \Pi_2, h_2, C_2, w_2), \]

\( T_1 \) represents transition system for Mike and \( T_2 \) represents transition system for Ben where

(i) \( Q_1 = \{ q_1^1, q_2^1, q_3^1, q_4^1 \} \) is the finite set of Mike’s states, either hand labeled by a user or generated automatically through a segmentation framework. \( Q_2 = \{ q_2^2, \ldots, q_6^2, q_9^2 \} \) is the similar set of Ben’s states. Superscripts indicate the robot (1 is for Mike, 2 is for Ben).

(ii) \( q_0 \) and \( q_0^2 \) are the initial states of Mike and Ben respectively;

(iii) \( \rightarrow_1 \subseteq Q_1 \times Q_1 \) is a reflexive transition relation of Mike (if \( i = 1 \)) or Ben (if \( i = 2 \)), where each state has a self-loop, allowing for one robot to transition to a new state without that requirement being imposed on the other robot;

(iv) \( \Pi_1 = \{ M_{initial}, M_{opengrip}, M_{detect}, M_{drillready}, M_{drill}, M_{closedgrip} \} \) is a finite set of atomic propositions for robot Mike. Similarly, \( \Pi_2 = \{ B_{initial}, B_{opengrip}, B_{Detect}, B_{Drop}, B_{grapplready}, B_{Grab}, B_{hold}, B_{IntermediatePos}, B_{DropReady} \} \) is a finite set of propositions for Ben.

These propositions represent the status of different sub-tasks performed by Mike and Ben respectively;

(v) \( h_1 : Q_1 \rightarrow 2^{\Pi_1} \) is a satisfaction (output) map, where state \( q_1^j \) satisfies the set \( h_1(q_1^j) \) of propositions from \( \Pi_1 \). \( 2^{\Pi_1} \) represents a set of all possible combinations of propositions of one robot. Thus, \( h_1 \) is a mapping of these combinations to each one of the states in the robot \( i \). It can be seen in Fig. 1 how each of the states has a combination of individual propositions;

(vii) \( C_1 \) and \( C_2 \) are sets of pairs of the form \( (f(x,t), \tau) \). For \( C_1 \) we have \( \{ f_1^1(x,t), \tau_1^1 \}, \ldots, \{ f_9^1(x,t), \tau_9^1 \} \) such that \( f_1^1(x,t) \) represents dynamics of a continuous parameter for duration of \( \tau_1^1 \). In the final pair, \( n = 6 \) and defines the number of degrees of freedom in Mike; \( r = 13 \) is and the number of motion primitives in Mike; \( e = 2 \) representing the two environmental cases e.g., loaded and unloaded condition, for Mike. Similarly for \( C_2 \) we have \( \{ f_2^1(x,t), \tau_2^1 \}, \ldots, \{ f_8^1(x,t), \tau_8^1 \} \);

(vi) \( w_1 : \rightarrow_1 \mapsto C_1 \) and \( w_2 : \rightarrow_2 \mapsto C_2 \). \( w_1 \) and \( w_2 \) are mapping from each transition for a respective robot to a pair in corresponding \( C_1 \) and \( C_2 \). More simply, it is a function that maps all the transitions of a robot to a corresponding wear dynamic.

The states correspond to the robot states while performing the tasks. For example, a state can be the idle state when the robot is waiting for the sensor to detect the box in front of it. The atomic propositions represent statements about the states of the robot and they can be either true or false. The linear temporal logic (LTL) specifications, as will be explained in the next subsection, are described in terms of these statements and the system evolves in terms of them.

The next task is to combine the representation of different robots to describe the whole system in terms of a single transition system. This can be achieved using the composition of the two transition systems. This composition is achieved by taking synchronous product of the transition systems for the individual robots.

The synchronous product of two transition systems \( T_1 \) and \( T_2 \), denoted as \( T_p = T_1 \otimes T_2 \), is a new transition system with \( \langle Q_p, q_p, \rightarrow_p, \Pi_p, h_p \rangle \). The states are Cartesian pairs of the single robot states, i.e., \( Q_p \subseteq Q_1 \times Q_2 \), likewise \( q_p = (q_1, q_2) \). Transitions exist between these joint states if and only if a transition existed between both single states, i.e., \( \rightarrow_p \subseteq Q_p \times Q_p \) is defined by \( (q, q') \in \rightarrow_p \) if and only if \( q \neq q' \) and \( (q_1, q_1') \in \rightarrow_1 \) and \( (q_2, q_2') \in \rightarrow_2 \), where \( q = (q_1, q_2) \) and \( q' = (q_1', q_2') \). The set containing atomic propositions for the composition of the two transition systems, denoted as \( \Pi_p \), is a union of the individual sets of propositions for the two robots that extends to include propositions which apply to situations where both robots are active.

Now we have the transition system for the two robots defined.
With a formal representation of the robots, we can now define high level tasks for the robots in terms of the states. This is accomplished with LTL specifications and their representation in the form of Büchi automaton. Next we describe the LTL based specifications.

4.2. Linear Temporal Logic (LTL) Specifications

What we want is a tailored transition system according to the high level objectives. This is where the LTL specifications come in. A brief introduction of the LTL operators is given as follows:

LTL formulas are described in terms of the set \( \Pi \) of atomic propositions. LTL specifications describe the high level objectives in the form of Boolean and temporal operators. The Boolean logic operators, that have been used, include, \( \neg \) (negation), \( \lor \) (disjunction), \( \land \) (conjunction), and \( \rightarrow \) (implication). The temporal operators include, \( X \) (next), \( U \) (until), \( F \) (eventually), and \( G \) (always). LTL formulas are defined over infinite words generated by the transition systems. In particular, the LTL specifications we define, describe the possible actions of our system of robots, \( T_p \).

An LTL formula \( \phi \) is said to satisfy a word of the transition system if the formula \( \phi \) is true for the first position of the word; \( X \phi \) states that at the next state, an LTL formula \( \phi \) is true; \( F \phi \) means that the LTL formula \( \phi \) eventually becomes true at some position of the word; \( G \phi \) means that the LTL formula \( \phi \) is true for all the positions of the word; \( \phi_1 U \phi_2 \) means \( \phi_2 \) eventually becomes true at some position in the word and \( \phi_1 \) is true until that position of the word comes. More complex and sophisticated specifications can be designed using a combination of Boolean and temporal operators. For details (Clarke, Peled, & Grumberg, 1999) can be consulted.

As an example, some high level objectives and their LTL representations are given below. We will only show the basic LTL form \( G(Proposition_1 \rightarrow Proposition_2) \), as this will be the most common form used in practice by manufacturers in specifying their high level objectives.

(i) Ben! Stay in initial position when Mike is performing drilling
\[ G(M_{drill} \rightarrow B_{initial}) \]

(ii) Mike! do not grip unless you are in the drilling position
\[ G(M_{closedgrip} \rightarrow M_{drill}) \]

(iii) Ben! do not open your hand while you are holding the box
\[ G(\neg B_{hold} \rightarrow B_{open}) \]

(iv) Mike! Stay in initial position when Ben is dropping the box
\[ G(B_{Drop} \rightarrow M_{initial}) \]

Figure 2. Büchi Automaton representation of an LTL specification

To check whether all words of the transition system, \( T_p \), satisfy an LTL formula \( \phi \) over the set of propositions \( \Pi_p \), we need to have Büchi Automaton that accepts only the words satisfying \( \phi \). By the help of a tool, LTL2BA (Gastin & Oddoux, 2001), we are able to get a Büchi Automaton \( B_\phi \) from the LTL specification \( \phi \). For example, the first specification can be given in the Büchi Automaton form as pictured in Fig. 2.

A tailored representation of the system can then be had by taking a product of the system transition system \( T_p \) and \( B_\phi \) to get the final automaton \( A \). Now this automaton as mentioned earlier represents all the allowed transitions between states of the system in light of the specifications defined in \( \phi \). The
LTL specifications are defined in such a way that they define the desired behavior of the whole system. We monitor the behavior of the system by monitoring the transitions in the system. If an error occurs, because of a sensor failure, robot motor failure etc., these specifications are not satisfied and the monitoring system returns a sequence that is not satisfied by $T_P \times B_Q$. We monitor and verify the desired movements of the robots based on the allowed transitions by using an interface between MATLAB and VREP.

5. Applications To PHM

Through the use of LTL we are able to build the discrete sensor oriented piece of the monitoring scheme. The transition system’s representation of the continuous parameter for each robot, $C_1$ and $C_2$, allows us to track differential wear functions over time. The two of these combine to create the complete system monitor for use in PHM.

5.1. Results of the LTL-Based Monitor

Figure 3 depicts the three dimensional model of the robotic work-cell in the VREP simulation environment. Figure 4 shows the MATLAB interface displaying continuous time wear parameters and the cycle time associated with the two robots along with the discrete system information. In the top figure, continuous information for the whole system has been presented. This includes wear information of all the joints of the robots according to the dynamic functions defined in the previous section. For each of the robots, wear has been computed for all of the six joints. It can be observed that wear curves for robot Ben are more evenly spread on to all the joints. In comparison, wear curves for robot Mike are mostly defined by joint 6. The third graph in Figure 4(a) represents the cycle time for each task that Mike and Ben are performing.

Figure 4(b) conveys information of the system’s discrete nature. The Motion Primitives section indicates the current motion primitive of Ben and Mike by filling the corresponding circle for the motion primitive. Discrete Objective states the high level overall objective of the system. Overall Status indicates if the high level objective specifications are satisfied or violated by toggling the color of the corresponding bubble.

A generalizable structure of the work is defined by Figure 5. The figure is specifically for the use-case where we have two robots that collaborate with each other, but could be extended to include any number of Robotic Transition Systems. The Robotic Transition Systems, which abstract the physical robots present on the factory floor, are subsequently transformed into the overall Manufacturing System Automaton. The plant maintenance team or robotics engineers create the high-level LTL specifications to monitor the discrete exceptions of the Manufacturing System, which is then mathematically written as the Büchi Automaton of the LTL Spec. The LTL Spec and Manufacturing System Automaton can then be represented in the same automaton, which finally becomes the Discrete System Monitor for PHM applications. The actuator wear is also projected for each joint with respect to the robotic systems to monitor continuous parameters. Discrete and Continuous Prognostic Indicators are finally realized, which is exemplified by the MATLAB interface in Figure 4.

5.2. Application to Adaptive Multi-Scale PHM

As previously stated, this paper is a part of a larger effort to create an adaptive multi-scale PHM scheme described in (Choo, Beling, LaViers, Marvel, & Weiss, 2015). Adaptive multi-scale prognostics and health management (AM-PHM) is a methodology designed to support PHM in smart manufacturing systems. AM-PHM is characterized by its incorporation of multi-level, hierarchical relationships and PHM information. AM-PHM utilizes diagnostic and prognostic information regarding the current health of the system and constituent components, and propagates it up the hierarchical structure. By doing so, the AM-PHM methodology creates actionable prognostic and diagnostic information along the manufacturing process hierarchy. This information includes the predicted health state upon completion of a task. The health estimates that flow up the hierarchy are based upon simulated operational directives that flow down it. Nodes at given levels along the system hierarchy consume operational profiles from adjacent, higher level nodes. These profiles describe the production goals under consideration by the decision makers (e.g., operators and supervisors) in the superior level. In addition to the traditional workload, bill of materials, and requirements of the manufacturing process, the operational profile may have a focused objective such as minimizing cost or maximizing reliability. Each AM-PHM mod-
(a) Continuous wear information for each robot, Ben and Mike, and their respective task cycle times

Figure 4. MATLAB interfaces for the continuous and discrete pieces of the monitoring framework

(b) Discrete information showing the active motion primitives, current system objective, and a status indicator showing if the high level objective is currently satisfied.

Figure 5. A more general representation of the LTL based monitoring system applied to the use-case where two robots are working together to accomplish a task.

The simulator framework described in this paper would provide the capability to estimate wear and other measures of system health with respect to given operational profiles, and so could be the basis for upward push of prognostics and health estimates. In an attempt to deliver true adaptable and scalable information for translating operational profiles into operational directives, LTL specifications can be hierarchical in nature to capture subtopic levels, or the individual motors, and head topic levels, which is the team process flow.

6. CONCLUSION

The paradigm shift in Advanced Manufacturing, where manufacturers are introducing the next generation of flexible and collaborative robotics, has the potential to further shape the sector. This shift, along with Prognostic and Health Management techniques, is a large part of what will enable Smart Manufacturing Systems. The novel LTL-based monitor reviewed in this work introduces a method for connecting continuous and discrete prognostics, and is immediately applicable to the robotic platforms that manufacturers seek to install in their factories.

We have applied this monitor to an industry-inspired use-case and showed in a three dimensional simulation environment how the methodology can be integrated on a robotic work-
cell. The differential wear functions can be installed to fit the manufacturer specific application, and handled by the automated computing environment for generating wear diagnostics. Intuitive high-level specifications can be applied by systems integrators or plant supervisors for filtering out discrete exceptions. This is especially important as production lines in the advanced manufacturing setting employ an increasing suite of sensors to observe their processes.

Therefore, we have laid the ground work for building intelligent control strategies to evenly spread wear of robotic platforms, ergo maximizing the life of the system. Future work will leverage the supervisory control and model checking found in the monitor to define the multiple ways motions can be performed, and then switch between styles of motion to best extend asset life. This automated flexibility continues to close the gap on waste, both in the form of time and capital expenditure.

The LTL-monitor serves as a blueprint for implementing PHM in robotics and all other forms of automation. The protocols can be written to allow for information flow into the larger supply chain systems scheme, further bolstering the Adaptive, Multi-scale PHM environment. The overall vision gives plant leadership teams and operations management alike the structure to seamlessly integrate their manufacturing capabilities with market demand. As pressures for profitability continue, this will undoubtedly be of interest to industry to ensure productivity, quality, and safety goals.

ACKNOWLEDGMENT

This research is supported by the National Institute of Standards and Technology. We would also like to thank the various industry partners whose support of our research proved invaluable.

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES


Biographies

Gerald Heddy is a M.S. student in the Department of Systems and Information Engineering at the University of Virginia. His research interests lie in the field of Robotics as applied to the Advanced Manufacturing sector and Smart Manufacturing Systems. Prior to UVa, he received his B.S. from Lehigh University in Industrial Engineering and worked in manufacturing at The Hershey Company as a Business Unit Team Leader.

Umer Huzaifa is a Ph.D. student in the Department of Systems and Information Engineering at the University of Virginia. He holds a B.S. degree in Electrical Engineering from University of Pakistan. He has a background in Electrical Engineering with a focus on control systems and robotics. His research interests lie in the field of supervisory control and hybrid systems. Specifically, he is focused on developing mathematical tools for complex motion planning of mobile robots.

Dr. Peter A. Beling is an associate professor in the Department of Systems and Information Engineering at the University of Virginia. Dr. Beling received his Ph.D. in Operations Research from the University of California at Berkeley. Dr. Beling's research interests are in the area of decision-making in complex systems, with emphasis on adaptive decision support systems and on model-based approaches to system-of-systems design and assessment. His research has found application in a variety of domains, including prognostics and health management, mission-focused cybersecurity, and financial decision-making.

Dr. Yacov Haimes is the Lawrence R. Quarles Professor of Systems and Information Engineering, Civil and Environmental Engineering, and Founding Director (1987) of the Center for Risk Management of Engineering Systems at the University of Virginia. He received his M.S. and Ph.D. (with Distinction) degrees in Systems Engineering from UCLA, and his B.S. degree in Mathematics, Physics, and Chemistry from the Hebrew University, Jerusalem. On the faculty of Case Western Reserve University (1970-1987), he chaired the Systems Engineering Department. As an American Association for the Advancement of Science-American Geophysical Union Congressional Science Fellow (1977-78), Dr. Haimes served in the Office of Science and Technology Policy, Executive Office of the President, and on the United States House of Representatives Science and Technology Committee. From 1990 to 2011 he served as a consultant to the Software Engineering Institute, Carnegie Mellon University, and for the last decade as a visiting scientist. He is a Fellow of seven societies: ASCE, IEEE, INCOSE, AWRA, IWRA, AAAS, and Society for Risk Analysis (SRA), (where he is a past President). The Fourth Edition of his most recent book, Risk Modeling, Assessment, and Management, will be published by Wiley and Sons in February 2015 (the first three editions were published in 1998, 2004 and 2009). Professor Haimes is the re-
Recipient of the 2014 ASCE American Academy of Water Resources Engineers Founders Award; the 2010 Distinguished Educator Award presented by SRA; the 2007 Icko Iben Award presented by AWRA; the 2001 Norbert Weiner Award, presented by IEEE-SMC; the 2000 Distinguished Achievement Award, presented by SRA; the 1997 Warren A. Hall Medal, the highest award presented by Universities Council on Water Resources; the 1995 Georg Cantor Award, presented by the International Society on Multiple Criteria Decision Making, and the 1994 Outstanding Contribution Award presented by the IEEE-SMC; following Hurricane Katrina, he was appointed as the 2007 Arthur Maass-Gilbert White Fellow under the U.S Army Corps of Engineers, Institute for Water Resources (IWR), among others. He is a registered Professional Engineer in Ohio and Virginia; Diplomate of the American Academy of Water Resources Engineers (and a Founding Trustee of the AAWRE); the Past Engineering Area Editor of Risk Analysis: An International Journal. He has authored (and co-authored) six books and 300 technical publications, over 200 of which were published in archival-refereed journals. He has served as dissertation/thesis advisor to 38 Ph.D. and over 80 M.S. students.

Dr. Jeremy Marvel is a project leader and research scientist in the Intelligent Systems Division of the National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Dr. Marvel received his Ph.D. in 2010 in computer engineering from Case Western Reserve University in Cleveland, OH. Since joining the research staff at NIST, he has established the Collaborative Robotics Laboratory, which is engaged in research dedicated to developing test methods and metrics for the performance and safety assessments of collaborative robotic technologies. His research focuses on intelligent and adaptive solutions for robot applications, with particular attention paid to human-robot collaborations, multi-robot coordination, safety, perception, self-guided learning, and automated parameter optimization. Jeremy is currently engaged in developing measurement science methods and artifacts for the integration and application of robots in collaborative assembly tasks for manufacturing.

Dr. Brian A. Weiss has a B.S. in Mechanical Engineering (2000), Professional Masters in Engineering (2003), and Ph.D. in Mechanical Engineering (2012) from the University of Maryland, College Park, Maryland, USA. He is currently the Associate Program Manager of the Smart Manufacturing Operations Planning and Control program and the Project Leader of the Prognostics and Health Management for Smart Manufacturing Systems project within the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). Prior to his leadership roles in the SMOPAC program and the PHM4SMS project, he spent 15 years conducting performance assessments across numerous military and first response technologies including autonomous unmanned ground vehicles; tactical applications operating on Android devices; advanced soldier sensor technologies; free-form, two-way, speech-to-speech translation devices for tactical use; urban search and rescue robots; and bomb disposal robots. His efforts have earned him numerous awards including a Department of Commerce Gold Medal (2013), Silver Medal (2011), Bronze Medals (2004 and 2008), and the Jacob Rabinow Applied Research Award (2006).

Dr. Amy LaViers is an Assistant Professor in Systems and Information Engineering and Director of the Robotics, Automation, and Dance Lab at the University of Virginia. She aims to extract useful features from human movement for robotic applications, such as, endowing co-robots the ability to work alongside human workers in manufacturing plants. She also works to classify organized human movement, such as, providing quantitative comparison between genres of movement styles. Her research began at Princeton University where she earned a certificate in Dance and B.S.E. in Mechanical and Aerospace Engineering. She went on to complete a M.S. and Ph.D. in Electrical and Computer Engineering at the Georgia Institute of Technology. She is currently enrolled in the Laban/Bartenieff Institute for Movement Studies Certification in Movement Analysis (CMA) Modular program.
Early Concepts for the Coupling of a Nuclear Plant Computer to a Computerized Maintenance Management System for Autonomous Prognostic Model Development

Zachary Welz\textsuperscript{1}, J. Wesley Hines\textsuperscript{1}, and Belle R. Upadhyaya\textsuperscript{1}

\textsuperscript{1}Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996, USA
zwelz@utk.edu
jhines2@utk.edu
bupadhy@utk.edu

\textbf{ABSTRACT}

Every day large amounts of process data are recorded in a variety of industries. For nuclear power plants, these data are stored within the Plant Computer (PC). As parts begin to degrade and components fail, maintenance personnel are responsible for making repairs and recording these repairs in a Computerized Maintenance Management System (CMMS). By coupling the information in the PC and CMMS, failure data can be extracted and repurposed for lifecycle prognostic models. Existing prognostic methods can be utilized to develop lifecycle models and predict the Remaining Useful Life (RUL). These efforts are currently done manually and require substantial amounts of time to develop. This results in offline predictions, which can drastically reduce response time for preventative maintenance. This paper outlines an early concept that uses data mining based on Big Data efforts in order to couple the plant computer data with the CMMS so that prognostic information can be gathered, sorted, and analyzed automatically. The extracted failure data can be used to autonomously update or build prognostic models based on component failure times, stressor information, and signal/residual values. An effective future implementation of this concept means that the results could be used as \textit{a priori} prognostic information in lifecycle prognostic models, and the updating and/or development of such models can be automated for improved response time.

1. INTRODUCTION

Lifecycle prognostics describes a set of data based models that can potentially give an accurate determination of the health of a system or component using several different forms of data such as usage time, usage stress, and degradation indicators. As more information is collected, knowledge of the system increases allowing for increasingly advanced models and the possibility of increased prediction accuracy. These data based models are valuable for condition-based maintenance efforts including preventive maintenance. In order to effectively extract and utilize prognostic information from existing operations, it is necessary to develop a semi-autonomous extraction routine. This algorithm would be responsible for repurposing maintenance information with the intent to retrieve failure data from process files. The result of such an algorithm is a detailed network of failure information on specific parts and systems, which is the main component necessary to update and develop predictive maintenance models. It is necessary for this extraction to be carried out autonomously to increase decision time and decrease uncertainty for the operator. The extraction algorithm is the first step of the process towards improving predictive maintenance in commercial applications, and will be loosely outlined in the following sections of this report. The second step of the process is the utilization of the extracted failure data to build or update prognostic models in a quick and efficient manner. The set of tools needed to achieve this goal must meet several requirements including near-to-full autonomy and high confidence decision-making in order to have utility in pre-existing commercial applications.

To make the concepts discussed in this paper easier to follow, examples will discuss how the proposed design might affect a system that includes a 3-phase motor and pump combination. The ideas discussed should be scalable to most nuclear plant components with modifications to aspects such as choice of measurements. Specifics on the application to assets and components will not be covered for this early discussion.
2. OVERVIEW OF LIFECYCLE PROGNOSTICS

The following section will highlight the important aspects of lifecycle prognostic models, and the types of information necessary to build them.

There are several steps involved in the development of a lifecycle prognostic model. The path from data collection to risk mitigation is outlined in Figure 1.

![Figure 1. Path of phases from data collection to risk mitigation including detection, diagnosis, and prognosis (Hines 2008).](image)

The first step of lifecycle prognostics is data acquisition. During the data collection process, monitoring is conducted that can detect anomalies or faults from on-line data samples. Faulted data is then compared to an analytical, empirical, or hybrid model to determine residuals that are related to degradation of the system. These residuals are combined into a system health indicator that is a measure of the total degradation in the system. Using a prognostic model, these health indicators, or prognostic parameters, are used to obtain RUL predictions. The RUL predictions are subsequently available for risk analysis and mitigation.

Prognostic models are typically divided into three different types depending on the failure data available. Type I prognostics is based on past failure time distributions and is often referred to as traditional reliability analysis. This type of prognostic model only utilizes past failure times and does not require any additional failure data. Therefore, it can be conducted before operation of additional cycles. During operation, as stressor information such as operating condition or load is obtained, the model transitions to a Type II prognostic model. In parallel to the Type II models, anomaly detection can be conducted on the failure data. When specific signals such as temperature or pressure are tracked over lifetime and show an increase in damage to the system, the Type I or II model transitions to a Type III prognostic model. Type III models use the tracked degradation across multiple signals to measure the overall system health. The transitions between prognostic model types can be seen in Figure 2.

![Figure 2. Transition between prognostic model types dependent on availability of failure information (Nam 2015).](image)

Only the component run time is needed to perform Type I prognostics, which can be mined directly from the CMMS. The ability to develop Type II and Type III prognostic models is dependent on the ability to collect stressor and degradation data for the component or system in question. With respect to nuclear power plant applications, these data are stored between the PC and CMMS. Data from both of these sources are necessary to develop these lifecycle models. Effectiveness of the proposed concepts will be dependent on the ability to obtain these data for future validation and development.

3. OVERVIEW OF PUMP-MOTOR SYSTEM

To show how the proposed concepts can be applied to a real world situation, a pump-motor system is being used. A simple diagram of this system is shown in Figure 3.

![Figure 3. Diagram of simple motor-pump system. The top of the figure contains a separated 3-phase motor with labeled cooling fan (1) and coil windings (2). The output shaft of the motor (3) spins the impeller (4) by feeding into the impeller eye (5) (Hernandez 2006) (Skvarenina 2004).](image)
Within this theoretical setup, there are several fault or failure modes to consider. For the 3-phase motor, the cooling fan (1) in Figure 3 can break a blade reducing cooling to the motor. This would be an example of a system fault Also) bearings that surround the motor shaft (3) can fail. The pump consists of one major failure mode, which occurs with degradation of the impeller fins. There are several ways that the fins can degrade, which are visually represented in Figure 4.

Figure 4. Representation of 6 types of impeller degradation: (1) fin-pitting, (2) vein tearing, (3) flattening, (4) vein ripping, (5) bowing, and (6) vein-pitting.

Impeller fins are used to regulate a vacuum within the pump, which creates the differential pressure needed to drive the fluid. The effect of each of the degradation forms listed in Figure 4 may have a unique effect on the pump-motor system. The distinctiveness of each degradation type is important for cataloging of CMMS entries.

Aside from degradation/failure modes, the availability of sensors in a system also affects decision-making. The assumption of what measurements are available both on-line and those taken during maintenance events has serious impact on the ability to effectively develop a data-mining algorithm. The sensor data available will change the degree to which Type II and III prognostic information can be discovered for the nuclear power plant, or more specifically, the system or component under investigation. With commercially available signals, effective lifecycle prognostics models have already been developed for pumps (Jeffries 2014), motors (Nam 2013), and other systems such as heat exchangers (Welz 2014). The practical development of these models is limited by the current availability of plant data. The coupling algorithm will be designed to increase the availability of necessary diagnostic data and consequently the ability to develop these prognostic models.

The choice of a pump-motor system is very meticulous with respect to this paper’s application to nuclear power plants. There are numerous pumps within the plant that are critical to plant operation. An example is the primary Reactor Coolant Pump (RCP) in a Pressurized Water Reactor (PWR).

This means that accurate prediction of pump-motor failure times could have a significant impact on planned and predictive maintenance. It is important to note, however, that the specifics of this system are arbitrarily chosen to provide insight into the purpose and functionality of the autonomous prognostic program.

4. Nuclear Plant CMMS Framework Challenges

To effectively mine out maintenance information from the CMMS, certain information in regards to the parts and components being serviced must be recorded (Bertolini 2013). For example, rather than a record stating that pump 6 was serviced at 12:03pm on cycle-day 4, the CMMS would require additional specific information such as pump model, reason for maintenance, activity (primary or redundant), recorded time of failure, etc. This would provide additional knowledge to increase the ability of a data mining algorithm to locate useful information.

Another aspect of the CMMS software that should be evaluated is the need for asset-specific maintenance information. Current CMMS work orders are tailored to an application, but not always a specific system. To gather useful information on a specific part or component, a pump may need a different CMMS record than a motor. Availability of additional information may directly affect the resulting models. This type of customized CMMS database may be necessary as the data mining algorithm is being developed, evaluated, and validated during future research.

As previously mentioned, specifics on the design of a CMMS standard for coupling with the nuclear plant computer will be decided based on future research needs. The CMMS design will be directly related to the specific needs of the data-mining program within the coupling algorithm with respect to the development of lifecycle prognostic models. The availability of these data, and the ability to manipulate existing CMMS frameworks are two of the major challenges in the development of a coupling algorithm.

5. Plant Computer Framework Challenges

Current nuclear plants have most of the information required to perform lifecycle prognostics on components, but access to the data is not always straightforward. Some components may need additional sensors, data collection systems, and data storage systems. Additionally, future plants could have/need data systems specifically designed to support lifecycle prognostics. These challenges will largely affect the success of a coupling algorithm. With the intent to apply concepts discussed in this paper to the nuclear fleet, any changes to the plant computer framework would likely need to be minimal. With the Nuclear Regulatory Commission (NRC) supervising critical plant design standards, substantial changes to the plant computer would be difficult to implement.
Other than challenges such as the need for additional transducers and data acquisition systems, there are a few minor changes that may need to be made depending on current plant computer operations. Similar to the CMMS, specifics on changes to the plant computer framework will be dependent on the coupling algorithm design. For example, one modification that may need to be made is the sampling frequency. There are several methods of frequency analysis that require a specific or large sample rate, therefore if the plant computer takes data from a sensor at 1 Hz, it may need to be increased to 100 Hz. Changes such as this may have minimal impact on plant computer operations, and will need to be carefully examined before implementation.

In the design of the coupling algorithm, several assumptions about the current software design of plant computers in the U.S. will have to be made. As a generic guideline for current American plant computer design, Westinghouse pressurized water reactor nuclear power plant documentation (Westinghouse 1984) will be used.

6. Early Coupling Algorithm Concept Design

The coupling algorithm is tasked with collecting useable prognostic information from the CMMS and combining it with information mined from the plant computer data. Extracting data from existing systems is the first step of predictive model development. Manual extraction from human efforts is ineffective and time consuming for many applications. The idea behind the coupling algorithm is a self-sustaining procedure with its own runtime that can remove the necessity for human facilitated data extraction. An early conceptual design of the algorithm is shown in Figure 5.

During this process, the user can either specify constraints on the data extraction, or the computer will choose constraints based on optimization efforts. The user will always be aware of the program’s status through the use of alert tools (progress bars, status beacons, etc.). Extracted data will be sent to a directory, which will be displayed to the operator in the event that the worker needs to intervene or wishes to catalogue specific files for offline evaluation. Visual aids will be used to display current system information to the operator, such as most recently extracted raw failure data, cross-correlation values between signals during the latest cycle, and even proximate fault detection results. These tools are grouped in a manner that will provide near-instantaneous information to the end-user.

After extraction, the data is sorted depending on the corresponding prognostic model type, and stored for later use in lifecycle prognostic models. Historical data has a different utility than current cycle data. As the algorithm strips out a current cycle, the data can be used for monitoring. Once the information has been sent to a prognostic database, it can be used to update existing models as a separate task of continuous model improvement. Current cycle data is of key importance for critical decision-making.

7. Prognostic Information Database

It is necessary to provide background into the inputs for prognostic models; the link between the coupling algorithm and the model inputs is a detailed prognostic information database. The coupling algorithm is responsible for sorting extracted data into this database, which has a structure similar to that in Figure 6.
By sorting different types of prognostic data into an isolated
database, the files are immediately available for model
development purposes. This database is standalone from the
coupling algorithm and model development to allow for each
to run simultaneously and independently from one another.
The utilization of the extracted data as input to prognostic
models is discussed below.

**8. AUTOMATED PROGNOSTIC MODEL TOOLKIT CONCEPTS**

The purpose of extracting useful prognostic data from
historical process data files is to improve maintenance efforts.
Two of the primary concerns in the development of predictive
maintenance models are quick response time, and high
prediction/model confidence. There are several stages in the
development of prognostic models when utilizing data
gathered from the coupling algorithm. Lifecycle prognostic
models can be updated/transitioned as additional information
is gathered. The first information stored in the database will
be simple failure times for the different components, which
is Type I prognostics. As additional failure times are
measured, monitoring efforts can be updated and prognostics
based on current information can be assessed. When stressor
information is incorporated with the preexisting failure times,
the model can be updated to Type II prognostics. As the
coupling algorithm reads in more data, it will be able to
extract useful signal values related to the failure times of
components. For example, if the inlet water temperature
sensor value for a pump increases over each failure cycle, the
coupling algorithm will identify that temperature signal as
useful. Once the coupling algorithm has identified several
useful degradation signals, it will store them as Type III
prognostic data. All three forms of prognostic data can be
used to update existing monitoring and prognostic efforts.

The main focus of this prognostic model “toolkit” is the
automation of model efforts. Automating the model process
results in a standard for data development, which reduces
variance in decision-making and increases model
development time. Outside of the transition from manual to
autonomous model development, current industry standard
and state-of-the-art prognostic methods are still utilized for
internal functionality. Additional sub-algorithms will be
included to improve the automated results, but will still rely
on existing state-of-the-art methods.

One example of a supplementary function is a model update
tool that distinguishes between assets repaired to “as good as
used (AGAU)” and “as good as new (AGAN)” conditions.
Based on the degree of repair, the model is altered in different
ways. In Figure 7, the failure distributions are provided for
different outcomes in order to highlight the differences
between a single run to failure, a run and repair to AGAU
condition, and a run and replacement to AGAN condition.

![Figure 6. Prognostic information database example structure](image)

![Figure 7. Representation of the differences in model
distributions for “as good as used” and “as good as new”
maintenance levels](image)
such as this within the toolkit, as well as industry standard prognostic model functionality.

Due to the complex nature of prognostic model developments, the prognostic model toolkit will contain a large number of supporting functions. It is important to note that with so many automated decisions, the user/operator is still in control of the process. Each stage of the program will allow the user to override options and set a desired path. This allows for the benefits of an automated process without the disadvantages of a “black box” program. Not only is the user able to specify individual options, but also certain supplementary options can be turned off for simple and fast model development requirements.

9. Novelty of Methods and Comparison to Current State-of-the-Art Efforts

To discuss the novelty of ideas presented in earlier sections, it is important to identify the focus of these new approaches. While significant attention is given to internal functions, the novelty of these ideas is in the process mechanism, specifically the automation of prognostic data extraction and model development. Never before has predictive maintenance been standardized with an automated model development process that begins with prognostic data extraction and ends with RUL predictions on a commercial scale. By automating these processes, predictive maintenance efforts may be improved through increased response time, reduced human interaction (decreased error), and decreased variability in prognostic model development. In many applications, predictive maintenance efforts will not be effective enough for implementation until these improvements have been satisfied. This makes the novelty of presented ideas very attractive. The utilization of the coupling algorithm and prognostics toolkit may allow for industry-wide implementation of prognostics and advanced diagnostics on a large scale. To companies, the novelty of these programs lies in their ability to reduce unexpected downtime for maintenance as well as improve scheduling of parts orders. The possibility for increased safety is another appealing outcome that may be achieved through the implementation of these programs. All of these efforts will increase the availability of information to the operator/user.

The novelty of automation with application to nuclear power is rather significant. While human operators will always be present in a nuclear power plant, industry is pushing towards a higher level of plant automation. Figure 8 shows the levels of automation from manual control to full autonomy.

These levels show the change in influence a machine or human has on an activity. The current level of automation for nuclear power plants is around level 5. Coupling of the CMMS and plant computer can push reactor operations closer to full autonomy. This not only simplifies the data extraction process, but also allows for improvements to reliability, safety, and most importantly decision-making for continued operation.

![Figure 1. Adaptation of Sheridan’s 10 levels of automation (Bradshaw 2011).](image)

It is important to reiterate that the methods discussed in this paper are not novel because they supersede or replace the current state-of-the-art prognostic methods. The automation of data extraction is necessary to increase response time for operators, but will rely on prognostic indicators and functions that are currently used during manual extraction. The automation of model development utilizes existing methods with individual runtime envelopes to facilitate autonomous control. This allows for new automation methods to retain existing validation of prognostic functions as a baseline, and provide increased confidence to the end-user.

10. Conclusions

With the presence of large pre-existing databanks for storing nuclear power plant process data and maintenance records, the coupling of the plant computer to a computerized maintenance management system could allow for the extraction of useful diagnostic, prognostic, and reliability information. This data can be passed to modified existing state-of-the-art prognostic functions and tools in order to autonomously create and update prognostic models for individual assets and components. With additional research applied to the methods described, effective application of predictive maintenance in commercial applications may be possible. The automation of model development and
predictions may lead to increased response time and decreased variability in the model development process. These potential benefits of these methods are immeasurable across the multitude of possible applications.

ACKNOWLEDGEMENT

The authors would like to acknowledge the Lloyds Register Foundation and the International Joint Research Center for the Safety of Nuclear Energy for partial funding of this research. Lloyd’s Register Foundation helps to protect life and property by supporting engineering-related education, public engagement, and the application of research.

REFERENCES


Skvarenina, T., & DeWitt, W., Chapter 7 Three-Phase Induction Motors, Electrical Power and Controls, 2004, Prentice Hall


Sparse Coding-Based Failure Prediction for Prudent Operation of LED Manufacturing Equipment

Jia-Min Ren\textsuperscript{1}, Chuang-Hua Chueh\textsuperscript{1}, and H. T. Kung\textsuperscript{2}

\textsuperscript{1}Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan
jmren@itri.org.tw
chchueh@itri.org.tw

\textsuperscript{2}Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge MA, USA
kung@harvard.edu

ABSTRACT

A sudden failure of a critical component in light-emitting diode (LED) manufacturing equipment would result in unscheduled downtime, leading to a possibly significant loss in productivity for the manufacturer. It is therefore important to be able to predict upcoming failures. A major obstacle to failure prediction is the limited amount of equipment lifecycle data available for training, as equipment failure is not expected to be frequent. This calls for machine learning techniques capable of making accurate failure predictions with limited training data. This paper describes such a method based on sparse coding. We demonstrate the prediction performance of the method on a real-world dataset from LED manufacturing equipment. We show that sparse coding can draw out salient features associated with failure cases, and can thus produce accurate failure predictions. We also analyze how sparse coding-based failure prediction can lead to significant efficiency improvements in equipment operation.

1. INTRODUCTION

Reducing costs and increasing productivity are crucial concerns in the competitive manufacturing industry. Many manufacturers are seeking to implement intelligent manufacturing processes including the use of automated data analysis techniques (Scoville, 2011) which can allow for cost- and time-saving predictive maintenance (Rothe, 2008). This paper addresses approaches to optimizing predictive maintenance methods for light-emitting diode (LED) manufacturing equipment.

Modern LEDs are multilayered structures of chemical materials in which the thickness and composition of the various layers determine the color and brightness of the emitted light and device energy efficiency. The layers are deposited sequentially through the metal organic chemical vapor deposition (MOCVD) process, a critical determinant process in LED performance. The crystalline structure of each new layer is epitaxially aligned with that of the underlying layer. This complex process is affected by the conditions of a multitude of components, such as pumps, heaters, mass flow controllers, and particle filters\footnote{Note that by a \textit{particle filter} we mean in this paper a \textit{physical} filter in MOCVD equipment. This is not to be confused with particle filtering methods in the diagnostics and prognostics prediction literature (see, e.g., Orchar & Vachtsevanos, 2007).}.

Unexpected component failure can reduce LED production yields, and finding and repairing the source of the failure can take engineers up to 5 days. For example, the failure of a particle filter will cause the pump to shut down, resulting in all the raw materials consumed in that run to be wasted. Here, we focus on developing a failure prediction algorithm for the particle filter to ensure continuous high-performance operation in the MOCVD process.

Learning features associated with failure cases plays a critical role in a data-driven prediction approach. The goal is to come up a compact yet discriminative feature representation, in which samples related to failure cases can be accurately expressed and easily differentiated from others. Many feature learning methods have been discussed in the literature; see, e.g., Huang & Aviyente, 2006. These include principal component analysis (PCA), linear discriminant analysis (LDA) and sparse coding (SC). The SC approach used in this paper has emerged as one of the most popular feature learning methods in recent years, in areas such as computer vision (Wright et al., 2010). It computes a sparse representation of input data in terms of a linear combination of atoms in an overcomplete dictionary (more details are given in Section 4.1). Compared to methods based on...
orthonormal transformations, SC has been shown to offer superior performance in a variety of applications including face recognition (Wright et al., 2009), emotion recognition (Chen et al., 2015), and wireless link prediction (Tarsa et al., 2015). Therefore, the proposed failure predictor is based on SC.

In evaluating our SC-based approach, we use real-world sensor data from LED MOCVD equipment. We found that the SC-based failure prediction method can improve F-measure about 39% over a conventional PCA-based approach. In terms of annual uptime of MOCVD equipment operation, the SC-based failure prediction method can increase it by about 600 hours over a traditional preventative maintenance policy. To the best of our knowledge, this work is the first application of SC to the prediction of component failure in MOCVD equipment.

The remainder of this paper is organized as follows: Section 2 provides an overview of failure prediction problem and prior work. Section 3 describes the dataset used in our experiments. Section 4 explains the basic concept of sparse coding and the proposed failure prediction pipeline for MOCVD equipment. In Section 5, we show the experimental results of PCA-based and SC-based failure prediction methods. Cost-benefit analysis for different maintenance and prediction policies is discussed in Section 6. Conclusions are given in Section 7.

2. Failure Prediction Problem and Prior Work

The goal of failure prediction is to predict an upcoming component failure in MOCVD equipment, and raise an alarm or a maintenance advice to the manufacturer who can then intervene to prevent unscheduled downtime. In this paper, we focus on the next-run failure prediction. In other words, following each run, the system provides a prediction of whether the particle filter will fail in the next run. Here a run denotes an execution of a fabrication task on MOCVD equipment. The next-run failure prediction can be simply considered as a decision problem of predicting a yes or no outcome. We thus address it as a binary classification task.

Failure prediction methods can be roughly categorized into model-based and data-driven approaches (Lee et al., 2014). Model-based approaches have been traditionally used to understand failure mode progression associated with equipment components. In training physics-model parameters, such as those in Kalman or particle filter methods (Orchar and Vachtsevanos, 2007), model-based methods usually require relatively smaller amounts of data. However, to achieve acceptable performance in prediction, building an appropriate physical model would require detailed mechanistic knowledge and could be time-consuming. In contrast, data-driven approaches build a machine learned model based on observed sensor data from equipment without relying strongly on domain knowledge.

![Figure 1. Data illustration of a particle filter replacement cycle. It is an example of the 22 cycles considered.](image)

(a) dp.filter raw data over runs in a replacement cycle and (b) dp.filter maximum values over runs in the same cycle. Here a run denotes an execution of a fabrication task on MOCVD equipment. (Runs may vary somewhat in their execution time.) The maximum value of dp.filter raw data is used to represent the feature of each run. Thus a replacement cycle can be represented as a sequence of these maximum values.

More details about dp.filter are given in Section 3.

3. Data Description

MOCVD processes produce powders and particulates which can cause unexpected and significant damage to costly pump equipment. A particle filter is therefore needed to avoid contamination of the pump. Among the sensors in MOCVD equipment, dp.filter is the most critical sensor in monitoring the particle filter status in the level of dust being accumulated there. This sensor measures the difference of pressure between the reactor and the pump. As fabrication tasks are carried out on MOCVD equipment, more powders and particulates are stacked on the filter, so dp.filter values usually increase gradually. Figure 1(a) shows an example of the degradation of the dp.filter signal in a replacement cycle. This cycle consists of multiple runs, with vertical lines denoting the boundaries between runs. Practically, the particle filter will be replaced when the maximum value of dp.filter over a run exceeds a configured threshold (e.g., 30). Accordingly, we extracted the maximum value for each run in our experiments. This means that each cycle was represented as a sequence of dp.filter maximum values as shown in Figure 1(b).

Note that the dips in Figure 1(b) were caused by executing “clean runs” on MOCVD equipment. Such clean runs are sometimes needed to clean up residual gases in the reactor mentioned earlier. The amount of gases used in a clean run is much less than those associated with a regular fabrication task execution, leading to dips in a sequence of the subsequent dp.filter maximum values.
In total, 22 replacement cycles were collected. In our experiments, the first 15 cycles and the remaining 7 cycles were respectively used to build the training and the test data. To predict whether the particle filter will fail in the next run, we labeled the previous run before the one whose dp.filter maximum value exceeded 30 as a faulty run. To incorporate historical information for failure prediction, we used a sliding window of size 10 runs (with adjacent windows overlapped by one run) to create the feature vector for each run. In other words, we used 10 dp.filter maximum values from the previous nine runs and the present run to represent the feature vector for the present run. Thus each created sample has a dimensionality of 10. Since each replacement cycle consists of a different number of runs (varying from 23 to 104 runs), different numbers of normal samples are thus collected for the training and the test data. Specifically, the training data consists of 387 normal and 15 faulty samples, and the test data is composed of 319 normal and 7 faulty samples.

4. SPARSE CODING BASED FAILURE PREDICTION

4.1 Basic Concept of Sparse Coding

Given \( N \) data samples \( x_i \in \mathbb{R}^{M \times 1} \), we first learn a dictionary \( D \) using the following mathematical optimization:

\[
\min_{D, \alpha \in \mathbb{R}^{K \times 1}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \| x_i - D\alpha_i \|_2^2 + \lambda \| \alpha_i \|_1
\]

\( C \hat{=} \{ D \in \mathbb{R}^{M \times K} \text{ s.t. } \| d_j \|_2^2 = 1, \forall j = 1, ..., K \} \) (1)

for certain \( \lambda > 0 \), where \( \alpha_i \) is the sparse code of \( x_i \), and \( D \) is the dictionary composed of \( K \) columns and \( d_j \) is the \( j \)th column (i.e., atom) in \( D \). We split samples into smaller patches of length four; therefore, we have \( M = 4 \) in our experiments.

Given the learned dictionary \( D \), we consider the following LASSO (Least Absolute Shrinkage and Selection Operator) formulated optimization problem:

\[
\min_{\alpha \in \mathbb{R}^{K \times 1}} \| x_i - D\alpha \|_2^2 + \lambda \| \alpha \|_1
\]

(2)

For a given data point \( x_i \), by solving the LASSO optimization using methods such as least angle regression (LARS) (Efron et al., 2004) and interior-point (Koh et al., 2007) we find the sparse code \( \alpha_i \) for \( x_i \).

4.2 LEARNING AND TRAINING PIPELINES

Figure 2 shows the learning and training pipelines for particle filter failure prediction via sparse coding. We use the following steps for dictionary learning and SVM classifier training.

1. **Patch generating.** To capture local variation within each sample, we split each sample into overlapping patches \( \{ x_1, ..., x_{10-p+1} \} \), where the patch size is \( p \) and the overlapping is one. In our experiments, \( p \) is typically set to four.

2. **Patch selecting and dictionary learning.** Note that for next-run failure prediction, only the run before the last run in a replacement cycle is labeled as faulty. All other runs are labeled as normal. Thus there are far more normal runs than faulty runs. To deal with this imbalance, two dictionaries \( D_N \) and \( D_F \) are then respectively learned on normal and faulty samples using (1). (Note that if all samples were used to learn a single dictionary, the dictionary would be dominated by normal samples.)

Specifically, to discriminatively learn these two dictionaries, only the patches whose values are all smaller than a threshold \( T_{N} \) are used to learn \( D_N \). On the other hand, if one value within the patches exceeds \( T_{N} \), these patches will be used to learn \( D_F \). Other patches that do not satisfy either of these two conditions are discarded.

3. **Dictionary concatenating.** A simple concatenation forms the final dictionary \( D = [D_N \vert D_F] \).

The following steps are used in SVM classifier training for failure prediction based on sparse code.

1. **Patch generating.** We split each sample into overlapping patches \( \{ x_1, ..., x_{10-p+1} \} \).

2. **Sparse coding.** Given the learned dictionary \( D \), we use (2) to compute sparse code \( \alpha_j \) of each patch \( x_j \). In total, \( 10-p+1 \) sparse codes are thereby obtained for each sample.

3. **Max pooling.** To incorporate local variation of patches to reflect global features of each sample, we perform max pooling over these \( 10-p+1 \) sparse codes to obtain a pooled sparse code \( z \) such that the \( k \)th element in \( z \), \( z_k = \max(\alpha_{1, k}, ..., \alpha_{p, k}, ..., \alpha_{10-p+1, k}) \) where \( \alpha_{j, k} \) is the \( k \)th element from \( \alpha_j \). Therefore, each sample is encoded as a pooled sparse code. The effectiveness of using pooled sparse code in classification as opposed to \( \alpha_j \) is well known (see, e.g., Chen et al., 2015, and Tarsa et al., 2015).

---

Figure 2. Dictionary learning and SVM classifier training pipelines.

---

659
4. Classifier training. We use pooled sparse codes as features to train a linear SVM classifier for failure prediction.

4.3 Failure Prediction on Test Samples
For a test sample, we first divide it into overlapping patches. Then, we compute sparse codes for each patch and perform max pooling on these sparse codes to obtain a pooled feature vector. Finally, the pooled feature vector is used as an input vector for the pre-trained SVM to obtain the prediction result.

5. Experimental Setup and Results

5.1 Baseline: PCA+SVM
The proposed method was compared against a PCA-based baseline method. In this baseline method, PCA was used to project data onto a lower dimensional space spanned by a relatively small number of dominant eigenvectors of the covariance matrix of the training data. Then, a linear SVM was used as the predictor.

5.2 Experimental Setup and Evaluation Metrics
To provide a fair comparison, we used the same setting of the penalty parameter in a linear SVM (Chang & Lin, 2011) for both PCA- and SC-based approaches. The maximal number of principal components (PCs) used in the experiments was six, which can explain over 95% of the training data variance. Different numbers of atoms in dictionary $D_N$ and dictionary $D_F$ were set for comparison. Since a particle filter will be replaced when the maximum value of dp.filter over a run exceeds 30, we set $T_{N}$ and $T_{F}$ to 10 and 20 respectively. As mentioned earlier, the patch size $p$ was empirically set to four. Sparse coding is set to use about three non-zero coefficients in our experiments.

Four standard metrics (Salfitri et al., 2010)—true positive rate (TPR), false positive rate (FPR), F-measure and the area under the receiver operating characteristic (ROC) curve (AUC)—were used to compare the performance of different methods. Note that a positive sample means a faulty sample in our experiments.

5.3 Patch Selection for Dictionary Learning
To learn the two dictionaries $D_N$ and $D_F$ that can respectively represent normal and faulty regularities, we generated two patch sets. For this, two patch selection schemes were compared. As shown in Figure 3(a), two patch sets were separately created by considering whether the patch belongs to the last sliding window in each cycle. (Note that we can create seven patches from a window consisting of 10 runs.) Clearly, some patches overlapped, making it difficult to differentiate between atoms in the dictionaries $D_N$ and $D_F$. In contrast, we defined two thresholds to select non-overlapped patches. As shown in Figure 3(b), only the patches satisfying the constraints described in Section 4.2 were used for dictionary learning. The other patches were discarded.

Figure 4 shows the sparse codes of a patch at a faulty run using dictionaries learned by different patch selection schemes. The dictionary $D_N$ and the dictionary $D_F$ are respectively indexed as 1–15 and 16–18. Obviously, when using the patch selection scheme 1, the overlapping of patches makes it difficult to train these two dictionaries discriminatively. Thus the patch at a faulty run can be incorrectly coded by atoms in $D_N$ (e.g., indices of 6, 10 and 12, as shown in Figure 4(a)). In contrast, when we separated patches into two sets without overlap, different atoms can be learned in $D_N$ and $D_F$. Accordingly, the same patch can be almost coded by only the atom in $D_F$ (e.g., the index of 18, as shown in Figure 4(b)). In summary, the patch selection scheme 2, as used in our pipeline, creates patches that can be used to train $D_N$ and $D_F$ more discriminately.

![Figure 3](image_url) Two patch selection schemes. (For simplicity of plot, black points denote other overlapping patches used to train $D_N$)

![Figure 4](image_url) Sparse codes of a patch at a faulty run using dictionaries learned by patch selection (a) scheme 1 and (b) scheme 2.
Table 1. Performance comparisons. For PCA+SVM, PCs=2 (in parenthesis) denotes that data is projected onto the first two PCs (a similar definition applies to PCs=3 and PCs=6). For SC+SVM, 10/3 (in parenthesis) means we have 10 atoms in $D_n$ and 3 atoms in $D_t$ (a similar definition applies to 15/3).

<table>
<thead>
<tr>
<th>Column Index</th>
<th>Method</th>
<th>Metric</th>
<th>TPR (%)</th>
<th>FPR (%)</th>
<th>F-measure</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCA+SV</td>
<td>PCA+SV (PCs=2)</td>
<td>85.71</td>
<td>16.93</td>
<td>0.179</td>
<td>0.916</td>
</tr>
<tr>
<td>2</td>
<td>PCA+SV</td>
<td>PCA+SV (PCs=3)</td>
<td>71.43</td>
<td>12.23</td>
<td>0.196</td>
<td>0.911</td>
</tr>
<tr>
<td>3</td>
<td>PCA+SV</td>
<td>PCA+SV (PCs=6)</td>
<td>71.43</td>
<td>5.64</td>
<td>0.333</td>
<td>0.847</td>
</tr>
<tr>
<td>4</td>
<td>SC+SVM</td>
<td>SC+SVM (10/3)</td>
<td>85.71</td>
<td>9.4</td>
<td>0.279</td>
<td>0.942</td>
</tr>
<tr>
<td>5</td>
<td>SC+SVM</td>
<td>SC+SVM (15/3)</td>
<td>100</td>
<td>13.17</td>
<td>0.25</td>
<td>0.983</td>
</tr>
</tbody>
</table>

5.4 Prediction Results

Table 1 compares the performance of the baseline method (PCA+SVM) and the proposed method (SC+SVM). For the PCA+SVM method (columns 1, 2 and 3), when more PCs were used, we obtained higher F-measure values. This means that reserving more PCs is useful for failure prediction. Under TPR equals to 85.71% (columns 1 and 4), the proposed SC+SVM method achieves a lower FPR (9.4%) than that of the PCA+SVM method (16.93%). In other words, the proposed method raised fewer false alarms than the PCA+SVM method. The proposed method (column 5) also achieves the best prediction performance in terms of AUC.

Figure 5 shows the receiver operating characteristic (ROC) curves of these two methods under the best AUC values (columns 1 and 5 in Table 1). From this figure, we observed that the PCA+SVM method raises more false alarms.

![Figure 5. ROC curves of the baseline method (PCA+SVM) and the proposed method (SC+SVM).](image)

In addition, when using a non-linear, radius basis function (RBF) SVM as a predictor rather than a linear SVM, the highest AUC value of PCA-based method achieves 0.925 (under PCs=2) and the highest AUC value of SC-based method achieves 0.965 (under 15 atoms in $D_n$ and 3 atoms in $D_t$). This experiment shows again that the use of SC-based features outperforms that of traditional PCA features when a non-linear SVM is used as a predictor.

Furthermore, to assess robustness of performance against various partitions of the data set into training and test cycles, we also evaluated the performance of two other random partitions. Similar results as mentioned above were found.

6 Cost-Benefit Analysis of Different Replacement Policies for MOCVD Equipment

In this section, we provide cost-benefit analysis of the proposed SC-based prediction method when compared with some other methods for MOCVD equipment. Given the difficulty of quantizing detailed factors of costs such as hardware and software design, engineering qualification and certification (Saxena et al., 2010), we only consider the annual uptime of MOCVD equipment operation. We use the following notations to facilitate the discussion.

$UT$: average uptime per cycle,

$DT$: average downtime for a replacement, and

$H$: the probability that the maintenance time provided by a certain replacement policy is before an actual failure.

Practically, $UT$ is calculated based on the maintenance time under a given replacement policy on the test cycles. $DT$ is calculated as

$$H * 1.5 + (1 - H) * 108$$

where 1.5 and 108 are average downtimes (in hours) for a scheduled and an unscheduled replacement, respectively. Note that in contrast, the execution time of a run is about 8 hours. Here these average downtimes and the run’s execution time were obtained from engineers who maintain MOCVD equipment.

The expected uptime in a year under a replacement policy is calculated by multiplying the number of operation units, each of which is the time duration for a pair $UT$ and $DT$, in a year and the average uptime per cycle:

$$\text{Expected uptime in a year} = \frac{\text{total hours in a year}}{UT + DT}$$

In addition to the SC- or PCA-based predictive maintenance policy, we consider two conventional replacement policies: (1) run-to-failure replacement policy, under which the particle filter will be used until it fails (i.e., $H = 0$), and (2) preventive maintenance policy, under which the particle filter will be replaced once the number of executed runs exceeds the average number of runs in training cycles.

Figure 6 compares uptime against the FPR under different replacement policies for the test data. The X-axis is the FPR of the PCA+SVM and SC+SVM methods. The Y-axis is the annual uptime of the MOCVD equipment. To facilitate the discussion, the total number of particle filters for which no alarm was raised under a replacement policy before they
failed is denoted as \#misses. For example, under the run-to-failure replacement policy, \#misses is 7 since these filters are used until they fail; under the preventive maintenance policy, only one filter is not replaced before it fails, so \#misses is 1. To study cost-benefit effects of \#misses for the PCA+SVM and SC+SVM methods, we consider the three FPR subintervals shown in Figure 6. In these FPR regions various methods exhibit their relative strengths.

Interval1 (\#misses is larger than 1 for both PCA+SVM and SC+SVM). When we allow a very low FPR, it is unlikely that an alarm will be raised. Then the performance of both the baseline method and the proposed method is worse than that of preventive maintenance in terms of uptime (as shown in the lower-right zoomed-in panel of Figure 6).

Interval2 (\#misses is 1 for SC+SVM, and \#misses is either 2 or 3 for PCA+SCM). We compare the proposed method with preventive maintenance under the same \#misses. We can see that the proposed SC-based method outperforms the preventive maintenance strategy with an increased uptime of 300+ hours (as shown in the upper-right zoomed-in panel of Figure 6).

Interval3 (\#misses is 0 for SC+SVM, and \#misses is either 1 or 2 for PCA+SVM). The proposed SC-based method successfully raises alarms before any particle filter fails, and therefore achieves the best uptime among all methods.

In summary, the proposed SC-based method outperforms the other replacement policies in Interval2 and Interval3. Particularly, the proposed SC+SVM method outperforms the preventive maintenance policy in annual uptime by 600+ hours under FPR equal to 5% (as shown in the upper-right zoomed-in panel of Figure 6). This suggests that under the given data set, when the proposed SC-based prediction method is used, the target FPR should be set at 5%. Note that if FPR is set too high (e.g., 50%), the proposed SC-based failure predictor would incur many false alarms, leading to a shortened uptime.

7 Conclusion and Future Work

In this paper, we propose a sparse coding-based failure prediction method for the particle filter in MOCVD equipment. Using a real-world dataset, our proposed SC-based method raises fewer false alarms than a PCA-based baseline method under the same TPR. Compared with the preventative maintenance strategy, the proposed SC-based method increases the annual uptime of MOCVD equipment by 600+ hours. To the best of our knowledge, this work is the first application of sparse coding to the prediction of component failure in MOCVD equipment, with performance demonstrated using a real-world dataset.

The paper focuses on classifiers rather than their ensembles. It is generally true that ensemble methods are often more accurate than their individual classifiers provided that the latter are accurate and diverse (Dietterich, 2000). As a future work, we plan to study ensemble methods based on the proposed SC-based classifiers of this paper.

Acknowledgement

This work is partially supported by the project of Big Data Technologies and Applications, Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan. H. T. Kung’s research at Harvard for this paper was supported in part by gifts from the Intel Corporation and in part by the Naval Postgraduate School Agreement No. N00244-15-0050 awarded by the Naval Supply Systems Command. The views expressed in this paper do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

References


**BIOGRAPHIES**

Jia-Min Ren received his Ph.D. from the Computer Science Department, National Tsing Hua University, Hsinchu, Taiwan. He currently works at the Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan. His research interests include machine learning, semantic analysis of musical signals, music information retrieval, and analysis of manufacturing data.

Chuang-Hua Chueh received his Ph.D. from the Computer Science Department, National Cheng Kung University, Tainan, Taiwan. He currently works at the Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan. His research interests include machine learning, pattern recognition and speech recognition.

H. T. Kung is William H. Gates Professor of Computer Science and Electrical Engineering at Harvard University. He is interested in computing, communications and sensing, with a current focus on machine learning, compressive sampling and the Internet of Things. He has been a consultant to major companies in these areas. Prior to joining Harvard in 1992, he taught at Carnegie Mellon University for 19 years after receiving his Ph.D. there. Professor Kung is well-known for his pioneering work on systolic arrays in parallel processing and optimistic concurrency control in database systems. His academic honors include membership in the National Academy of Engineering in the US and the Academia Sinica in Taiwan.
Testing Diagnostics Components Supervising Functional Safety Requirements

Mihai Nica¹, Ingo Pill², and Franz Wotawa³

authors are listed in alphabetical order

¹ AVL GmbH, 8020 Graz, Austria
mihai.nica@avl.com

²,³ Institute for Software Technology, TU Graz, Inffeldgasse 16b, 8010 Graz, Austria
{ipill,wotawa}@ist.tugraz.at

ABSTRACT

For safety-critical applications, safety diagnostics components are an attractive safeguard for meeting some specified safety requirements under operation. Like a monitor, such a software artifact shall supervise a system under operation, and furthermore, if needed, it overrides the system’s control software in order to maintain safety. In this paper we contribute to testing such a component, suggesting an approach that draws on fault injection and, in order to enhance deployability, accommodates also needs in respect of business issues like intellectual property disclosure and resource efficiency. The required testing oracle we directly obtain from the defined and formalized functional safety requirements, for the purpose of assessing that the safety diagnostic component indeed maintains safety also under faulty conditions.

1. INTRODUCTION

Scientific evolution has been allowing us to continuously step ahead and develop solutions tackling problems that priorly seemed intractable. Consequently, the technology to assess and manage risks involved with our designs and their possibly faulty behavior has been facing constantly rising demands. For instance, in respect of effectiveness in highly dynamic environments, efficiency at handling complex designs, and robustness in order to name just a few challenges. For many a project, we seek to proactively address related safety concerns, as is demanded by public regulations via standards like IEC 61508¹ and its adaptation ISO 26262² (Automotive Safety Integrity Level (ASIL) as used in the automotive industry) on one side, and customers on the other one. For instance, in electronic design automation industry, logics like the Linear Temporal Logic LTL (Pnueli, 1977) or the Property Specification Language PSL (Eisner & Fisman, 2006) have been used to describe desired system requirements (properties) to be used for automated verification like model-checking (Clarke, Grumberg, & Peled, 1999). Recent work assists designers in formalizing these requirements (Bloom, Cavada, Pill, Roveri, & Tchaltsev, 2007). Also an AI-based diagnosis approach for diagnosing faults in such formalized requirements has been proposed (Pill & Quaritsch, 2013).

For the design and operation of nuclear power plants, space applications, or avionics, the importance of designing and satisfying safety requirements is obvious to each and everyone of us. However, also for more mundane systems like private cars, such concerns and corresponding requirements are, and have been, of utmost importance in their design and operation. Imagine, for instance, an automated parking brake. Certainly we would like it to be released only on a driver’s command (manual release, tipping the gas pedal, ...). The assistance systems available in today’s car would take many variables into account for deciding about this (e.g. driver present, doors closed, seatbelts fastened, ...), which makes it important to clearly specify safety targets like the one above that have to be met under all circumstances. Such specific requirements to be verified during the design stages and monitored under operation shall help that safety (or related functionality aspects) are ensured, no matter the system’s complexity or unexpected issues in “live” real-world situations.

Diagnostic reasoning (de Kleer & Williams, 1987; Reiter, 1987) as explored by the Artificial Intelligence community (i.e. the DX community) is a powerful asset for tackling related issues (Weber & Wotawa, 2010). That is, finding a fault and identifying its root cause(s) certainly is an important and essential step towards keeping a system operating as optimal

¹e.g., http://www.iec.ch/functionalsafety/
²e.g., http://www.iso.org/iso/catalogue_detail?csnumber=43464

Mihai Nica et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
as possible. Complementing the theoretical and application-specific research on solutions to the diagnosis problem itself, in order to enhance deployability of corresponding solutions, we also have to address the question of how to accommodate diagnostic reasoning components in the system design process.

For our current work, we consider this very question and focus on the specific issue of testing a diagnostic component’s effectiveness. That is, we assume the scenario of a safety diagnostic component \( SDC \) supervising some control software \( CS \) (like a controller for the automated parking brake of above) and that can even override it in case of the software violating specified functional safety requirements. The task we face is testing that, indeed, \( SDC \) maintains safety as defined by the requirements, even if the control software would show some potentially hazardous behavior.

The testing concept that we propose easily integrates into an overall system’s development process. Aiming to minimize needed resources, we reuse the input parts of the test cases available for the supervised component \( CS \), since they were already designed to exercise the system thoroughly (also covering a wide selection of interesting scenarios). Since our test purpose is to assess whether \( SDC \) indeed lives up to our expectations of maintaining compliance to some safety requirements \( REQ \), we use the concept of mutation testing in order to inject faults and simulate control software faults to be considered and dealt with by the diagnostic engine. The product of the reused input sequences and designed fault scenarios (mutants) shall be used to evaluate \( SDC \)’s capabilities. In this context, we judge safety conformance with a test oracle directly derived from the safety requirements \( REQ \) that are specified (or need to be translated) in a concise logic in order to support automated reasoning.

Before discussing our concept in Section 4 and related work in Section 3, we introduce a motivating example in Section 2 where we elaborate also on our problem description. A discussion of our approach, as well as corresponding conclusions and directions for future work are provided in Section 5.

2. A Motivating Scenario and Task Analysis

The basic task that the considered diagnostic component \( SDC \) has to tackle is to ensure that some defined safety requirements are met even if the control software \( CS \) would show some hazardous behavior. The latter could result either from some fault(s) in the software design like forgotten corner cases, faults in the implementation, some memory corruption (e.g. due to (Kim et al., 2014)) or other hardware faults, compiler errors, and many other issues. Certainly software designers add assertions and other sanity checks to their control software, possibly complemented with some elaborate code that aims at keeping the control software in a safe state (e.g. fault tolerant control concepts).

Now let us assume that we do have also some “external” \( SDC \) that sort of acts as last defense. As is illustrated in Figure 1, it supervises the control software \( CS \) in that it collects all of \( CS \)’s relevant inputs \( I \), its outputs \( O \), and then decides whether the I/O scenario complies with defined safety goals \( REQ \), using diagnostic reasoning to investigate encountered issues. If necessary, it can override \( CS \) by altering the output from \( O \) to \( O' \) in order to maintain safety. The latter (depicted by an abstract selection function \( SELFCT \) in Fig. 1), obviously, can be implemented in many ways. One option would be a dedicated component on the main bus, another one the use of individual priority signal lines to the final actuators controlled by \( CS \) (like a gear box or a clutch). In practice, the choice of implementation might also be made individually for each isolated signal or component controlled by \( CS \).

**Definition 1** A Safety Diagnostics Component (SDC) (see also Figure 1) observes the inputs \( I \) and outputs \( O \) of the supervised system \( CS \), monitors for given requirements \( REQ \) the property of \( I \cup O \cup REQ \) being consistent (satisfiable), and for a violation designs a new output \( O' \) delivered via a given component \( SELFCT \) such that \( I \cup O' \cup REQ \) is consistent (satisfiable)

Let us discuss this concept in the context of a simple WUMPUS-like scenario. The main character in this sce-
nario is a small robot that explores a grid-based world with
walls. It may rotate 90-degree-wise in both directions, shift
gears to forward, backward, or neutral, can move step-wise
by one grid-element (see Figure 2 for an example world),
and the drive unit can be shut off. The corresponding com-
mands at control software level would be $RL / RR$ to rotate
left or right, $SGF / SGB / SGN$ to shift the gearbox, $M$ to
move, and $CP$ for cutting the drive unit's power. Let us
assume that the robot can sense its position, and that the
current task is to move from $A1$ to $C3$ in the world de-
picted in Figure 2. Assuming the robot is oriented towards
the east, a possible command sequence to achieve this is
$SGF, M, M, M, SGF, RR, SGF, M, SGN, RR, SGF, M, M,$
$SGN, RL, SGF, M (CP)$. Now let us assume that for some
reason, like a bit flip or memory corruption, for the first com-
mand $SGF$, the gearbox changes to backward instead. De-
pending on where the exact fault occurred, besides assertions
that check position, orientation, and other available internal
knowledge, the robot might not be aware of the fact that when
set in motion it will not proceed to $A2$, but hit the western
wall in $A1$. Thus, when the robot executes the next activity
$M$, without some interference it would actually hit a wall.

However, a safety requirement any designer would definitely
derive for this application scenario is that the robot should
never hit a wall. The safety diagnostics component imple-
menting $REQ$ then most likely would monitor continuously
the robot’s proximity to obstacles, and, e.g., in two violation
levels could first shift the gearbox to neutral and if this does
not suffice (gearbox malfunction) could even cut the power
from the drive. An elaborate reasoning engine might try to de-
terminate (diagnose) the detailed reason for the violation (e.g.
a second moving robot vs. a wall) in order to derive the best
course (like triggering an emergency protocol in $CS$ for flee-
ing from another swarm robot that seems out of control) and
degrade task performance in a sensible way.

Even the simple WUPMUS-like scenario shows why having
an external component supervising the main software could
be of advantage. For instance,

- it can consider information at a different level of abstrac-
tion or at different frequencies.
- while the main software’s complexity could ask for state-
of-the-art components, the diagnostic engine could run
on special, e.g., radiation-hardened, hardware that has
been showing reliability in the designated environment in
the past (concerning special demands in respect of heat,
humidity, radiation, electric/magnetic fields,...).
- for $SDC$’s development, conceptional details from the
main software (data handling, operation frequencies,
real-time issues) represent no restrictions, so that one can
concentrate on a requirements-based development (fo-
cusing on $REQ$) vs. a system-design oriented adaption
of relevant functionality in $CS$.
- flexibility is offered also in the direction of collecting $I$
and $O$. That is, we could possibly grab signal values
from the software itself, or also at some location in the
hardware lines so that a malfunction in the lines would
also be covered.

Obviously, an $SDC$ offers many advantages, best combined
with afore-mentioned functionalities in the control software
itself. For actual deployment, confidence in $SDC$’s capabil-
ities is however of utmost importance. The specific chal-
ge we consider in this paper is related to this very issue,
in that we propose a testing concept for the purpose of as-
suming whether $SDC$ actually lives up to the task of avoiding
violations of specified safety requirements $REQ$, even if the
control software would malfunction. More formally, we aim
to test whether $SDC$ conforms to $REQ$ as of Def. 2.

**Definition 2** Let $SDC$ be a safety diagnostics component
as of Definition 1 which supervises some system $CS$. Then
for some given requirements $REQ$ and possible fault modes
$FAULTS$ of $CS$, $SDC$ conforms to $REQ$ if and only if for all
inputs $I$, we have that $REQ \cup I \cup O'$ is consistent (is satis-
fiable).

Obviously, a formal proof of such a conformance would be
optimal, but is not a likely scenario since (a) the system
is most likely too complex to apply techniques like model-
checking (Clarke et al., 1999), and (b) we cannot assume the
system to be a white box for us. Furthermore, considering a
model only, would not support us in finally checking the very
implementation “live” on actual hardware and in the actual
environment (relevant e.g. in respect of radiation or other
straining effects that could possibly affect timings, memory
contents, and other electronics). Thus, we propose to im-
plement a testing concept for evaluating the desired confor-
mane. This concept then can be implemented at various ab-
straction levels (from specification level to the final product).

Aside academic questions like those regarding test design and
coverage, for deployment in industry, e.g., for an automotive
application, we have to consider also issues and restrictions
originating from business issues to be faced in the develop-
ment process. For instance, the availability of detailed system
information (white vs. grey vs. black box) certainly is of an
issue for such an automotive scenario. Thus it is also unlikely
that we can inject faults for purposefully exercising $SDC$ at
will (see Sec. 3 for a discussion of mutation testing) or have
access to all the internal models for imagining and deriving
test cases. However, we can assume that a function enabling
specific faults in the software can be made available, since
this suits also testing the individual components themselves.
Also the safety requirements to be enforced should be avail-
able (although not necessarily in a formal syntax), since they
are essential for the design of the system.
To this end, in Section 4 we outline an approach for testing SDC in the exemplary context of an automotive application, which takes also business-related issues into account and addresses the conformance problem as of Definition 2. The underlying concept is to execute the system under test (SUT) using the test inputs originally designed for testing CS (or the overall system). We inject faults \( f \in \text{FAULTS} \) into CS using provided fault injection functions, and then record the resulting outputs \( O' \). A test oracle derived from the given (safety) requirements \( \text{REQ} \) then classifies the input and recorded output in respect of conformance to \( \text{REQ} \) as of Definition 2.

Please note that our testing concept is independent from the concepts used for developing and implementing the SDC. Depending on the actual system and requirements (and their structure), one might be able to implement individual monitors for the requirements that then trigger some signal that per construction results in a safe system state. More sophisticated solutions that perform some abstract model-based diagnosis (de Kleer & Williams, 1987; Reiter, 1987) reasoning and/or try to alter the system to degrade in a graceful way (maintaining the best possible functional performance) (Weber & Wotawa, 2010) can also be implemented, which we account for in using the term SDC.

3. RELATED RESEARCH

In automated software testing, we can distinguish between two different kinds of methods: active testing and passive testing. Active testing (see (Broy, Jonsson, Katoen, Leucker, & Pretschner, 2005)) is a method that makes use of a SUT’s model for deriving test suites and a corresponding oracle. The model represents a formalization of the SUT’s essential behavior, from which tests, i.e., the required input and the expected output, can be obtained more or less directly. This usually results in abstract test cases, requiring a transformation into executable ones where parameters and actions are mapped to concrete values and actions.

Passive testing (Arnedo, Cavalli, & Nunez, 2003), or monitoring, is a testing methodology that mainly applies in situations where a SUT is at least not supposed to be fully controlled. The input then solely comes from users or the environment, and from real scenarios at that. Corresponding traces for these interactions are monitored. The passive tester then takes a system specification as reference model, and evaluates the collected traces with respect to this model in order to qualify a certain trace as correct or incorrect. Since we do not derive new test cases but reuse existing ones (possibly extended by a full live operation where we do not provide any inputs at all) and classify the results according to an abstract (safety) requirements specification, rather than checking the detailed functionality, our work could be classified as passive testing approach with an active “touch”. Obviously, also the concepts for the SDC component have to tackle many challenges of passive testing.

A method central to our approach is fault injection. Software fault injection (Voas & McGraw, 1999) is a common technique used in software testing as a modality to verify the application’s robustness and also tolerance of selected faults. This technique assumes the availability of a selection of operators that inject faults into the application. Among fault injection techniques commonly used for software applications, mutation testing is the oldest one, being introduced for the first time in 1971 (DeMillo, Lipton, & Sayward, 1978). Different types of software systems may use the benefits of mutation testing, since it can be successfully applied to different levels of testing and for different programming environments. In the context of mutation testing, we evaluate the quality of a test suite, i.e., a set of test cases, by injecting small software changes, i.e., mutations, at source code or byte code level, and then verify whether there is some anomalous response. That is, the generated test suite is run against all the mutants generated (the altered software versions), and we investigate whether there is some test case in the suite s.t. the output differs for the original program and a mutant. The mutation score (i.e., the percentage of mutants for which the test suite offers such a test case) serves as metric for assessing a test suite’s quality.

There are two major drawbacks with mutation testing. For one, there are time complexity issues in respect of the resources needed to run all the tests for all the mutants, specifically if we include many mutation operators. The other issue is related to interpreting the mutation score. That is, since, most likely, the functional equivalence between a mutant and its original program is not investigated beforehand, we run into the problem of identifying those mutants for which the test suite does not offer a killing test case, but which are functionally (semantically) equivalent to the original program (and thus represent an alternative correct implementation). If this cumbersome process is not executed, such equivalent mutants result in a lower mutation score, which has to be taken into account. There are many tools available for mutation testing, e.g.: FIAT (Fault Injection-based Automated Testing) (Segall et al., 1988), PROTEUM (Delamaro, 1993; Agrawal et al., 1989; Ghosh, 2000) tools for C source code mutation testing, MUJAVA (Ma, Offutt, & Kwon, 2006) a Java based mutation tool, and SQLMutation (Tuya, Suárez-Cabal, & la Riva, 2007) and JDAMA (Zhou & Frankl, 2009) for SQL. In this paper we assume that there is such a mutation testing tool for the desired development environment, that in turn allows us to inject faults into the control software.

4. TESTING THE SAFETY DIAGNOSTICS COMPONENT

In Sec. 2, we formalized the purpose of a safety diagnostics component in Def. 1 (see also Fig. 1). Informally, its task is
to monitor all the supervised system CS’s activities, and, if it
detects hazardous behavior (such that the safety requirements
would not be met), actions shall be taken in order to maintain
safety. To this end, the SDC collects all the inputs $I$ as con-
sidered by the control software, as well as CS’s output $O$, and,
if needed, issues special signals overriding $O$ and resulting in
output $O’$ s.t. the scenario $I \cup O’$ actually implements $REQ$.
The obvious question then is whether SDC indeed lives up to
this task and actually conforms to $REQ$ as of Def. 2.

For practical purposes, rather than considering a completely
unrestrained fault model, we included in Def. 2 also a set of
faults $FAULTS$ that should be considered for our conformance
tests. For those faults $f \in FAULTS$, we can derive faulty mu-
tants as of Definition 3 in order to exercise SDC. A relevant
fault mode for the WUMPUS example could be that a gear
switch would result in the wrong gear, and for the parking
brake a bit-flip in the derived wheel blocking signal such that
the wheels would be released without the driver requesting it.

**Definition 3** For some program CS, a mutant $CS’$ is an al-
tered version of the original program. A mutant is equivalent
to CS if and only if they do not differ in their behavior.

An advantage of our setting is that the identification of equiv-
alence in the traditional sense is not of importance in respect
of achieved scores. Since SDC shall remedy the effects of a
mutation (at the least in respect of the requirements $REQ$,
but possibly also concerning graceful degradation), we have
the situation that the original program and all the mutants are
even expected to be equivalent in respect of conformance to
$REQ$. Verifying whether all the mutants’ behavior, as de-
scribed by $I/\hat{O}$ scenarios, implements $REQ$ thus directly
translates to addressing our question.

For judging whether some $I/\hat{O}$ scenario actually implements
$REQ$, we can directly take the requirements’ formalization
and use a SAT or constraint solver to implement an oracle
and check the satisfiability of $I \cup \hat{O} \cup REQ$ as of Def. 2.

This leaves us with the question of which inputs to use. Ob-
viously, an exhaustive solution is impractical. The motivation
to effectively and efficiently exercise CS with our input sce-
narios is however shared with those test cases that were de-
designed for functionally evaluating CS itself. Thus we propose
to actively exploit this and reuse those inputs in our context,
minimizing test design efforts.

Now that we have established the basic concepts for our ap-
proach, let us briefly consider possible deployment issues to
be faced in industrial applications. An automotive applica-
tion, for instance, certainly qualifies for implementing some
SDC due to the complexity of related products (like cars) and
designated operating environments, and it is a domain sen-
sitive to safety concerns. There, for business and complex-
ity reasons, we cannot assume, for instance, the system to
be available to us as a white box, but rather a box with sub-
boxes, interfaces, and intellectual property cores in varying
shades of grey. Furthermore, an easy integration into existing
and proven development concepts is essential in order for an
approach to be attractive enough for actual deployment. In
order to accommodate such concerns, we make the following
assumptions in respect of the data available to us.

- First, the requirements $REQ$ to be monitored and en-
forced by $SDC$ have to be available. If we have to con-
vert their informal characterization into a formal syntax
(as accessible by automated tools) first, research like (Pill
& Quaritsch, 2013) can identify mistakes in this process
and provides the means to investigate unexpected results
- but is out of the scope of this paper. The formalized
requirements $REQ$ will be used to assess whether an in-
dividual test scenario passes or fails our expectations in
that it complies with $REQ$ or not. Please remember that
the requirements, most likely, implement abstract safety
requirements and do not encode the system’s detailed
functionality (if the latter is taken to the extreme, $SDC$
could become redundant to $CS$). That is, some safety
requirement like that a robot should never hit a wall, or
that an automated parking brake should never block the
wheels when the car is still moving (i.e. at high velocity).

- Second, we assume that the control software $CS$ offers us
controls to inject / simulate / enable faults $f \in FAULTS$
via a function $\mu(CS, f)$. Thus secrecy about detailed in-
tellectual property can be maintained, and the developers
working on the very components and system aggregation
can integrate the best fault injection (mutation) opera-
tors for the individual context. What is needed, however,
is (a) some guidelines in order to assist these designers
in providing meaningful faults (b) a full list $FAULTS$ of
available fault models, and (c) a guideline for interpreting
the impact of each fault $f \in FAULTS$ so that we, on
one hand, gain confidence in the results, and on the other
hand can act on encountered issues (e.g., when a test case
produces unexpected results - see also the discussion of
Algorithm 2).

- Third, we assume the availability of the test cases de-
dsigned for $CS$ (or the overall system). This test suite $TS$
was derived in order to extensively exercise the system
and ensure its correctness, so that there is no need to
come up with entirely new input scenarios. We, how-
ever, are interested only in the input part of these test
cases, since the evaluation of the system’s response is
done with respect to the abstract system safety require-
ments $REQ$ instead of the original functional or perfor-
mance concerns. To this end, the formalized and thus
executable requirements are used as testing oracle in or-
der to classify the test output.

Now let us propose an easily integrable testing approach.
The underlying concept of the algorithm as depicted in Algorithm 1 is as follows. For any fault $f \in \text{FAULTS}$, we create a corresponding mutant $\text{SUT}'$ using the injection function $\mu$. Then, for any test case $t \in \text{TS}$ we execute the mutant for the corresponding inputs of $t$, and record the trace of this execution. This trace is checked for compliance with the requirements $\text{REQ}$ and classified accordingly. Naturally, this raises the question of whether to include also the unmodified program in the tests, which we allow the test engineers to answer for Algorithm 1 such that they can include a corresponding fault $\epsilon$ in $\text{FAULTS}$ that results in an unaltered program.

Note that our concept is orthogonal to the decision of whether the system/environment (or possibly a mock-up) is part of a $\text{SUT}$'s model or not - which might depend on the test setup and design stage. Thus, while the $\text{SUT}$ might contain it, we will omit it in our algorithmic presentations.

Algorithm 1 TEST-SDC($\text{SUT}$, $\text{FAULTS}$, $\mu$, $\text{REQ}$, $\text{TS}$)

Input: The system under test $\text{SUT}$ ($CS + \text{SDC} + \text{SELFCT}$), the set $\text{FAULTS}$ of faults to be considered, the fault injection function $\mu$, the safety requirements $\text{REQ}$, and the original test suite $\text{TS}$ for $\text{CS}$ (or $S$).

Output: PASS if the system under test behaves as demanded by the given requirements $\text{REQ}$, and FAIL otherwise.

1: for all $f \in \text{FAULTS}$ do
2:  $\text{SUT}' = \mu(\text{CS} + f) + \text{SDC} + \text{SELFCT}$
3: for all $t \in \text{TS}$ do
4:  $\text{res} := \text{EXECUTE(} \text{SUT}', \text{input}(t))$
5:  if $\text{res} \cup \text{input}(t) \cup \text{REQ} \models \bot$ then
6:    return FAIL
7: end if
8: end for
9: end for
10: return PASS

Algorithm 2 answers the most basic question of whether there was a scenario (a combination of a test case $t$ and an injected fault $f$) that violated the safety requirements, or if we have that all the scenarios complied with $\text{REQ}$. A viable alteration to the algorithm would be to return for an encountered failed scenario, both the test case $t$ and the injected fault $f$. Another variant could run all the tests (archiving the results) instead of stopping on the first unvelled violating scenario.

Reusing the test cases that were designed to extensively exercise $CS$, for one, allows us to compare and connect test results for $CS$ (or the overall system $S = \text{system/environment} + CS$) with our special purpose tests of $S^+ = \text{CS} + \text{SDC}$ (or $S^+ = S^+ + \text{system/environment}$) for evaluating the effectiveness of $\text{SDC}$, and furthermore we do not require additional resources for test design (not counting the mutation function $\mu$). Comparing the evaluation of testing $CS$ and our special tests for $S^+$, however, is not as simple as it might sound, since the testing purposes (and thus the oracles for classifying the results) differ significantly. That is, when we test the safety diagnostics component $\text{SDC}$, the focus is solely on the safety requirements, while for testing $CS$ such concerns are mingled with others like functionality and performance. In other words, regardless of whether a test scenario in $\text{TS}$ was originally intended to meet or fail some other goal, $\text{REQ}$ is to be met anyhow. Nevertheless, reusing the test inputs helps us in making the connections when interpreting the results in more detail.

While Algorithm 1 focuses on verifying whether $\text{SDC}$ lives up to the expectations encoded in $\text{REQ}$, it shall be noted that it does not aim at verifying whether adding $\text{SDC}$ to the system would result in degraded functionality or performance (in some respect other than conformance to the safety requirements $\text{REQ}$). To this end, a system integration test could be of interest also for $S^+ (S^+)$, Also there it makes sense to reuse the test cases generated for $S$, and even to reuse the same evaluation principles (oracle). That the runs for these system integration tests and our testing purpose could be executed in unison (with multiple or varying evaluations/oracles) is most evident and implemented by Algorithm 2.

Algorithm 2 TEST-SDC-EXT($\text{SUT}$, $\text{FAULTS}$, $\mu$, $\text{REQ}$, $\text{TS}$)

Input: The system under test $\text{SUT}$ ($CS + \text{SDC} + \text{SELFCT}$), the set of faults to be considered $\text{FAULTS}$, the fault injection function $\mu$, the safety requirements $\text{REQ}$, and the original test suite $\text{TS}$ for $\text{CS}$ (or $S$).

Output: The test results $\text{TR}$, which is a list of tuples $\{t, f, \text{res}, r\}$ such that $t \in \text{TS}$, $f \in \{0 \cup \text{FAULTS}\}$ tells us the injected fault $f$ (0 indicates that we did not inject a fault), $\text{res}$ is the output obtained when executing the scenario (the combination of $t$ and $f$), and $r \in \{\text{REQPASS}, \text{REQFAIL}\}$ indicates whether for the described scenario the requirements $\text{REQ}$ are violated or not.

1: $\text{TR} \leftarrow \emptyset$
2: for all $t \in \text{TS}$ do
3:  $\text{res} := \text{EXECUTE(} \text{SUT}, \text{input}(t))$
4:  if $\text{res} \cup \text{input}(t) \cup \text{REQ} \models \bot$ then
5:    $\text{TR} = \text{TR} \cup \{t, 0, \text{res}, \text{REQFAIL}\}$
6:  else
7:    $\text{TR} = \text{TR} \cup \{t, 0, \text{res}, \text{REQPASS}\}$
8: end if
9: for all $f \in \text{FAULTS}$ do
10:  $\text{SUT}' = \mu(\text{CS} + f) + \text{SDC} + \text{SELFCT}$
11:  $\text{res} := \text{EXECUTE(} \text{SUT}', \text{input}(t))$
12:  if $\text{res} \cup \text{input}(t) \cup \text{REQ} \models \bot$ then
13:    $\text{TR} = \text{TR} \cup \{t, f, \text{res}, \text{REQFAIL}\}$
14:  else
15:    $\text{TR} = \text{TR} \cup \{t, f, \text{res}, \text{REQPASS}\}$
16: end if
17: end for
18: end for
19: return $\text{TR}$

Each test case is executed for the correct $\text{SUT}$ as well as all the mutated versions $\text{SUT}'$ obtained by injecting the individual faults $f \in \text{FAULTS}$. Then the obtained outputs as well as the verdict whether the outputs satisfy or contradict the given re-
uirements $REQ$ are archived. Please note that one could easily add to the archived tuple also any further functional/non-functional evaluation results. With Algorithm 2, we thus can directly compare the results for the unmodified $SUT$ $S^+$ ($S'$) (including the $SDC$) with those for the original system (excluding the $SDC$) in order to assess the impact of $SDC$ on the performance. Furthermore, we do store also the outputs for the test scenarios with injected faults, to the end of supporting later inspections of further aspects.

While the two algorithms offer varying details to the user, the main question addressed is whether the $SDC$ component is indeed able to keep the overall system for the test scenarios within the boundaries defined by $REQ$. If this is not the case, there is the obvious question of how to debug this situation. While answering this question is not in the scope of this paper, dynamic slicing techniques (Korel & Rilling, 1998; Zhang, He, Gupta, & Gupta, 2005) could certainly be of help if we start from the signals involved in the violated requirement and reason backwards to the inputs. An urgent question for every failed scenario will be whether the $SDC$ did not catch the issue at all, s.t. its monitoring capabilities are insufficient, or whether its response to the situation was inadequate. For an enhanced automated support of debugging which individual parts of the requirements were involved in their violation, adopting the work on requirements/specification diagnosis presented in (Pill & Quaritsch, 2013) could be of interest.

5. DISCUSSION AND CONCLUSIONS

In this paper, we propose a testing concept tailored towards gaining confidence that a functional safety diagnostic component indeed lives up to our expectations of monitoring a system at an abstract level and maintaining safety for malfunctions in the more detailed and highly complex main control system. We depict an initial attempt at such an approach that does neither demand for the design of additional tests, nor requires us to entirely restructure our testing efforts in the given development cycle. Instead, we reuse the input part of those test cases designed for the monitored system, and the oracle is implemented directly from the given formalized requirements. Thus we minimize needed resources and efforts.

The only thing we have to rely on is a fault injection function for the main control software that allows us to inject faults so that we can effectively trigger actions from the safety diagnostics component. This accommodates also business concerns related to intellectual property rights and the availability of detailed system models, such that designers only have to add corresponding functionality that they can also use for their testing of the individual components. In the worst case of having no such function, we could revert to simple mutations on the software’s output signals. We outline two algorithms that (1) offer a quick search for a scenario violating the requirements, or (2) run the complete test suite for the unmodified $SUT$ as well as all the mutated versions achievable via the given fault injection function, storing all the obtained results. While the first offers us some quick check whether everything is fine in respect of compliance to the safety requirements, the second variant is more tailored towards a full inspection where the obtained data are also stored for future evaluation in respect of other aspects.

Future work will have to show the practical viability of our concept. While the reuse of tests does have its advantages as discussed, we will also explore ideas for (additional) tests solely derived for testing an $SDC$ and their impact on general fault detection performance. A specific aspect of this research will be to isolate means that can help us in selecting test scenarios (combinations of faults and test inputs). That is, while we showed with Algorithm 2 that our work can be integrated easily into the system integration tests that we certainly would like to run, the available testing resources might not suffice to run all the combinations of mutated $SUT$ variants and input sequences in the test suite. Identifying an effective prioritization scheme would then certainly be of interest.

Also the process of identifying an effective selection of mutation operators/functions to be used for some project will be a target of our future research. That is, mutation testing relies on the Competent Programer Hypothesis and the Coupling Effect, which assume that (a) a faulty program is close to the correct one and suggest that (b) a test suite that catches simple mutations is also effective at catching more complex faults (see e.g. the discussion in (Offutt, 1992)). Newer findings like the discussion in (Gopinath, Jensen, & Groce, 2014) investigate the impact of the selection of mutation operators on the performance for a specific project, and suggest that for a specific project, further empirical data should provide a solid empirical footing for the underlying hypotheses’ validity. In this context, we limit our algorithms currently to injecting single faults, where the accommodation of multiple faults for more complex scenarios is subject to future work. In the latter respect, combinatorial testing like it was used in (Wotawa & Pill, 2014) for configuration testing in an automotive context will be of interest.

Enhancing the support for a user debugging those scenarios that violated the requirements, adopting the work on specification/requirements diagnosis in (Pill & Quaritsch, 2013) could provide interesting stimuli, as could the work for diagnosing failed test cases for service-oriented software architectures (Hofer, Jehan, Pill, & Wotawa, 2014).

Last but not least, we currently consider functional safety requirements only. In the future, also $SDC$ solutions for non-functional requirements should profit from an improved concept accommodating also non-functional requirements.
ACKNOWLEDGMENT

Our research was conducted as part of the eDAS project funded by the European Commission under FP-7 grant agreement number: 608770.

REFERENCES


Direct analysis of non-quadratic phase coupling for detection of linearly modulated signals

Fidel Hernández¹, and Aitzol Iturrospe²

¹,²University of Mondragon, Mondragon, Guipuzcoa, 20500, Spain

alaina@upr.edu.cu
aiturrospe@mondragon.edu

ABSTRACT

The detection of a linearly modulated signal is currently accomplished by applying the Bispectrum. This technique is capable of detecting quadratic phase coupled spectral components, and consequently, can be used in order to reveal a linearly modulated signal presence. However, a linear modulation by itself does not exhibit quadratic phase coupled spectral analysis. Then, the application of the Bispectrum for detecting linearly modulated signals could be unsuccessful. In this paper a general method for detection of linearly modulated signals, which can be applied whether the signals comprise quadratic phase coupled spectral components or not, is proposed. This method is evaluated through numerical simulations and it is applied for detecting a local fault in rolling element bearings. The achieved results are compared with those obtained by the traditional spectral analysis and the Bispectrum, revealing the effectiveness obtained by the application of the proposed method.

1. INTRODUCTION

Second-order signal interactions or transformations analysis is a common issue in Signal Processing (Chaari, Bartelmus, Zimroz, Fakhfakh, and Haddar (2012); Chen & Zuo, 2009). Second order interactions are characterized by Quadratic Phase Coupled (QPC) spectral components, and consequently by linear modulations. Although this paper is focused on linear modulation, the analysis proposed here can be extended to other kind of modulations (e.g., exponential modulations) whenever the phases of the spectral components at both sides around the frequency of the so called carrier signal are phase coupled.

Some published papers have been focused on the problem of detection of phase coupled spectral components, in particular QPC spectral components. In these publications QPC detection has been applied to different signals; such as biological signals (Venkatakrishnan, Sukanesh, and Sangeetha (2011)), telecommunication signals (Sanaullah, 2013), mechanical vibration signals (Bouillaut & Sidahmed, 2001; Raad & Sidahmed, 2002), etc. In order to detect QPC spectral components, higher-order statistical signal processing, more specifically Bispectrum (Venkatakrishnan et al. (2011); Sanaullah, 2013; Fackrell & McLaughlin, 1995), has been successfully applied. Bispectrum, by definition, detects the quadratic phase coupling among spectral components. However, a linearly modulated signal by itself is not a signal with QPC spectral components necessarily.

This paper proposes a general method for detecting linearly modulated signals based on the analysis of phase coupling among spectral components located at both sides around a center frequency.

The rest of the article is organized as follows. Section 2 presents the well-known Bispectrum’s capability of detection of modulated signal once the QPC condition is fulfilled and the problem concerning the application of the Bispectrum on the detection of a linear modulation for a signal without QPC spectral components; in Section 3, the theoretical foundations of the proposed method are explained; Section 4 corroborates these theoretical foundations by numerical simulations; Section 5 presents the results of the application of this method on rolling element bearing fault detection and also a comparison with results obtained by the application of the traditional spectral analysis and the Bispectrum.

2. BISPECTRUM AND THE DETECTION OF LINEARLY SIMULATED SIGNALS

A QPC interaction is produced by a second-order transformation system as follows (Sanaullah, 2013; Gallego, Urdiales, and Ruiz (1999); Seydnejad, 2007):

\[ y(t) = ax^2(t) + bx(t) + c, \]  

where \( x(t) \) and \( y(t) \) are the signals at the system input and output, respectively.
For example, let

\[ x(t) = \cos(2\pi f_1 t + \varphi_1) + \cos(2\pi f_2 t + \varphi_2), \quad (2) \]

where \( \varphi_1 \) and \( \varphi_2 \) are independent random variables with uniform probability density function between \(-\pi\) and \(\pi\). Then,

\[ y(t) = c + b \cos(2\pi f_1 t + \varphi_1) + b \cos(2\pi f_2 t + \varphi_2) + 2a \cos(2\pi f_1 t + \varphi_1) \cos(2\pi f_2 t + \varphi_2) + a \cos^2(2\pi f_1 t + \varphi_1) + a \cos^2(2\pi f_2 t + \varphi_2) \quad (3) \]

As a result of the second-order transformation process, a linearly modulated signal, given by the term

\[ 2a \cos(2\pi f_1 t + \varphi_1) \cos(2\pi f_2 t + \varphi_2) \]

in Eq. (3), is obtained at the system output. The signal \( y(t) \) comprises spectral components at frequencies \( f_1, f_2, f_1 + f_2, f_1 - f_2, 2f_1 \) and \( 2f_2 \). Given that QPC spectral components are defined as spectral components at frequencies \( f_a, f_b \) and \( f_a + f_b \) with phases equal to \( \varphi_a, \varphi_b \) and \( \varphi_a + \varphi_b \), respectively [3, 4, 5, 6, 7, 9], then it can be stated that \( y(t) \) fulfills the QPC condition.

Bispectrum (third-order spectral cumulant) is a common technique applied for detecting QPC spectral components. The Bispectrum of a signal \( z(t) \) can be obtained as follows (Venkatakrishnan et al. 2011; Sanaullah, 2013; Fackrell & McLaughlin 1995; Rivola & White 1998):

\[ \mathcal{B}(f_a, f_b) = E\{Z(f_a)Z(f_b)Z^*(f_a + f_b)\} \quad (4) \]

where \( Z(f) \) is the Fourier transform of \( z(t) \).

Let \( z(t) \) denote the signal obtained from Eq. (3) corrupted by a zero mean Gaussian noise, \( w(t) \), as

\[ z(t) = y(t) + w(t) \quad (5) \]

The Bispectrum of signal \( z(t) \) is in theory different from zero only for the bispectral components \( (f_a - f_b, f_b) \) and \( (f_a, f_b) \). Such nonzero bispectral components are indicating that certain spectral components in \( z(t) \) fulfill the QPC condition, and consequently the signal \( z(t) \) comprises a linearly modulated signal.

An important question must be taken into consideration: a linearly modulated signal by itself does not necessarily comply with the QPC condition; then the application of Bispectrum for detecting linearly modulated signal can be unsuccessful.

### 3. General Method Proposed for Linearly Modulated Signal Detection

In order to detect a linearly modulated signal, the calculation of the phase coupling among spectral components at both sides around a center frequency can be carried out depending on the characteristics of the modulated signal to detect. For example, the following equations calculate the spectral components phase coupling for detecting a linearly modulated signal with unsuppressed carrier, a linearly modulated signal with suppressed carrier, and a linearly modulated signal with only odd spectral components around the carrier frequency, respectively:

\[ D_0(f_c, \alpha) = E\{Z(f_c - \alpha)Z(f + \alpha)\} \cdot |Z(f_c)|e^{-j2\alpha \text{ang}[Z(f_c)]} \quad (6) \]

\[ D_2(f_c, \alpha) = E\{Z(f - 2\alpha)Z^*(f + 2\alpha)\} \quad (7) \]

\[ D_3(f_c, \alpha) = E\{Z(f - 3\alpha)Z^*(f + 3\alpha)\} \quad (8) \]

where \( E\{\cdot\} \) is the expected value operator, \( Z(f) \) is the Fourier transform of the signal under study and \( \text{ang}[w] \) is the angle of the complex number \( w \).

For example, given a linearly modulated signal with unsuppressed carrier at frequency equals to \( f_c \), there will be an \( \alpha = f_m \), provided that \( f_c + f_m \) and \( f_c - f_m \) are the frequencies of two nonzero modulated signal spectral components, such that \( D_0(f_c, f_m) \neq 0 \). An example of a linearly modulated signal spectrum is shown in Figure 1.

![Figure 1. Example of a linearly modulated signal spectrum.](image)

Moreover, the modulation signal detection algorithm is immune to the corrupting noise, then it can be applied under low signal-to-noise rates. For example, given a signal \( z(t) \) comprising an amplitude modulation signal, \( a(t) \), with unsuppressed carrier at frequency equals to \( f_c \) and modulating signal with frequency equal to \( f_m \), plus a random signal (noise), \( a_n(t) \), the Eq. (6), at frequencies \( f_c, f_m \) can be written as follows:

\[ D_0(f_c, f_m) = E\{Z(f_c - f_m)Z(f_c + f_m)\} \cdot |Z(f_c)|e^{-j2\text{ang}[Z(f_c)]} \quad (9) \]
The spectral components in \( z(t) \) at frequencies \( f_c - f_m \), \( f_c + f_m \) and \( f_c \) correspond with spectral components of modulation signal and noise, then:

\[
D_0(f_c, f_m) = \mathbb{E}\left[ \left| A_{-m} e^{j(\varphi_c - \varphi_m)} + A_{n} e^{j\varphi_c} \right| \right.
\]

\[
\left. \left| A_{m} e^{j(\varphi_c + \varphi_m)} + A_{m} e^{j\varphi_c} \right| \right].
\]

(10)

where \( A_{-m} \) and \( \varphi_c - \varphi_m \), \( A_{+m} \) and \( \varphi_c + \varphi_m \), \( A_c \) and \( \varphi_c \), are the amplitudes and phases of spectral components of the modulation signal at frequencies \( f_c - f_m \), \( f_c + f_m \) and \( f_c \), respectively, and \( A_{-n-m} \) and \( \theta_{n-m} \), \( A_{n-m} \) and \( \theta_{n-m} \), \( A_{n} \) and \( \theta_{n} \), are the amplitudes and phases of spectral components of the random signal at frequencies \( f_c - f_m \), \( f_c + f_m \) and \( f_c \), as well.

The development of Eq. (10) results in

\[
D_0(f_c, f_m) = \mathbb{E}\left[ A_{-m} A_{+m} A_c \right]
\]

(11)

the significant magnitude of which will be indicating the modulation signal presence. Furthermore, Eq. (11) reveals the independent nature of the algorithm with respect to the corrupting noise magnitude.

4. NUMERICAL SIMULATIONS

In order to evaluate the proposed method, 1000 realizations of a discrete linearly modulated signal with non-quadratic phase coupled spectral components, corrupted by a zero mean gaussian noise with variance equal to 4, are generated by using the software Matlab\textsuperscript{®}, according to the following equation:

\[
y[n T_s] = \cos(2\pi f_1 n T_s + \varphi_1)
+ \cos(2\pi f_2 n T_s + \varphi_2)
+ \cos(2\pi f_1 n T_s + \varphi_3) \cos(2\pi f_2 n T_s + \varphi_2)
+ w(n T_s)
\]

(7)

where \( T_s = 1 \text{ ms}, f_1 = 10 \text{ Hz}, f_2 = 200 \text{ Hz} \), and \( \varphi_1, \varphi_2, \text{ and } \varphi_3 \) are independent discrete random variables with uniform probabilistic density function between \(-\pi \) and \( \pi \). It should be noted that a linear modulation signal is produced but the spectral components are not quadratic phase coupled.

The Bispectrum of \( y[n T_s] \) does not reveal any information about the linearly modulated signal. This can be observed in Figure 2. The magnitude of the bispectral components at frequencies \( (f_1, f_2) = (190, 10) \) and \( (f_1, f_2) = (200, 10) \), calculated for the simulation signal, are shown in Figure 3. The sketch of function \( |B(f_1, 10)| \) for a linearly modulated signal is presented in Figure 3.
modulated signal with non-quadratic phase coupled spectral components.

As expected, it can be observed in Figure 3 that the application of the Bispectrum for detecting a linearly modulated signal is not effective.

If $D_0(f, \alpha)_3$ is computed for the simulation signal, a nonzero value at $(f, \alpha) = (200, 10)$, indicating that a modulated signal is present, is obtained (see Figure 4). The amplitude of the component at $(f, \alpha) = (200, 10)$ can be observed in Figure 5, where the magnitude of the function $D_0(f, \alpha)_3$, for $\alpha = 10$ Hz, is plotted.

Figure 4. Sketch of $|D_0(f, \alpha)_3|$ for a linearly modulated signal with non-quadratic phase coupled spectral components.
5. DETECTION OF A VIBRATION MODULATED SIGNAL PRODUCED BY A ROLLING ELEMENT BEARING WITH A LOCAL FAULT

When a rolling element bearing has a local fault produces a vibration with an amplitude modulation waveform (Seydnejad, 2007). This signal does not satisfy the QPC condition necessarily. The modulating signal is periodic (with main frequency known as “fault characteristic frequency”, $f_{ca}$), it depends on the physical characteristics of the bearing components, and it is proportional to the rotating frequency of the shaft supported by the bearing (Seydnejad, 2007). The detection of this fault is performed by identifying the modulated signal spectral components, spaced in the fault characteristic frequency that arise around the frequencies associated to the natural frequencies of the mechanical system (Randall & Antoni (2011); Hernandez & Caveda (2008)). The accurate identification of such modulated signal can be affected either by spectral components of vibrations from sources unrelated to the bearing fault mechanism or the background noise. Under these conditions, the application of the proposed method represents a suitable tool.

In order to test the effectiveness of the application of the proposed method on bearing fault detection, $D_0(f, \alpha)_3$ is employed for analyzing a vibration signal experimentally obtained from a rolling element bearing with a local fault. $D_0(f, \alpha)_3$ is chosen because the modulated signal has a significant spectral component magnitude at the carrier frequency. The Spectrum and the Bispectrum are also calculated for comparison purposes.

A vibration signal produced by a rolling element bearing with an incipient local fault in the bearing outer race ($f_{ca} = 98$ Hz), supporting a shaft rotating at 1500 RPM (25 Hz), is sampled at 50 kHz and analyzed.
The Amplitude Bispectrum of the vibration produced by the incipient local fault in the bearing outer race is shown in Figure 7a and the Amplitude Bispectrum contour is shown in Figure 7b. As it can be seen in Figure 7b, the modulated signal is not detected by the Bispectrum and the fault characteristic frequency, in this case \( f_{ca} = 98 \) Hz, is not identified either. The representation of the magnitude of the Bispectrum along the frequency \( f_2 = 98 \) Hz (see Figure 7c) shows clearly the unsuccessful application of the Bispectrum for detecting the modulated signal.

\( D_0(f, \alpha)_3 \) is calculated for the vibration produced by the same incipient local fault in the bearing outer race. The absolute value of the computed \( D_0(f, \alpha)_3 \) and its corresponding contour are shown in Figure 8a and Figure 8b, respectively. As it can be seen in Figure 8a and Figure 8b, nonzero components at \( (f, k \cdot 98) \), \( k = 1, 2, \ldots \), are obtained, which indicates that several phase coupled spectral components, spaced in 98 Hz, have been excited. Therefore, it is possible to assure that the modulation has been detected and that the fault characteristic frequency has been identified. The sketch of the magnitude of the function \( D_0(f, \alpha)_3 \) along the frequency \( \alpha = 98 \) Hz is shown in Figure 8c, where the achieved effectiveness is better revealed in comparison with that obtained by the Bispectrum (see Figure 7c).

6. CONCLUSION

A method for detecting linearly modulated signals has been proposed. It was demonstrated that this method is suitable to be applied for detecting the distinct phase coupling among the modulated signal spectral components, whether such a coupling is quadratic or not. This method could also be applied for detecting other kind of modulations (e.g., exponential modulations) since spectral components at both sides of the carrier frequency are found to be phase coupled. The proposed method for detecting a linearly modulated signal is based on the phase relationships among spectral components around a center frequency.

This method has been experimentally evaluated, and its applicability has been verified by detecting a local fault in rolling element bearings.

It has been demonstrated through the numerical and experimental works that better results can be achieved by applying the proposed method, in comparison with the use of the conventional spectral analysis and the Bispectrum analysis. In fact, the Bispectrum analysis is one of the currently most used technique for detecting phase coupled spectral components.
Figure 7. Bispectrum of the vibration produced by an incipient local fault in a bearing outer race ($f_{ca} = 98$ Hz). 
a) Amplitude Bispectrum. b) Amplitude Bispectrum contour. c) Amplitude Bispectrum along the frequency $f_2 = 98$ Hz.

Figure 8. $D_0(f, \alpha)_3$ for the vibration produced by an incipient local fault in a bearing outer race ($f_{ca} = 98$ Hz). 
a) Sketch of $|D_0(f, \alpha)_3|$. b) Contour of the magnitude of $D_0(f, \alpha)_3$. c) Sketch of $|D_0(f, \alpha)_3|$ along the frequency $\alpha = 98$ Hz.

REFERENCES


**Aitzol Iturrospe** received his an Engineer in Automation and Electronics (MSc) and PhD degrees from the University of Mondragon. After finishing his PhD studies, which focused on signal processing applied to industrial processes and systems monitoring, he was a visiting researcher at the Laboratory for Manufacturing and Sustainability (LMAS) at the University of California in Berkeley, where he worked on the development of wireless micro-sensors for monitoring micro-manufacturing processes. After returning from California, he worked as a researcher for industrial and aeronautical industries. He is currently a member of the Signal Theory and Communications area at the University of Mondragon. His research focuses on signal processing applied to Non-Destructive Testing (NDT) and health monitoring applications (as well as on the development of sensors and measurement systems).

**Biographies**

**Fidel Hernández.** Born in Pinar del Rio, Cuba, in 1972. Engineer in Telecommunications and Electronic by the University of Pinar del Rio, Cuba, in 1995. M.Sc. in Digital Systems by CUJAE, Havana, Cuba, in 2000, Ph.D. in Electronics and Industrial Automation by University of Mondragon, Spain, in 2006. Researcher and Assistant Professor at the University of Pinar del Rio, Cuba, since 1995 until 2014. Head of the Research Group for Advanced Machine Diagnosis (GIDAM), University of Pinar del Rio, Cuba, since 2000 until 2014. Currently, he works as Researcher at the University of Mondragon, Spain. He has coordinated several international and national projects involving eye image processing for medical diagnostic applications, higher-order statistical signal processing applied on mechanical vibrations, classification and demodulation of communication signals, and others. He is a CYTED expert, and is associate editor and reviewer of various international journals. He is member of the administration committee of the Cuban Association of Pattern Recognition.
Valid speed signal is essential for proper condition monitoring of modern variable speed wind turbines. Traditionally, a tachometer mounted on the high speed shaft provides reference for tracking speed dependant frequency components, such as generator speed harmonics and gearbox tooth mesh frequencies. The health assessment of drive train components is limited to broadband measurements when the speed signal is invalid. This condition results in reduced fault detection capabilities and consequently decreased lead time. In this work, a new speed estimation algorithm is presented in order to overcome the above mentioned issues. The high speed stage shaft angular velocity is calculated based on the maximum correlation coefficient between the 1st gear mesh frequency of the last gearbox stage and a pure sinus tone of known frequency and phase. The proposed algorithm utilizes vibration signals from two accelerometers for cross-referencing purposes. The method is tested in three drive train configurations, where 720 sets of vibration signals of 10.24s length, sampled at 25.6kHz are analysed. Consistent speed estimation reaches approximately 98% when two vibration sources are utilized, whereas it is lower when only one source is taken into account. No apparent patterns arise between speed variation levels or power production and the number of invalid outputs, showing the independence of the method from operational parameters.

1. Introduction

Condition monitoring of wind turbine drive trains can be considered as a procedure carried out in two layers (Andersson, Gutt, & Hastings, 2007). In the first layer, extracted condition indicators (CI), corresponding to characteristic frequencies describing the state of a component, are employed for long time trending and alarming. Gearbox tooth mesh frequencies (Bartelmus & Zimroz, 2009; Taylor, 2000), energy in the high or low frequency range (Marhadi & Hilmisson, 2013) and running speed harmonics (Wu, Lin, Han, & Ding, 2009) are typical CIs used by condition monitoring experts. The diagnostic process is moved to the second layer when an alarm is generated, where detailed analyses in time and frequency domains take place in order to verify the presence and nature and consequently assess the severity of a developing fault.

The above described condition monitoring scheme relies on the capability of generating alarms based on both speed and not-speed related condition descriptors. If the generator speed signal is invalid, due to either hardware problem or improper installation, monitoring of the drive train components is limited to broadband measurements, which provide vague information regarding the nature of the fault. Hence, failure modes manifested as developing speed related frequencies, such as the second running speed harmonic in case of misalignment between the generator and gearbox, are challenging to be detected in early stage resulting in reduced lead time for inspection and correction.

Several speed estimation techniques have been developed and
proposed the past few years utilizing vibration signals such as resampling in the angular domain (Bonnardot, Badolou, Randall, Danie’re, & Guillet, 2005; Villa, Reñones, Perán, & De Miguel, 2011; Urbanek, Zimroz, Barszcz, & Antoni, n.d.), Chirplet transform (Peng et al., 2011), combination of Chirplet transform and Vold-Kalman filtering (Zhao, Lin, Wang, Lei, & Cao, 2013), short–time scale transformation (Bonnardot, Badaloui, et al., 2005; Villa, Reñones, Perán, & Antoniadis, 2009). Furthermore, tachless approaches are adopted in order to mitigate the reduction in effectiveness of the speed sensor in variable speed conditions (Borghesani, Pennacchi, Randall, & Ricci, 2012; Coats, Sawalhi, & Randall, & Antoniadis, 2009). The present paper investigates the feasibility of extracting the speed signal by tracking a running speed multiple in the time domain, such as the mesh frequency of the gearbox high speed stage. The method is based on locating the maximum correlation coefficient between the selected spectral component and a pure sinus tone. This technique constitutes an effort to utilize the raw vibration signal while decoupling the estimation of the instantaneous frequency from the frequency and angular domains.

The paper is organised as follows. The mathematical background of the maximum correlation coefficient method is presented in section 2. The challenges regarding the speed estimation process in general are discussed in section 3. Section 4 presents the developed algorithm utilizing two vibration sources. The validation of the method is tested offline in 120 turbines of three different drive train topologies and the results are shown in section 5. Finally, the main conclusions are discussed in section 6.

2. USE OF MAXIMUM CORRELATION COEFFICIENT FOR FREQUENCY ESTIMATION

2.1. Method Description

The extraction of frequency $f_a$, within a predefined range $[f_{low}, f_{high}]$, and angle $\phi_a$, where $\phi_a \in (-\pi/2, \pi/2)$, of a signal $x_a(n)$, where $n$ is the sample number, sampled at $F_s$ can be achieved in the time domain via the maximum correlation coefficient method (MCC) (Bellini, Franceschini, & Tassoni, 2006). Based on this technique, a test signal $x_b(n, k, l)$ of frequency $f_b(k)$ and phase $\phi_b(l)$ generated and the correlation coefficient $C_{a,b}(k,l)$ between the two signals $x_a(n)$ and $x_b(n,k,l)$ is calculated. The values of $f_b(k)$ and $\phi_b(l)$ which yield the maximum absolute correlation coefficient $C_{a,b}(k,l)$ provide the estimation of the initial signal frequency $f_a$ and angle $\phi_a$ respectively.

$$x_b(n, k, l) = A_b \sin \left( \frac{2\pi f_b(k)n}{F_s} + \phi_b(l) \right)$$

where $f_b(k) \in [f_{low}, f_{high}]$ and $\phi_b(l) \in (-\pi/2, \pi/2]$. Parameters $k$ and $l$ refer to the frequency and phase difference between two consecutive test frequencies and angles respectively and they depend on the desired accuracy.

$$C_{a,b}(k,l) = \frac{\text{cov}(x_a(n), x_b(n,k,l))}{\sqrt{\text{cov}(x_a(n), x_a(n)) \cdot \text{cov}(x_b(n,k,l), x_b(n,k,l))}}$$

$$f_a \approx f_b(k) \text{ and } \phi_a \approx \phi_b(l) \text{ for } k, l \mid |C_{a,b}(k,l)| = \max$$

The approximation sign is used in order to denote that the calculated quantities are approximations of the true values depending on the specified resolution.

Figure 1 illustrates the signal described in equation 4 of total length equal to 0.5 seconds, which consists of a strong sinusoidal tone of 20Hz and initial phase 60°, a weaker signal at 8Hz and white Gaussian noise $\nu$. The objective is to estimate the dominant frequency and corresponding phase within this short time interval.

$$x_a(n) = \sin (2\pi 10n + 60°) + 0.4 \sin (2\pi 4n) + \nu$$

Assuming that a rough estimation of the frequency band is available ($|f_a - f_{range}, f_a + f_{range}|$), and selecting the frequency and phase resolutions to be 0.05Hz and 9 degrees respectively, the calculated test frequency and phase are

$$f_b = 20Hz \text{ and } \phi_b = 61.6°$$

Figure 2 shows a contour plot of the absolute correlation coefficient as function of the test frequencies and phases. There are two sets of maxima, for the two components of $x_a(n)$, i.e. 8Hz and 20Hz. The optimum value is indicated by a data cursor, corresponding to frequency and phase of 20Hz and 61.6°; the maximum correlation coefficient value is 0.9242. The maximum tracked frequency is moderately dependant on phase, varying approximately 3.5% from the true value at angle approximately minus 90° off the actual phase.

In the following time section, e.g. next 0.5 seconds, the same process would be repeated and the proper frequency estimation is yielded if the the frequency of interest alters. In wind turbine applications, the expected frequency is assessed to be close to the former result, hence making the process faster by keeping a small frequency range of search.

The maximum correlation coefficient method does not pro-
3. SPEED ESTIMATION CHALLENGES

The commonest location of vibration signals utilized for speed reconstruction is on the gearbox high speed stage, mainly due to the high energy content of the generated vibrations. However, numerous factors may complicate the process and jeopardize the accuracy of speed estimation. Some of the major challenges are listed below.

1. initial selection of the search bandwidth of the frequency of interest
2. identification of a known suitable speed related frequency, which is adequately represented within the available time interval and is dominant in the aforementioned frequency range
3. verification of results and redundancy in case of invalid signal dynamics of the selected accelerometers
4. guarantee of consistent results regardless the operating condition

The first three points are discussed in the following subsections, whereas the fifth is discussed in section 5, where the results are presented.

3.1. Selection of initial search frequency range

The core of speed estimation based on the maximum correlation coefficient method is the selection of the initial search range. If the frequency of interest is not within the specified bandwidth, the method yields invalid results introducing high uncertainty in the fault diagnosis procedure.

So, it is essential to select a relatively broad frequency range for the running speed regardless the operating conditions. A parameter which could assist on the bandwidth selection and is simultaneously available in all wind turbines is the instantaneous power production. Of course, this approach shall not be applied globally, but fine tuning is suggested for all gearbox types and ratios.

Figure 3 illustrates the mean running speed of approximately 70000 10.24 seconds files as function of the active power production from approximately 500 3.0 MW geared wind turbines with a gearbox ratio close to 1 : 110, operating from 2009 to present. In order to ensure that the power generation is relatively constant within the above mentioned time length, the condition that the speed variation of the acquired file is less than 1 Hz has to be met in the present application. In order to ensure that the power generation is relatively constant within the above mentioned time length, the condition that the speed variation of the acquired file is less than 1 Hz has to be met in the present application. Furthermore, any power values above 3.3 MW are neglected, so the cases where the turbine controller parameters are invalid, do not affect the distribution. The same applies for the recorded generator speeds over 35 Hz.

A wide range of running speed values applies for the same power production, fact which complicates the selection of the initial bandwidth. The latter must include all potential cases.
on one hand, but on the other hand a valid calculation has to be executed within the specified computational time.

If a wide dynamic range is considered for the initial estimation of the running speed, as shown in Figure 3 with green circles, then it can be guaranteed that the actual rotational speed is within this band rendering the method independent of any special considerations for each wind park or turbine. It can be observed in Figure 3 that by following the above described concept for the estimation of the initial speed, the number of outliers on the low power bins is minimized while keeping an acceptably low frequency band.

Figure 3. Raw data and dynamic frequency range for initial speed estimation.

### 3.2. Identification of speed related frequency

The speed related frequency component which serve as fingerprint for the identification of the high speed shaft (generator) running speed shall fulfil two prerequisites:

1. adequate representation within the specified time interval
2. it has to be dominant in the search bandwidth

First and foremost, it is mandatory to specify the rate of update of the running speed ($n_{HSS}$) estimation. For a geared high speed turbine, where $n_{HSS}$ varies roughly between $17Hz$ to $30Hz$, one shaft revolution corresponds to 0.03 to 0.06 seconds. Bearing in mind that the generator rotor inertia of a multi-megawatt scale turbine is high and that the wind conditions do not change dramatically within 1 second, a fair selection would be to update the speed value every 10 revolutions, namely 0.3 to 0.6 seconds.

The component which has been utilized by the vast majority of researchers is the gearbox high speed stage first tooth mesh frequency (1TMF) and its harmonics (Zimroz et al., 2010, 2011). This selection is fully justified based on the following facts. Assuming that the the high speed shaft speed is between $17Hz$ to $30Hz$ and that the high speed stage pinion has 35 teeth, 1TMF is approximately between $600Hz$ to $1000Hz$, as shown in 4. This means that in the worst case scenario, at least $180 (=0.3s \cdot 600Hz)$ cycles of the first tooth mesh frequency fit in the specified time interval offering the required representation of the selected frequency component. On the contrary, if the 1st running speed harmonic is used (approximately $17Hz$ to $30Hz$), then only 5 cycles correspond to 0.3 seconds which is considered insufficient.

Figure 4. Power spectrum of gearbox High Speed Stage Front Accelerometer. The first two harmonics of the High Speed and Intermediate Speed Stage tooth mesh frequencies are shown by arrows. The objective is to track HSS.1TMF in intervals of 0.3 seconds.

### 3.3. Validation of results and redundancy

In order to obtain an accurate estimation of the running speed, the result has to be verified so as to eliminate any invalid outcomes which will potentially produce unrealistic trends of the speed related frequencies. Furthermore, the robustness of the method depends on the presence of multiple sources of vibration data where a suitable frequency speed related component can be utilized for the extraction of the actual speed.

Figure 5 illustrates the drive train and the sensor location of a conventional geared turbine. The 3rd stage (also referred as High Speed Stage) of a gearbox is usually monitored by two sensors in order to identify faults on the meshing gears, such as broken teeth and excessive wear, and the bearings supporting the high speed shaft. It should be noted that the sensors installed close to the 2nd stage can be also utilized by detecting the corresponding 1st tooth mesh frequency. The consistency of the results depends highly on the presence of a clear vibration path between the 2nd and 3rd stages’ gears and the sensors’ location.

### 4. RUN OF ALGORITHM

In order to have a reliable running speed reconstruction which enhances the capabilities of the condition monitoring system when the speed sensor (tachometer) does not function prop-
In order to illustrate the functionality and robustness of the method, the speed variation is presented in Figure 7 as: 1) recorded from the speed sensor and 2) it is calculated from the High Speed Stage Front (HssFr) and High Speed Stage Rear (HssRr) accelerometers separately and 3) combining the two vibration sensors. The latter is labelled as "Virtual".

It can be seen that the speed values generated when tracing 17\( TMF \) using the HssRr accelerometer are invalid for Time > 20s, which is due to the presence of frequency components around 17\( TMF \) of comparable amplitude resulting to the occasional collapse of the method on them. If the speed estimation based on the HssRr accelerometer is taken into account, then the uncertainty regarding the correct value will be high. Therefore, this case shows the advantage of having two sources for the reconstruction of the speed signal and how temporary effects affect the tracking of the maximum correlation coefficient.

An important aspect is the performance of the method under large speed fluctuations. Figure 8 shows a case where the speed variation is 14.27\% or 3.65Hz (219rpm). In more details the turbine accelerates during the first 6.65s and slowly decelerates giving a variation of 33rpm/s for the first portion.

5. **Validation of Speed Estimation Method**

The method described in section 2.1 is tested in three drive train configurations where the number of main bearings and gearbox layout are different. The three topologies are:
Figure 6. Flowchart showing the basic steps of the proposed speed evaluation algorithm.

1. Two main bearings, three stage gearbox (one planetary and two helical – 1P2H)
2. One main bearing, three stage gearbox (two planetary and one helical – 2P1H)
3. One main bearing, four stage gearbox (three planetary and one helical – 3P1H)

The speed estimation algorithm is tested in 40 turbines per topology installed worldwide – 120 turbines in total. Six 10.24 second gearbox or generator vibration signals sampled at 25.6kHz are analyzed for each turbine. The time interval between two consecutive files is approximately two days, offering the possibility of investigating the efficiency of the method at different wind conditions. Furthermore, the aforementioned 120 turbines fulfil two conditions: 1) the actual speed signal is valid for comparison purposes and 2) the signals generated by the accelerometers mounted on the gearbox are valid.

The random selection of turbines ensures that the method is independent of the rotor diameter, turbine wind class and con-
controller settings as long as the same gearbox is installed on them. The illustrated figures in the following sections consist of three bars; the first two bars present the cases where only one source of vibration data is used, whereas the third bar corresponds to the combined utilization of two sensors and it serves as indicator of the robustness of the method. The acceptable error limit is selected to be 0.25Hz throughout the following analysis.

5.1. First topology – Three stage gearbox (1P2H)

The first gearbox layout consists of one slow rotating planetary and two helical stages. The utilized vibration signals are recorded by two accelerometers mounted on strategically selected positions adjacent to the high speed stage bearing housings in order to achieve optimum vibration path.

The used abbreviations in the following figures are HssFr (High Speed Stage Front), HssRr (High Speed Stage Rear) and Fr-Rr suggesting that the response of both accelerometers is accounted. It can be observed that although the success rate is above 90%, the percentage of valid estimations is relatively lower when only the High Speed Stage Front accelerometer is used. This behaviour is assessed to be linked to the presence of a mechanically driven oil pump located close to this sensor introducing high noise interference around the high speed stage 1TMF. Of course, the vibration signal from the High Speed Stage Rear sensor suffers also from noise but the influence is limited.

The invalid outcomes of the third column in Fig. 9 count for 1.25% of the test set which suggests that the algorithm yields incorrect estimations for 9 out of 720 files. The latter can be translated as that for the considered 10.24s, a frequency component close to the 1TMF is of higher amplitude and therefore the method collapses to this maximum. However, it should be underlined that the invalid results are not originated by the same turbine implying that temporary local phenomena influence the speed estimation.

Figures 10 and 11 shows the maximum absolute error distributions as function of the mean power production and the speed variation within 10.24s for the two sensors. The following remarks can be made:

- the method generates valid estimations even in the extreme cases where the speed variation is approximately 11% (approximately 300rpm) rendering it robust to large speed changes.
- the HssRr sensor presents less incorrect estimations in the high power range which is most likely associated with the operation and influence of the oil pump.
- the maximum absolute difference, when only the High Speed Stage Front is used, is approximately identical for all the incorrect estimations. This is again assessed to be directly linked to the oil pump operation.
5.2. Second topology – Three stage gearbox (2P1H)

The high speed stage of the second topology (two planetary and one helical stages) is also monitored by two sensors mounted adjacent to the high speed stage support bearings offering the capability of validating the speed estimation from two vibration sources. Fig. 12 presents the results for the error limits under examination. It is observed that the certainty level that at least one sensor yields consistent speed estimation approaches is in the range between 90 to 100%. The main sources of interference which have resulted to invalid results for this gearbox type are:

- increased sideband activity due to misaligned/eccentric gears, issues such as macropitting, stand-still marks, cracked and broken teeth or bearing defects (located mainly at the inner race)
- low amplitude of the 1TMF, which is assessed to be related to the vibration signature of the gearbox and not to a specific failure mode

![Figure 12. Validation of speed estimation for the 2\textsuperscript{nd} topology based on absolute error.](image)

As in the first topology, the speed estimation algorithm generates reliable results for both low and high speed variations. Furthermore, no straight correlation was observed between the power production and the number of invalid results fact, showing that any mismatches are random.

5.3. Third topology – Four stage gearbox (3P1H)

The third gearbox topology consists of four stages in total, three planetary and one helical. The maximum absolute errors generated by the two accelerometers monitoring the high speed stage are presented in Figure 13. The third bar shows the results when both vibration sensors are used to cross-reference the speed estimation; it is reminded that the third bar presents the certainty level that the algorithm can identify a correct or incorrect result.

![Figure 13. Validation of speed estimation for the 3\textsuperscript{rd} topology using the high speed stage gearbox vibration sensors based on absolute error.](image)

The following observations can be made:

- the speed estimations from the HssFr sensor are considerably poorer compared to the HssRr one, which is assessed to be related to the construction of the gearbox and location of the sensor. For this gearbox, a clear vibration path applies not only from the helical stage to the sensor but also from the last planetary stage to it.
- although the speed estimation success rate from each sensor individually is low, the valid results when both are accounted are acceptable. This implies that only for approximately 3 – 4% of the files the algorithm regards an invalid result as valid.

6. CONCLUSIONS

The present paper presents the application of the maximum correlation coefficient on the estimation of the high speed shaft speed utilizing two sources of vibration data for validation and cross referencing purposes. Three drive train topologies and gearbox layouts are studied in this work, where 720 files in total serve as test sample. Statistical analysis reveals that the technique offers reliable results for more than 98% of the cases when the performed analysis is based on two accelerometers. The success rate is lower, close to 90%, when the speed estimation algorithm input is only one vibration signal, emphasizing the necessity of validating the outcome. In specific, in one of the gearbox configurations, the presence of auxiliary equipment, i.e. a mechanically driven oil pump, influences the speed estimation performance based on the accelerometer located closer to it. In the third topology, the presence of strong vibration path between one of the accelerometer under consideration and more than one gearbox stages, results in incorrect speed estimations for approximately 30% of the cases, which again supports the employment of two vibration sources. Finally, it has not been ob-
served any direct dependency between the algorithm’s efficiency and large speed variations or power output dependency, rendering the method robust.

**References**


**Biographies**

**Georgios Alexandros Skrimpas** received the Diploma in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2009 and the M. Sc. in wind energy from the Technical University of Denmark (DTU) in 2012. He joined Brüel and Kjær Vibro in 2012 and since 2013 he is pursuing the Industrial Ph.D. degree at the Centre of Electric Power and Energy at DTU in cooperation with Brüel and Kjær Vibro. His research interests are diagnosis and prognosis of electrical and mechanical faults in wind turbines.

**Kun S. Marhadi** is an engineer in the Remote Monitoring Group at Brüel and Kjær Vibro. He joined Brüel and Kjær Vibro in 2012. Previously, he worked as a postdoctoral fellow in the Department of Mathematics at the Technical University of Denmark (DTU). He received his PhD in computational science in 2010 from San Diego State University and Claremont Graduate University. He has M.S. and B.S. in aerospace engineering from Texas A&M University. His expertise is in structural vibration and analyses, probabilistic methods, and design optimization.

**Bogi Bech Jensen** received the Ph.D. degree from Newcastle University, Newcastle Upon Tyne, U.K., for his work on induction machine design. He was in various engineering and academic positions in the marine sector from 1994 to 2004.
He was at Newcastle University from 2004 to 2010 first as a Postgraduate, then Research Associate and finally as a Lecturer. From 2010 to 2014 he was Associate Professor and later Head of Research Group at the Technical University of Denmark (DTU), Lyngby, Denmark. He is currently Professor of Energy Engineering at the University of the Faroe Islands (UFI), where he is responsible for education and research in energy. Prof. Jensen is Associate Editor of IEEE Transactions on Industry Applications and a Senior Member of IEEE.

Christian Walsted Sweeney received his B.Sc. degree from the University of Southern Denmark in 2006 and M.Sc degree from the Technical University of Denmark in 2008 both in mechanical engineering. From 2008 to 2010 he was employed by Bruel and Kjaer Vibro as diagnostic engineer and since 2010 he is the team leader of the diagnostic services group. His research focus is on the development of condition monitoring systems and handling of large data quantities.

Nenad Mijatovic received his Ph.D. degree from the Technical University of Denmark for his work in superconducting machine. After obtaining his Dipl.Eng. education at University of Belgrade, Serbia, he enrolled as a doctoral candidate in 2012. Upon completion of the PhD, he has continued to work in the same field of machine research - superconducting machines, as an Industrial PostDoc. The 3 year industrial PostDoc grant has been provided by Hojteknologifonden and supported by Envision Energy Aps., Denmark. Dr. N. Mijatovic is a member of IEEE from 2008 and his field of interest and research includes novel electrical machine design, operations and diagnostic.

Joachim Holboell is associate professor and deputy head of center at DTU, Department of Electrical Engineering, Center for Electric Power and Energy. His main field of research is high voltage components, their properties, condition and broadband performance, including insulation systems performance under AC, DC and transients. Focus is also on wind turbine technology and future power grid applications of components. J. Holboell is Senior Member of IEEE.
Sliding Wear Particle Mass Distribution Assessment for Wear Mode Diagnosis

Surapol Raadnui

Department of Production Engineering, Faculty of Engineering
King Mongkut’s University of Technology North Bangkok (KMUTNB)
1518, Pracharaj 1 Road, Bang-Sue, District, Bangkok
Postal Code 10800, Bangkok, Thailand
Email address: raadnui_s@yahoo.com, s_raadnui@yahoo.co.uk

Abstract
Condition based maintenance has been coming to the fore especially in recent decades and it is at the expense of conventional maintenance strategies. Wear particle tribology-based predictive maintenance is based on continuous monitoring, evaluating its condition and uses knowledge of technical diagnostics and prognostics. The use of sliding wear particle mass distribution as a means for distinguishing different modes of wear and determining its transitory behavior is evaluated in terms of the quantitative analysis of the multi-filtergram slides produced from a series of wear tests from a multiple point contact sliding wear tester. In this particular research, a four ball machine was used throughout. At the end of each test the wear debris generated was collected and then separated using a multi-filtergram maker which resulted in the wear debris being extracted due to their specific size ranges. Each filtergram patch of each specific size range was subsequently weighed to obtain “wear particle mass distribution” which in turn can be used to produce a histogram plot of the particle mass distribution. Various distribution functions have been fitted with the data. The results obtained from different sliding wear modes are presented; they confirm that changes in the mean and the variance of the selected statistical distributions provide clear indications of the type and extent of the sliding wear as it progresses.

1. Introduction
The introduction of ferrograph in the 1970’s allows wear debris to be studied in detail (Bowden and Westcott (1976) and Seifert and Westcott (1972)). It was suggested that examination of the wear debris produced by a tribo-system would allow the wear mode/mechanism operating to be established. It was generally accepted that each wear mechanism produced typical characteristic wear debris morphology. The use of the ferrograph was refined until it could be successfully used as one of a condition monitoring tools for oil and grease lubricated machinery. Recent development in the field of wear debris extraction/separation result in the introduction of the multi-filtergram patch maker which can be used to extract “total” solid debris, rather than ferrous debris as normally done by the ferrograph, from used lubricant samples into specific size range. During the original work with ferrography only ferrous wear debris can be extracted into different size range along the ferrogram slide. It was therefore, in this particular study, decided to reproduce mild-severe sliding wear debris on a four ball machine to establish whether wear debris mass distribution can be used to diagnose/prognosis the wear generating mode. It was considered that the identification of a wear debris mass distribution characteristic for mild-severe scuffing would be beneficial to that using wear debris analysis as a tool for condition diagnostic/prognostic monitoring technique.

2. Experiments
The four ball machine was used to perform the tests throughout. The relevant feature of the machine is four 12.7 mm, diameter AISI 52100 ball bearings which are arrange in the form of equilateral tetrahedron. The three lower balls which form the base of the tetrahedron are held stationary which the top ball is free to rotate at 1470 rpm under a fixed specific load at 40 kgf. Dry sliding wear tests were conducted throughout. The duration of each test was varied between 3 to 30 minutes. At the end of each test the wear debris generated was thoroughly rinsed with heptane and subsequently collected using a multi-filtergram maker which resulted in the wear debris being extracted due to their specific size ranges. Each filtergram patch of each specific size range was weighed to obtain “wear particle mass...
distribution” which in turn can be used to produce a histogram plot of the particle mass distribution. The three lower balls were collected and cleaned. Wear scar diameter of each lower ball was measured and micrograph was taken by an optical microscope. A new protocol for solid debris extraction from used lubricating oil is proposed (Raadnui (2011)). The Particle Separating Disk (PSD) is designed to separate solid particles from used oil samples for viewing under a microscope (Raadnui (2011 and 2012)). The PSD system for particle processing, more specifically particle separation based upon particle size. The PSD system includes a processing module that includes a plurality of separation ports, each separation port within the plurality of separation ports configured to receive samples including particles. The system enables multiple samples to be simultaneously processed during operation, more specifically centrifugation, of the processing module. Each separation port is fluidly communicable with a space (e.g. sample collection space) defined by a housing module of the device. Introduction of samples into the plurality of separation ports can be controlled. Residual sample received or collected by the housing module can be removed or drained from the housing module during centrifugation of the processing module. A typical PSD device is shown in Figure 1. A simple centrifuge unit, centrifuges the diluted sample through a set of filter patches and dries them quickly to allow for immediate examination. One oil sample provides multiple patches (large particles, medium size particles & small size solid particles). In addition multiple oil samples can be processed simultaneously as shown in Figure 2. This is accomplished with the use of centrifugal force for solid particle separation. The Relative Centrifugal Force (RCF) can be calculated from the expression:

$$ RCF = \omega^2 r g \text{……………………………….}(1) $$

$\omega$ is the angular velocity, $r$ is the radial distance, and $g$ is the acceleration due to gravity. Generally, it is inconvenient to measure the angular velocity ($\omega$), and so it is more convenient to express the RCF in terms of revolutions per minute (rpm), $N$, and this gives the expression:

$$ RCF = 11.18 r [N/1000]^2 \text{…………………………………………………}(2) $$

The centrifugal force is usually given in terms of ‘$g$’ and is written as such or as ‘$xg$’. From equation (2), it can be seen that the centrifugal force acting on the particles is related to the square of the speed and hence doubling the speed increases the centrifugal force by a factor of four. The centrifugal force also increases with the distance from the axis of rotation ($r$). Hence particles in a homogeneous medium will accelerate as the radial distance increases. A typical procedure for the using of PSD in solid debris separation process is shown in Figures 3 and 4.
3. RESULTS AND DISCUSSION

A good number of methods are well developed for the separation of solid debris from used oil samples. The selection of the best suitable method depends on the result desired. For example, if the interest is on only ferrous wear particles, one of the best separation methods by ferrograph must be selected. However, for this specific research work, the main objective is on the total surveys about the different materials which normally encounter in the real world applications, filtration and/or centrifugation processes are generally preferred. The separation by filtration can follow two different goals. One is to determine the total amount or total mass of solid particles in used oil sample. The other serves for information about the “morphological characteristics” of individual particles. A great advantage of the filtration technique is to separate particles into different size ranges. For this purpose, in this work though, the first step is a filtration of the greater particle size over 100 µm. the second step is a filtration over 40 µm. The third step is a filtration over 0.45 µm. As a result the range lies between 0.45 -40, 40-100 and over 100 µm. This is generally called “fractionated filtration” or the multi - filtergram patch maker is possible in every range were filters are available in different pore sizes. Consequently, the filter patches can then be weighed with a precision scale (Sartorius LE Pro Analytical Balances (100g x 0.01mg)) for the quantity or mass variation due to each specific sized range. Typical patches are shown in Figures 5 to 7 below.
The investigation of the mass distribution of the particles from each specific size range is the second step after their separation. In all cases, investigation means to look at the quantity or weight of each specific size range of the captured particles. Figure 8 shows the two and three dimension histogram plots of the wear debris mass distribution from all the sliding wear test duration between 3 to 30 minutes.

Figures 9 and 10 illustrate the correlation between the mean wear scars diameters of the three lower balls from the series of test conducted. A fair correlation between the calculated wear volume and total weight gain from the multi-filtergram patch produced was achieved where $R^2$ was 0.9130 or 91.30%. Typical wear scar from dry sliding four ball wear tests are shown in Figures 11 (a) to 11 (c).
All raw data from each consecutive dry sliding four ball wear tests were fitted by multiple statistical distributions, namely, exponential distribution, 2-parameter exponential distribution, larger extreme value distribution and normal distribution. Typical probability paper plots are shown in Figure 12. Table 1 summarizes the governing parameters for each specific statistical distribution. It can be clearly seen for the less severe wear mode, i.e. 3 and 10 minutes test duration, transition severe wear mode at 10 minutes test duration and the more severe wear mode, i.e. the test duration higher than 15 minutes have quite a distinctive governing parameters whatever the distribution are. The lower value of governing parameters may imply to the less severe sliding wear mode and the higher values reflects for the higher severe wear mode, although it may be applied for only this particular work. The statistical analysis results seem to correlate well with physical feature of the lower ball wear scar diameters. Typical wear scar diameters are shown in Figures 11 (a) to (c). In addition, examination of the wear debris from multipatch filtergram revealed two basic type of debris (the less severe and more severe patterns from wear debris morphology). The first type of wear debris is as shown in Figure 7 and consists of thin platelet of ferrous wear particles with a size of approximately 20-30 µm. It is generally known as rubbing wear debris which is similar to that describe in the Wear Particle Atlas (Bowden and Westcott (1976)). It is the most commonly found debris in normal lubricated machinery component. The second type of wear debris found is shown in Figures 5 and 6. The debris size could be up to 300 µm. The debris surface texture showed evidence of striation and discolor feature from high temperature mode of wear.

Figure 11. (a) wear scar diameter at 3 minute test duration

Figure 11. (b) wear scar diameter at 15 minute test duration

Figure 11. (c) wear scar diameter at 30 minute test duration

Figure 11. Typical wear scar diameter of lower balls from 4-ball dry sliding wear tests

Figure 12. 2-parameter exponential probability paper plot for mass distribution from 4-ball sliding tests
Table 1. Statistical mass distribution analysis of sliding four ball test data

<table>
<thead>
<tr>
<th>Test duration (minutes)</th>
<th>Exponential</th>
<th>Normal</th>
<th>Largest extreme (Gumbel Distribution)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Scale</td>
<td>Threshold</td>
<td>Mean SD</td>
</tr>
<tr>
<td>3*</td>
<td>0.8333</td>
<td>0.7343</td>
<td>0.8333</td>
</tr>
<tr>
<td></td>
<td>0.099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10**</td>
<td>0.9333</td>
<td>0.8343</td>
<td>0.9333</td>
</tr>
<tr>
<td></td>
<td>0.099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15***</td>
<td>2.4670</td>
<td>2.368</td>
<td>2.4670</td>
</tr>
<tr>
<td></td>
<td>0.099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20***</td>
<td>5.4000</td>
<td>3.816</td>
<td>5.4000</td>
</tr>
<tr>
<td></td>
<td>1.584</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30***</td>
<td>10.830</td>
<td>7.764</td>
<td>10.830</td>
</tr>
<tr>
<td></td>
<td>3.069</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*: less severe sliding wear mode, **: transition mode, ***: high severe sliding wear mode

From Figures 12 and Table 1, in which it is seen that an increasing in the statistical governing parameters (mean, standard deviation, location parameter, scale parameter, threshold value) occurs in the transition region. This is associated with the appearance of thin flat platelets in the less severe form of sliding wear mode, typically less than 40 µm in size. In the more severe sliding wear mode condition, the indications from visual inspection of the particles are that the thin platelets have been replaced by a larger severe sliding particle while the mass of smaller particles has increased. This intensity activity produces a particle population which extends over a larger size range but also contains many small particles which have been broken down in size either as a result of their generation process or during subsequent wear processing. Anderson-Darling statistic (AD) was used to measure the area between the fitted line (based on chosen distribution) and the nonparametric step function (based on the plot points). More precisely, the Anderson-Darling statistic is a squared distance that is weighted more heavily in the tails of the distribution. Smaller Anderson-Darling values indicate that the distribution fits the data better.

4. CONCLUSION

The mass distribution of particles extracted by a new protocol have been determined for different wear mode associated with dry sliding four ball wear tests. Several statistical distributions, namely; Normal, Exponential, 2-parameter Exponential and Largest Extreme Value Distribution, provide good fits to the mass distribution data with 95% confidence interval. The variation in the distribution characteristics i.e. mean, standard deviation, scale parameter, shape parameter, threshold value confirms that they are importance indicators of changing in the type and extend of wear mode. An increasing in the mean particle mass from about 10 minutes test duration resulted during a transition period from less severe sliding wear to the more severe sliding wear mode. A transition from the less severe to the higher severe sliding wear mode was accompanied by a similar increase in the governing parameters from each specific statistical distribution. An increase in mass distribution of wear particles (in the other words, “the wear rate”) was marked by a corresponding increase in the number and size range, hence the mass distribution, being generated, and the latter resulting in an increasing in the variance of the distributions. The particle mass distributions covered by this technique are mainly in the range 0.45 to over 100 µm, and this is particularly relevant to the development of a diagnostic/prognostic approach to wear monitoring because particles within this range are generated at all stages of the wear process. Thus the means exist for determining the wear mode and monitoring the progress of wear by quantitative (mass distribution) analysis of the particles. What appears to be lacking is the appropriate other wear mechanisms information associated with each mode which is necessary for the establishment of a realistic prediction of life expectancy. A detailed study of the specific wear mechanism may prove to be rewarding in this respect.

To summarize:

1. Less severe sliding wear mode produces small to medium size platelet up to approximately 50 µm.
2. In a more severe sliding wear mode generates larger size wear particles which have size range up to 200-300 µm.
3. Debris mass distribution has a potential to be used as a tool to elucidate the wear mode severity through the application of statistical analysis.

ACKNOWLEDGEMENT

The work reported here is funded by the Thailand Research Fund (TRF), grant no. RMU-5380033.

REFERENCES

Raadnui, S. (2012). Between “ferrogram” and “filtergram” makers is the centrifiltergram maker: a new technique


**Biography**

Surapol Raadnui had his Ph.D. from the University College of Swansea, U.K., in 1995. He received his Bachelor Degree (B.Sc. – Industrial Engineering) from Prince of Songkhla University in 1985 and subsequently went on to get his Master Degree (M.E. – Industrial Engineering) from Chulalongkorn University in 1989. He worked for six years as a Maintenance Engineer in the Royal Thai Naval Dockyard before taking up a position with King Mongkut’s University of Technology North Bangkok (KMUTNB) as a lecturer. His research interests are: Fundamental and Applied Tribology, Used Lubricant Analysis, Wear Particle Tribology, Proactive Maintenance (Contamination Control) and Maintenance Management. He has contributed 60 international publications and three international patents.
Poster Papers
Creep Mechanisms vis-à-vis Power Law vs. Grain Boundary Sliding in α-β Titanium Alloys for Physics Based Prognostics

Amar Kumar1, Alka Srivastava1, Nita Goel1, Avi Banerjee2 and Ashok K Koul2

1Tecsis Corporation, 210 Colonnade Road, Ottawa, ON, K2E 7L5, Canada
2Life Prediction Technologies Inc., 23-1010 Polytek Street, Ottawa, ON, K1J 9J1, Canada

ABSTRACT

This work is performed in support of our continued physics-based prognostics system development using a life cycle management-expert system (LCM-ES) framework. The physical damage model developed involves global behavior and localized response of a component at the microstructural level. The current research aims at constructing parts of a deformation mechanism map (DMM) for α-β Ti alloy. The appropriate constitutive equations are used for power law creep and grain boundary sliding mechanisms. Simulations are performed using the Newton-Raphson method using Matlab software code in order to obtain contour lines corresponding to strain rates ranging from 10^{-8} to 10^{-12} over the homologous temperature ranges of 0.10 to 0.655. The dominance of power law creep and grain boundary sliding over a wider range of stresses and temperatures in Ti-64 alloy is studied. The simulation results are validated using experimental data points. The predicted contour lines in the map match fairly well. The structure-creep mechanism relationships in α-β Ti alloy under different stress, temperature and strain rate conditions are discussed.

1. INTRODUCTION

Creep deformation involving significant plastic strain accumulation in titanium alloys is the single major cause for compressor and turbine failure. It is known to occur by a number of alternative and competitive mechanisms. Creep failure resistance remains a big challenge for all high temperature design and service considerations. The problem has assumed greater significance in titanium alloys for physics based approach and analysis for the development of diagnostic prognostic and health management systems. Titanium alloys are used in a wide variety of microstructures and properties and are subjected to wide ranges of operating conditions in aeroengines (Badea et al., 2014; Evans and Harrison, 1983; Seco and Irisarri, 2001). Deformation mechanism maps (DMM) summarize information regarding dominance of creep plasticity mechanisms in temperature and stress space varying as a function of strain rate in a given microstructure. Both creep mechanism and the strain rate are the major variables for assessing the rate of damage accumulation in physics based prognostic analysis. The DMMs are constructed with axes of normalized shear stress σ/μ (μ being the shear modulus) and T/Tm where Tm is the melting temperature of the alloy. This space is divided into various fields which show the regions of stress and temperature over which each deformation mechanism is dominant. (Frost and Ashby, 1982). The dominant deformation mechanisms include dislocation glide, dislocation creep and Harper-Dorn creep, power law creep (dislocation glide plus climb) and diffusion creep mechanisms, i.e., Nabarro–Herring creep and Coble creep (Frost and Ashby, 1982; Langdon, 2006; Janghorban and Esmaeili, 1991). This understanding of constructing DMMs arose from using experimental data and microstructural evidence collected on simple metals and alloys. The applicability of these concepts to complex engineering alloys is now being questioned. Instead of diffusion creep mechanisms, grain boundary sliding accommodated by different thermally activated processes is considered likely as a dominant deformation mechanism in complex engineering alloys (Janghorban and Esmaeili, 1991; Briquet et al., 2012; Bano, Koul and Nganbe, 2014).

In physics-based prognostics, use of physically realistic damage models is a prerequisite for accurate life prediction of cold and hot section turbine components. A life cycle management-expert system (LCM-ES) framework involves the integration of both global behavior and localized response at the microstructural level (Banerjee et al., 2013). Microstructural variability in a damage model was considered earlier for the prediction of the component reliability upfront (Banerjee et al., 2013; Metzer and Seifert, 2012).
The primary objective of this paper is to construct a DMM for an α-β Ti alloy showing the dominance of power law creep (PLC) and grain boundary sliding (GBS) over a range of stresses and temperatures. The work emphasizes the importance of GBS mechanism leading to grain boundary cavitation and wedge cracking. The commonly used constitutive rate equations and a newly modified Matlab soft code are considered for the reconstruction of the PLC and GBS regimes. The other interest of the work is to identify the power law deviation separated from the region of power law dominance. The theoretically reconstructed DMM is further validated with experimental data points available in the literatures.

2. α-β Ti Alloy Systems
The IMI 834 is one of the latest near-α titanium alloys developed for compressor discs, blades and vanes of modern aircraft jet engines. High-temperature creep resistance is one of the prime requirements of IMI 834 since the alloy is targeted for application up to about 600°C (T/Tm = 0.45). The desirable microstructure of this alloy for compressor application with good combination of fatigue and creep resistance has been found to be about 15% equiaxed primary α phase embedded in transformed β matrix. The aeroengine components operating at high temperatures are prone to creep failure due to high sustained rotational speeds. Also, the components during service are frequently subjected to varying stresses and temperatures. Thus these become susceptible to localized creep or low cycle fatigue (LCF) damage, which leads to initiation of cracks early in life. Modern aircraft engines have portions of compressor that operate in the temperature range of 325°C (T/Tm = 0.31) to 575°C (T/Tm = 0.44) and have Ti alloys for the stressed rotors (Es-Soumi, 2001, Lutjering and Williams, 2007; Donachie, 2002).

The Ti-6Al-4V (Ti-64) is the most widely used and studied alloy. The aluminum stabilizes the alpha phase while the vanadium stabilizes the beta phase. These alloys can be solution treated, quenched and aged for increased strength. The microstructure of the alloy depends on its composition and heat treatment. Creep strength is not as good as most near Alpha alloys. The properties of Ti-64 vary smoothly with increasing temperature, covering the range from minus 196°C up to 750°C. Although the alloy retains useful short-term properties up to 500°C, but its properties over the longer term tend to limit its use to 300°C. The LCF behavior for Ti-64 is highly structure dependent as beta forged has the least resistance, while 10% alpha–beta structure shows the highest resistance (Leyens and Peters, 2003; Peters et. al., 2003).

3. Creep Deformation Mechanisms
A brief account of creep mechanisms considered in the work is discussed in order to highlight their importance and implications. The generic creep mechanisms in Ti alloys covers different sections of the map as separated by the boundaries. Fig. 1 is reproduced here from earlier work for illustration (Janghorban and Esmaeili, 1991). The whole map is subdivided into various regime where the dominant mechanism prevails as a function of stress and temperature. The diffusional creep as well as power law creep are shown to be prevailing over 0 < T/Tm < 0.6, however the PLC appears to be at much higher stress level. The strain rate contour lines go over different regimes depending on the stress applied and temperatures. However, earlier work did not consider the GBS mechanism.

3.1. Power Law Creep
The rate at which the dislocations can overcome a series of obstacles within the crystal determine the glide/climb rate and consequently strain accumulation (Wu, 2010). At higher temperature, movement of the dislocations become easier. Thermal activation enables edge dislocations to climb from one glide plane to another by either absorbing or emitting vacancies at a sufficient rate to bypass the obstacles. A large number of models have been proposed for PLC (Leyens and Peters, 2003; Peters et. al., 2003). The generic form of power law creep (PLC) is represented by

\[ \varepsilon_{ss} = A\sigma^n \exp \left( \frac{-Q}{RT} \right) \]

(1)

In general, in stress dependence ‘n’ is termed as stress exponent, \( \sigma \) is the stress and \( \varepsilon_{ss} \) is the creep strain rate. In temperature dependency, Q is the activation energy, R is the gas constant, T is temperature and A is the material dependent constant. Prediction of the value of n from first principles is not easy, but its value does depend on which mechanism is operating. Uncertainties lie with selection of appropriate values of apparent activation energy (Q) and stress exponents (n). High values of Q and n have led to confusion about rate controlling mechanisms and hence a number of models have been proposed. These anomalies have been reported for various alloy systems including titanium (Barboza et. al., 2006). However, several works on these materials have shown that the anomalous behavior can be rationalized by considering the presence of a threshold stress (\( \sigma_0 \)) opposing creep flow and the creep rate is related to an effective stress (\( \sigma - \sigma_0 \)), where \( \sigma \) is the applied stress (Badea et. al., 2014; Barboza et. al. 2006). For diffusion creep the value of exponent is approximately 1, while for deformation governed by transgranular dislocation glide/climb it is usually in the range 3-8. At high temperature, the supply of vacancies for the climb of dislocation is assumed to occur through volume diffusion resulting in values for the stress exponent, n, ranging from 3 to 5. At lower temperatures vacancy diffuses predominantly along the dislocation cores. The rate at which vacancies are supplied to or removed from the climbing dislocations is then dependent on the dislocation density.
which is itself a function of the applied stress. Stress exponents of 5 to 8 are then predicted (Wu, Koul, 1999; Wu, Dickson and Koul, 1998; Wu, 2010). Unusual higher values greater than 15 are also reported (Evans and Harrison, 1983).

Power law dominated creep was reported in near α Ti duplex microstructures with 17% primary α for IMI 834 and fully transformed microstructures for IMI 829 and 685. Stress exponents are reported to be 4.2 to 5.2 for the steady-state and from 3.2 to 6.2 for the transient creep rates. The activation energy for the alloys are in the range of 300 to 345 kJ/mol. (Es-soumi 2001).

3.2. Grain Boundary Sliding (GBS)

At high temperature, grain boundaries tend to slide to cause creep deformation. Langdon developed a rate equation for GBS taking into account the movement of dislocations within, or adjacent to, boundary planes, through a combination of dislocation glide and climb steps (Langdon; 2006). The rate of sliding was considered to be controlled by the rate of accommodation through intragranular slip. The model showed good agreement with experimental results under creep and superplastic deformation conditions. Wu and Koul (Wu 2010; Wu and Koul, 1995) modified the Langdon’s GBS model to develop a constitutive rate equation for GBS in the presence of grain boundary precipitates by incorporating physically defined back stresses opposing dislocation glide and climb. They further concluded that the grain size dependence of the creep rate was grain boundary precipitate distribution dependent. They refined their theoretical model to consider GBS at serrated grain boundaries, using the dynamics of grain boundary dislocation pile-ups, by averaging the sliding rate over the characteristic dimensions of grain boundary serrations. Grain boundary sliding can be accommodated by two mechanism, namely by dislocation activity and by diffusion (Wu, 2010; Wu and Koul, 1995; Xu, et. al., 1999; Wu, Dickson and Koul, 1998).

4. DMM Construction and Analysis

A modified DMM is constructed between normalized stress (tensile stress /elastic modulus) as the y-axis and homologous temperature (T/Tm, Tm is melting temperature) as the x-axis. A Matlab code developed for this work is used for the construction of Ti alloy DMM. Fig. 2 display the part of the DMM of our interest.

The data used is given in a separate section (table 1). In view of the application range of the alloy, our interest is in the T/Tm limit up to 0.65 and so the diffusional creep mechanism like bulk (Nabbarro-Herring) and boundary (Coble) diffusions and Harper-Dorn creep are not considered (Fig.1). As indicated earlier, two separate modules are considered and discussed separately. Plots and results under the following modules are integrated finally.

a) Power law creep (PLC) and
b) Grain boundary sliding (GBS)

The strain rate range considered here varies from 10^{-12} to 10^{14} while the homologous temperature range considered is from

Figure 1. Illustration of dominance of various creep mechanisms in Ti-6Al alloy of grain size 100 microns (Janghorban and Esmaeili, 1991).

Figure 2. PLC-GBS regime of DMM for Titanium alloy as emerged out of the present simulation work. Symbols (in red) are the experimental data points from literatures [Badea et. al, 2014; Evans and Harrison, 1983; Nono et. al., 2006; Barboza, Neto, Silva, 2004]). Symbols (in blue) are the power law deviation data points as computed from experimental data.
0.10 to 0.65. The higher temperature regime above \( T/T_m = 0.4 \) is dominated by GBS mechanisms and depending on stress and temperature, IG creep fracture occurs by nucleation, growth and linking of voids. Two cavitation damage and cracking mechanisms are generally observed and known as wedge (W) type and round/elliptical (r) type (Campbell, 2012). The distinction between the two are primarily based on geometrical shape. W-type cracks initiate along grain boundaries aligned in shear, while r-type form along grain boundaries in tension. W-types are associated with cracks at triple points due to GBS. W-types tend to form at relatively higher stress, lower temperature and larger grain size. Though no specific temperature-stress regimes are specified in literatures in view of a large number of influencing factors, an arbitrary demarcation line is chosen here at \( T/T_m = 0.50 \) to distinguish between two dominating creep cavitation damages under GBS.

4.1. Ideal Shear Strength

Ideal strength of solids or upper limit varies widely depending on interatomic bond strength, modulus, temperature etc. Theoretical strength of solids is defined as the ultimate strength beyond which plastic deformation, fracture, or decohesion would occur. A number of theoretical models are developed to explain the ideal strength of materials. Some of models displayed enormous differences with the actual strength of materials, like Frenkel’s and Orowan’s models (Frenkel, 1926; Orowan, 1948). In our analysis, in DMM a line at the top is always marked to indicate a region where ideal materials collapse. This line is shown in Fig. 2 at normalized value of \( 7 \times 10^{-3} \) on y-axis.

4.2. Newton-Raphson Simulation

Newton-Raphson method for solving equations numerically is used in this work. It is based on the simple idea of linear approximation with a starting root value of \( x \). The method is a powerful iterative process and follow a set guideline to approximate one root of a function of the form \( f(x) = 0 \). An initial guess for the root we are trying to find needs to be assumed first, and we call this as initial guess, \( x \). A large number of simulation studies performed with various values of \( x \), strain rates and temperature and iteration number in order to arrive at a converged value of normalized stress on y-axis. The normalized stress is estimated as the ratio of applied tensile stress to elastic modulus.

In our approach, we have focussed on simulations with two differential equations and these are for PLC and GBS as given in Table 1. The parameters and the values considered are also shown.

4.3. Power Law Creep (PLC) Module

Simulation work is performed using Newton Raphson method for the power law creep using the respective constitutive equations as given in Table 1. Small adjustment in the assumed values of \( x \) was required in some of the strain rate contour lines. Simulations were carried out for each strain rate from the starting \( T/T_m \) as may be seen in Fig. 2. Strain rates from \( 10^4 \) to \( 10^{-2} \) are considered in this work. The PLC line for strain maximum lies at the top, while the line for lowest strain rate lies at the bottom.

4.4. Grain Boundary Sliding Module

Major plastic deformation mechanisms at medium temperature (0.4) and above are considered to be slip, grain boundary sliding (GBS) and diffusional creep. These tend to occur simultaneously or sequentially and the damage is accumulative. Slip and diffusional creep generally results under higher and lower stress levels respectively, but GBS contributes to plastic deformation at all grain size alloys (Wadsworth et al., 1999). At finer grain sizes, GBS tends to be the dominant ones. GBS accommodation can occur by the glide or climb of dislocation. A GBS model for creep at elevated temperature is introduced in which accommodation by slip is controlled by lattice diffusion. The simple form of GBS dependency on stress level, grain size and diffusion rate is shown below (Wadsworth et al., 1999).

\[
\dot{\varepsilon} = A \left( \frac{D_b}{d^2} \right) \sigma^2 \quad \text{for GBS by lattice controlled and}
\]

\[
\dot{\varepsilon} = A \left( \frac{D_{gb}}{d^3} \right) \sigma^2 \quad \text{GBS by grain boundary controlled}
\]

where \( d \) is the grain size, \( D \) is diffusion rate, \( \sigma \) is the stress.

4.5. PLC-GBS Transition

By equating the constitutive equations as given in Table 1 for PLC and grain boundary sliding (GBS) mechanisms, a transition line between the two mechanisms, namely PLC and GBS are outlined as shown in Fig.2. As may be seen, \( T/T_m \) above 0.40, the GBS regime is further subdivided into two, namely W type and R type cracking mechanisms. It is assumed in this work that R-type void formation begins at \( T/T_m \) greater than 0.50 (Fig. 2).

5. INTEGRATED DMM

The deformation modules as discussed in earlier sections need to be integrated to produce the complete deformation mechanism map for Ti alloys covering wide ranges of stress, temperature and strain rates. The modules integrated DMM is displayed in Fig. 2. Elastic range is typically assumed to dominate below \( T/T_m = 0.25 \), while failures dominated by SCC, corrosion fatigue etc., expected to fall over \( 0.25 < T/T_m < 0.40 \). As may be seen in Fig. 2, GBS mechanisms dominates above \( T/T_m \) greater than 0.40.

5.1. Alloy Data for Ti-64

Material data from several sources is used in this work for titanium alloy system and this is tabulated in Table 1. These data are shown in respect of mechanisms considered for Ti-
alloy components, namely power law creep and grain boundary sliding. (Janghorban and Esmaeili, 1991; Bano, Koul and Nganbe, 2014). The table also show the constitutive boundary sliding. (Janghorban and Esmaeili, 1991; Bano, al. et. al., 2014, Barboza et.al., 2004, Seco et. al., 2001). A recent relevant creep deformation data for the titanium alloy (Badea et. al., 2006, Ro et. al 1989, Chakraborty et. al, 2001). The data used for each of the module mechanisms.

### Table 1: Ti alloy data for Deformation Mechanism Map

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power law creep (climb), $\dot{\varepsilon} = A_{PLC} \frac{\sigma_{clim} \exp(\frac{Q_{clim}}{kT})}{E}$</td>
<td></td>
</tr>
<tr>
<td>Stress exp. $n$</td>
<td>4.3</td>
</tr>
<tr>
<td>Dorn constant, $A_{PLC}$</td>
<td>2.382 x 10$^{18}$</td>
</tr>
<tr>
<td>Act. Energy for vol. diffusion, $Q_{v}$</td>
<td>501 KJ/mol</td>
</tr>
<tr>
<td>Pre exp constant, $D_{av}$</td>
<td>3.26 x 10$^{-4}$</td>
</tr>
<tr>
<td>Power law breakdown $(\sigma_{plc}, E_o)$</td>
<td>3.5 x 10$^{-3}$</td>
</tr>
<tr>
<td>Grain Boundary sliding, $\dot{\varepsilon}<em>{GBS} = A</em>{GBS} \frac{D_{gb} \sigma_{gb} b^2}{kT \alpha} \frac{\sigma_s a}{E}^2$</td>
<td></td>
</tr>
<tr>
<td>Constant, $A_{gb}$</td>
<td>6.259 x 10$^6$</td>
</tr>
<tr>
<td>Act Energy for GBS, $Q_{gb}$</td>
<td>311 KJ/mol</td>
</tr>
<tr>
<td>Pre exp const, $D_{gb}$</td>
<td>3.26 x 10$^{-4}$</td>
</tr>
<tr>
<td>Stress exponent, $n$</td>
<td>2.2</td>
</tr>
<tr>
<td>Grain size, $a$</td>
<td>10 microns</td>
</tr>
</tbody>
</table>

5.2. Experimental Validation

The model based DMM is validated using experimental data points. A survey of literatures was made earlier to look for relevant creep deformation data for the titanium alloy (Badea et. al., 2014, Barboza et.al., 2004, Seco et. al., 2001). A recent work clearly demonstrate a transition in stress exponent values $n$ with stress over a temperature range from 450°C to 600°C and stress from 100 to 500Mpa. For stress lower than 0.9x yield strength (YS), steady-state creep strain rate dependency with stress follows a power law close to diffusion model ($n = 4.6$). For stress higher than 0.9xYS, a power law breakdown domain is observed when $n$ becomes high around ($n = 11-22$). Similar result was also reported for 500°C and 600°C for Ti-64 alloy (Nono et. al., 2006; Barboza, Neto, Silva, 2004). At low stress, the value of $n$ is 4.2 while $n$ becomes around 8.9 at higher stress indication the power law deviation. These results are collected and suitably converted for $x$ and $y$ axis and shown on the DMM maps in Fig. 2. The experimental points correspond to the transition region lie scattered, though most transition points are largely close to the predicted boundary between PLC and GBS regimes.

6. DISCUSSION

In view of the wide ranging microstructures expected in α-β Ti alloys and the various creep mechanisms and models reported for steady state creep regime in Ti alloys, an attempt is made here to develop some general relationships among them. The discussion here is presented in the light of four experimental cases (C1…C4) for validation of the model. Fig. 3 represents a typical α-β micrograph of the alloy in mill annealed (MA) condition after creep testing along and consequences of creep deformation at the alpha-beta interfaces and at alpha grains triple points. High triaxiality at triple points induces crack formation. The beta annealed Ti-64 alloy was tested at 455°C ($T/Tm = 0.38$), at a normalized stress level of 4.25x10$^{-3}$ and at strain rate at 2.8x10$^{-5}$ s$^{-1}$ (Seco et. al., 2001).

At $T/T_m$ of 0.38, referring to Fig. 2, a combination of dislocation creep (PLC) and GBS is perhaps the rate controlling mechanism. This test case is marked as C1 in Fig. 2. Failures occurred by the mechanism of creep cracking and coalescence in the MA alloy.

![Figure 3](image_url)
recrystallization-annealed and β-annealed Ti–6Al–4V were studied earlier (Sastry et al., 1980). Widmanstatten α/β interfaces act as obstacles to dislocation motion. The relatively large initial average grain size of 395 μm decreases the creep rate due to a reduction in grain boundary sliding, dislocation sources and the rate of oxygen diffusion along the grain boundaries (Barbouza et al., 2006). The testing conditions for above discussion refers to homologous temperature of 0.22, applied normalized stress of 5-10 x 10^{-3} and creep steady state strain rate around 2-6 x 10^{-8} s^{-1}. The creep test temperature was low enough to fall in the elastic region as marked C3 on Fig.2. In this case, the specimen simply deforms heavily and eventually fails.

Dependency of creep rates on microstructure has also been attributed to the colony size, spheroidization of the β films for the near α alloys. The dislocation substructures deformed at 200 MPa and 550°C to a strain of 10^{-2} s^{-1} show the formation of a stable dislocation configuration in the primary α grains leading to cell formation in IMI834 (Es-Soumi, 2001). Presence of fine microstructure and presence of primary α phase and a fully transformed β structure resulted in a higher creep resistance in IMI834. Climb controlled dislocation creep (PLC) in the α phase is proposed as the creep mechanism in the near α alloy. As the future work, effects of grain size on the predicted mechanisms map and comparison with experimental findings will be interesting.

7. Summary

Simulations with Matlab software code are performed in order to obtain contours lines for power law creep and grain boundary sliding deformation mechanisms for a titanium alloy. The study aims to develop a PLC and GBS based damage accumulation processes over strain rates ranging from 10^{4} to 10^{12} per second over the homologous temperature range of 0.10 to 0.655. A deformation mechanism map is generated for alpha-beta Ti alloy of 10 micron grain size. Two dominating mechanisms, namely a) Power law creep and c) Grain boundary sliding accommodated by different processes are considered. Appropriate constitutive equations are used for two mechanisms and simulation work is performed using the Newton- Raphson method. The modified and reconstructed map is validated using experimental data points. The experimental points support the theoretical
contours lines over the homologous temperature ranges 0.40 to 0.50. Four experimental validation cases are discussed in the context of predicted map of creep deformations. The test cases support the predicted map reasonably well in view of large differences in grain sizes and other structural conditions.

Figure 5(b). Formation and linkage of microcracks at the primary α particles observed in crept fracture surface of near α-Ti alloy (Omprakash et al., 2010).

REFERENCES


Es-Souni, M. (2001). Creep behaviour and creep microstructures of a high-temperature titanium alloy Ti–5.8Al–4.0Sn–3.5Zr–0.7Nb–0.35Si–0.06C (Timetal 834), Part I. Primary and steady-state creep, Materials characterization 46, 365– 379


with Titanium Matrix Composites, Materials Science and Engineering, 428, pp. 319-326.

Biographies

Dr. Amar Kumar has more than 25 years of research and consulting experience in the fields of structural materials characterization and development, fracture mechanics, failure analysis and applications. Dr. Kumar is currently working as senior research scientist in the R&D project of diagnostics, prognostics and health management of aeroengine components. He specializes in both data driven approaches and physics-based modeling and simulations.

Alka Srivastava graduated with a B.A.Sc. (Electrical Engineering) from the University of Ottawa. She joined Nortel Networks in 1990 as a Member of Scientific Staff. She joined Tecsis Corporation in 2003 as a member of the Research and Development team. Her research interests are in the areas of project management, software quality assurance, fault tolerant computing, numerical analysis and statistical methods.

Nita Goel graduated with a B.E. (Electronics & Telecommunications in 1985 from the University of Jodhpur, India. She completed her M.A.Sc (Electrical Engineering) from the University of Ottawa in 1994. She founded Tecsis Corporation in 1994 and is currently engaged in Research and Development. Her research interests are in the areas of software engineering, fault tolerant computing, numerical analysis and statistical methods.

Dr. Avi Banerjee is the Systems Development and Services Manager at Life Prediction Technologies Inc. He manages the development of diagnostics and prognostics tools for turbo-machinery and avionics at LPTi. His broad areas of research interest are performing physics based prognostics case studies, ENSIP, data trending for failure prediction, development of parts life tracking systems and the development of PHM framework. He works extensively with end users requiring prognostics services. Dr. Banerjee is a registered professional engineer in Ontario, Canada.

Dr. Ashok Kouli is the President of Life Prediction Technologies Inc. (LPTi), Ottawa, ON and also acts as an overall technical advisor. He has 25 years of experience in the field of materials engineering and life prediction with extensive experience in managing and supervising research and development activities in gas turbine structures, materials and life cycle management strategies. Over the years he has made key contributions in identifying and applying existing as well as emerging technologies in the field of gas turbine engineering.
Hardware Development for the Controlled Fault Injection into a Turbofan Engine Air-Bleed Valve

George E. Gorospe1, Donald L. Simon2, and Kai F. Goebel3

1 SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA
george.e.gorospe@nasa.gov

2 NASA Glenn Research Center, Cleveland, OH, 44135, USA
donald.l.simon@nasa.gov

3 NASA Ames Research Center, Moffett Field, CA, 94035, USA
kai.goebel@nasa.gov

ABSTRACT

Gas path fault diagnostics assists operators in determining, and managing the health of gas turbine engines. Engine data depicting fault progression under realistic operating conditions is useful for the maturation of these diagnostic methods. In this paper we present hardware created to inject a progressive fault in an air bleed valve of a high bypass turbofan engine during on-wing engine testing. The developed hardware interrupts and overrides the engine control computer’s command of the valve and allows for the nondestructive, progressive off-schedule operation of the air bleed valve. Numerical simulation results based on NASA’s Commercial Modular Aero-Propulsion System Simulation 40k are presented to illustrate representative changes in measured engine parameters that can be expected during such an experiment.

1. INTRODUCTION

Data-driven diagnostics and prognostics endeavor to determine the health of a system and predict the remaining useful life of that system. Data illustrating nominal operation and off-nominal operation is key to the development of these methods. However, run-to-failure data collected under typical operational environments and loads can be rare. In many cases, in-service data related to the failure of a specific component or system is either unavailable or opportunistic, and laboratory bench testing of the hardware may be infeasible or may not fully represent system-level effects.

In such cases, non-destructive failure testing of the integrated system may be necessary. This paper will present progress towards the successful seeded fault testing of an air-bleed system within an aircraft gas turbine engine for the benefit of diagnostic and prognostic methods.

Under the NASA Aviation Safety Program, NASA, in collaboration with the U.S. Air Force and other external partners, is demonstrating and maturing aircraft engine health management technologies through a series of stationary ground-based on-wing engine demonstrations known as Vehicle Integrated Propulsion Research (VIPR) testing (Hunter, Lekki, & Simon, 2014). VIPR provides a means to test and evaluate emerging health management technologies on an aircraft engine including new sensors and diagnostic algorithms. This series of tests is ongoing at the NASA Armstrong Flight Research Center / Edwards Air Force Base on a C-17 aircraft equipped with Pratt & Whitney F117 high bypass turbofan engines. The first and second VIPR tests, which occurred in 2011 and 2013 respectively, introduced non-damaging gas path fault scenarios into one of the engines installed on the aircraft. A third VIPR test (VIPR 3), is scheduled to occur in 2015.

During the VIPR 3 tests, a Prognostics and Decision Making (PDM) experiment will off-schedule the test engine’s station 2.5 air-bleed valve while the engine is at realistic operational levels. This PDM experiment will require new hardware designed to interrupt and override the command of the valve from the engine’s control computer. The hardware presented here allows for the controlled injection of a fault in the engine’s station 2.5 air-bleed valve system. The fault is a progressive degradation of the air bleed valves pneumatic actuator.

Previous work within the aircraft engine health management community has shown how gas path diagnostics can assist in the identification of rapid or abrupt gas path system faults...
through the observation and analysis of available engine gas path measurement parameters ((Li, 2002) & (Volponi, DePold, Ganguli, & Daguan, 2003)). The intent of the PDM research to be conducted in VIPR 3 will focus on the diagnosis of incipient or gradual gas path system faults along with prognostic forecasting of evolution of the fault over some future time horizon. Aircraft engine simulations can be used for the initial development and evaluation of gas path fault diagnostic methods.

For the purpose of this paper, the NASA Commercial Modular Aero-Propulsion System Simulation 40k (C-MAPSS40k) generic turbofan engine model will be used to illustrate the expected fault signature of a station 2.5 bleed fault. While not identical to the Pratt & Whitney F117 turbofan engine that will be tested during VIPR 3, C-MAPSS40k does provided representative outputs suitable for the development of gas path fault diagnostic methods for an engine in the same thrust category as the F117. Rinehart and Simon used C-MAPSS40k for the development of an integrated model-based architecture for performance monitoring and gas path fault diagnostics (Rinehart & Simon, 2014). Their use of a model-based approach for the diagnosis of engine gas path fault conditions, when applied to experimental data depicting normal station 2.5 air bleed valve operation and failed open conditions yielded the accurate detection of a steady state fault. Additionally, Boyle used far field acoustics with limited success in the identification of a fully failed station 2.5 air bleed valve (Boyle, 2014).

The structure of the paper is as follows. Section II describes the air-bleed valve system. Section III presents the approach to fault seeding, the overall setup of the air-Bleed valve Override Box (BOB), and its integration within the aircraft-engine system. Section VI discusses the fault modeling effort under taken prior to the experiment for the benefit of operations and data analysis. The paper closes with a conclusion in Section V.

2. The Air-Bleed Valve System

The station 2.5 air-bleed valve system allows airflow to exit the gas path of the engine at the low pressure compressor exit, ensuring proper airflow between the low and high pressure compressor. The 2.5 bleed system improves engine start-up and ensures operability over the operating range of the engine. The 2.5 air-bleed valve is also used by the engine’s Electronic Engine Control (EEC) system to assist in the recovery from stall or surge conditions (Linke-Diesinger, 2008).

The system is comprised of a closed loop controller, an electrohydraulic servo valve (EHSV), a pneumatic actuator, a linear variable displacement transducer (LVDT) and the valve itself. The valve is an axial ring which can cover or expose the circumferential slots at the exit of the low pressure compres-

3. The Fault and the Fault Injection Hardware

The fault to be introduced to the air-bleed valve system will simulate the progressive degradation of the valve actuator. Under the simulated fault condition, the actuator will be unable to maintain the desired valve position. This will produce a drift towards the failsafe position of the actuator, 100% open. This condition will degrade engine efficiency, but will not compromise the operability of the engine.

To begin the Prognostic and Decision Making portion of the testing, in which the station 2.5 bleed fault will be injected, the engine speed will be increased from idle to the experimental test point at which, the air-bleed valve position will be approximately 50% open. Next the operator of the Bleed valve Override Box (BOB) located in the cargo bay of the aircraft will override the command of the valve from the EEC and begin issuing new commands with an independent controller located within the BOB. Manually inputting each point in the degradation profile, the operator will take the valve from the
initial position commanded by the EEC based on N1C2 speed to a fully open failed state progressing at 5% increments with 3 minutes for the engine to stabilize between steps.

The BOB as shown in Figure 1 has been designed and developed to inject and control the progression of a fault in the overridden the air-bleed valve system. Enclosed within a vibration dampening, 10 unit high, 19” rack case, the BOB is composed of a power supply, a series of switches, an EHSV controller and error display, a positioning knob, and an ARINC 429 interface device. The power supply provides operational voltage for the controller, error display, and ARINC 429 interface device. The BOB includes an override switch which allows the control loop between the EEC and the EHSV to be overridden with a new control loop featuring the BOB’s controller, the BD101 from Parker Controllers, providing proportional control of the EHSV. The controller generates a 10VDC reference voltage, which is attenuated by the desired position potentiometer/knob circuit to create a voltage signal ranging from 0VDC to 5VDC representative of the desired position of the valve. Finally, the ARINC 429 interface device, the Aeroflex DT400 ARINC 429 database analyzer, is used to monitor the aircraft’s database for the data-word containing the valve’s LVDT position. The ARINC 429 interface device interprets the data for the valve’s current percent open position, and produces an analog voltage representative of that position. This signal is compared with the desired position voltage within the BOB’s modified closed loop control of the valve. The controller error is displayed on a voltmeter mounted above the desired position knob.

From the BOB, located within the aircraft’s cargo bay, a wiring harness with channels for the ARINC 429 databus, bleed-valve actuator inputs, EEC outputs, and LVDT circuit control will be routed across the wing of the aircraft to the engine pylon and down into the engine. With this implementation, the BOB operator is able to manually control the position of the station 2.5 bleed valve on the engine.

**4. Fault Simulation**

Results from the NASA C-MAPSS40K turbofan engine simulation are presented in this section to examine the representative change in engine performance characteristics expected at each step in the station 2.5 bleed fault progression. C-MAPSS40k simulates the operation of a typical twin spool, high-bypass turbofan engine in the 40,000 pound take-off thrust class, and has been tuned to produce data matching the characteristics of actual flight data (May, Csank, Lavelle, Litt, & Guo, 2010). While of the same thrust category as the Pratt & Whitney F117 turbofan engine to be used in the VIPR 3 test, C-MAPSS40k is not the same engine and therefore performance differences are expected. As such, the C-MAPSS40k results presented here are simply representative of the observations expected during the test. In conducting this assessment, C-MAPSS40k is run to a sea level static standard day condition (i.e., altitude = 0, Mach = 0, temperature = 59F) at an Engine Pressure Ratio (EPR) power setting of 1.24. At this operating condition the C-MAPSS40k station 2.5 bleed valve is 50% open.

Within the simulation routine, the above operating point for the model is first specified, then the engine model is allowed to reach a steady-state at this setting. Next, in order to simulate degradation of the air-bleed valve actuator, the valve is operated off-schedule from the 50% open position towards 100% open in 5% increments. Between each increment, the engine is allowed to return to a stabilized steady state condition at the specified EPR power setting. Furthermore, during this progression, no other modifications where made to the model’s closed-loop control system other than operating the station 2.5 bleed off-schedule.

In general, the take away from the data generated by the C-MAPSS40K simulation is a representation of the type of measurement changes that are to be expected during the PDM research and the relative magnitude of those perturbations. The station 2.5 bleed bias from 50% open to 100% open is most evident in percentage change of fuel flow (Wf) which
showed a 3.3% increase, exhaust gas temperature (T50) which showed a 3.2% increase, and the station 2.5 pressure and temperature (P25 and T25) which decreased by 6.8% and 9.01% respectively. Fig. 3 plots the change in exhaust gas temperature (EGT) as a function of station 2.5 (BLD25) position as predicted by C-MAPSS40K.

5. Conclusion

The work reported in this paper describes a fault injection apparatus that allows for the creation of precisely controlled fault conditions within the station 2.5 air-bleed valve system of an aircraft integrated turbofan engine. The operator can not only inject a fault in the air-bleed valve system but also control the progression of the fault to a fully failed condition in a nondestructive manner. This will allow an analysis of the engine response signatures which in turn can be used for development of diagnostic and prognostic methods. An simulation of the experiment has shown that multiple engine parameters are indeed perturbed by the valve failure. A lot of emphasis was placed on assuring non-interference with other operation of the aircraft and the engine through a rigorous requirements and engineering process. This required close collaboration with the owners of the engine and the aircraft to assure that all needed operational and procedural rules were followed. To avoid a tedious and costly process for software verification and validation, the entire apparatus was realized through hardware components.

Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOB</td>
<td>Bleed-valve Override Box</td>
</tr>
<tr>
<td>C – MAPSS40K</td>
<td>Commercial Modular Aero-Propulsion System Simulation</td>
</tr>
<tr>
<td>EGT</td>
<td>Exhaust gas temperature</td>
</tr>
<tr>
<td>EHSV</td>
<td>Electrohydraulic Servo Valve</td>
</tr>
<tr>
<td>EPR</td>
<td>Engine Pressure Ratio</td>
</tr>
<tr>
<td>LVDT</td>
<td>Linear Variable Displacement Transducer</td>
</tr>
</tbody>
</table>

REFERENCES


Biographies

George E. Gorospe Jr. received the B.E. degree in Mechanical Engineering from the University of New Mexico, Albuquerque, New Mexico, USA, in 2012. Since October 2012, he has been a research engineer at NASA Ames Research Center. In May 2013 he joined Stinger Ghaffarian Technologies and the Prognostic Center of Excellence at NASA Ames Research Center. His current research interests include space mission design, systems engineering, and autonomous mobile robot control and control systems design.

Donald L. Simon is a Controls Engineer at the NASA Glenn Research Center. He holds a B.S. degree from Youngstown State University (1987) and a M.S. degree from Cleveland State University (1990), both in electrical engineering. During his career he has focused on the development of advanced control and health management technologies for current and future aerospace propulsion systems. Mr. Simon’s specific research interests are in aircraft gas turbine engine performance diagnostics and performance estimation. He currently leads the propulsion gas path diagnostics effort ongoing under the Vehicle Integrated Propulsion Research (VIPR) effort.
**Dr. Kai Goebel** works at NASA Ames Research Center where he is the Area Lead for Discovery and Systems Health. He received the degree of Diplom-Ingenieur from Technische Universitaet Muenchen in 1990 and the Ph.D. from the University of California at Berkeley in 1996. Dr. Goebel worked at General Electrics Corporate Research Center in upstate New York where he was also an adjunct professor at Rensselear Polytechniq Institute. He has been on the dissertation committee of seven Ph.D. students at RPI, Syracuse University, University of Cincinnati, Vanderbilt University, Georgia Institute of Technology and Stanford University. He is currently guest professor at University of Cincinnati. He holds eighteen patents and he has published more than 300 technical papers.
Distributed Fault Detection and Estimation for Cooperative Adaptive Cruise Control System in a Platoon

Zoleikha Abdollahi Biron¹, and Pierluigi Pisu²

¹²Department of Automotive Engineering Clemson University, Greenville, SC, 29607, USA
zabdoll@clemson.edu
pisup@clemson.edu

ABSTRACT

Wireless vehicle to vehicle communication in a vehicle platoon is proposed in intelligent transportation systems to increase safety of the transportation system and assist drivers for improved decision making. However, similar to any networked system, cooperative connected vehicles in a platoon are vulnerable to malfunction due to failures in network communication. In addition to the possible data corruption, sensor and actuator faults can have significant effects on the control strategy for cruise control. This paper considers a platoon of connected vehicles equipped with cooperative adaptive cruise control and presents a reconstructive method based on sliding mode observer to estimate and reconstruct the faults in the sensors and actuators of vehicles.

1. INTRODUCTION

These Traffic congestions, limited road throughput and safety concerns in the past two decades led the automobile industry towards the idea of traffic control using intelligent vehicles. Consequently, the connectivity concept in vehicles has become a hot topic of research. Current smart vehicles are equipped with more than 70 electronic controls units (ECUs), Bluetooth, and Wi-Fi enabling them to communicate with the external networks (Larson, and Nilsson, 2008). Research in automotive control led to the development of radar-based adaptive cruise control (ACC) which can have a positive impact on vehicle safety, driver comfort, and highway efficiency (Kester, Willigen, and Jongh, 2014). Research and development in cruise control focuses on enabling more and better cooperation between ACC systems using communication and information transmission between vehicles using specific communication protocols such as Dedicated Short Range Communication (DSRC). Adaptive cruise control is operated without wireless communication link to enhance driving ability in traffic throughput, while maintaining a sufficient level of safety distance between vehicles (Ploeg, Semsar-Kazerooni, Lijster, Wouw, and Nijmeijer, 2013). Cooperative adaptive cruise control is indeed a step ahead of the ACC system in the vehicles and can be considered as a major development in recent research on intelligent transportation systems (ITS). CACC takes the ACC to the next level by receiving the other vehicle’s information through wireless communication, and tries to minimize the distance between vehicles in the range of couple of meters.

In addition to reducing the traffic congestion and increasing the road throughput, the CACC system can cause significant reduction in aerodynamic drag, especially for heavy-duty vehicles, thereby decreasing fuel consumption (Al Alam, Gattami, and Johansson, 2010).

As it is mentioned before, cooperative vehicles regarding to their Wi-Fi communications and also in-vehicle network communications as in a CAN bus, are vulnerable to network communication failures and faults in sensor and actuators. Numbers of papers have addressed the significant vulnerabilities of the vehicles and possible hacking in the vehicle network which can lead to significant security issues for all vehicles connected to the victim vehicle (Nilsson, and Larson, 2008).

Several researches on platooning vehicles with the main focus of control strategy design for CACC system and fault tolerant control is available in the literature (Lygeros, Godbole, and Broucke (2008), Zhang, Gantt, Rychlinski, Edwards, Correia, and Wolf (2009), Han, Chen, Wang, and Abraham (2013)). In (Nunen, Ploeg, Medina, and Nijmeijer, 2013), authors discuss that the CACC system cannot rely on the driver as a backup and is constantly active, therefore more prominent to the occurrences of faults (such as packet loss in the wireless communication). Hence, they present an algorithm which uses the availability of sensor-data in each moment in time to calculate in real-time a safe distance for the CACC system.
In this paper, a distributed fault reconstructive method based on sliding mode observer has been presented. In this method an estimation of faults on the sensors can be derived from the observers and consequently, each vehicle can reconstruct the faulty data before transmitting the data to the neighboring vehicles. Therefore, all incoming data from the network to the vehicles are considered to be healthy data, which enables the second observer to detect the faults on the actuators.

The rest of this paper is organized as follows: Section II presents modeling of the vehicle platoon. Fault diagnosis problem statement is described in section III. In section IV, we provide observer based fault reconstructive method and finally, simulation results for a vehicle platoon with 3 vehicles are shown and discussed in section V.

2. System modeling

In the current existing ACC system, the range (i.e., relative distance) and range rate to the preceding vehicle are measured with a radar or LIDAR sensor (Bu, Tan, and Huang, 2010). While, cooperative adaptive cruise control (CACC) is essentially a vehicle-following control methodology that automatically accelerates and decelerates so as to keep a desired distance to the preceding vehicle (Rajamani, and Zhu, 2002). To do this, in addition to onboard sensors, such as radars, vehicles should be equipped with wireless communication devices, such as DSRC, to receive extra information of the preceding vehicle(s) e.g., the desired acceleration is received through a wireless communication link.

2.1. Vehicle Dynamics

A nonlinear model has been considered for each vehicle in the platoon as shown in Eq. (1).

\[
\begin{align*}
\dot{x}_i &= v_i \\
\dot{v}_i &= u_i - \frac{1}{2M_i} C_d \rho_a A_v v_i^2 - C_r g \cos(\theta) - g \sin(\theta)
\end{align*}
\]

where, \( x_i \) and \( v_i \) are two states of each vehicle representing the absolute position and velocity of the \( i \)th vehicle. \( u_i \) is the acceleration control input per unit mass generated by CACC.

2.2. Control Strategy

A cooperative adaptive cruise control is considered for this paper as the control strategy for each vehicle in the platoon. This controller works as a higher level of controller generating the demanded acceleration for the vehicle based on the information exchange with the preceding vehicle and the current states of the vehicle. The block diagram of the whole platoon model is depicted in Fig. 2.

The control strategy used in this paper can be modeled as a discrete event system containing three main states as gap filling, gap regulating and suitable distance state. Each state has its own specific control rule (dynamic) based on the relative distance and speed between the host vehicle and its target (which is its preceding vehicle) Fig. 3.

In each state, control command is generated such that all constraints on the vehicle’s speed limits; acceleration and deceleration limits are satisfied.

\[
\begin{align*}
\nu_{\min} &< v_i < \nu_{\max} \\
\alpha_{\min} &< \alpha_i = u_i < \alpha_{\max}
\end{align*}
\]

where, \( \nu_{\min} \) and \( \nu_{\max} \) are minimum and maximum speed limitations respectively. Similarly, \( \alpha_{\min} \) and \( \alpha_{\max} \) are the minimum and maximum acceleration limitations for each vehicle.
1- Gap filling state:

When the relative distance between host vehicle and target, \(x_{rel}\), is more than maximum distance limit \(x_{max}\), the CACC controller of host vehicle enters the gap filling state and based on two vehicles’ current states (velocity and position), and by considering the system constraints, a satisfactory acceleration for the host vehicle is determined to fill the gap between the two vehicles. The preceding vehicle is identified as the vehicle directly in front of the host vehicle.

2- Gap regulating state:

When the relative distance between two vehicles is less than the desired gap, \(x_{min}\), CACC gap regulation controller is engaged immediately. The gap regulating controller should be precise enough to satisfy stringent performance criteria under various constraints such as limitation on speed and brake actuation.

3- Suitable distance state:

When the relative distance between target and host vehicle is between the gap filling and gap regulating conditions, both vehicles cruise with zero relative speed keeping similar acceleration profile.

3. Problem statement

As mentioned before, the cooperating vehicles are vulnerable to communication failures and faults in sensors and actuators. These faults can be physical faults on the system or can be a consequence of a virus on the CAN bus in the vehicle internal network due to the open source ECUs, Bluetooth and Wi-Fi connection (Nilsson et.al, 2008).

In this paper, we consider faults in the sensors (velocity and position) and the actuator (the accelerating or braking pedal). To this end, the main goal of this paper is fault detection and isolation using advantages of communication and extra information in the connected vehicles. To achieve this, we propose an observer-based diagnostics strategy to estimate and reconstruct faults in the sensors in each individual vehicle. This strategy will enable each vehicle in the platoon to detect, isolate and reconstruct the fault in its position or velocity sensor. Therefore, each vehicle can correct the sensor reading rejecting the effect of the faults before transmitting data through the network. Consequently, each vehicle in the platoon will receive correct sensor data from its preceding vehicle which in turn will make the reliable control input command available to the observers. Therefore, the controllers’ output will not be corrupted and the observers in each individual vehicle will be able to detect the actuator faults. The following section describes the observer design strategy and fault signature for each vehicle in the platoon.

4. Observer design

Considering vehicle dynamics in Eq. (1), sensors fault can be injected as measurement fault for each state. As a result, faults in sensors and actuator can be modeled as:

\[
\begin{align*}
x_{r-m} &= x_i + \Delta x_i \\
v_{r-m} &= v_i + \Delta v_i \\
u_{r-m} &= u_i + \Delta u_i
\end{align*}
\]

(3)

Where \(x_{r-m}\), \(v_{r-m}\) and \(u_{r-m}\) are measured values of position, velocity and control command of \(i^{th}\) vehicle respectively. \(\Delta x_i\), \(\Delta v_i\) and \(\Delta u_i\) represent measurement faults on position sensor, velocity sensors and actuator.

The observer structure is chosen based on sliding mode methodology and is given by:

\[
\begin{align*}
\dot{x}_i &= v_{r-m} + \eta_i \text{sgn}(x_{r-m} - \hat{x}) \\
\dot{v}_i &= u_i - \alpha \hat{v}_i^2 - \beta_i + \eta_i \text{sgn}(v_{r-m} - \hat{v}_i)
\end{align*}
\]

(4)

(5)

Where \(\hat{x}\) and \(\hat{v}\) present the estimated position and velocity of \(i^{th}\) vehicle respectively. \(\text{sgn}(.)\) stands for sign function and \(\eta_i\) is sliding mode observer gain. Consequently, the error dynamics will be:

\[
\begin{align*}
\dot{x}_i &= \dot{x} - \dot{x}_i = v_i - v_{r-m} - \eta_i \text{sgn}(x_{r-m} - \hat{x}) \\
\dot{v}_i &= \dot{v} - \dot{v}_i = -\alpha \hat{v}_i^2 - \beta_i - \eta_i \text{sgn}(v_{r-m} - \hat{v}_i)
\end{align*}
\]

(6)

(7)

Where \(\hat{x}_i\) and \(\hat{v}_i\) stand for position estimation error and velocity estimation error respectively.
4.1. Velocity Sensor Fault

In presence of fault on the velocity sensor in each vehicle, by considering the second observer equation, Eq. (5), the sliding surface can be written as:

\[ s_{vi} = v_{i-m} - \hat{v}_i = v_i + \Delta v_i - \hat{v}_i \]  (8)

We choose the Lyapunov candidate as:

\[ V_{vi} = \frac{1}{2} s_{vi}^2 \]  (9)

Using Eq. (7) as error dynamic, the derivative of the Lyapunov function can be written as:

\[ \dot{V}_{vi} = s_{vi} \dot{s}_{vi} = s_{vi} \left[ \dot{v}_{i-m} - \dot{\hat{v}}_i \right] 
= s_{vi} \left[ -\alpha_i (v_{i-m}^2 - \hat{v}_i^2) - \eta_{vi} \text{sgn}(v_{i-m} - \hat{v}_i) \right] \]  (10)

\[ \dot{V}_{vi} = s_{vi} \left[ -\alpha_i (v_{i-m} - \hat{v}_i) (v_{i-m} + \hat{v}_i) - \eta_{vi} \text{sgn}(v_{i-m} - \hat{v}_i) \right] 
= s_{vi} \left[ -\alpha_i (s_{vi}) (v_{i-m} + \hat{v}_i) - \eta_{vi} \text{sgn}(s_{vi}) \right] 
= -\alpha_i (v_{i-m} + \hat{v}_i) s_{vi}^2 - \eta_{vi} s_{vi} \quad \text{sgn}(s_{vi}) 
\leq -\alpha_i (v_{i-m} + \hat{v}_i) |s_{vi}|^2 - \eta_{vi} |s_{vi}| \]  (11)

By selecting \( \eta_{vi} \) as a large positive constant, \( \dot{V}_{vi} \) can be made negative definite and the sliding surface can be achieved in finite time.

On the sliding manifold, we have \( s_{vi} = 0, \dot{s}_{vi} = 0 \) from which we can write the following:

\[
\begin{align*}
\dot{s}_{vi} & \to 0 \quad \Rightarrow \\
\dot{\hat{v}}_i & = -\Delta v_i \\
\dot{\hat{v}}_i & = -\Delta \hat{v}_i
\end{align*}
\]  (12)

4.2. Position Sensor Fault

In occurrence of position sensor fault, the first sliding mode observer, Eq. (4), is considered and the sliding surface is defined as the following:

\[ s_{si} = x_{i-m} - \hat{x}_i = x_i + \Delta x_i - \hat{x}_i \]  (16)

A positive function \( V_{si} = \frac{1}{2} s_{si}^2 \) is chosen as a Lyapunov candidate to analyze the stability of the first observer error dynamics.

\[ \dot{V}_{si} = s_{si} \dot{s}_{si} 
= s_{si} \left[ \dot{x}_{i-m} - \dot{\hat{x}}_i \right] 
= s_{si} \left[ v_{i-m} - v_{i-m} - \eta_{si} \text{sgn}(x_{i-m} - \hat{x}_i) \right] \]  (17)

\[ \dot{V}_{si} = s_{si} \left[ -\eta_{si} \text{sgn}(x_{i-m} - \hat{x}_i) \right] = -\eta_{si} s_{si} \quad \text{sgn}(s_{si}) \]  (18)

Therefore, by choosing sliding mode gain, \( \eta_{si} \), as a large positive constant, \( \dot{V}_{si} \) can be made negative definite and the sliding surface can be achieved in finite time.

On the sliding manifold,

\[ s_{si} \to 0, \dot{s}_{si} \to 0 \]
\[ s_{si} = x_i + \Delta x_i - \hat{x}_i = 0 \]  (19)
\[ \dot{x}_i = -\Delta x_i = -K_{si} \]  (20)

Where \( K_{si} \) is filtered version of switching term \( \eta_{si} \quad \text{sgn}(x_{i-m} - \hat{x}_i) \) and it is referred as equivalent output error.

Based on Eq. (20), integrating \( K_{si} \) gives estimation of the fault on position sensor.

Residual \( R_2 \) is defined equal to the equivalent output error, \( K_{si} \), on sliding surface. As it is expected from Eq. (20), fault in the position sensor will show up in this residual.

4.3. Actuator Fault

Assuming the correct data received from preceding vehicle, second sliding model observer is used to detect the actuator fault.

The error dynamic is described in the following set of equations

\[
\begin{align*}
\dot{v}_i &= u_i - \alpha_i v_i^2 - \beta_i \\
\dot{\hat{v}}_i &= u_i - \alpha_i (v_i^2 - \hat{v}_i^2) + K_{vi} \\
\hat{v}_i &= u_i - u_i - \alpha_i (v_i^2 - \hat{v}_i^2) + K_{vi} 
\end{align*}
\]  (21)

\[
\begin{align*}
\dot{u}_i &= \Delta u_i - \alpha_i (v_i^2 - \hat{v}_i^2) + K_{vi} 
\end{align*}
\]  (22)

Therefore, choosing the sliding surface as Eq. (8) and Lyapunov candidate as Eq. (9), the stability of the error dynamic can be described via Eq. (11).
Hence, on the sliding surface, fault on the actuator will show up in the residual $R_2$ which is generated by filtering the switching term of second sliding mode observer.

4.4. Fault Signature

As it is explained in the previous sections, each fault shows up in different residual. Based on the residuals and threshold setting for each residual, fault signature table is generated as table 1.

<table>
<thead>
<tr>
<th>Residual</th>
<th>Velocity sensor fault</th>
<th>Position sensor fault</th>
<th>Actuator fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

As it can be inferred from the table, using the distributed fault diagnostics algorithm in each vehicle, faults in sensors and actuator can be detected and isolated.

5. SIMULATION AND RESULTS

To simulate faults in the connected vehicles, a platoon of three vehicles equipped with cooperative adaptive cruise controller has been considered. A distributed fault diagnostics algorithm is implemented for each following vehicle. The leader vehicle follows the velocity profile of US06 as highway driving profile. The length of driving cycle is 600 seconds and during the whole cycle single fault scenario is simulated for different faults on sensors and actuator for each car. In the flowing the results for the second vehicle will be explained as an example.

As it can be inferred from the table, using the distributed fault diagnostics algorithm in each vehicle, faults in sensors and actuator can be detected and isolated.

A position sensor fault as a bias of 0.05% of current position with constant slop occurs at $t=100$ seconds and it remains for 75 seconds. A velocity sensor fault is injected at time $350$ second and remains for 75 seconds. This fault is simulated as a bias of 0.5 m/s which is equal to 2% of maximum velocity. At the end, the actuator sensor fault occurs at $t=500$ seconds as a bias of 1% of normal actuator value and it remains for 50 seconds in the system. Fig. 4 shows both residuals $R_1$ and $R_2$. As it can be seen, faults in velocity sensor and actuator show up in $R_1$ while faults in position and velocity sensors will change the value in $R_2$.

In order to have a satisfactory low false alarm for each residual, a fix threshold based on probability distribution method is chosen. Fig. 5 shows threshold setting in second residual as an example.

For each residual threshold is set as (23). Whenever one of these residuals surpasses its own threshold, a fault detection signal will be triggered to declare that fault is detected in each vehicle in the platoon.

$$|R_i| > \gamma_i = \mu_i + 1.1\sigma_i$$  \hspace{1cm} (23)

As it is expected, faults on the velocity sensor and actuator will trigger $R_1$ in real time, while faults on position and velocity sensors can trigger the $R_2$. Therefore, faults in each vehicle in the platoon will follow the signature mentioned in the table 1. These faults can occur in a system due to GPS, radar or laser measurements. Fig. 6 depicts fault signature when three different faults happen in the system, as it be inferred from the figure, the signature is as described in table 1.
6. CONCLUSION

In this paper, a distributed fault diagnostics scheme is designed based on sliding mode observer to estimate and isolate faults on position and velocity sensors. In addition, assuming the ideal network connection between vehicles in the platoon and reconstructed faults on the sensors before data transmission, fault on the actuator can be detected for each vehicle. Future work in this research includes fault diagnostics with no-ideal network connection and packet dropout.

REFERENCES


A Probabilistic Machine Learning Approach to Detect Industrial Plant Faults: PHM15 Data Challenge

Wei Xiao1

1 SAS Institute Inc. Cary, NC, 27513, USA
Wei.Xiao@sas.com

ABSTRACT

Fault detection in industrial plants is a hot research area as more and more sensor data are being collected throughout the industrial process. Automatic data-driven approaches are widely needed and seen as a promising area of investment. This paper proposes an effective machine learning algorithm to predict industrial plant faults based on classification methods such as penalized logistic regression, random forest and gradient boosted tree. A fault’s start time and end time are predicted sequentially in two steps by formulating the original prediction problems as classification problems. The algorithms described in this paper won first place in the Prognostics and Health Management Society 2015 Data Challenge.

1. INTRODUCTION

Fault detection in industrial plants is a hot research topic as more and more sensor data are being collected throughout the industrial process, and standard systems based on univariate statistical process control lack power in these more complex systems. Early detection of faults can help to avoid system shut-down and component failure or even catastrophes (Korbicz, Koscielny, Kowaleczuk, & Cholewa, 2012).

Many machine learning algorithms used in pattern classification are now being utilized in fault detection. Dimension reduction techniques, such as principal component analysis, partial least squares, and Fisher’s discriminant analysis have been applied to detect faults in chemical processes (Chiang, Russell, & Braatz, 2000; Chiang, Kotanchek, & Kordon, 2004; Yin, Ding, Haghani, Hao, & Zhang, 2012). Support vector machine and artificial neural networks are also widely used methods for fault detection; they have been applied to gearbox failure detection (Samanta, 2004) and chemical process fault diagnosis (Wang & Yu, 2005). K-Nearest Neighbor and fuzzy-logic are two other powerful methods that have been used to detect faults in semiconductor manufacturing processes (He & Wang, 2007) and mechanical systems (Korbicz et al., 2012). Tree based algorithms such as random forest and gradient boosted tree are useful machine learning algorithms in situations where one expects nonlinear and interactive effects between covariates. They have been applied to fault detection in aircraft systems (Lee, Park, & Jung, 2014). This year’s Prognostics and Health Management (PHM) Society data challenge focused on plant fault detection. We try many of the above machine learning techniques and ultimately use a combination of several in our final detection strategy described herein. The rest of the paper is organized as follows. Section 2 discusses the data challenge problem. Section 3 introduces the relative methodologies and our algorithm. Finally, Section 4 concludes the paper and discusses future work.

2. PROBLEM STATEMENT

The objective of this year’s challenge is to design an algorithm to predict plant faults. Correct prediction involves predicting the type of fault (one of five), as well as the start and end time of each fault, within one hour.

Three datasets are given, training, test, and validation; they contain information on 33, 15, and 15 plants, respectively. For each plant three files are provided: plant-#a.csv, plant-#b.csv, plant-#c.csv, where # is the plant id. File (a) contains readings from 4 sensors ($S_{1-4}$) and 4 control reference signals ($R_{1-4}$) from each plant component (denoted $mi$). The number of components varies by plant; data on $S_{1-4}$ and $R_{1-4}$ for each plant component are denoted $mi_{Sj}$ and $mi_{Rj}$, respectively. File (b) contains time series data for cumulative energy consumed ($E_1$) and instantaneous power ($E_2$) from a fixed number of zones (denoted $ni$) within a given plant. Each plant zone covers one or more of the plant components and the number of zones varies by plant. The notation $ni_{Ej}$ is used to represent the reading of $E_j$ for the $ith$ zone. File (c) contains plant fault events, each characterized by a start time, an end time, and a failure type. Data are given on 6 different fault types ($F_{1-6}$), but only faults 1-5...
are to be predicted. The training dataset has complete fault event data, and is used to train the model. The test dataset has complete fault event data for the first half of the sample, but approximately 50% of the events in the second half of the data have been randomly removed. The boundary between the first and second half of the data is given, and referred to as the boundary time. Our goal is to predict the deleted fault events. The validation dataset is similar in structure to the test dataset.

Each team participating in the contest is permitted to submit their predictions of the missing faults in the test data (fault type, and start and end time) once each week to assess their prediction performance and use the score as feedback to improve their model. The final team rank is determined by the score of a submission of predictions based on the validation dataset (Rosca, Song, Willard, & Eklund, 2015).

### 2.1. Data Description and Preprocessing

We began our analysis by first studying the data to garner any information that would be useful in predicting the faults. Not only do the number of both zones and components vary by plant, but the proportion of each fault type ($PF_i$) varies quite dramatically. To illustrate, Table 1 summarizes the data for the first five plants in both the training and test datasets. Note that $F_3$ (fault type 3) never occurs in plants 2, 3, 5 and 42, and $F_5$ never occurs in plants 2, 3, 5, 41, 42 and 45. We also notice that $S_3$, $R_1$, $R_2$, $R_3$, $R_4$ appear to be categorical variables, and $S_1$, $S_2$, $S_4$ appear to be continuous variables. The number of unique levels of all categorical variables for the same sample plants are summarized in Table 2. Given the above differences across plants and variables, we build a separate model for each plant.

The sampling interval for the data provided was theoretically 15 minutes, however some logging delays resulted in irregular intervals. To preprocess the data, we round all timestamps to obtain regular 15-minute gaps, and then combine all three files. We define new variables $TTF_{Fk}$, $k = 1, \ldots, 6$, to represent time to failure of fault type $k$. A negative value, $-i$, means the next fault is $i$ intervals in the future (1 interval is 15 minutes), and a positive value, $i$, means the current fault started $i$ intervals ago and has not yet ended. We define $E_3$ as the first order difference of $E_1$, i.e., $E_3(t) = E_1(t) - E_1(t-1)$. We also define $start_{Fk}$, $k = 1, \ldots, 6$, as a binary indicator of whether any type $k$ fault starts within one hour of the corresponding timestamp, and define $end_{Fk}$, $k = 1, \ldots, 6$, as the binary indicator of whether any type $k$ fault ends within one hour of the corresponding timestamp. Occasionally observations of covariates on some timestamps are missing. Forward imputation is applied to all covariates to impute these missing values, except for $TTF_{Fk}$, $start_{Fk}$ and $end_{Fk}$, which are imputed with values -999, 0 and 0, respectively. To illustrate these preprocessing steps, a small sample of data from plant 1 is shown in Table 3.

<table>
<thead>
<tr>
<th>Plant</th>
<th>Nm</th>
<th>Nn</th>
<th>PF1</th>
<th>PF2</th>
<th>PF3</th>
<th>PF4</th>
<th>PF5</th>
<th>PF6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
<td>0.25</td>
<td>0.18</td>
<td>0.12</td>
<td>0.04</td>
<td>0.11</td>
<td>0.29</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>2</td>
<td>0.46</td>
<td>0.03</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.49</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>2</td>
<td>0.17</td>
<td>0.01</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.80</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>2</td>
<td>0.30</td>
<td>0.16</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.48</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>2</td>
<td>0.27</td>
<td>0.07</td>
<td>0.00</td>
<td>0.07</td>
<td>0.00</td>
<td>0.58</td>
</tr>
<tr>
<td>41</td>
<td>5</td>
<td>2</td>
<td>0.29</td>
<td>0.10</td>
<td>0.02</td>
<td>0.05</td>
<td>0.00</td>
<td>0.54</td>
</tr>
<tr>
<td>42</td>
<td>10</td>
<td>3</td>
<td>0.39</td>
<td>0.15</td>
<td>0.00</td>
<td>0.07</td>
<td>0.00</td>
<td>0.40</td>
</tr>
<tr>
<td>43</td>
<td>6</td>
<td>2</td>
<td>0.08</td>
<td>0.16</td>
<td>0.02</td>
<td>0.22</td>
<td>0.11</td>
<td>0.41</td>
</tr>
<tr>
<td>45</td>
<td>7</td>
<td>2</td>
<td>0.17</td>
<td>0.47</td>
<td>0.05</td>
<td>0.06</td>
<td>0.03</td>
<td>0.22</td>
</tr>
<tr>
<td>46</td>
<td>5</td>
<td>2</td>
<td>0.19</td>
<td>0.21</td>
<td>0.14</td>
<td>0.05</td>
<td>0.00</td>
<td>0.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plant</th>
<th>S3</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>38</td>
<td>6</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>26</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>30</td>
<td>7</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>34</td>
<td>7</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>12</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

| Table 3. A sample of data from plant 1 after preprocessing. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| Timestamp       | m1_R1           | m1_S1           | TTF_F1          | start_F1        | end_F1          |
| 2009-09-04 09:00:00 | 739 763          | -7              | 0               | 0               | 0               |
| 2009-09-04 09:15:00 | 739 763          | -6              | 0               | 0               | 0               |
| 2009-09-04 09:30:00 | 739 759          | -5              | 0               | 0               | 0               |
| 2009-09-04 09:45:00 | 700 711          | -4              | 1               | 0               | 1               |
| 2009-09-04 10:00:00 | 700 711          | -3              | 1               | 0               | 1               |
| 2009-09-04 10:15:00 | 700 712          | -2              | 1               | 1               | 1               |
| 2009-09-04 10:30:00 | 700 720          | -1              | 1               | 1               | 1               |
| 2009-09-04 10:45:00 | 700 714          | 0               | 1               | 1               | 1               |
| 2009-09-04 11:00:00 | 700 716          | 1               | 1               | 1               | 1               |
| 2009-09-04 11:15:00 | 700 711          | 2               | 1               | 1               | 1               |
| 2009-09-04 11:30:00 | 700 720          | -4              | 1               | 1               | 1               |
| 2009-09-04 11:45:00 | 700 716          | -40             | 1               | 1               | 1               |
| 2009-09-04 12:00:00 | 700 712          | -39             | 0               | 0               | 0               |
| 2009-09-04 12:15:00 | 700 711          | -38             | 0               | 0               | 0               |
| 2009-09-04 12:30:00 | 700 716          | -37             | 0               | 0               | 0               |

Table 3. Counts of unique levels of all categorical variables, (Plants in training set: 1, 2, 3, 4, 5; plants in test set: 41, 42, 43, 45, 46)
correlated, and $S_2$ and $S_4$ are highly negatively correlated, across all components in all plants. To illustrate this finding, Figure 1 shows the correlation heatmap of plant 1 for the first two components.

Second, by observing the correlation heatmap of either $m_iR_2$, $m_iR_3$, or $m_iR_4$, across all components for a given plant, one can identify which components are in the same zone; components in the same zone are highly correlated. For example, Figure 2 shows the correlation heatmap of $m_iR_4$ across all 6 components in plant 1. Based on the heatmap, it seems components 1, 3, 5 of plant 1 belong to one zone, components 2, 4 belong to another zone, and component 6 itself belongs to the third zone. Although one can identify which components are zoned together, the groups of components could not always be linked to a specific zone, so this information was ultimately not utilized in our modeling approach.

We also find that month and hour are important categorical variables to predict the faults. Count plots of $F_2$ by month and hour are shown in Figure 3 and 4 to illustrate this point. $F_2$ starts most frequently between May and November and between 6 o’clock and 23 o’clock. But its distribution varies across plants.

Lastly, we observe that, before a fault happens, sensor readings are often increasing or decreasing. These unique patterns can be utilized to predict the start time of the fault. See Figure 5 for an example, where the mean value of $m_2R_2$ and its corresponding 95% confidence bands are plotted against time to failure of $F_1$.

3. METHODOLOGY

In this section we introduce our approach and the related methodologies utilized for the PHM competition. The overall approach consists of two parts: preprocessing and modeling. Figure 6 provides an overview of the process implemented. Details of the data preprocessing have been discussed in Section 2.1. After preprocessing, we divide the training data into two parts: cross validation training data and cross validation test data. Our basic approach is to try various models using the cross validation training data and then evaluate their performance based on their ability to forecast faults in the cross validation test data. The winning model is then applied to the test data and the subsequent predictions submitted to PHM for assessment.

There are two steps to the modeling process: predict fault start times and then, given these start times, predict fault end time. A detailed flowchart of the modeling process is shown in Figure 7. The modeling procedure outlined is implemented for each fault type, plant by plant. Given a fault type and plant, $F_1$ in plant 5 for example, we translate the prediction problem into a classification problem ($start_F1=1$ vs $start_F1=0$). From the classification model we estimate the probability that $F_1$ starts during each time interval.
(\text{start}_F1=1). We derive the set of predicted fault start times, \( \Omega_{F1} \), based on these estimated probabilities. For each start time in \( \Omega_{F1} \), we then solve another classification problem \( (\text{end}_F1=1 \text{ vs } \text{end}_F1=0) \) and estimate the probability the \( F1 \) will end in the next 1 to \( t_{\text{max}} \) time intervals, where \( t_{\text{max}} \) is an estimated upper bound of fault \( F1 \)'s duration. These estimated probabilities are used to find the fault end time.

Various machine learning algorithms were tried to solve these classification problems: K-nearest neighbors (KNN), naive bayes, gradient boosting machine (GBM), random forest, penalized logistic regression (with \( \ell_2 \) penalty), etc. In the final algorithm, we use gradient boosting machine, random forest and penalized logistic regression.

In Section 3.1, we give a quick review of the machine learning algorithms used. Section 3.2 discusses how we evaluate the effectiveness of these different approaches, and finally, the specific algorithm details to predict a fault’s start time and end time are given in Section 3.3 and 3.4, respectively.

### 3.1. Machine Learning Algorithms

Data-driven or statistical approaches based solely on historical data are seen as the most cost-effective approach for fault detection in complex systems (Aldrich & Auret, 2013). Machine learning is the key to any data-driven algorithm.

Machine learning algorithms can be categorized as either supervised or unsupervised. In supervised learning, the goal is to predict a response \( Y \) based on input features \( X \). All methods in our algorithm belong to supervised learning.

K-nearest neighbor is an instance-based learning algorithm which has a very simple form but works extremely well on many problems. When used in classification problems, it can learn very flexible decision boundaries. However, when dealing with high dimensional data, it is likely to suffer from overfitting and perform poorly due to the curse of dimensionality.

Naive Bayes is a classification technique based on applying Bayes’ theorem. It assumes conditional independence between features given a class \( Y = i \). Despite its oversimplified and sometimes unrealistic conditional independence assumption, it often outperforms other more sophisticated algorithms. Naive Bayes is widely used in text mining and natural language processing.

Logistic regression is widely used in classification problems. However, when the number of input features is large, it performs poorly due to overfitting. Penalized logistic regression avoids the overfitting problems of logistic regression by imposing a penalty on large fluctuations in the estimated parameters. In this paper, we use a penalized logistic regression with \( \ell_2 \) penalty. Besides avoiding overfitting and improving prediction accuracy, this ridge type penalty is also very computationally efficient.

Random forest is an ensemble learning method which averages over a large collection of de-correlated decision trees. Random forest allows for interaction effects among features just like any tree based algorithm, but it corrects for the likely overfitting of decision trees. The performance of random forest is comparable to boosting, and they are easier to train and tune (Friedman, Hastie, & Tibshirani, 2001).

Gradient boosting machine (GBM) is an ensemble method which combines weak classifiers to form a strong classifier. It works in a forward “stagewise” fashion. GBM is very flexible as users can provide their own loss function. GBM has been implemented in many data mining competition winning
strategies. We implement GBM using the decision tree as our weak classifier.

3.2. Evaluation

Evaluation is the key step to obtain feedback and find the approach that works well predicting faults with the data at hand. In this competition, each team was allowed to submit a set of predictions only once a week to score their model. However, this is not frequent enough given that there are a large number of possible models and tuning parameters to try. To remedy this problem and allow us to try many approaches, we built our own evaluation system, based on the idea of cross validation. Our cross validation system was basically designed to mimic the competition evaluation/scoring system.

To build our own scoring system, we randomly remove 50% of the faults in the second half of each training dataset. We build the model using the remaining fault data and attempt to predict the deleted events. We then compare the predicted faults $E_P$ with the deleted true events $E_T$, and score our model. If a fault event in $E_T$ has been correctly predicted in $E_P$ (i.e., there exists an event in $E_P$ with start time and end time within one hour of actual start time and end time, and fault type also matches), it is a true positive and receives 10 point. If a fault event in $E_P$ has correct start time and end time, and incorrect fault type, it is a misclassification and that prediction receives -0.01 point. If a fault event in $E_P$ has incorrect start time or end time, it is considered as a false positive and receives -0.1 point. If a fault event in $E_T$ has not been identified in $E_P$, it is considered a false negative and receives -0.1 point.

We found that the above scoring system worked very well in the sense that the order of magnitude of improvement of one classification algorithm over another based on our scoring system was similar to the improvement seen on the leaderboard. In this way, we could use our scoring system and experiment with many different algorithms and tuning parameters.

3.3. Predict Start Time

For every plant and fault type in the cross validation training, test or validation datasets, we built a separate classification model to predict the start time of deleted events. The binary indicator of whether a type $k$ fault starts within

![Figure 5. Plot of $m_2.R2$ against time to failure of F1.](image)

![Figure 6. Overall flowchart.](image)

![Figure 7. Modeling flowchart.](image)
to gain the optimal performance.

- We include covariates R1-R4, S1-S4, E2 and E3 in our models.
- We do feature engineering and add features like month, hour, weekday and time (the number of minutes since 00:00 of the first day of the corresponding year / (60*24)) in our classification models.
- We add lagged covariates of all sensor readings to the model. Specifically, for any sensor reading, $X(t)$, we include $X(t-k)$, for all nonzero $k$, where $k \geq \min_{\text{lag}}$ and $k \leq \max_{\text{lag}}$. We try different $\min_{\text{lag}}$ and $\max_{\text{lag}}$ combinations, and the best one we found is $\min_{\text{lag}} = -8$ and $\max_{\text{lag}} = 4$. To define these new variables we introduce the following notation: $L_k S_j(t) = \min_{\text{lag}} S_j(t-k)$, where $k > 0$ and thus, represents the lagged covariate of $\min S_j$. In contrast, $R_k^* S_j(t) = \max_{\text{lag}} S_j(t-k)$, where $k < 0$ and $k^* = -k$, and thus, represents the lead (future) covariate of $\max S_j$.
- All covariates are standardized to have mean 0 and variance 1.
- For $k$ consecutive estimates of $p_{\text{test}}$ (i.e., the estimated probability a fault starts for $k$ consecutive time intervals), we find the largest probability and compare it to a threshold $p$. If it exceeds $p$, the corresponding time stamp is saved as a predicted start time. We tested different combinations of values for $k$ and $p$. The best performing combination is $k = 6$ and $p = 0.75$.
- We compared the performance of the various algorithms modeling covariates (month, hour, S3, R1, R2, R3, R4, etc) as categorical versus continuous variables. No real improvement was made modeling them as categorical variables, so all covariates are treated as continuous.
- We experimented with various different classifiers including KNN, Naive Bayes, GBM, random forest, and penalized logistic regression. We found that random forest and penalized logistic regression performed the best. Our final algorithm is an ensemble of these latter two models.

One can determine which covariates are most important in predicting fault start times by looking at the random forest results. Each covariate can be scored based on mean decrease in impurity. Table 4 lists the top 15 most important covariates and their corresponding score for each one of the five fault types in plant 1 and 6. The importance of covariates vary from fault to fault and from plant to plant. S3 is the most important covariate in predicting the start time of F1 in plant 1, while both R1 and S3 seem to be important in predicting F1’s start time in plant 6; S3 is the most important covariate in predicting F2’s start time, while E3 is the most important covariate in predicting F4’s start time in plant 1.

Table 5 shows the percentage of times that each covariate ranked in the top 15 importance score (random forest to predict fault start time) averaged over all plants. S1-S4 seem to be more important than R1-R4, and the covariates time, month, E2 and E3 are also important features in the models.

3.4. Predict End Time

To predict the end time of a plant fault, we built another classification model. As with the start time prediction problem, we estimate a separate model for each fault type and plant. To explain our modeling approach, suppose we want to find the end time of a predicted type 1 fault (F1). We first estimate an upper bound for the duration of a F1 fault ($t_{\max}$) based on all known F1 events. We estimate $t_{\max}$ as follows: $t_{\max} = \max(8, q_{0.95})$, where $q_{0.95}$ is the 95% upper quantile of all historical F1 durations.

Intuitively, we could predict the fault end time by calculating the elapsed time since the fault first occurred. We denote it as $\text{elapsed}_t$, which is measured in units of 15 minutes. We find that the model only based on $\text{elapsed}_t$ isn’t powerful enough, so we add other covariates to the model.

The classification model is trained using data from the $t_{\max}$ intervals following the onset of each known F1 event. These data, stacked together, form the matrix of covariates for the training model ($X_{\text{train}}$, $\text{end}_F$). Once the classification model is trained, it is used to estimate the probability that each of our predicted events will end in any one of the $t_{\max}$ time periods following the predicted event start time. These predictions are based on $X_{\text{test}}$, formed by stacking the $t_{\max}$ intervals following the onset of each predicted F1 event.

Given the small penalty for false negative predictions relative to the reward for a true positive prediction, we allow our system to predict as many as two end times for each predicted event. The first estimated end time is made by finding the time period within the $t_{\max}$ periods following our predicted event with the largest estimated probability that $\text{end}_F=1$. This is our first end time prediction. To look for a possible
Table 4. Top 15 most important covariates to predict fault start time for each of five faults in plant 1 and 6.

<table>
<thead>
<tr>
<th>Plant Rank</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R4_m4_S3</td>
<td>R1_m5_S3</td>
<td>R7_m2_S4</td>
<td>R5_n3_E3</td>
<td>R4_m3_R1</td>
</tr>
<tr>
<td>2</td>
<td>R5_m4_S3</td>
<td>R4_m5_S3</td>
<td>R7_m2_S2</td>
<td>n3_E3</td>
<td>R8_m1_S1</td>
</tr>
<tr>
<td>3</td>
<td>R3_m4_S3</td>
<td>R8_m5_S3</td>
<td>R6_m2_S4</td>
<td>R1_n3_E3</td>
<td>R5_m5_S4</td>
</tr>
<tr>
<td>4</td>
<td>time</td>
<td>R3_m5_S3</td>
<td>R1_m5_S3</td>
<td>R3_m3_E3</td>
<td>R7_m3_R1</td>
</tr>
<tr>
<td>5</td>
<td>R8_m6_S3</td>
<td>R2_m5_S3</td>
<td>R1_m5_S3</td>
<td>R3_m3_E3</td>
<td>R5_m5_R1</td>
</tr>
<tr>
<td>6</td>
<td>R2_m5_S3</td>
<td>R6_m5_S3</td>
<td>R6_m5_S3</td>
<td>R6_n3_E3</td>
<td>R7_m5_S2</td>
</tr>
<tr>
<td>7</td>
<td>R7_m4_S3</td>
<td>R7_m5_S3</td>
<td>R6_m3_S3</td>
<td>R8_m3_E3</td>
<td>R5_m3_R1</td>
</tr>
<tr>
<td>8</td>
<td>L4_m6_S3</td>
<td>R5_m5_S3</td>
<td>R7_m3_S2</td>
<td>R2_n3_E3</td>
<td>R4_m5_S2</td>
</tr>
<tr>
<td>9</td>
<td>month</td>
<td>L1_m5_S3</td>
<td>R8_m4_S2</td>
<td>R3_n3_E3</td>
<td>R1_m3_S3</td>
</tr>
<tr>
<td>10</td>
<td>R5_m6_S3</td>
<td>m5_S3</td>
<td>R3_m3_S3</td>
<td>R1_m2_S2</td>
<td>m6_R1</td>
</tr>
<tr>
<td>11</td>
<td>R6_m4_S3</td>
<td>L4_m5_S3</td>
<td>m5_S3</td>
<td>L1_m5_S2</td>
<td>R1_m1_R1</td>
</tr>
<tr>
<td>12</td>
<td>R1_m4_S3</td>
<td>R7_m2_S2</td>
<td>L4_m5_S3</td>
<td>L1_m5_S2</td>
<td>R1_m1_S1</td>
</tr>
<tr>
<td>13</td>
<td>m4_R1</td>
<td>R2_m9_S3</td>
<td>R4_m3_S3</td>
<td>R2_m3_S1</td>
<td>R4_m5_S1</td>
</tr>
<tr>
<td>14</td>
<td>m5_R1</td>
<td>R1_m9_S3</td>
<td>R4_m3_S3</td>
<td>R4_m8_S1</td>
<td>R4_m5_S2</td>
</tr>
<tr>
<td>15</td>
<td>R4_m5_S3</td>
<td>R7_m2_S4</td>
<td>L4_m3_S1</td>
<td>R8_m8_S1</td>
<td>R4_m8_S2</td>
</tr>
</tbody>
</table>

Table 5. Percentage that each covariate rank top 15 in the importance score (random forest to predict fault start time) averaged over all plants.

<table>
<thead>
<tr>
<th>Covariate</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>60%</td>
<td>48%</td>
<td>31%</td>
<td>32%</td>
<td>18%</td>
</tr>
<tr>
<td>month</td>
<td>27%</td>
<td>14%</td>
<td>17%</td>
<td>11%</td>
<td>4%</td>
</tr>
<tr>
<td>hour</td>
<td>5%</td>
<td>12%</td>
<td>8%</td>
<td>4%</td>
<td>0%</td>
</tr>
<tr>
<td>weekday</td>
<td>2%</td>
<td>0%</td>
<td>2%</td>
<td>0%</td>
<td>2%</td>
</tr>
<tr>
<td>S1</td>
<td>32%</td>
<td>40%</td>
<td>81%</td>
<td>82%</td>
<td>64%</td>
</tr>
<tr>
<td>S2</td>
<td>54%</td>
<td>48%</td>
<td>58%</td>
<td>36%</td>
<td>88%</td>
</tr>
<tr>
<td>S3</td>
<td>65%</td>
<td>78%</td>
<td>33%</td>
<td>34%</td>
<td>20%</td>
</tr>
<tr>
<td>S4</td>
<td>49%</td>
<td>52%</td>
<td>56%</td>
<td>41%</td>
<td>90%</td>
</tr>
<tr>
<td>R1</td>
<td>19%</td>
<td>5%</td>
<td>10%</td>
<td>14%</td>
<td>10%</td>
</tr>
<tr>
<td>R2</td>
<td>17%</td>
<td>19%</td>
<td>23%</td>
<td>16%</td>
<td>14%</td>
</tr>
<tr>
<td>R3</td>
<td>19%</td>
<td>12%</td>
<td>8%</td>
<td>12%</td>
<td>12%</td>
</tr>
<tr>
<td>R4</td>
<td>2%</td>
<td>3%</td>
<td>6%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td>E2</td>
<td>19%</td>
<td>31%</td>
<td>12%</td>
<td>25%</td>
<td>8%</td>
</tr>
<tr>
<td>E3</td>
<td>25%</td>
<td>41%</td>
<td>17%</td>
<td>39%</td>
<td>16%</td>
</tr>
</tbody>
</table>

The following list details the specifics of our final algorithm for the end time classification problem:

- The optimal threshold probability for deciding on a second end time is $p_2=0.2$.
- The model includes features R1-R4, S1-S4, E2, E3 and elapsed_t. The model also includes features like month, hour, weekday and time (the number of minutes since 00:00 of the first day of the corresponding year / (60*24)).
- We add lagged covariates of all sensor readings to the model. The optimal lag choice is min_lag = −8 and max_lag = 8.
- All covariates are standardized to have mean 0 and variance 1.
- We model all covariates as continuous variables.
- We compared the performance of various different classifier methodologies including GBM, random forest, and penalized logistic regression. We find that GBM has the best performance. We choose tree_number=200 and tree_depth=5 for the GBM.

As with the start times, we find the most important covariates in predicting fault end time by calculating a score based on...
mean decrease impurity in GBM. In Table 6 we list the top 15 most important covariates and their corresponding score in predicting fault end time for each one of the five fault types in plant 1 and 6. The most important covariate is \( \text{elapsed}_t \), which ranks number one in all cases.

Table 7 shows the percentage of times that each covariate ranked in the top 15 importance score (GBM to predict fault end time) averaged over all plants. \( \text{elapsed}_t \) is the most important covariate. S1-S4 are again more important than R1-R4, and the covariates time, hour, weekday, E2 and E3 are also important features in the models.

4. Conclusion

In this paper, we proposed and implemented a machine learning based algorithm to detect industrial plant faults. The encouraging results demonstrated the usefulness of data-driven algorithms in fault detection of complex systems. Several extensions to our algorithms were considered but not implemented due to the time constraints of the PHM Society Data Challenge. These additional approaches are left as future work.

One such approach would be to not model each plant independently. Alternatively, we could try to first group the plants into clusters of like plants (based on like distributions and/or timing of faults, for example), and then model plants in each cluster together.

Another untried approach is deep learning neural networks. Convolutional neural networks or recurrent neural networks, which have been shown to be powerful tools when modeling with large and complex datasets, may yield good results. Convolutional neural network can automatically consider lagged observations by modeling temporal contiguous observations jointly together. Recurrent neural network can create an internal state of the network which allows it to exhibit dynamic temporal behavior. These facts make deep learning neural networks potentially very useful in fault detection for the PHM data.

Lastly, in our approach, lagged covariates are added to the model which creates high dimensional features. Curse of dimensionality may damnify the classifiers’ performances. Techniques such as principle component analysis and functional data analysis can be applied to extract key features from time series covariates and reduce the feature dimension.

Acknowledgment

The author wants to give thanks to SAS colleges Anya Menguirk, Sergiy Peredriy, Arin Chaudhuri, Alex Chien, De-ovrat Kakde and Gul Ege for their help in the Prognostics and Health Management Society 15 data challenge competition.

References


Table 6. Top 15 most important covariates to predict fault end time for each of five faults in plant 1 and 6.

<table>
<thead>
<tr>
<th>Plant Rank</th>
<th>Plant</th>
<th>elapsed_t</th>
<th>elapsed_t</th>
<th>R1 elapsed_t</th>
<th>L1 elapsed_t</th>
<th>L2 elapsed_t</th>
<th>L3 elapsed_t</th>
<th>L4 elapsed_t</th>
<th>L5 elapsed_t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0.0707</td>
<td>0.1406</td>
<td>0.2616</td>
<td>0.2444</td>
<td>0.1472</td>
<td>0.0636</td>
<td>0.0672</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>0.0285</td>
<td>0.0338</td>
<td>0.0989</td>
<td>0.0374</td>
<td>0.0574</td>
<td>0.0321</td>
<td>0.0342</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>0.0118</td>
<td>0.0121</td>
<td>0.0101</td>
<td>0.0131</td>
<td>0.0118</td>
<td>0.0101</td>
<td>0.0118</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4</td>
<td>0.0155</td>
<td>0.0116</td>
<td>0.0094</td>
<td>0.0074</td>
<td>0.0065</td>
<td>0.0083</td>
<td>0.0065</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>5</td>
<td>0.0110</td>
<td>0.0110</td>
<td>0.0092</td>
<td>0.0104</td>
<td>0.0110</td>
<td>0.0104</td>
<td>0.0110</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>6</td>
<td>0.0093</td>
<td>0.0099</td>
<td>0.0076</td>
<td>0.0088</td>
<td>0.0083</td>
<td>0.0065</td>
<td>0.0083</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>7</td>
<td>0.0076</td>
<td>0.0091</td>
<td>0.0074</td>
<td>0.0085</td>
<td>0.0093</td>
<td>0.0085</td>
<td>0.0093</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8</td>
<td>0.0070</td>
<td>0.0083</td>
<td>0.0065</td>
<td>0.0083</td>
<td>0.0065</td>
<td>0.0083</td>
<td>0.0065</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9</td>
<td>0.0069</td>
<td>0.0079</td>
<td>0.0059</td>
<td>0.0075</td>
<td>0.0083</td>
<td>0.0075</td>
<td>0.0083</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
<td>0.0067</td>
<td>0.0075</td>
<td>0.0056</td>
<td>0.0075</td>
<td>0.0083</td>
<td>0.0075</td>
<td>0.0083</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>11</td>
<td>0.0066</td>
<td>0.0073</td>
<td>0.0055</td>
<td>0.0074</td>
<td>0.0083</td>
<td>0.0074</td>
<td>0.0083</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>12</td>
<td>0.0064</td>
<td>0.0067</td>
<td>0.0051</td>
<td>0.0073</td>
<td>0.0083</td>
<td>0.0073</td>
<td>0.0083</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>13</td>
<td>0.0063</td>
<td>0.0063</td>
<td>0.0048</td>
<td>0.0066</td>
<td>0.0083</td>
<td>0.0066</td>
<td>0.0083</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>14</td>
<td>0.0062</td>
<td>0.0060</td>
<td>0.0046</td>
<td>0.0063</td>
<td>0.0083</td>
<td>0.0063</td>
<td>0.0083</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>15</td>
<td>0.0061</td>
<td>0.0060</td>
<td>0.0046</td>
<td>0.0063</td>
<td>0.0083</td>
<td>0.0063</td>
<td>0.0083</td>
</tr>
</tbody>
</table>

Table 7. Percentage that each covariate rank top 15 in the importance score (GBM to predict fault end time) averaged over all plants.

<table>
<thead>
<tr>
<th>Covariate</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>elapsed_t</td>
<td>98%</td>
<td>95%</td>
<td>93%</td>
<td>93%</td>
<td>95%</td>
</tr>
<tr>
<td>time</td>
<td>94%</td>
<td>89%</td>
<td>79%</td>
<td>79%</td>
<td>63%</td>
</tr>
<tr>
<td>month</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>hour</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>weekday</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>S1</td>
<td>65%</td>
<td>86%</td>
<td>98%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>S2</td>
<td>90%</td>
<td>91%</td>
<td>91%</td>
<td>73%</td>
<td>98%</td>
</tr>
<tr>
<td>S3</td>
<td>89%</td>
<td>82%</td>
<td>20%</td>
<td>12%</td>
<td>14%</td>
</tr>
<tr>
<td>S4</td>
<td>97%</td>
<td>93%</td>
<td>93%</td>
<td>73%</td>
<td>95%</td>
</tr>
<tr>
<td>R1</td>
<td>11%</td>
<td>18%</td>
<td>15%</td>
<td>25%</td>
<td>9%</td>
</tr>
<tr>
<td>R2</td>
<td>2%</td>
<td>2%</td>
<td>4%</td>
<td>7%</td>
<td>5%</td>
</tr>
<tr>
<td>R3</td>
<td>2%</td>
<td>7%</td>
<td>11%</td>
<td>9%</td>
<td>7%</td>
</tr>
<tr>
<td>R4</td>
<td>0%</td>
<td>2%</td>
<td>7%</td>
<td>7%</td>
<td>7%</td>
</tr>
<tr>
<td>E2</td>
<td>22%</td>
<td>33%</td>
<td>35%</td>
<td>30%</td>
<td>16%</td>
</tr>
<tr>
<td>E3</td>
<td>29%</td>
<td>37%</td>
<td>37%</td>
<td>32%</td>
<td>40%</td>
</tr>
</tbody>
</table>
Feature Extraction and Ensemble Decision Tree Classifier in Plant Failure Detection

Cong Xie, Donglin Yang, Yixiang Huang, and Donglai Sun

Maxtropy Technology, Shanghai, 201199, China
xiecong,yangdonglin,huangyixiang,sundonglai@maxtropy.com

ABSTRACT
This paper describes a set of algorithms used to tackle the plant prognostic problem provided in the IEEE 2015 PHM Data Challenge. The task is to detect failure events by analyzing a dataset including sensor measurements and control reference signals of multiple plants without prior knowledge. There are two main difficulties lies in the data challenge. One is to identify which of the faults will occur. And the other is when the fault is going to happen. In this study, the authors tried to transform the task issue into a classification problem by three key steps including: 1) data cleansing and event time alignment; 2) feature extraction; 3) application of the ensemble decision tree classifiers. Results show that the proposed data-driven methods can effectively detect several types of the failure events, which may be promising in the real world plant prognostic applications.

1. INTRODUCTION

In recent years, fault detection (Isermann, 1984; Isermann & Ballé, 1997; Samy, Postlethwaite, & Gu, 2011; Isermann & Isermann, 2011; Gao, Cecati, & Ding, 2015) has attracted increasing attention in both academic and industrial fields. The complexity of plants and large amount of integral subsystems necessitates automated prognostic system, which is used to replace the human supervision. Researchers and engineers now confront challenges which require advanced technologies rather than the traditional model-based methods.

The approaches to fault detection can be generally classified into two categories: model-based methods and data-driven methods. The traditional model-based methods (Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003; Isermann, 2005; Frank, 1990) are based on the physical models of the systems and the related experience and expertise. However, the complexity of modern industrial systems sets obstacles for devising a practical model. Furthermore, in some cases we simply have no idea about the structure of the system. What we have is the mere data recorded from the sensors, which motivates us to use data-driven methods (Yin, Wang, & Karimi, 2014; Schwabacher, 2005; Qin, 2009). Data-driven methods tend to introduce technologies of machine learning and data science to prognostics.

This paper presents the methods developed by the team Maxtropy for the IEEE 2015 PHM Data Challenge. The prognostics topic focuses on the operation of a plant and the capability to detect plant failure events. The dataset is composed of the following three parts:

(a) time series of sensor measurements and control reference signals for each of a number of control components of the plant (e.g. 6 components);
(b) time series data representing additional measurements of a fixed number of plant zones over the same period of time (e.g. 3 zones), where a zone may cover one or more plant components;
(c) plant fault events, each characterized by a start time, an end time, and a failure code.

Only faults of type 1-5 are of interest, while code 6 represents all other faults not in focus. The frequency of measurements is approximately one sample every 15 minutes, and the time series data spans a period of approximately three to four years. The goal is to predict the beginning time and end time of failure events of types 1-5. More detailed information could be found on the website of PHM Society (PHM Society, 2015). The dataset can be downloaded from NASA Ames Prognostics Data Repository (J. Rosca, 2015).

In this paper, data-driven methods are adopted under the circumstance that we have no prior knowledge about the structure or physical characters of the plants. More specifically, we extract several unordered features from the raw data. Pieces of time series are then sliced. Finally, we train classifiers and predict to which kind of fault each piece belongs. We use ensemble decision tree methods, including RF (Random Forest) and GBDT (Gradient Boosting Decision Tree), as the classifiers.
The rest of the paper is organized as follows. In Section 2, we formally define the problem to be solved. The detailed solution of our team is proposed in Section 3. Empirical results are presented in Section 4. Finally, we conclude the paper in Section 5.

2. PROBLEM FORMULATION

In this section, we formally describe the problem in PHM 2015 data challenge.

2.1. Notations

We use bold letters for vectors, and normal fonts for scalars, sets. Bold capital letters are used for matrices. All the vectors in this paper are column vectors. And $S^T$ denotes the transpose of matrix $S$. $1(\text{condition})$ denotes the indicator function, whose value is 1 when the condition is satisfied and 0 otherwise. Each observation is represented as a pair of features and a time stamp: $(x, t)$, where $x = [x_1, \ldots, x_k]^T$ is a $k$-dimensional feature vector and $t$ represents the time when such sample is observed. A fault event is composed of a pair of time stamps and a fault label: $(t_{\text{start}}, t_{\text{end}}, y)$, where $t_{\text{start}}$ is the beginning time of the fault, $t_{\text{end}}$ is the end time of the fault, and $y \in \{1, 2, 3, 4, 5\}$ is the corresponding fault label.

The sequence of observations in dataset $X$ corresponding to a fault event $(t_{\text{start}}, t_{\text{end}}, y)$ is $s = \{(x, t) | (x, t) \in X, t \geq t_{\text{start}}, t \leq t_{\text{end}}\}$.

2.2. Problem Definition

Two groups of data are provided in PHM 2015 Data Challenge: component sensor measurements along with control reference and zone sensor measurements. Although both groups of data are supposed to be useful, we only adopt the component sensor measurements along with control reference to simplify our model. Hence, the raw data have totally 9 features: 1 integer indicates the component number, 4 sensor measurements and 4 control reference. We represent the data of the a specific plant as $X^{\text{raw}} = \{(x_i, t_i) | i \in \{1, \ldots, N\}\}$, where $N$ is the number of observations of the plant. The set of corresponding fault events can then be represented as $E = \{(t_{\text{start}}^i, t_{\text{end}}^i, y_i) | i \in \{1, \ldots, M\}\}$, where $M$ is the number of fault events of the plant. $E^- = \{e = (t_{\text{start}}^i, t_{\text{end}}^i, y) | e \in E, t_{\text{end}}^i \leq T\}$ denotes all the events before $T$. $E^+ = \{e = (t_{\text{start}}^i, t_{\text{end}}^i, y) | e \in E, t_{\text{end}}^i \geq T\}$ denotes all the events after $T$. To simplify the problem, we assume that $E^- \cap E^+ = \emptyset$.

For a specific plant, we have the data $X^{\text{raw}}$ and complete $E^-\cup E^+$ along with a specific $T$. $E^-\cup E^+$ is incomplete. The task is to find all the missing elements in $E^-\cup E^+$.

3. METHODOLOGY

In this section, we propose the solution to the problem defined in Section 2.2.

3.1. Feature Extraction

The first step is to extract useful features from the raw data to facilitate the detection. Note that the fault events can overlap. Hence, we draw an assumption that the indicator of a fault event may involve a subset of all the components. Furthermore, such indicator may be independent on the order of the components.

The simplest way of feature extraction is to connect the raw data vector in a specific order. In particular, raw data vectors are aligned by the time stamps. Observations with nearly the same time stamps are collected into a group and combined into a longer vector, which is the feature vector. The combination is according to a specific order of component number. The component numbers are then useless and can be removed from the feature vector. In addition, some other features can also be combined into the feature vector introduced above. The time series of fault events follow a periodical pattern. Month, day, weekday and hour can be extracted from time stamps. Then features are extracted by getting the maximum, minimum, average and standard deviation of each column of $S$ from observations in the same month. Similarly for the same day, weekday and hour. Another similar aspect is that plant fault events are independent of one another but a fault is possible to be dependent of data i inside a three hour time window before the fault start time. Then features are extracted by getting the maximum, minimum, average and standard deviation of each column of $S$ from observations in the one hour. Similarly for two hours and three hours time window. (Faloutsos, Ranganathan, & Manolopoulos, 1994; Huang et al., 1998)

According to our assumption, the model should be independent on the order of the components. To fulfill this goal, we can simply consider all the permutations of the observations in the same group. All such permutations are then combined into feature vectors. Such way of feature extraction does make sense. Nevertheless, when the number of components of a plant is large, the number of data after feature extraction grows dramatically. For example, if there are 10 components in some plant, $10! = 3,628,800$ permutations should be considered, which makes the problem intractable.

Another way rather than permutes the components is to draw histograms from each dimension of the group of raw data which are aligned by the time stamps and then connect the histograms into a longer vector. Note that this solution can also remove the order of components from the features. However, the bins of the histograms should be carefully selected via inspecting each dimension of the observations. Small
number of bins may eliminate the discrepancy between failures and normal status. Too many bins may result in overfit or bad generalization.

In our solution, a list of all unique values is drawn out of each dimension of raw data. One out of every 8 elements in the list is selected as a bin. A region of higher density of elements also possess more bins.

Now we formally define the feature vector. For a specific sequence of \( k \) observations \( S = \{(x_i, t_i) | i \in \{1, \ldots, k\}\} \), we construct a corresponding \( k \times 8 \) matrix \( S = [x_1, \ldots, x_k]^T \), each row of which is an observation. Note that we assume that the component numbers are already removed from the observations. For a certain dimension \( j \in \{1, \ldots, 8\} \), we have the set of bins \( B_j = \{b_{j,1}, \ldots, b_{j,p_j+1}\} \). The corresponding histogram of the \( j \)th dimension is a \( p_j \times 1 \) vector \( f_j = [f_{j,1}, \ldots, f_{j,p_j}] \), where \( f_{j,i} = \frac{1}{\sum_{b \in \{b_{j,1}, \ldots, b_{j,p_j+1}\}} 1(b_{j,i} \leq S) - 1} \) for \( i \in \{1, \ldots, p_j\} \). The final feature vector of \( S \) is simply the combination of all \( f_j \), which is \([f_{1,1}, \ldots, f_{1,p_1}, \ldots, f_{8,1}, \ldots, f_{8,p_8}]^T\).

3.2. Ensemble Decision Tree Classifier

Ensemble methods are powerful tools for classification. In our solution, we adopt decision tree classifier to predict the type of fault. More specifically, we use RF (Random Forest) and GBDT (Gradient Boosting Decision Tree) as the classifiers.

RF (Random Forest) build an ensemble of classifiers and it takes advantage of two powerful machine-learning techniques: bagging and random feature selection (Liaw & Wiener, 2002; Breiman, 2001). Bagging constructs new training sets by resampling from the original data set by randomly select \( k \) samples. Note that the sample selected will not be removed from the data set in the next draw. Bootstrap sampling technique makes some of the training samples be chosen more than once while some others will not be chosen at all in a new training set. For each training set, instead of using all features, RF randomly selects a random subset of the input features to split at each splitting node when growing a tree. Since only a subset of the features is utilized at each node, computational load of RF is comparatively light. To assess the prediction performance, an out-of-bag (OOB) method can be used. For each training set, one-third of the samples are randomly left out and two-thirds of the samples are used for building a tree. For accuracy estimation, votes for each sample in OOB samples can be used to estimate the performance of prediction. In the end, a simple majority vote is taken for prediction. (Svetnik et al., 2003)

GBDT (Gradient Boosting Decision Tree) is a data mining technique that has achieved considerable success in data mining (Dieterich, 2000; J. Friedman, Hastie, Tibshirani, et al., 2000; J. H. Friedman, 2001, 2002). There are examples of gradient boosting applications in other fields including refinement of classification tree analysis in a remote sensing problem (Lawrence, Bunn, Powell, & Zambon, 2004), discrimination of freshwater residency in a coastal fishery from scales collected from subadult fish (McCulloch, Cappo, Aumend, & Müller, 2005), microscopy image analysis of bread (Lindgren & Rousu, 2002), graphical estimation of a slate deposit (Diener et al., 2004), and calibrating spectroscope measurements of organic chemicals in plant samples (Shepherd, Palm, Gachengo, & Vanlauwe, 2003).

Boosting creates a series of decision trees which together form a single predictive model. Trees are built sequentially from pseudo-residuals, which is the gradient of the loss function of the previous tree. At each iteration, a tree is built from a random sampling of the original data set, producing an incremental improvement in the model. This process is similar to a bootstrap technique in that many trees are generated. With each successive tree, it is hoped that gradient boosting will reduce the error. Using only a fraction of the training data increases both the computation speed and the prediction accuracy, while also helping to avoid over-fitting the data. An advantage of stochastic gradient boosting is that it is not necessary to select predictor variables ahead of time or transform predictor variables. It is also resistant to outliers, as the steepest gradient algorithm stresses points that are close to the correct classification (Moisen et al., 2006; B. Roe et al., 2005).

Algorithm 1 Training of a plant

\[
\text{Algorithm 1}
\]

**Input:** \( X^{\text{raw}}, E^{-T} \), set of bins \( B = \{B_1, \ldots, B_8\} \)

**Output:** A function \( \text{label} := \text{predictor}(\text{sample}) \)

1: \( S = \emptyset \)
2: \text{for all} \( e \in E^{-T} \) \text{do}
3: \text{Extract the feature vector} \( f \) \text{from} \( e \) \text{by using} \( b \)
4: \text{Add} \( f \) \text{into} \( S \)
5: \text{end for}
6: \text{for all} \( e \notin E^{-T} \text{ and before the last event in} E^{-T} \) \text{do}
7: \text{Extract the feature vector} \( f \) \text{from} \( e \) \text{by using} \( B \)
8: \text{Add} \( f \) \text{into} \( S \)
9: \text{end for}
10: \text{predictor} = \text{train_trees}(S)

3.3. Algorithm

Now we can formally describe our algorithms. The solution is mainly separated into two parts: training and prediction. For training, features are extracted from all the fault events in \( X^{\text{raw}} \) indicated by \( E^{-T} \) as well as an adequate number of normal (without any fault) events. All these samples are used to train the classifier. The detailed algorithm is shown in Algorithm 1. Note that \( \text{train_trees} \) can be either RF or GBDT. For prediction, moving windows of different lengths are used to draw events from \( X^{\text{raw}} \) after the last event in
\(E^T\). All such events are labelled by the classifier trained before. Overlapping fault events with same label will then be combined into a single event. Finally we recover the set \(E^T\). Detailed algorithm is shown in Algorithm 2.

Algorithm 2 Prediction of a plant

Input: \(X_{raw}, E^T\), set of bins \(B = \{B_1, \ldots, B_8\}\), predictor, a set of lengths of events \(l\)

Output: \(E^T\)

1: \(S = \emptyset\)
2: An empty label 0 is initially assigned to each event \(e\)
3: for all \(e = (t_{start}, t_{end}, 0)\) after the last event in \(E^T\) of length in \(l\) do
4: Extract the feature vector \(f\) from \(e\) by using \(b\)
5: \(label = predictor(f)\)
6: \(e_{new} = (t_{start}, t_{end}, label)\)
7: Add \(e_{new}\) into \(S\)
8: end for
9: repeat
10: for all \(g \in S\) do
11: for all \(h\ in S\ and \ h \neq g\) do
12: if \(g\ and \ h\ overlap\ and\ have\ same\ label\ then\)
13: Combine \(g\ and \ h\ into\ a\ single\ event \ e_{gh}\)
14: Delete \(g\ and \ h\ from \ S\)
15: Add \(e_{gh}\ into \ S\)
16: end if
17: end for
18: end for
19: until No overlapping events with same label are found
20: \(E^T_+ := S\)

Note that each plant is processed separately according to the way we normalize the histograms, though samples from different plants have the same number of features. Some other kinds of features may prevent us from differentiating samples from various plants, but we will not discuss such details in this paper.

4. EXPERIMENT

In this section, we present the empirical results of our methods. We compare the performance of RF and GDBT. Furthermore, we report the predicting result of each plant according to the scoring criterion.

4.1. Dataset

The dataset we use is presented in Table 1. Totally 32 plant is used. Plant 2 is not included for the missing of many fault events after a certain time point. Fault events of the remaining plants are complete. We separate the data of each plant into two parts: training data and testing data. In Table 1, \# samples 1 and \# faults 1 denotes the numbers of observations and faults before a specific time point, which are used for training. And \# samples 2 and \# faults 2 denotes the numbers of observations and faults after such time point, which are used for testing.

### Table 1. Dataset

<table>
<thead>
<tr>
<th>Plant</th>
<th># samples 1</th>
<th># faults 1</th>
<th># samples 2</th>
<th># faults 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1728</td>
<td>1155</td>
<td>1546</td>
<td>1157</td>
</tr>
<tr>
<td>2</td>
<td>1393</td>
<td>411</td>
<td>3148</td>
<td>442</td>
</tr>
<tr>
<td>3</td>
<td>1724</td>
<td>914</td>
<td>1811</td>
<td>916</td>
</tr>
<tr>
<td>4</td>
<td>574</td>
<td>189</td>
<td>1269</td>
<td>491</td>
</tr>
<tr>
<td>5</td>
<td>1619</td>
<td>1152</td>
<td>2780</td>
<td>1753</td>
</tr>
<tr>
<td>6</td>
<td>3555</td>
<td>2124</td>
<td>3297</td>
<td>2126</td>
</tr>
<tr>
<td>7</td>
<td>1614</td>
<td>1064</td>
<td>1660</td>
<td>1065</td>
</tr>
<tr>
<td>8</td>
<td>917</td>
<td>734</td>
<td>1262</td>
<td>735</td>
</tr>
<tr>
<td>9</td>
<td>748</td>
<td>486</td>
<td>748</td>
<td>487</td>
</tr>
<tr>
<td>10</td>
<td>221</td>
<td>25</td>
<td>78</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>862</td>
<td>466</td>
<td>522</td>
<td>468</td>
</tr>
<tr>
<td>12</td>
<td>2047</td>
<td>1824</td>
<td>2245</td>
<td>1825</td>
</tr>
<tr>
<td>13</td>
<td>1423</td>
<td>936</td>
<td>1671</td>
<td>937</td>
</tr>
<tr>
<td>14</td>
<td>2395</td>
<td>1472</td>
<td>2406</td>
<td>1473</td>
</tr>
<tr>
<td>15</td>
<td>498</td>
<td>238</td>
<td>2311</td>
<td>239</td>
</tr>
<tr>
<td>16</td>
<td>1647</td>
<td>1399</td>
<td>1954</td>
<td>1401</td>
</tr>
<tr>
<td>17</td>
<td>1462</td>
<td>182</td>
<td>788</td>
<td>183</td>
</tr>
<tr>
<td>18</td>
<td>1488</td>
<td>652</td>
<td>2393</td>
<td>653</td>
</tr>
<tr>
<td>19</td>
<td>380</td>
<td>319</td>
<td>1165</td>
<td>320</td>
</tr>
<tr>
<td>20</td>
<td>1820</td>
<td>1275</td>
<td>1525</td>
<td>1277</td>
</tr>
<tr>
<td>21</td>
<td>1401</td>
<td>349</td>
<td>2385</td>
<td>359</td>
</tr>
<tr>
<td>22</td>
<td>382</td>
<td>207</td>
<td>535</td>
<td>208</td>
</tr>
<tr>
<td>23</td>
<td>1180</td>
<td>559</td>
<td>856</td>
<td>561</td>
</tr>
<tr>
<td>24</td>
<td>1977</td>
<td>1136</td>
<td>1248</td>
<td>1138</td>
</tr>
<tr>
<td>25</td>
<td>375</td>
<td>141</td>
<td>187</td>
<td>142</td>
</tr>
<tr>
<td>26</td>
<td>1815</td>
<td>1565</td>
<td>1833</td>
<td>1566</td>
</tr>
<tr>
<td>27</td>
<td>548</td>
<td>415</td>
<td>535</td>
<td>416</td>
</tr>
<tr>
<td>28</td>
<td>951</td>
<td>697</td>
<td>1378</td>
<td>698</td>
</tr>
<tr>
<td>29</td>
<td>1483</td>
<td>1268</td>
<td>1466</td>
<td>1269</td>
</tr>
<tr>
<td>30</td>
<td>925</td>
<td>619</td>
<td>754</td>
<td>620</td>
</tr>
<tr>
<td>31</td>
<td>1561</td>
<td>689</td>
<td>1301</td>
<td>690</td>
</tr>
<tr>
<td>32</td>
<td>1679</td>
<td>1592</td>
<td>1844</td>
<td>1594</td>
</tr>
</tbody>
</table>

4.2. Scoring

In this section, we evaluate the classifiers. We simply extract all the fault events and randomly pick up 2000 normal events out of each plant for testing. The number of ensemble trees is 256 for each classifier. The ensemble method we use for GDBT is called \textit{RUSBoost} in MATLAB. Accuracy, recall and true negative of both algorithms are shown in Figure 1, 2 and 3. Accuracy is the percentage of events which are correctly labelled. Recall is the percentage of fault events which are correctly labelled. True negative is the percentage of normal events which are correctly labelled. RF shows competitive performance in recall. And RF avoids more false positive. Generally, RF overshadows GDBT. Note that for RF, the number of normal events for training must be carefully tuned, while such number affects the performance of RUSBoost slightly.

The scoring criterion is as follows:

\[
Score = TP \times 10 - MC \times 0.01 - FP \times 0.1 - FN \times 0.1, \]

where \(TP\) is the number of faults identified correctly with the start and end time estimated within 1 hour, \(MC\) is the
Figure 1. Classification result: accuracy
Figure 2. Classification result: recall
Figure 3. Classification result: true negative
number of faults identified with correct start and end times but with the wrong fault code, $FP$ is the number of faults identified that did not actually occur in the data, $FN$ is the number of faults in the real data that were not identified.

Table 2. Scoring GDBT

<table>
<thead>
<tr>
<th>Plant #</th>
<th>TP</th>
<th>MC</th>
<th>FP</th>
<th>FN</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>307</td>
<td>1302</td>
<td>14163</td>
<td>616</td>
<td>1579.08</td>
</tr>
<tr>
<td>3</td>
<td>206</td>
<td>57</td>
<td>30028</td>
<td>222</td>
<td>-970.71</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>578</td>
<td>10226</td>
<td>739</td>
<td>-932.28</td>
</tr>
<tr>
<td>5</td>
<td>158</td>
<td>527</td>
<td>36575</td>
<td>146</td>
<td>-2097.37</td>
</tr>
<tr>
<td>6</td>
<td>277</td>
<td>1118</td>
<td>3842</td>
<td>545</td>
<td>2320.12</td>
</tr>
<tr>
<td>7</td>
<td>272</td>
<td>3135</td>
<td>41052</td>
<td>1016</td>
<td>-1518.31</td>
</tr>
<tr>
<td>8</td>
<td>34</td>
<td>257</td>
<td>1315</td>
<td>936</td>
<td>112.35</td>
</tr>
<tr>
<td>9</td>
<td>249</td>
<td>1577</td>
<td>10704</td>
<td>201</td>
<td>2654.73</td>
</tr>
<tr>
<td>10</td>
<td>137</td>
<td>739</td>
<td>13634</td>
<td>289</td>
<td>-31.69</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>14</td>
<td>1569</td>
<td>18</td>
<td>-128.84</td>
</tr>
<tr>
<td>12</td>
<td>93</td>
<td>517</td>
<td>4368</td>
<td>256</td>
<td>462.43</td>
</tr>
<tr>
<td>13</td>
<td>73</td>
<td>1947</td>
<td>7398</td>
<td>1169</td>
<td>-146.17</td>
</tr>
<tr>
<td>14</td>
<td>31</td>
<td>481</td>
<td>2594</td>
<td>740</td>
<td>-28.21</td>
</tr>
<tr>
<td>15</td>
<td>325</td>
<td>835</td>
<td>12262</td>
<td>1016</td>
<td>1913.63</td>
</tr>
<tr>
<td>17</td>
<td>42</td>
<td>294</td>
<td>12036</td>
<td>145</td>
<td>-800.51</td>
</tr>
<tr>
<td>18</td>
<td>42</td>
<td>417</td>
<td>3109</td>
<td>1210</td>
<td>-16.01</td>
</tr>
<tr>
<td>21</td>
<td>26</td>
<td>2</td>
<td>8762</td>
<td>8</td>
<td>-62.76</td>
</tr>
<tr>
<td>23</td>
<td>201</td>
<td>61</td>
<td>13805</td>
<td>318</td>
<td>591.53</td>
</tr>
<tr>
<td>24</td>
<td>93</td>
<td>494</td>
<td>38880</td>
<td>169</td>
<td>-2979.84</td>
</tr>
<tr>
<td>25</td>
<td>52</td>
<td>1078</td>
<td>7417</td>
<td>851</td>
<td>-317.58</td>
</tr>
<tr>
<td>27</td>
<td>82</td>
<td>501</td>
<td>18535</td>
<td>675</td>
<td>562.19</td>
</tr>
<tr>
<td>28</td>
<td>11</td>
<td>224</td>
<td>7467</td>
<td>134</td>
<td>-652.34</td>
</tr>
<tr>
<td>29</td>
<td>127</td>
<td>586</td>
<td>6174</td>
<td>533</td>
<td>613.94</td>
</tr>
<tr>
<td>30</td>
<td>272</td>
<td>1574</td>
<td>5987</td>
<td>368</td>
<td>1448.76</td>
</tr>
<tr>
<td>31</td>
<td>43</td>
<td>252</td>
<td>1717</td>
<td>60</td>
<td>239.78</td>
</tr>
<tr>
<td>32</td>
<td>104</td>
<td>14</td>
<td>6471</td>
<td>941</td>
<td>284.06</td>
</tr>
<tr>
<td>33</td>
<td>111</td>
<td>519</td>
<td>8842</td>
<td>236</td>
<td>197.01</td>
</tr>
<tr>
<td>34</td>
<td>21</td>
<td>140</td>
<td>1421</td>
<td>648</td>
<td>1.7</td>
</tr>
<tr>
<td>35</td>
<td>84</td>
<td>591</td>
<td>2957</td>
<td>992</td>
<td>441.19</td>
</tr>
<tr>
<td>36</td>
<td>204</td>
<td>126</td>
<td>40801</td>
<td>133</td>
<td>-2946.07</td>
</tr>
<tr>
<td>39</td>
<td>3</td>
<td>295</td>
<td>5703</td>
<td>586</td>
<td>-601.85</td>
</tr>
<tr>
<td>40</td>
<td>70</td>
<td>502</td>
<td>3565</td>
<td>1376</td>
<td>-220.88</td>
</tr>
</tbody>
</table>

The result of GDBT is shown in Table 2. And the result of RF is shown in Table 2. It can be seen that RF outperforms GDBT in most cases. Although changing some parameters of GDBT may help to improve the performance, RF works well without much effort of tuning.

The major problem of our solution is that it is still difficult to tell apart the overlapping fault events. And in many cases, a long-time fault event may be detected as many short events, which results in many false positives. A more effective linking strategy is needed. Furthermore, the bins used for feature extraction may not be the optimal one. Further tuning may be necessary.

5. CONCLUSION

In this paper, we have proposed a solution to the problem of the IEEE 2015 PHM Data Challenge. A useful feature extraction technology as well as ensemble decision tree classifiers have been utilized. The empirical results have also been presented. More efforts will be taken in diminishing the false positives in our future work.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and editors for their helpful comments, as well as the organizers of the PHM data challenge competition for their efforts to give the opportunity for us to present the potential solutions to industrial applications.

REFERENCES


Diener, H.-C., Bogousslavsky, J., Brass, L. M., Cimminiello, C., Csiba, L., Kaste, M., ... others (2004). Aspirin and clopidogrel compared with clopidogrel alone after recent ischaemic stroke or transient ischaemic attack in high-risk patients (match): randomised, double-blind, placebo-controlled trial. The Lancet, 364(9431), 331–


Fault Log Recovery Using an Incomplete-data-trained FDA Classifier for Failure Diagnosis of Engineered Systems

Hyunjae Kim, Jong Moon Ha, Jungho Park, Sunuwe Kim, Keunsu Kim, Beom Chan Jang, Hyunseok Oh and Byeng D. Youn

Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Republic of Korea

secutas07@snu.ac.kr, billyhm@gmail.com, hihijang@snu.ac.kr, lunashisun@gmail.com
keunshu@gmail.com, bob1333@naver.com, hyunseok52@gmail.com, bdyoun@snu.ac.kr

ABSTRACT

In the 2015 PHM Data Challenge Competition, the goal of the competition problem was to diagnose failure of industrial plant systems using incomplete data. The available data consisted of sensor measurements, control reference signals, and fault logs. A detailed description of the plant system of interest was not revealed, and partial fault logs were eliminated from the dataset. This paper presents a fault log recovery method using a machine-learning-based fault classification approach for failure diagnosis. For optimal performance, it was critical to be able to utilize a set of incomplete data and to select relevant features. First, physical interpretation of the given data was performed to select proper features for a fault classifier. Second, Fisher discriminant analysis (FDA) was employed to minimize the effect of outliers in the incomplete data sets. Finally, the type of the missing fault logs and the duration of the corresponding faults were recovered. The proposed approach, based on the use of an incomplete-data-trained FDA classifier, led to the second-highest score in the 2015 PHM Data Challenge Competition.

1. Introduction

Failure diagnosis of engineered systems plays a critical role in industrial plant systems. A robust and accurate failure diagnosis system helps prevent fatal accidents, saves costs and increases manufacturing efficiency (Hu, Youn, Wang, & Yoon, 2012; Wang, Wang, Youn, & Lee, 2105; Oh, Han, McCluskey, Han, & Yoon, 2015). Developing a high-performance failure diagnosis system for a particular system requires mainly two kinds of information: (1) a profound understanding of the target system or (2) condition monitoring / fault log data. An ample level of knowledge about system failures (i.e., mechanisms, root causes) can facilitate effective fault diagnosis for industrial plant systems. On the other hand, a significant amount of monitoring / fault log data – if available – can provide excellent information for data-driven diagnosis (e.g., data mining, machine learning). Unfortunately, having thorough knowledge of the target system is nearly impossible in real plant systems in field applications because such systems are composed of numerous components and operate in a variety of conditions (Kim, Hwang, Park, Oh, & Youn, 2014). Therefore, most fault diagnosis methods focus on securing accurate condition monitoring / fault log data. In reality, however, most available data contains incomplete or missing fault logs due to human factors or monitoring systems that provide poor (e.g., obsolete format) data.

The Prognostics and Health Management (PHM) Society addressed the topic of failure diagnosis of industrial plant systems with incomplete failure log data in the 2015 PHM Data Challenge Competition. The problem in this competition was to identify (1) the types of faults, and (2) the start and end times of the corresponding faults. The problem partially reflects real-world situations because failure logs are often missing in actual real-world industrial applications.

Several approaches for failure diagnosis using incomplete data have been researched (Lee, C., Choi, S. W., Lee, J. M., & Lee, I. B. 2004; Negnevitsky, M. & Pavlovsky, V. 2005; Razavi-Far, R., Zio, E., & Palade, V. 2014; Wu, Y., Jiang, B., Lu, N. Y., & Zhou, Y. 2015). Li et al. (2006) introduced a method for dealing with an incomplete data set using data mining based on rough set theory. In Li’s method, a two-stage data mining technique is implemented to extract a diagnostics rule. Li applied the method to a pump system fault diagnosis problem. Marwala and Chakravery (2006) investigated fault classification in structures with incomplete measured data. They proposed a method based on an autoassociative neural network and a genetic algorithm. First, the neural network is trained with the incomplete data and the genetic algorithm is then used to determine missing input values. Yongli et al. (2006) proposed an approach based on Bayesian networks to deal with uncertain or incomplete data for power system...
diagnosis. He et al. (2009) developed robust fault detection for networked systems with communication delay and missing data. He designed a robust fault detection filter for incomplete measurements using H infinity filtering and a Markovian jumping system.

This paper presents the failure diagnosis method used by the SNU-SHRM team and presents the team’s results. The key idea for failure diagnosis is to recover the missing fault log data from the industrial plant system of interest. The rest of this paper is organized as follows. Section 2 defines the Data Challenge problem by describing the data set and its structure. Section 3 shows the analysis of the given dataset and extracts the key ideas of the proposed method. The incomplete-data-trained FDA method, along with features that the team suggests for enhancing accuracy, is presented in Section 4. Section 5 presents the results of the fault log recovery. The paper concludes with a summary of the proposed research and suggestions for future work.

![Schematic diagram](image)

Figure 1. Descriptions of the released data sets

2. PROBLEM AND DATASETS

The problem of the 2015 PHM Data Challenge Competition is described in Section 2.1. The details of the released datasets and the scoring procedures are presented in Sections 2.2 and 2.3, respectively.

2.1. Problem Definition

The goal of the 2015 competition problem is to develop a method that can use incomplete data to (1) determine the type of faults present in the system of interest and (2) predict the start and end times of the faults for unknown industrial plants. The committee provided data sets from 48 plants that included sensor signals and fault logs. Data from 33 plants was complete; however, second half data from the fault logs of 15 plants was partially eliminated in a random manner.

2.2. Description of the Data Sets

As shown in Figure 1, three files for each of 48 plants were released to the participants (Rosca, J., Song Z., Willard, N., & Eklund, N. 2015). Each plant has a different number of components and zones. “File (a)” contains the time series of four sensor signals and four reference signals for N components in that particular plant. The components in a plant are controlled by a feedback loop system. “File (b)” includes the cumulative energy consumption and instantaneous power measured in M zones. “File (c)” contains fault starting times, fault ending times, and fault codes. Each File (c) contains one to six independent fault codes. Among them, fault code 6 is considered trivial as it only includes sensor signals and control references were sampled every 15 minutes. The total time span of data collection of the sample data is approximately three to four years.

2.3. Scoring Process

The score metric is defined as:

\[
\text{Score} = 10 \times N_{TP} - 0.01 \times N_{MS} - 0.1 \times N_{FP} - 0.1 \times N_{FN} \quad (1)
\]

where \(N_{TP} \), \(N_{MS} \), \(N_{FP} \), and \(N_{FN} \) are the number of true positives (TP), misclassifications (MS), false positives (FP), and false negatives (FN), respectively.

The score varies with the number of correct or false predictions. The scoring system awards ten points for true positives. If the fault prediction is placed within the one-hour tolerance of the actual fault time and has the correct fault code, the prediction is accepted as a true positive. The scoring system penalizes misclassifications, false positives, and false negatives. A misclassification corresponds to faults identified with correct start and end times, but with the wrong fault code. False positives indicate faults identified by a submission that did not actually occur in the data. False negatives mean faults in the actual data that are not identified in the submission.

3. DATA ANALYSIS

This section investigates the characteristics of the given data in such a way that the findings in the section can be used for fault log recovery, as proposed in Section 4. The correlation between sensor measurements and reference control signals is identified in Section 3.1. To define the dataset for training a classifier, seasonality analysis is performed in Section 3.2. In Section 3.3, statistical analysis of sensor signals and fault data is shown to identify the distribution of fault durations. In Section 3.4, rule-based fault diagnostics is presented to verify the applicability of machine-learning-based fault log recovery.

3.1. Sensor Data Analysis based on Inference of the System Type from a Physical Interpretation

Specifications and details about the industrial plants were not revealed. Therefore, it was impossible to identify the characteristics of the exact system and the collected data. However, there were some clues from which we could infer...
the type of system. For example, the terms used to describe the problem, including ‘sensor signal,’ ‘control reference,’ and ‘energy consumption’ provided us keywords for a literature review. Thus, we attempted to find a plant system with similar terminology, signals, and operating modes, as described below.

A correlation between sensor measurements and reference control signals was observed in studies of air handling units from heating, ventilation, air conditioning (HVAC) systems, as discussed by Schein (2006). For example, in Figure 2, we recognized that signals from Sensors 2 and 4, which can be related to those from heating and cooling sensors of air handling units (AHU), show behaviors opposite to each other. When the amplitude from Sensor 2 rises, that of Sensor 4 falls. The correlation coefficient between them was almost minus one. Thus, we assumed that these sensor values had an inversely-proportional relationship.

Furthermore, the reference signals operated in a way in which they controlled the valve position or pre-determined temperature, as described by Salsbury (2001). The value of the reference signal usually changed periodically both day and night. From this, we suspected that Sensor 1 could be an object value of the system. In the daytime, it was observed that the magnitude from Sensor 1 fluctuated after that from Reference 4 changed. The instantaneous power consumption value was also related to signals from Sensor 1 and Sensor 2. The value of Sensor 1 has a trend that approximately correlates the Sensor 2 value, especially working time. However, this trend does not hold at the moment of transition from the working time to night time. From this evidence, it is reasonable to assume that Sensor 1 shows the target temperature. Sensor 2 is a representative value of the heating operation function. Sensor 4 indicates the cooling operation function in the temperature regulation system. These findings are used to extract proper features, as outlined in Section 4.2.

3.2. Seasonality Analysis for Sensor and Zone Data along with Fault Data

In Section 3.1, we inferred the type of the system of interest from the perspective of the physical mechanisms. This section investigates the characteristics of a time series in sensors and fault logs that recurs every calendar year. Figure 3 shows a representative example including data from Sensors 1, 2, and 3, and data on instantaneous power for two years. Both sensor magnitudes and instantaneous power have a local minimum in winter, and a local maximum in summer. This predictable pattern existed over a one-year period. For
this reason, it was assumed that the industrial plant data was from a temperature controlling system.

Table 1 summarizes the number of faults for each quarter over three years. Despite the seasonal characteristics in the sensor and zone data, the occurrence of faults does not show a seasonality pattern. Even though some faults frequently occurred in a particular quarter of a year (e.g., F3 in Q2 of 2011), these same faults may not be found in the same quarter of the next year (e.g., F3 in Q2 of 2012). Meanwhile, the opposite situation can also be observed. Even though no fault occurred in one quarter of a particular year (e.g., F3 in Q2 of 2010), a fault may be found in the same quarter of the next year (e.g., F3 in Q2 of 2011).

Based on observations of the seasonality analysis, it was identified that the occurrence of faults does not show seasonality, whereas the sensor data does. This indicates that use of the sensor data from the first year and the corresponding fault logs for training purposes may not be the best solution to accurately detect faults in the second year. The selection of relevant sensor and fault data for training is critical. In Section 4.3, we present our proposed strategy for designing and training a relevant classifier using an incomplete dataset.

### 3.3. Fault Duration Analysis

Analysis of the duration of faults can provide information about the general characteristic behaviors of a faulty condition in a plant. Figure 4 shows the fault duration time for all logged faults contained in the data from the 33 training plants—those with complete data. Similar to the sensor signals sampled every 15 minutes, fault times were also resampled at 15-minute intervals. Figure 4 shows that 15-minute and 60-minute-long faults occur most frequently, accounting for 25% and 12% of all faults, respectively. 77% of all faults were 180 minutes or less in duration, and only 1.45% of all faults lasted longer than a day.

![Fault duration times observed in the sample data](image-url)

In the data-driven approach, the basic assumption for fault classification is that a detectable change in health conditions can be observed from a system of interest. Based on the assumption, the empirical PDF of sensor signals can help distinguish normal conditions from faulty conditions. Using the data collected from sensor signals, empirical PDFs are compared in Figure 5. S1, S2, and P in the left side of the figure indicate the data from Sensors 1 and 2 and the average of the instantaneous power, respectively. The indicators in the first row of the figure represent the number of the plant and of the component, respectively. For example, “P10/C1” means that component one in plant 10 was used for the statistical analysis. The distributions with blue and red colors correspond to normal and faulty conditions, respectively. Visual inspection shows that empirical PDF for sensor signal data from normal and faulty conditions are partially separated in some examples. In the highlighted box shown in Figure 5, for example, normal and faulty conditions in plant 10 were partially separable in terms of data from Sensor 2 and the average of the instantaneous power. In this case, most data under the faulty condition had a Sensor 2 value smaller than 470, and an average of instantaneous power larger than 60.

When the time series of the signal from Sensor 2 and the average of the instantaneous power of plant 10 are analyzed, relevant features can be more clearly found, as shown in Figure 6. Data for “Fault Code 5” is marked with red circles in Figure 6. In this example, the fault was detected around one hour before the following simple rules were satisfied: (1) Sensor 2 data was lower than 470, and (2) the average of the instantaneous power in the zone was greater than 60. Based on the abovementioned rules, a rule-based fault detection was attempted for Fault Code 5 of plant 10, as shown in Figure 7. The faults predicted by the rule are represented by blue cross marks in the figure. As a result, 86.28% of Fault Code 5 events in plant 10 were successfully predicted.

Although the abovementioned rule-based fault diagnostics approach successfully identified the existence of Fault Code 5 in plant 10, it was not generally applicable for other plants. Most plants have an irregular number of components and zones. As a result, there may exist hundreds of rules that define faults of these systems based on the combination of multiple signals from several components and zones. In this case, it is impossible to identify general rules for diagnostics of most faults. Because of this challenge, the following section of this paper presents a machine-learning-based fault log recovery that can substitute for rule-based fault diagnostics.

4. Fault Log Recovery for Failure Diagnosis

In this section, a fault log recovery technique is proposed for failure diagnosis. Relevant features are extracted based on physical interpretation of the data. Then, an FDA-based classifier is proposed to incorporate incomplete data. Finally, the procedures of fault log recovery for failure diagnosis are presented.

4.1. Processing Fault Logs for Machine Learning

The fault logs in the original file consist of start and end times for each fault. We discretized fault log data every 15 minutes to run a machine-learning algorithm. After processing the fault logs in this manner, the logs have discrete values every 15 minutes. For example, for a fault log that starts at 1:00 pm and ends at 1:45 pm, the log is converted into four discrete fault logs corresponding to fault analysis at 1:00 PM, 1:15 PM, 1:30 PM, and 1:45 PM. This process makes fault logs match signal data.
4.2. Feature Extraction based on Physical Interpretation of Datasets

In this section, we introduce features in conjunction with the physical interpretation of sensor signals and fault data. As discussed in Section 3.1, we observed that Reference 4 values were discretized with two values, one and two. The two values repeatedly occur during the day and night. In the given data, more faults occurred during the day, when Reference 4 values are one. Therefore, the relationship between sensor values and faults could be enhanced by multiplying each sensor value by the Reference 4 value, as:

\[ F_{Si,R4} = S_i \times R_4 (i = 1, 2, 3, \text{and } 4) \]  

(2)

Most instances of Fault Codes 2 to 5 happen when Sensor 1 and instantaneous power are high, as mentioned in Section 3.2. This implies that a faulty condition can be separated from a normal condition if data from Sensor 1 and instantaneous power are integrated into a single feature. The ratio of Sensor 1 to the instantaneous power is defined as:

\[ F_{S1,P} = \frac{S_1}{P_{\text{inst}}} \]  

(3)

From (2) and (3), the features, \( F_{S1,R4}, F_{S2,R4}, F_{S3,R4}, F_{S4,R4}, \) and \( F_{S1,P} \) at time \( t \) are shown in a vector form:

\[ F_t = [F_{S1,R4}, F_{S2,R4}, F_{S3,R4}, F_{S4,R4}, F_{S1,P}] \]  

(4)

Equation (4) shows the features at time \( t \). Features at \( t-15, t-30, \ldots \) can be also be presented. In this study, to incorporate the features from the past three hours (180 minutes), the features are stacked like this:

\[ F_{\text{stack},t} = [F_t, F_{t-15}, \ldots, F_{t-180}] \]  

(5)

\( F_t \) consists of five components. Thirteen feature vectors at \( t, t-15, \ldots, t-180 \), thus becomes a 1 by 65 matrix, as illustrated in Figure 8.
incomplete dataset with mislabeled data is used for training, the accuracy of the fault classification will be lower than when using a complete data set.

The use of an irrelevant classifier has more impact on data points with incorrect labels than for those with correct labels. For example, in Figure 9, a widely accepted classifier like support vector machine (SVM) does not provide high classification accuracy when used with the incomplete dataset. The SVM is designed to find a hyperplane that has a good separation ability by making the hyperplane with the largest margin to the nearest training data for each label. With an incomplete dataset, there was the possibility that the support vector was misplaced due to several mislabeled data points. Thus, the hyperplane did not separate the normal conditions from faulty conditions.

Similarly, another popular memory-based learning technique, k-nearest neighbors (KNN) is also not appropriate for training with this incomplete dataset. KNN has a shortcoming in this setting because it is very sensitive to the data’s local structure. If a single mislabeled data point exists in the middle of another label, most of the classification results near the mislabeled data are labeled to the wrong one. Therefore, a classifier insensitive to the existence of mislabeled data should be identified for use in settings like this one, where incomplete data training is needed.

Unlike these previously described classifiers, the Fisher discriminant analysis (FDA) classifier has robust characteristics for working with incomplete data. FDA can classify normal and faulty data while ignoring a small number of outliers, i.e., mislabeled data (Jeon, Jung, Yoon, Kim, & Bae, 2015). This characteristic relies on two facts. First, FDA requires a training set in which only a small portion of the dataset describes faulty conditions. In other words, the training data must consist of a significant amount of normal data and a relatively small amount of faulty data. Second, FDA chooses its separation plane based on each label group’s mean and variance. Thus, even if some faulty data are mislabeled as normal, the mean and variance values do not change much. We believe that this characteristic makes the FDA classifier the most suitable classifier for the given incomplete dataset. Figure 10 shows the robustness of FDA to the incomplete dataset. As shown in Figure 10, the value of the separation plane and the classification accuracy trained from the complete dataset and that derived from the incomplete dataset with a few mislabeled data was almost identical. The separation plane does not change in any significant way between the first and second case, while keeping the level of accuracy.

![Figure 9](image1.png)  
**Figure 9.** Fault classification by radial basis SVM:  
(a) trained using a complete dataset  
(b) trained using an incomplete dataset

![Figure 10](image2.png)  
**Figure 10.** Fault classification by FDA: the results with complete and incomplete datasets are almost identical.

### 4.4. Procedures of Fault Log Recovery for Failure Diagnosis

The procedures for fault log recovery consist of three steps, as shown in Figure 11: (1) feature values are calculated using the training data set. The selection of a proper training data set enhances the separability of the FDA classifier; (2) a trained FDA classifies normal and faulty conditions for the test data set. In this step, incomplete data missing from the fault logs are recovered through the characteristics of FDA and datasets; (3) the adjacent logs are merged and then
converted back into the original log form and divided into one hour units, as explained in Section 3.3. The fault log recovery technique can also be used in real-time condition monitoring to diagnose failure of plant systems.

5. RESULTS AND DISCUSSION

To validate the proposed method, we randomly eliminated half of the faults in the 33 plants with full fault logs, and evaluated the performance of the method using the score metric in Equation (1). On average, it scored 1663 points per plant with 213 TPs and 4483 FPs per plant. 427 faults were eliminated from the plant fault logs and the method correctly recovered about half of the faults for the individual plants. Although FP values are about 20 times larger than TP values, the FP has a less significant impact on the score than does the TP value.

We incorporated the data from the second half of the data (the data with missing fault logs) to recover the incomplete data as well as other data from various plants. It should be noted that the proposed method can be used for fault log recovery of any industrial plant system, and eventually, for failure diagnosis using real-time condition monitoring data.

6. CONCLUSIONS

This study addressed failure diagnosis of industrial plant systems in real applications. The key idea was to recover missing data from incomplete fault logs. The recovery of missing fault data was accomplished through comprehensive analysis of sensor measurements, control reference signals, and fault log data. Data analysis provided correlation between sensor signals and fault logs. A strategy was proposed to recover the missing fault log information and, thus, enable the use of incomplete training data. Compared to other classifiers such as SVM and KNN, the incomplete-data-trained FDA classifier was superior at classifying normal and faulty conditions. The results from the selected features and the FDA-based fault classification method ranked second-highest in the 2015 PHM Data Challenge Competition.

There is a room for further improvement of the proposed method. It would be possible to improve the fault log recovery performance by optimally combining the first and second half of the datasets for use in training the FDA. In addition, it is expected that greater accuracy would be accomplished if more system details become available so that physical interpretation-based features could be defined.

REFERENCES


**BIOGRAPHIES**

**Hyunjae Kim** received his B.S. degree from Seoul National University, Seoul, Republic of Korea, in 2012. He is a Ph.D. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research topic is battery thermal and power management. He received two awards including the IEEE PHM Data Challenge Competition Winner (2014) and the PHM Society Data Challenge Competition Winner (2014).

**Jong Moon Ha** received his B.S. degree from Hongik University, Seoul, Republic of Korea, in 2011. He is a Ph.D. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research topic is prognostics and health management for gearboxes. He received an award including the IEEE PHM Data Challenge Competition Winner (2014).

**Jungho Park** received his B.S. degree from Seoul National University, Seoul, Republic of Korea, in 2012. He is a Ph.D. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research topic is fault diagnostics of planetary gears. He received an award including the PHM Society Data Challenge Competition Winner (2014).

**Suneuwe Kim** received his B.S. degree from Korea University, Seoul, Republic of Korea, in 2014. He is an M.S. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research topic is prognostics and health management for lithium-ion batteries.

**Keunsu Kim** received his B.S. degree from Seoul National University, Seoul, Republic of Korea, in 2013. He is a Ph.D. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research area includes prognostics and health management for machine tools. He received an award including the IEEE PHM Data Challenge Competition Winner (2014).

**Beom Chan Jang** received his B.S. degree from Seoul National University, Seoul, Republic of Korea, in 2014. He is a Ph.D. student at the Department of Mechanical and Aerospace Engineering in Seoul National University. His research topic is prognostics and health management of lithium-ion batteries.

**Hyunseok Oh** is received the B.S. degree from Korea University, Seoul, Republic of Korea, in 2004, the M.S. degree from KAIST, Daejeon, Republic of Korea, in 2006, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2012. He is a Research Professor in the Laboratory for System Health & Risk Management, Seoul National University. His current research area includes prognostics and health management and model verification and validation. He was with the Hyundai MOBIS Technical Research Institute as a Research Engineer from 2006 to 2007. He was a Research Associate at the Center for Advanced Life Cycle Engineering, University of Maryland from 2012 to 2014. Dr. Oh received the A. James Clark Fellowship (2007). He received two awards including the IEEE PHM Data Challenge Competition Winner (2012) and the PHM Society Data Challenge Competition Winner (2014).

**Byeng D. Youn** received the B.S. degree from Inha University, Incheon, South Korea, in 1996, the M.S. degree from KAIST, Daejeon, Republic of Korea, in 1998, and the Ph.D. degree from the University of Iowa, Iowa City, IA, USA, in 2001. He is an Associate Professor of mechanical and aerospace engineering at Seoul National University (SNU), Seoul, Republic of Korea. Before joining SNU, he was an Assistant Professor in the Department of Mechanical Engineering, University of Maryland, College Park. His research goal is to develop rational reliability and design methods based on mathematics, physics, and statistics for use in complex engineered systems, mainly focused on energy systems. His current research includes reliability-based...
design, prognostics and health management (PHM), energy harvester design, and virtual product testing. Dr. Youn’s dedication and efforts in research have garnered substantive peer recognition resulting in four notable awards including the ASME IDETC Best Paper Awards (2001 and 2008), the ISSMO/Springer Prize for a Young Scientist (2005), the IEEE PHM Competition Winner (2014), the PHM society Data Challenge Competition Winner (2014), etc.
Author Index

A
Acuña, David ........................................ 478
Aizpurua, Jose Ignacio ................................. 196
Alessi, Allegra ........................................ 253
Amyot, Normand ..................................... 624
Anger, Christoph ...................................... 543
Ardakani, Hossein Davari ........................... 334
Ardis, Paul A ........................................... 360
Astigarraga, Daniel ................................. 253
Avram, Remus C ...................................... 315

B
Bagheri, Behrad ....................................... 334
Baijwa, Anupa ......................................... 11
Banerjee, Avi .......................................... 697
Baraldi, Piero .......................................... 253
Bechhoefer, Eric ......................................... 138, 590
Belegundu, Ashok ...................................... 154
Beling, Peter A ......................................... 210, 428, 640
Bird, Ross ............................................... 115
Biron, Zoleikha Abdollahi ......................... 711
Biswas, Gautam ........................................ 75
Blancke, Olivier ........................................ 624
Boldyrjew, Roman .................................... 242
Bole, Brian M ........................................... 381
Bortman, Jacob ........................................ 532
Bower, Gregory ......................................... 115
Brandt, Sebastian ..................................... 632
Bregen, Anibal .......................................... 57
Brown, Barbara ........................................ 11, 21, 466
Bruck, Maira A ......................................... 288
Bruton, K .................................................. 343
Buehner, Michael ..................................... 373

C
Cao, Yongcan ........................................... 234
Carter, Jim ............................................... 525
Carter, Matt ............................................. 525
Catterson, Victoria M ................................ 196
Cecilfo, Inês M .......................................... 448
Celaya, José R ........................................... 262, 381
Ceschini, Giuseppe .................................... 612, 632
Chakravarthy, Satyanarayanan R ............ 353
Chen, Yan ............................................... 373
Choi, Joo-Ho ............................................. 553
Choo, Benjamin Y ..................................... 428
Chueh, Chung-Hua ................................... 657
Cohen, Evyatar ........................................ 532
Cruz, José A ............................................. 597

E
Ewanchuk, Jeffrey .................................... 242

F
Fan, Ip-Shing ......................................... 390
Feng, Wenquan ....................................... 75
Fishkin, Alexey .......................................... 632
Flores, Paulina ......................................... 478
Frank, Jeremy ........................................... 154
Fuentes, Pablo .......................................... 478

G
Galarza, Ainhoa ........................................ 253
Goebel, Kai F ............................................ 706
Goel, Nita ................................................ 697
Gollnick, Volker ........................................ 509
Gomez, Robert ........................................... 280
Gorospe, George E ..................................... 706
Goudarzi, Navid ........................................ 288
Grissom, Michael ...................................... 154
Guan, Xiumei .......................................... 75

H
Ha, Jang M ............................................... 553, 618
Ha, Jang Moon ......................................... 736
Haines, Yacov .......................................... 210, 640
He, Yuning ............................................... 580
Heaton, Andrew E .................................... 390
Heddy, Gerald .......................................... 640
Helu, Moneer .......................................... 323
Hernández, Fidel ........................................ 672
Hill, Boyd L ............................................... 381
Hines, J. Wesley ........................................ 650
Hogge, Edward F ....................................... 381
Holboell, Joachim ..................................... 280, 680
Honda, Tomonori ...................................... 223
Hood, Adrian .......................................... 560
Hopkins, Brad .......................................... 363
Huang, Yixiang ........................................... 727
Hudon, Claude .......................................... 624