Online Abnormality Diagnosis for real-time Implementation on Turbofan Engines and Test Cells

Jérôme Lacaille\textsuperscript{1}, Valério Gerez\textsuperscript{2}

\textsuperscript{1} Sncema Algorithm Expert Villaroche, France
\texttt{jerome.lacaille@sncema.fr}

\textsuperscript{2} Sncema System Engineer, Villaroche, France
\texttt{valerio.gerez@sncema.fr}

ABSTRACT

A turbofan used in flight or in a bench test cell produces a lot of data. Numeric measurements describe the performance of the engine, the vibration conditions and more generally the behavior of the whole system (engine + bench or aircraft). It seems reasonable to embed an application capable of health diagnosis. This inboard monitoring system should use very light algorithms. The code need to work on old fashion FADEC calculators (Fault Authority Digital Engine Control) built on a technology dating more than 20 years. Snecma, as an engine manufacturer, has a great knowledge of the engine design and its behavior. This knowledge helps to select the best inputs for a good abnormality detection process, hence limiting the need of a too complex solution. In this article, I describe a very simple anomaly detection algorithm designed for embedding on light computers. This algorithm was validated on a bench test cell running a military engine.

1. THE CONTEXT

During the development process, engine parts or prototypes are tested in a bench cell environment. The engine and the bench itself are connected to many sensors (more than a thousand). Afterward, during standard missions, only a subset of those sensors is kept; but the engine continues to send a lot of messages carrying a potential knowledge about its behavior. Our goal is to fetch a part of this information to be able to detect potential abnormalities. An application to detect anomaly should work during the test process on benches but also during real flights. One of our main concerns is to develop a code that may be implemented on current computers such as FADEC or eventually ACMS (Aircraft Condition Monitoring System). Today’s calculators in use do not have a big computing power, neither much memory. For example, it may be difficult to implement complex signature classification algorithms (Cômes 2010a, 2010b, 2011, Cottrell 2009, Lacaille 2009b, 2011). Most of the work should be done on the fly using mathematic filters with little amount of memory for calibration.

The volume of data available during bench tests is really huge. Analyzing simultaneously too many sensors will damage the quality of a mathematic computation, so we choose to build many small instances of the same algorithm. Each instance deals with a small (but reasonable) amount of measurements; it produces its own diagnostic outputs (detection and anticipation or prognostic). Each separate result is an indication of the behavior of a specific component according to specific faults. All such results are merged together by a higher decision layer. This complex fusion algorithm embeds a selection step which gives a great indication of what detectors (instances) should be implemented for on board work. This article presents the first part of the whole process: the detection layer called CES for \textit{continuous empirical score}.

This work presents a specific code that is both light and efficient. Much solutions proposed to monitor engines and detect abnormalities are built around specific components and the code is generally cut in two parts for embedding acquisition and ground analysis (Lacaille and Nya Djiki 2009, Flandrois 2009, Lacaille 2009a, 2009c, 2011). Other types of monitoring algorithms are fully dedicated to ground analysis; in general they work on a larger time scale using successive flights to detect trends and build prognostics (Cômes 2010a, 2010b, 2011, Lacaille 2011). Those solutions need large databases and even use some
datamining analysis to preprocess the observations (Seichepine 2011). The current proposition is a general detector of unusual behavior and it is built in one standalone application dedicated for embedded systems.

2. THE DETECTION ALGORITHM

2.1 Inputs

Looking at a specific component, one finds probable faults, corresponding signature indicators and running constraints. All this data is reported in the FMECA (Failure Mode, Effects, and Criticality Analysis) document which lists all failure modes of all components of each engine system with corresponding occurrence probabilities. Then for one specific component and a small list of potential faults it is possible to isolate a small amount of indicators divided in two subsets.

- The first subset describes the fault signatures; we call it the endogenous subset.
- The second one gives the constraints or the description of the execution context during which such fault may occur. This second set of inputs is called the exogenous subset because it describes the working conditions that should apply for a valid detection.

The two input subsets are not used the same way: the exogenous subset serves the identification of an acceptable context when the endogenous subset is only studied for abnormality detection when the context is accepted.

Each subset is made of indicators that do not necessary correspond to the raw measurements. A small preprocessing stage should be implemented. We select a set of online linear filters (moving averages and autoregressive filters) with the help of company experts. Eventually, mathematic computations, relations between sensor outputs, are also used in place of the initial measurements (Lacaille 2007). On more powerful computers one may try to mathematically reduce dimensionality using PCA (Principal Component Analysis) (Lacaille 2009c) or other advanced algorithms like the LASSO (Lacaille 2011b).

2.2 Outline

This algorithm is based on a very simple assumption: “most of the time the engine is working under normal conditions; then when something unusual happens, it may be easily detected as an outlier”.

How does it work? Look at a new input, the exogenous part of the input describes the context, then look at what happened for the endogenous indicators when this context “nearly” applies. If the endogenous observation resembles the already observed ones, everything is usual: no abnormality. Otherwise the behavior is unusual: this is an anomaly.

Such algorithm needs some sort of memory to store normal conditions, a distance computation to compute proximity of context observations and a score which is another distance or likelihood to see if an observation is an outlier.

The computation is controlled by quality estimations:

- To define a context as usual or normal, a minimum amount of observations is needed. If the engine is often in such “running context” the quality of the “normal” flag is high otherwise it is not clear. The adequacy measurement computes an indicator of neighborhood. If new measurements are really new, for example “never observed”, the adequacy should be low (distance from current context to other measurements is high); otherwise the adequacy is high when the current and some already observed contexts are similar (distance between current and past context observations is low).
- When the context is clearly identified, the local variance of the endogenous indicators (on similar context/exogenous data) gives a precision indication. This precision value indicates the quality of the outlier detection. When the precision is high, the endogenous indicators should be almost constant for a given running context. Hence the detection of an outlier is easy. However, if the precision is low, the variance of the endogenous measurements is high so the detection is a little fuzzier.

Adequacy and precision are used together to build some global quality indicator.

2.3 Definition of the proximity

In the exogenous domain, the main computation is the proximity. This is a distance between a current (exogenous) context and some stored observations. We will note \( \mathbf{u} \) a vector of exogenous measurements. Let \( \mathbf{u}^* \) be the current observation and \( \mathbf{H}_e \) the set of historic exogenous measurements stored in a little database. The distance between a current \( \mathbf{u}^* \) and any \( \mathbf{u} \in \mathbf{H}_e \) is noted \( d(\mathbf{u}^*, \mathbf{u}) = \| \mathbf{u}^* - \mathbf{u} \|_u \), where the \( u \)-norm corresponds to an Euclidian norm straighten according to the distribution of the exogenous measurements stored in the historic database (Eq. 1).

\[
d(\mathbf{u}, \mathbf{u}^*)^2 = (\mathbf{u}^* - \mathbf{u})^\top \Sigma_u^{-1} (\mathbf{u}^* - \mathbf{u})
\]  

(1)

---

1 High precision corresponds to low local variance.

2 This database will be automatically updated; it is not a temporal buffer but a selection of interesting templates.
Where \( \Sigma_u = \text{cov}(U) \) is the correlation matrix of the stored exogenous context observations\(^4\).

This distance is coded easily without computation of the correlation matrix and its inverse as shown in the following algorithm:

Let \( U = (u_1, \ldots, u_n)' \) be the matrix of all exogenous measurements stored in the database and \( \mu \) be its mean.

Compute \( QR = U-\mu \) the unary and upper triangular decomposition of the centered observations.

Compute \( r = (u^*-U)R \) which will be a rectangular matrix with \( n \) rows (the number of stored observations) and \( m_u \) columns (if \( m_u \) is the dimension of the exogenous vector). As \( R \) is a triangular matrix the computation of \( r \) is straightforward.

Finally make \( d \) the vector of \( n \) rows by computing for each row the Euclidian norm of the corresponding \( m_u \)-dimension vector in \( r: d^2 = \langle r, r \rangle \).

\[
d_i^2(u^*) = \|u^*-u_i\|^2 = \sum_{j=1}^{m_u} r_{ij}^2 \text{ for } i=1\ldots n.
\]

The proximity value is defined as a given quantile of the computed distances \( d_i(u^*) \). Eq. 2 defines \( prx(u^*,H_u) \) as the proximity of \( u^* \) to the history \( H_u \) with percentile parameter \( \rho_u \).

\[
P(d(u,u^*) \leq prx(u^*)) < \rho_u \tag{2}
\]

As the number of observation is finite, this quantile is just an approximation. For example we may select the first value of the “sorted” distances \( d'_{i} < d'_{i+1} \) which realizes the preceding constraint: \( i/n \leq \rho_u \) and \( (i+1)/n > \rho_u \).

### 2.4 Definition of the adequacy

The adequacy is an indicator of novelty according to the exogenous observations. It should increase when new observations are already seen, or equivalently if new observations are common to the observations stored in our database.

We keep a buffer of the last observed data \( B_u \). For each observation \( u^* \) in this buffer its proximity to the history \( H_u \) is \( prx(u^*) \). This list of proximities should be compared to distances accepted in the history. We also have an equivalent list of proximities \( prx(u) \) for all \( u \) in \( H_u \) but each computed with all observations in \( H_u \) excepted the singleton \( \{u\} \). The proximity value has the dimension of a distance so the sum of all squared proximities follows a statistic law equivalent to a \( \chi^2 \).

The ratio \( f \) (Eq. 3) of the two corresponding sums (local distances over normal distances) approximately follows a Fisher law \( F(#B_u,#H_u) \) (where \( #B \) denotes the cardinal of the set \( B \)).

\[
f = \frac{\frac{1}{#B_u} \sum_{u \in B_u} prx^2(u^*)}{\frac{1}{#H_u} \sum_{u \in H_u} prx^2(u)} \tag{3}
\]

The numerator is high if the new observations are far from the stored historic data. The way it is “far”, is normalized by a standard measure of proximity done of normal observation (the denominator).

An adequacy value may be defined as the p-value associated to this statistic test (Eq. 4):

\[
\text{adequacy} = 1 - P(F < f) \tag{4}
\]

### 2.5 Risk of abnormality

Each time a new observation is available, and if the current adequacy is high, the endogenous measurements should be analyzed. We want to compare this new observation with the ones already observed when the context was similar.

We extract a subset of the stored input database with observations close to the current context. This is inferred from the computation of the proximity values. Our subset is the set of historic observations corresponding to a proximity percentile \( \rho_x \) (Eq. 5\(^5\)).

We denote \( H \) the stored set of all historic observations including exogenous and endogenous indicators (\( H_u \) and \( H_x \) are the respective projections of \( H \) on the exogenous and endogenous indicators):

\[
H(u^*) = \left\{ (u^*,x_\ast) \in H / \frac{1}{#B_x} \sum_{u \in H_u} 1_{d(u\ast,\bar{u}; d(u\ast,\bar{u}) \leq \rho_x} \right\} \tag{5}
\]

This subset of the historic storage contains couples of exogenous and endogenous indicators, but it is defined only from computations on context (exogenous) observations.

The score is computed from the likelihood of the endogenous observations according to a local Gaussian law defined empirically by the selected subset of endogenous historic data. If \( x^\ast \) is the current endogenous observation, for each \( x \in H_x(u^*) \), we compute \( d(x^\ast,x)=||x^\ast-x||_k \) as we did previously on exogenous observations:

\[^5\text{ The bold } 1 \text{ here (Eq. 5, 10 and 12) denotes the indicator function.}\]
\[ d(x^*, x)^2 = (x^* - x)^T \Sigma_x (u^*)^{-1} (x^* - x) \] (6)

This time \( \Sigma_x(u^*) \) refers to the covariance matrix of the selected endogenous observations in \( H_x(u^*) \). The \((u^*)\) notation is only there to recall that the subset \( H_x(u^*) \) contains only endogenous measurement that were chosen with approximately similar context.

Finally, for each couple \((u, x)\) in \( H(u^*) \) we have a distance measure \( d(x) = d(x^*, x) \) on exogenous data (Eq. 6) and an equivalent proximity distance \( d_i(u^*) \) on endogenous data (Eq. 1). To take into account the proximity in the score computation we weight the endogenous distances by the context proximity\(^\dagger\):  
\[
\text{score}(u^*, x) = \sum_{(u, x) \in H(u^*)} \frac{d_i^2(u^*)}{1 + d_i^2(u^*)} \] (7)

As this measure has the dimension of a \( \chi^2 \) with \( m_x \) degrees of freedom (\( m_x \) is the dimension of the endogenous vector) the anomaly risk indicator is defined by
\[
\text{risk} = P(\chi^2 \leq \text{score}) \] (8)

### 2.6 Risk precision

The preceding computation gives a risk of abnormality, but this essentially depends on the observations already observed. Hence it is necessary to follow another quality indicator that gives a precision for this result. Our choice is to use an indicator based on the local variance of the endogenous data according to the current context.

Let \( \sigma_j \) be the variance of one of the endogenous observations \( x_j \) on \( H \) and the equivalent \( \sigma_j(u^*) \) on \( H(u^*) \). The ratio of those two variances is a Fisher, so we take the corresponding p-value and the mean on all components \( j \). This is given by (Eq. 9) where generic variable \( F \) is a Fisher stochastic variable with the adequate number of freedom degrees \#\( H(u^*) \) and \#\( H \):

\[
p_j(u^*) = P \left( F < \frac{\sigma_j^2(u^*)}{\sigma_j^2} \right) \text{ for all endogenous variable } x_j \] (9)

and \( \text{precision} = 1 - \frac{1}{m_x} \sum_{j=1}^{m_x} p_j(u^*) \)

This precision value is a number between 0 and 1 that increases when the variance on the local context decreases\(^\dagger\).

### 2.7 Update of the database

Each time a new observation \((u^*, x^*)\) is acquired one first computes the adequacy, the risk and its precision; but the database should also be updated. The observations stored in this database represent the normal behavior of the monitored system. As this set should maintain a small size, one must focus only on the “best” observations and store them as templates.

Hence a test is made to check if the new observation is more relevant than the worse one already stored. If this is the case the new observation replaces the other.

A “worse” observation is selected as the one that is the least useful for our purpose. That’s an observation which may be suppressed from the history without “much” loss in the evaluation of the proximities and risks\(^\dagger\). At first, this observation is selected among the ones (set \( H \)) with the lower values of proximity. A new percentile \( \rho \) is defined for this purpose (Eq. 10):

\[
H^- = \left\{ (u_-, x_-, u) \in H / \frac{1}{|H^-|} \sum_{j=1}^{m_x} 1_{\text{prox}(u_j) \leq \text{prox}(u) \leq \rho} \right\} \] (10)

Then the observation with the lowest risk is selected.

\[
(u^-, x^-) = \arg \min_{(u, x) \in H} [\text{risk}(u, x)] \] (11)

This observation is replaced by the current one \((u^*, x^*)\) if the current context is still unknown (belong to the set \( H^* \) defined by Eq. 12) and if the current score is greater than the “worse” one. The first constraint limits the number of templates belonging to the same context. The second condition ensures that the new added observation corresponds to something really different.

A fourth percentile threshold \( \rho^+ \) is used for the context constraint:

\[
H^+ = \left\{ (u_+, x_+, u) \in H / \frac{1}{|H^+|} \sum_{j=1}^{m_x} 1_{\text{prox}(u_j) \leq \text{prox}(u) > 1 - \rho^+} \right\} \] (12)

\(^\dagger\) We certainly may find a better multivariate solution here. It should take endogenous correlation into account.

\(^\dagger\) As a first implementation of this algorithm, a stochastic metropolis algorithm was programmed to update the database. It adds more freedom and converges to better solutions but the random process makes it difficult to validate on a real time implementation. It was then decided to temporarily replace the stochastic method by a least precise but more easily controllable deterministic rule.
Then the new observation \((u^*, x^*)\) replaces the “worse” one if it belong to \(H^+\) and if \(\text{score}(u^*, x^*) > \text{score}(u, x)\).

At the beginning, the database is initialized with the first observations. Intermediates states of the history \(H\) may be stored to help the engineers understand the behavior of the detector and optimize the configuration thresholds.

3. FIELDED IMPLEMENTATION

There are two reasons for the deployment of HM algorithms in the test benches:

To monitor the installation and the tested machine: in spite of automated and human monitoring of safety parameters, a slow degradation of a body of the machine or the bench cell (engine, gearbox, torque transmissions ...) may lead to a sudden and unexpected failure.

The economic impact may therefore prove prohibitive regarding the program developments underway. Indeed, apart from the exorbitant repair cost, time penalizes programs. It is therefore essential to anticipate such events by deploying a system that allows, not to replace what already exists in terms of real-time monitoring, but defensively to detect any abnormality, known or not, which may lead to a destructive event.

Maturation of algorithms: by definition, the HM algorithms must evolve continuously. Indeed, even if their developments for embedded applications require a high TRL, the unexpected, related to new applications or exceptional operating conditions ever encountered, requires them to evolve in light of this new experience. Similarly, these same algorithms deployed on ground applications need to be matured. This maturation in the test cells is beneficial both for the monitoring of these facilities themselves and the embedded systems.

![Figure 1: Example of a Health Monitoring system deployed in a test bench.](image)

We distinguish the machine from the facility bench because it may be subject to special supervision regardless of the tested machine. The sensors are installed on the machine and on the equipments of the bench. Depending on their type, signals they deliver are digitalized and stored in databases at:

- High frequency for dynamic measurements issued by the accelerometers, displacement sensors, strain gauges, microphones and unsteady pressure sensors.
- Low frequency for temperature, static pressures, rpm measurements and dynamic signals processed. In the example using the CES algorithm, these types of parameters are exploited.

In the following sample application we focus on the health of a shaft for transmitting torque from low frequency settings. (CES algorithm, left branch of Figure 1). Other implementations of the LF algorithm are also under investigation (Lacaille 2010b). For HF implementations see (Klein 2009) and (Hazan 2010a and 2010b).

3.1 Current benchmark

One accelerometer and one thermocouple are mounted on the ball bearing which bears the shaft to be monitored. One room microphone is located in the test cell near the bench equipment. Sensors measure the radial displacement of the shaft relative to a fixed structure.

The context parameters which influence the above parameters are:

- The shaft speed, calculated from the signal of a phonic wheel linked to the rotating shaft.
• The position of an air intake valve which directly affects the torque felt by the shaft.
• The torque measurement itself.
• The pressure of a piston chamber which loads axially the ball bearing of the shaft and hence affects its dynamic response.

Conditioning systems of sensors and storage bases can be either specific or common to several benches.

A supervisor computer, in turn:
• Hosts and executes the algorithms from a dedicated development and maturation platform (Lacaille 2009b, 2010b).
• Receives and manages the high and low frequency data.
• Generates alert messages.

3.2 Implementation
The algorithm consists of standard modules, all customizable and available from a tools library of the development and maturation studio (see implementation graph on Figure 2).
• The “Read File” module reads the low frequency data files as they arrive in a file directory managed by the supervisor.
• In this example, the “Average” module splits each original signal into indicators using moving averages. This has the effect of smoothing the original signal.
• The “Instances” blocks contain the heart of the CES algorithm and are run in parallel.
• The “Demultiplexing” module separates endogenous and exogenous parameters.
• The CES module delivers the abnormality risk and a quality indicator computed as the geometric mean of the previously computed adequacy and precision.
• The “Message” module issues anomalies whenever the adequacy is above a given threshold (confidence regarding the current situation) and when the risk exceeds another threshold. (A third threshold limits the number of detections waiting the risk to down-cross its value before launching a new alert.)

3.3 Algorithm’s instances
In the implementation, each instance of parameters is dedicated to a special supervision such as:

Change of the dynamic behavior of the transmission shaft: in this case (Figure 3), the endogenous parameters are by-revolution tracked signals vibrations levels in low frequency computed from the bearing’s accelerometer and sensors that directly measure the radial displacement of the shaft. These levels are representative of the dynamic response of the shaft according to its imbalance, which can itself be a consequence of geometric imperfections or thermal expansion.

Change of the mechanical behavior of the ball bearing: (Figure 4) the endogenous parameters are low frequency signals coming from the bearing temperature and energy levels by frequency bands. The energy levels are computed from the wideband signals of the accelerometer and the room microphones. For this instance setting, assumption is made that whether a gradual bearing spalling occurs, levels of endogenous parameters will increase (energy levels in specific frequency bands).
For these two instances, the context is given by the shaft speed, the position of an air intake valve and the torque transiting through a transmission shaft between the output of a multiplier box and the machine under test.

It is worth noting an additional context parameter for the instance in Figure 3 compared to Figure 4: the piston pressure of the axial loading for the ball bearing which has a direct influence on the tracked levels coming from the accelerometer and the displacement sensors.

The amount of endogenous parameters is not exhaustive and is linked to the amount of sensors that can be much greater than what is presented for those two instances.

(A third instance shown on Figure 2 is not developed here.)

3.4 Experimental observations

Early in the test campaign, when the algorithm starts from zero, the adequacy was never up and fluctuates as the context data have never been encountered. However, when the adequacy exceeds a preset threshold, it stays high. At this point when the risk reached the detection threshold, a message is generated (the star on Figure 5).

The algorithm finished its calibration when no new templates are added to the database. The adequacy keeps the maximum value and almost all observed detections correspond to acknowledged anomalies.

4. CONCLUSION

In this article we described a light algorithm able to detect unusual behavior of a system made from an engine and/or a bench cell. The original point in this code is the management of the input data as a couple of sensors subsets dedicated to the context identification and the monitoring itself. We may also quote the way the detection is controlled by different quality indicators taking into account both the context identification and the precision of the estimation. This algorithm was first installed on a bench for maturation of the code but also to monitor the bench.

An offline test was build for statistic computation of the main performances indicators for such detection algorithm. This test was build from data recorded during a machine test bench of 15 days. We registered all real problems observed during the test (such as stall) and we add synthetic degradations based on expert knowledge. The process was repeated 28 times with random positioning of the simulated degradation. A cross validation scheme was applied: it gives a false alarm rate of less that 1% (with a precision of ± 4%) and a detection probability of more than 55% (± 20%).

The definitions of the KPI are given below:

\[
PFA = \frac{P(\text{Healthy} | \text{Detected})}{P(\text{Detected})} = \frac{P(\text{Healthy})}{P(\text{Healthy})} \quad (13)
\]

which gives

\[
PFA = \frac{\alpha(1-P(\text{Faulty}))}{\alpha(1-P(\text{Faulty}))+ (1-\beta)P(\text{Faulty})} \quad (14)
\]

and

\[
POD = P(\text{Detected} | \text{Faulty})= 1- \beta \quad (15)
\]

where \( \alpha \) is the type I error and \( \beta \) the detection test power (Lacaille 2010a).

The PFA result corresponds to the requirements for such algorithm on bench test application, however the POD is a little low but is greatly improved by the fusion layer and an optimization of the threshold parameter is in progress.
NOMENCLATURE

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACMS</td>
<td>Aircraft Condition Monitoring System</td>
</tr>
<tr>
<td>AQV</td>
<td>Adequacy Quality Value</td>
</tr>
<tr>
<td>CES</td>
<td>Continuous Empirical Score</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>FADEC</td>
<td>Fault Authority Digital Engine Control</td>
</tr>
<tr>
<td>FMECA</td>
<td>Failure Mode, Effects, and Criticality Analysis</td>
</tr>
<tr>
<td>HM</td>
<td>Health Monitoring</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>LASSO</td>
<td>Least Absolute Shrinkage and Selection Operator</td>
</tr>
<tr>
<td>PFA</td>
<td>Probability of False Alarm</td>
</tr>
<tr>
<td>POD</td>
<td>Probability of Detection</td>
</tr>
<tr>
<td>TRL</td>
<td>Technical Readiness Level</td>
</tr>
</tbody>
</table>

NOTATIONS

- \( \mathbf{u} \): Vector of context (exogenous) indicators
- \( \mathbf{x} \): Vector of endogenous indicators
- \( m_u, m_x \): Dimensions of exogenous and endogenous vectors
- \( \mathbf{H} \): History storage database
- \( (\mathbf{H}_u, \mathbf{H}_x) \): DB projection on respectively exogenous and endogenous indicators
- \( (\mathbf{u}^*, \mathbf{x}^*) \): Current observation
- \( (\mathbf{u}^-, \mathbf{x}^-) \): “Worse” observation in the DB (the least useful observation)
- \( \mathbf{H}(\mathbf{u}^*) \): Observations in the neighbor of the current context
- \( prx \): A quantile distance to the current history DB
- \( adequacy \): Confidence to be already observed
- \( risk \): Probability of abnormality
- \( precision \): Reliability of the risk value
- \( d(\mathbf{u}), d(\mathbf{x}) \): Component of the proximity (distance to one observation of the DB) in exogenous or endogenous projection
- \( \rho_u \): Percentile threshold for the definition of the proximity value
- \( \rho_x \): Percentile threshold for the definition of the context neighborhood
- \( \rho^- \): Percentile threshold for the selection of the “worse” stored observation in update process of the DB
- \( \rho^+ \): Percentile threshold for context replacement constraint in update process
REFERENCES


(Hazan, 2010b) A. Hazan, M. Verleysen, M. Cottrell, J. Lacaille, Linear smoothing of FRF for aircraft engine vibration monitoring, ISMA 2010, Louvain.


(Lacaille 2011b) J. Lacaille, E. Côme, Sudden change detection in turbofan engine behavior. In proceedings of the 8th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, Cardiff, UK.

(Seichepine, 2011) N. Seichepine, J. Ricordeau, J. Lacaille, Datamining of flight measurements. In proceedings of AIAA@Infotech 2011, Saint Louis, MO.

Jérôme Lacaille is senior expert in algorithms for Snecma. He joined the company in 2007 with responsibility for developing a health monitoring solution for jet engines. Jérôme has a PhD from the Ecole Normale Supérieure, France in Mathematics. 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.

Valerio Gerez is a mechanical engineer who works for Snecma since 1982. He has almost 30 years of experience in aircraft engines in the areas of quality and especially in Engine dynamics, both in test cells and in Aircrafts. In 2006, he joined the Diagnostic and Prognostic Department and now manages R&D HM projects for Snecma future applications and the deployment of algorithms in test cells.