An Adaptive Anomaly Detector used in Turbofan Test Cells

Jérôme Lacaille[1], **Valério Gerez**[2], **Rafik Zouari**[3]

[1] *Snecma Algorithms Expert, Villaroche, France*
jerome.lacaille@snecma.fr
[2] *Snecma System Engineer, Villaroche, France*
valerio.gerez@snecma.fr
[3] *Safran Engineering Services, Montigny, France*
rafik.zouari@safran-engineering.com

ABSTRACT

Airplane engines use sophisticated technologies to improve their efficiency, reduce their weight, reduce fuel consumption, limit NOx generation and reduce the generated noise. On another hand, airlines want to decrease their maintenance costs. These changes may have an effect on engine reliability and there is a greater need to understand and control the behavior of the engine. This is the goal of PHM algorithms. However if such algorithms are "easy" to build, V&V stay a challenge. To increase their readiness level, Snecma, as engine manufacturer, tests all engines on bench cells during development phases and before reception. Now Snecma chooses also to use PHM algorithms on bench tests. It helps the maturation of the code itself but it is also a way to monitor the bench cells.

The present document describes an implementation on a partial bench test cell of a generic abnormality detector. The first section gives an outlook at the implementation of some algorithms on a real test cell. The second section is the description of the main algorithm: an online abnormality detector able to automatically update when new recurrent usual observations appear. Finally the last section sketches some results obtained during the execution of the algorithm.

1 ARCHITECTURE

During a bench test, the engine (or engine tested components) as well as the test cell itself are monitored using a wide set of sensors. During some tests procedures in development phases we may register more than a thousand sensors, including performance measurements (pressures, temperatures, flows, gauges...) up to 100Hz, high frequency dynamic measurements (tip timing, accelerometers, microphones) up to 50kHz, and some context information that describes the test procedure.

Snecma's test cells are monitored with specific SPC (Statistic Process Control) tool that is able to register each sensor at different acquisition frequencies and presents in real time graphs with alert bounds.

This SPC software also uploads all data to real-time databases.

Our PHM (Prognostic and Health Monitoring) application is connected to the control system. Data is transfered from the control system to the monitoring system in real-time. Afterward the HM system benefits of the database's storages, hence it is possible to redo computations in the lab.

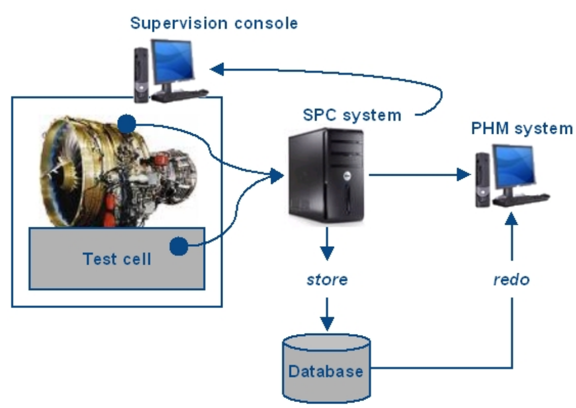

Figure 1: Deployment. The PHM algorithms are on a specific computer and communicate with the control system and its database.

1.1 The currently implemented algorithms

They are in number of three. One of them uses low frequency inputs. It is looking for performance abnormalities, and is instantiated many times to cover the biggest set of patterns known to experts. The two others are high frequency vibration anomaly detectors. One is looking at known patterns of bearing abnormalities and the other is seeking for unknown fleeting events.

The performance abnormality detector is of our main concern in this article. Giving some contextual inputs, it automatically infers the current testing conditions, and according to this specific category of test context, it searches for an unusual behavior of some endogenous variables. Each measurement is observed at a given scale. The patterns of the curves and their concurrent relations are explored. An abnormality is detected when a curve pattern appears unusual according to the other curves in the current contextual conditions. This detector uses an **unsupervised classification algorithm** to identify current operating conditions, a **curve compression algorithm** to automatically identify curve patterns and a **scoring algorithm** made of two parts: a normalization according to local context and a fault identification based on a dependency matrix and a Bayesian fusion for high level decision. This last scoring algorithm was presented in (Lacaille, 2009c). At least, a prognostic is proposed, based on the detection of increasing score trends.

Due to the different possibilities to select the curves and the scales at which one wants to explore the patterns, a lot of configurations are possible. The current application exploits the help of Snecma experts to select some configurations that may be representative of particular system faults[1].

The bearing abnormality detector searches for very specific anomaly patterns. Each pattern is identified by a set a vibration pointers corresponding to frequencies in order domain for a given operational mode or shaft speed (acceleration, deceleration or stationary at given speeds). The way the pointers are linked together for a given operating mode, a specific bearing component, and an acquisition sensor may be complex.

This algorithm was described in (Klein, 2009). Each pattern is compared to a normal baseline producing scores for each bearing components. It looks for specific operating conditions and merges the results obtained during a test procedure for different sensors to confirm detection. This method is able to detect abnormalities days before a bearing breaks; thus it is launched only once the test fulfilled, in general at the end of the day. As it is not possible to record all information for a late analysis during the night, one first detects the interesting operating conditions to store only the usable parts of the signal.

The fleeting event detector looks for vibrations anomalies like FOD (Foreign Object Damages) and rotor/stator contacts. It computes in real time some frequency energies corresponding to specific bands and looks for impact patterns.

This algorithm scans the high frequency signals in real time.

[1]A future work will provide a solution based on information theory to automatically select the best configurations using some complex agent based genetic algorithm.

1.2 The process scheduler

Each algorithm is embedded into an executable process with a common interface. This interface allows a supervisor program to create and manage instances of each algorithm (fig. 2). The supervisor launches the algorithms and ensures that the computations are working fine. Errors are intercepted and managed; recovery is implemented. The algorithms communicate with the supervisor using a message scheme. When an anomaly is detected, a message is sent to the supervisor that is able to inform the SPC system.

This implementation deploys the algorithms on a specific computer (the HM system) different from the one running the test cell control procedures (the control system). In the future we thought of deploying very mature solutions on the control computer, but for V&V (Verification and Validation) purpose this method seems sufficient even if some communication delay may be considered.

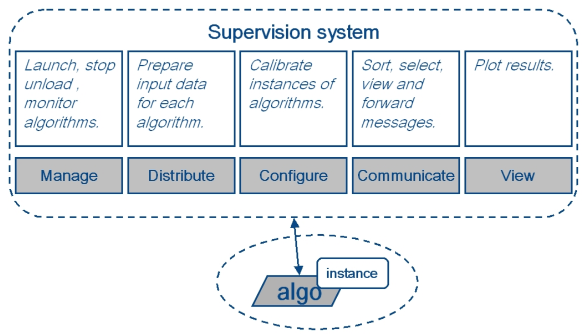

Figure 2: The supervision system manages the algorithm's instances. Each instance runs a process with its specific configuration of the algorithm.

1.3 Anomaly messages

The message content is not only a boolean indication of abnormality. Each message contains a **risk probability** but also a quality information corresponding to a **precision** of the result (PQV: Prediction Quality Value) and a global **adequacy** of the current observations (AQV: Adequacy Quality Value) to the score model (Lacaille, 2010). Let U be stochastic vector of exogenous (contextual) measurements like bench control, speeds, and let X be the vector of parameters to monitor (for example, shaft torque, vibration energy, exhaust gas temperature, pressure...). At instant t, if the observations are (u_t, x_t), then the risk, precision and adequacy are defined as follow:

$$\left\{ \begin{array}{rcl} \text{Risk}(t) & = & 1 - \text{P}(X = x_t | U \approx u_t) \\ \text{Precision}(t) & = & \text{tr}[\text{var}(X|U \approx u_t)] \\ \text{Adequacy}(t) & = & P(U \approx u_t) \end{array} \right. \quad (1)$$

The adequacy is the probability that the current context measurements resemble to some context observation seen previously. It may be computed as the likelihood of the context observation if one has a model of the context stochastic law.

The $\approx$ sign in the equations means that for robustness purpose the computation smooths the notion of equality because we may assume that the behavior of the engine is continuous near the measurement point and that one can consider also observations with very similar contexts. An empirical point of view is to constrain the statistics to past observations with similar contexts defined by a classification of operating modes.

The precision in equation (1) is nothing else but the local dispersion of the endogenous parameters when the context is near the current one.

According to the adequacy computation, the supervisor knows when it should launch new calibration of models. Some algorithms may also be able to automatically update by themselves (see section 2.2 and results one figures 10 and 11).

A general quality indicator is defined from combination of precision and adequacy. Each algorithm will also use risk computation and quality information to send messages to the supervisor. Such alarm message is raised when the risk crosses a minimum threshold and when the measured quality is over another threshold. In the following equation the continuous g function is the CDF (Cumulative Distribution Function) of the precision law. In practical applications[2], we assume that it is possible to model this law with a χ^2.

$$\begin{aligned} \text{Quality}(t) &= \text{Adequacy}(t)\times \\ &\qquad (1 - g(\text{Precision}(t))) \\ g(\rho) &= \text{P}_{\chi^2}(\text{Precision} \leq \rho) \end{aligned} \quad (2)$$

To get more information about messages, the supervisor may also ask each algorithm to present a view of the corresponding results. The plot method is specific for each algorithm, thus this is the charge of the algorithm itself to locally store enough data when an abnormality is detected. Through the supervisor, the expert asks some display to investigate each detection.

2 GENERIC ANOMALY DETECTOR

The generic anomaly detector detects unusual behavior. It analyzes observations according to the context of acquisition. The first step of the algorithm is to classify the context measured from exogenous data into different **operating modes** (OM). (Typically we detect 5 to 15 different operating conditions during one specific type of test.) The exogenous data, for operational application when the engine is under the wing, corresponds to aircraft attitude, flight configuration, bleeds and pilot commands. During the test they are the result of maneuver, rotation speed commands, and other input configurations. Once an operating mode isolated, a local diagnostic is applied, followed by a prognostic on a given horizon (fig. 3).

[2]The precision is computed as a variance of compressed coefficients that will be computed as regression residuals in a scoring algorithm (see section 2.3). The scoring algorithm use the hypothesis of Gaussian residuals, hence the χ^2 model.

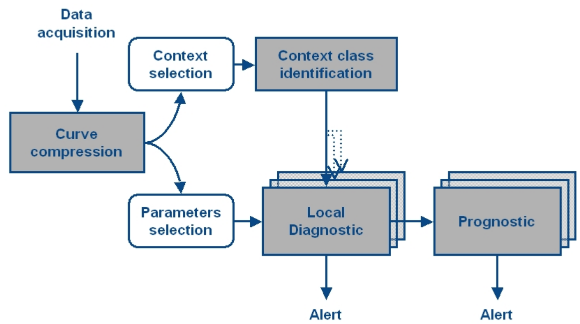

Figure 3: The diagnostic algorithm is applied with models defined for each context class. A context class is identified with an OM.

To make the classification possible, as well as the diagnostic, the input data must be converted to indicators. The system works in real time: at every instant one registers a small curve (20 seconds to 10 minutes) of past measurements for each parameter (exogenous or endogenous). Each curve is observed at a given scale and is compressed into pattern combination. The selection of the scale and the patterns on which to project the curve is part of the input configuration of the algorithm's instance.

2.1 The compression of curves

Any given number of input parameters may be used for this algorithm. Each input is registered at a given frequency. Then it is smoothed and sub-sampled according to the scale one wants to analyze the curve (fig. 4). The compression algorithm works curve by curve and produces pattern indicators. All indicators for all different inputs and different scales are joined together to form a big input vector.

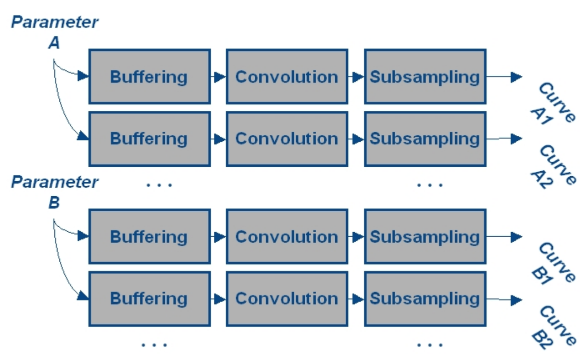

Figure 4: A scaling process, multiple and different for each parameter is applied before curve the compression.

For one parameter, the scaling process is defined by a buffer of size n, a sub-sampling rate r and averaging polynomial filter a of rank p: $a = [a_0, a_1 \dots a_p]$. At time t, if x_t is the observed parameter, then the selected curve is defined by $\mathbf{y}_t = [y_t, y_{t-r} \dots y_{t-(n-1)r}]$ where $y = a * x$ is the convolution of x by filter a:

$$y_t = \sum_{i=0}^{p-1} a_i x_{t-i} \tag{3}$$

The simplest (linear) compression algorithm uses a decomposition in singular values (Mardia *et al.*, 1979). The first step is a normalization of each curve. Let $\bar{\mathbf{y}}_t = (\mathbf{y}_t - \mu_t)/\sigma_t$, where for each instant t, μ_t and σ_t are the mean and standard deviation of all the curve points (computed with the n points of curve $\mathbf{y}_t$)[3]. Each normalized curve can be projected on a well chosen orthonormal basis $\{v_1, v_2 \ldots v_n\}$ where each v_i is a vector of length n that may be interpreted as *a curve template*.

$$\mathbf{y}_t = \sum_{i=1}^{k} \alpha_{t,i} v_i + \epsilon_t \tag{4}$$

In equation (4), $\alpha_t = [\alpha_{t,1}, \alpha_{t,2} \ldots \alpha_{t,k}]$ is the vector of projection coefficients for the curve $\bar{\mathbf{y}}_t$ on the orthonormal basis. The compressed vector of indicators is $\tilde{\mathbf{y}}_t = [\mu_t, \sigma_t, \alpha_t]$ (mean and variance must be seen like any indicator). Finally ϵ_t is the residual error of compression when only the first k base vectors v_1 to v_k ($k \geq 0$) are used.

The selection of template curves (v_i) and size k optimizes the projection by minimization of the error norm $\|\epsilon_t\|$. The calibration procedure to find the most representative set of templates that minimizes the error of projection is obtained by principal component analysis (PCA) of a set $\{\bar{\mathbf{y}}_{t_1}, \bar{\mathbf{y}}_{t_2} \ldots \bar{\mathbf{y}}_{t_N}\}$ of normalized curves that were selected at N instants $t_1, t_2 \ldots t_N$ randomly chosen.

Let $Y = [\bar{\mathbf{y}}_{t_1} \bar{\mathbf{y}}_{t_2} \cdots \bar{\mathbf{y}}_{t_N}]'$ be the $(N \times n)$ matrix formed by all those curves juxtaposed so that each line of the matrix corresponds to one curve. The singular value decomposition (SVD) of symmetric $(n \times n)$ matrix $Y'Y$ is

$$Y'Y = VSV' \text{ with } V'V = I \text{ and } S \text{ diagonal.} \tag{5}$$

The diagonal elements of S are the singular values of $Y'Y$: $(\sigma_1^2 \cdots \sigma_n^2)$ which may be sorted in decreasing order. The columns of V form an orthonormal basis and the first columns are a good choice for the basis (v_i) because in that case the error norm of the compression for each $\bar{\mathbf{y}}_{t_i}$ will be less than the sum off all unselected singular values.

However it may not be the case for any normalized curve $\bar{\mathbf{y}}_t$ different from the ones used in calibration phase. So the best value for k (and for the selected columns of V) is obtained with a cross-validation method. Our choice is the leave-one-out algorithm: for any curve $\bar{\mathbf{y}}_{t_i}$ in the learning set, one computes matrix V_i and S_i using all curves except the i^{th}, and one selects the minimal k_i as the number of first columns of V_i that let the compression error on curve i under a given threshold. Finally, the value of k is chosen as the maximum values of all k_i: $k = \max_{i=1\cdots n} k_i$.

[3] If a signal is constant, it's certainly coming from an unconnected sensor, then no compression is needed, only the mean is relevant.

2.2 The unsupervised classification

For a given configuration (input selection and scale), $\tilde{\mathbf{u}}_t$ will denote the exogenous, context, vector of indicators[4]. The classification algorithm automatically selects an operating mode for the observation vector.

- During a calibration phase, a first learning-set of input vectors $\{\tilde{\mathbf{u}}_{t_1}, \tilde{\mathbf{u}}_{t_2} \ldots \tilde{\mathbf{u}}_{t_N}\}$ is used to define a first set of classes (or initial OM).
- During execution, each new input may be classified or not. If no class corresponds to input, the vector is memorized until enough similar inputs are detected, hence making it possible to build a new class. Regularly, recent inputs (corresponding to already defined classes) are also stored for further update of any class according to a slow evolution of the system (fig. 5).

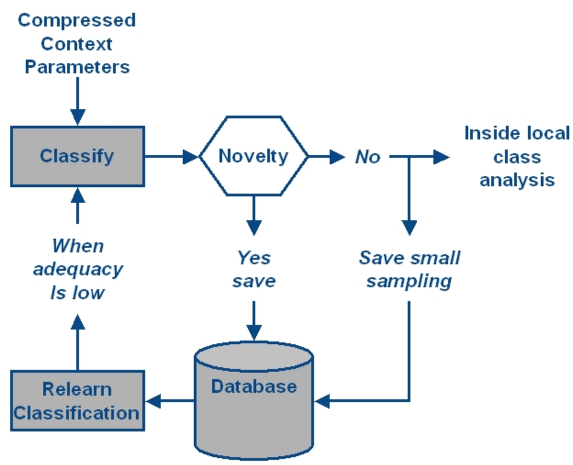

Figure 5: Recalibration of the classification algorithm. When the classification cannot target a known operating mode, it is a novelty and the input data is memorized. When it recognizes a given OM, only a small amount of new observations are memorized with part of the original data that were used to calibrate the algorithm. Finally, when the adequacy is low, a relearning procedure is executed.

The classification algorithm in use there is the EM (Expectation-Maximization) method for the identification of a mixture of Gaussian densities (Dempster *et al.*, 1977; Bilmes, 1998). Our hypothesis is that the stochastic vector of compressed indicators $\tilde{U}$ follows a normal distribution inside each operating mode. The EM algorithm is an iterative process that converges to the model coefficients of each Gaussian law (mean, variance and mixture a priori in the space of compressed indicators).

This algorithm is initialized with a first number of classes. In fact we use a loop on the number of classes (fig. 6) and select the one that optimizes the BIC

[4] According to the compression method, the temporal index t may only take multiple values of the sub-sampling ratio.

(Bayesian Information Criterion) which is approximately the conditional likelihood of the observations (here the exogenous context data) knowing the number of estimated parameter (or the number of classes) (Akaike, 1978).

$$BIC = -2\log(\mathcal{L}(\tilde{\mathbf{u}}_1 \dots \tilde{\mathbf{u}}_N)) + K\log(N) \quad (6)$$

where

$$\begin{cases} \mathcal{L}(U) & = & \text{likelihood of observations} \\ K & = & \text{number of classes} \\ N & = & \text{number of observations} \end{cases}$$

and the likelihood $\mathcal{L}$ is exactly what is optimized in the EM classification. The use of this criteria limits the risk of over-parameterization bounding the number of classes.

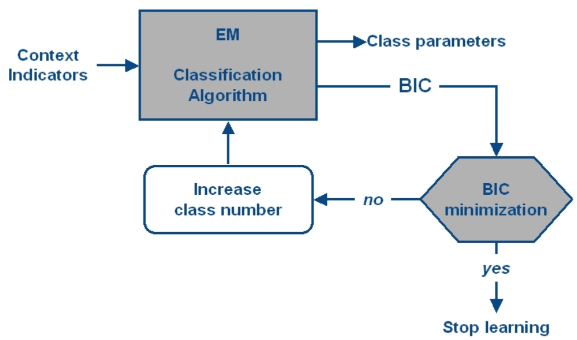

Figure 6: The classification learning algorithm. A loop that minimizes the BIC criterion selects the best number of initial classes. Inside each loop the classification algorithm EM is also an iterative process.

Once a class is recognized for a given input, the process may continue: in our case it will be a new intra-class compression of the original input curves followed by the scoring algorithm (fig. 7). The new compression uses only curves that belongs to the same operating mode, hence the corresponding indicators become specific and more sensitive.

After the scoring, a **prognostic algorithm** detects trends in the risk computation and estimates a probability to cross a threshold at a future horizon.

2.3 The scoring algorithm

The scoring algorithms was already described in (Lacaille, 2009c). It is designed in two blocs:

- Context removal and normalization (CRN) suppresses the local dependency of the exogenous context and replaces each measurement with the difference of the observation to an estimation[5].
- Fault detection and identification (FDI) is the main scoring algorithm that computes the likelihood of the vector of previous residuals according to a normal behavior.

[5]Even inside a unique OM, the normalization gets rid of the local variations of context and uses dependencies between endogenous parameters

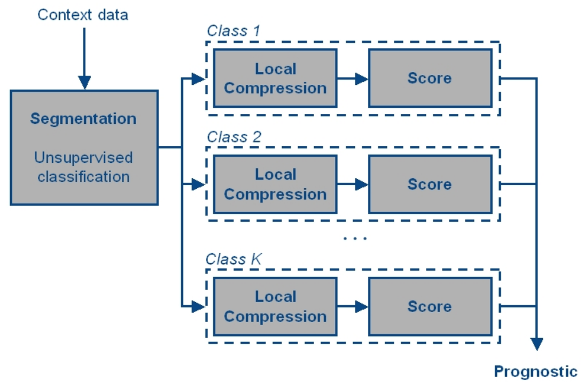

Figure 7: Separate scoring processes execute for each operating mode. Once the operating mode detected by classification algorithm, the process continue with a more sensitive compression of the input curves (local to OM) and a scoring.

Let $x = [x_1, x_2 \dots x_p]$ be the measurement indicators to analyze and $u = [u_1, u_2 \dots u_q]$ be the vector of compressed context indicators. The normalization phase uses an estimation model (a regression) to replace each x_i by a corresponding residual $y_i = x_i - f(u, x_{-i})$ where f is the regression function and $x_{-i} = [x_j; j \neq i]$ is the vector of all endogenous indicators except the i^{th}.

The regression function is calibrated on a set of inputs. We use the indicators coming from the same set of curves that were used to calibrate the corresponding OM class. The regression model is a generalized linear model (GLM). It is a sort of three-layers perceptron: the last layer is a linear projection and the intermediate layer is the result of the selection of specific and physical computations[6].

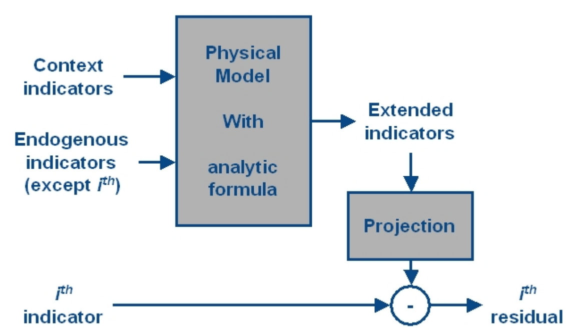

Figure 8: The generalized regression uses the knowledge of the experts to propose a model for each endogenous indicator. This knowledge is given either by explicit physical formulation or with a numeric model of physical behavior.

[6]This is not a kernel method, instead it uses mechanical expert knowledge.

When a system expert in Snecma analyzes a specific engine component it searches for usual faults that may be described analytically by a set of formula or modeled with a finite element program. Our intermediate layer comes from selection of members of theses equations or result obtained from the physical simulation. A cross-validation procedure helps to select the meaningful indicators and get rid of outliers during the calibration.

The fault and identification part produces a global score of the previous residuals. It is also able to classify the different damages according to Bayesian rules but we didn't implement this feature yet for bench test cells. Our main goal is to detect unusual behavior, the identification part will be addressed in the future. The score model makes the assumption that the vector of residuals is Gaussian with zero mean. Then the calibration of the scoring algorithm is limited to the computation of the covariance matrix $\Sigma = \text{var}(y)$, which is computed with the same previous set of observations OM by OM. Finally the global score is the Mahalanobis distance $y'\Sigma^{-1}y$ and the risk probability is obtained using the corresponding χ^2 distribution with the adequate freedom degrees.

3 RESULTS

The application is tested on a Snecma test cell where sensors register either data from the engine and from the test cell itself. The measurements to analyze are dynamic strength gauges for a specific shaft. The corresponding exogenous context measurements are the shaft speed, input air pressure, and temperature. Figure 9 gives some examples of input data. The commands (top line) show the different maneuvers executed by the pilot. Some of the endogenous corresponding observations are on the bottom line. The algorithm detects unusual behaviors of endogenous measurements according to the executed maneuvers.

The scaling process extracts time intervals of 20 seconds at 10Hz and compress all the segments the same way.

The figures 10 and 11 (page 9) present the risk, precision and quality computed by the algorithm.

On figure 10 is the beginning of the execution (first two hours of execution). The red curve (-o-) is the adequacy result (AQV); at the beginning this quality indicator is very low, then when the application starts to learn new observations it increases. The blue line which stays most of the time on the bottom is the risk probability; and the green one (between 0.7 and 0.8) corresponds to the precision (PQV) which is almost stable in our case because the same amount of observations is used at any time.

The clear blue (cyan) vertical dashed lines represent the repartition of observations stored in the database as learning set (0 corresponds to the oldest data and 1 to the new ones at any instant). During the first hours of treatment the diagnostic often detects some abnormalities, but as soon as those observations are common they are added to the learning set and will not be diagnosed as unusual again.

After less than 3 hours, the system stabilizes and the adequacy raises. The algorithm doesn't add a lot more observations in the database and less unusual behavior are detected. However one sees some crisis around 14000s (4h) and a short detection just before 21000s (5h30). These detections need investigation by experts.

4 CONCLUSIONS

Using PHM algorithms on bench test cells is a great way to validate the codes. It is also a way to understand how the system will behave once embedded on aircrafts but with the possibility to maintain the code without the cost of certification.

The currently deployed solution (detection of unusual behavior on transitory measurements) needs a lot of improvement in term of configuration. The challenge now that we are in a position to implement and test this diagnostic is to validate FMECA specific to system components and known faults.

But to begin with this application, we want to reach some confidence from the test engineers. Answering their need for a supervision of the test cell itself, and be able to detect abnormalities without much false alarms. Thus the next point is to defined some key performance indicators (KPI) like false alarm rate (PFA) and probability of detection (POD).

ACKNOWLEDGMENTS

This work is a collaboration of the PHM team and the bench test department of Snecma; without such collaboration, this work should not have been possible.

NOMENCLATURE

BIC	Bayesian Information Criterion
CRN	Context removal and normalization
EM	Expectation-Maximization
FDI	Fault Detection and Identification
$FMECA$	Failure Modes, Effects and Criticality Analysis
FOD	Foreign Object Damage
GLM	Generalized Linear Model
HM	Health Monitoring
IQR	Inter Quantile Range
KPI	Key Performance Indicator
OM	Operating Mode
PCA	Principal Component Analysis
PFA	Probability of False Alarm
PHM	Prognostic and Health Monitoring
POD	Probability Of Detection
SPC	Statistic Process Control
SVD	Singular Value Decomposition
$V\&V$	Verification and Validation

REFERENCES

(Aaron, 2004) C. Aaron. Algorithme em et classification non supervisée. In *ACSEG*, 2004.

(Akaike, 1978) H. Akaike. A bayesian analysis of the minimum aic procedure. *Ann. Inst. Statist. Math.*, 30, part A:9–14, April 1978.

(Ausloos and all., 2009) A. Ausloos and all. Estimation of monitoring indicators using regression methods; application to turbofan start sequence. In *ESREL*, Prague, Poland, 2009.

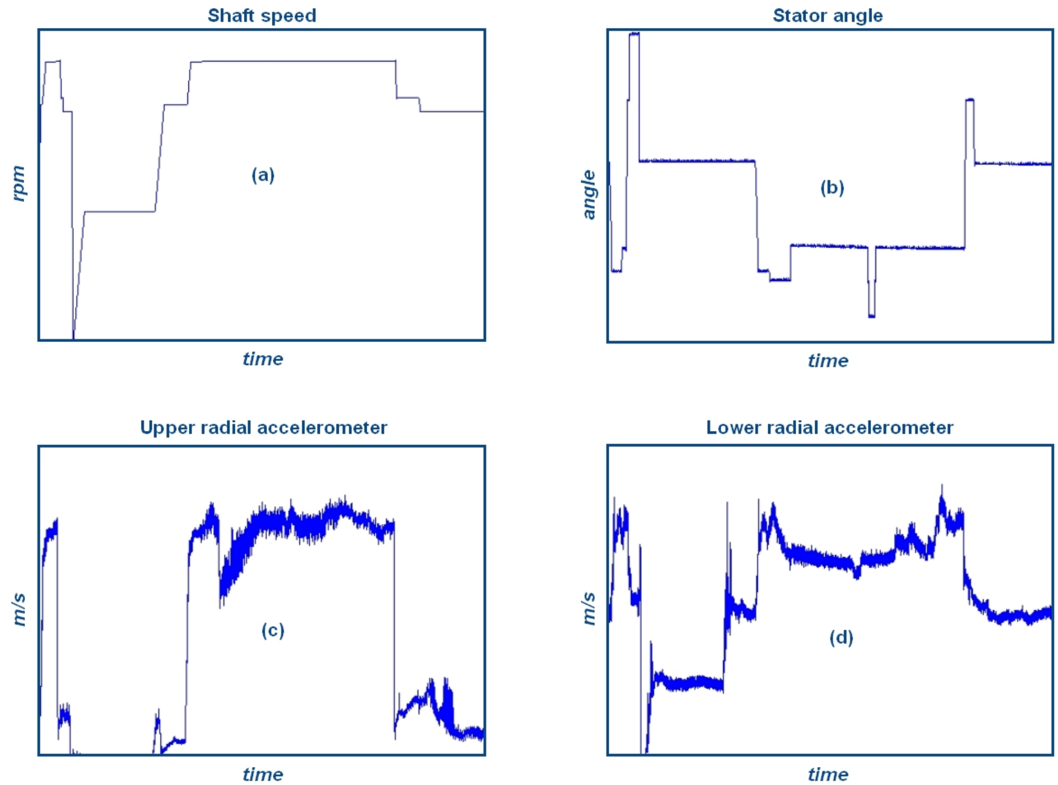

Figure 9: Some inputs. The top line graphs (a) and (b) show context commands: (a) core shaft speed and (b) bleed angle. On the bottom line are measurement sensors: (c) and (d) give radial accelerometers outputs.

(Azencott, 2003) R. Azencott. A method for monitoring a system based on performance indicators, 2003. U.S. Patent 6594618B1.

(Basseville *et al.*, 2009) Michèle Basseville, Laurent Mevel, and Rafik Zouari. Variations on cusum tests for flutter monitoring. In *Proceedings of the 2nd International Workshop in Sequential Methodologies (IWSM)*, Troyes, FR, June 2009.

(Bilmes, 1998) J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, April 1998.

(Blanchard *et al.*, 2009) S. Blanchard, J. Lacaille, and M. Cottrell. Health monitoring des moteurs d'avions. In *Les entretiens de Toulouse*, France, 2009.

(Cômes *et al.*, 2010a) E. Cômes, M. Cottrell, M. Verleysen, and J. Lacaille. Aircraft engine health monitoring using self-organizing maps. In *ICDM*, Berlin, Germany, 2010.

(Cômes *et al.*, 2010b) E. Cômes, M. Cottrell, M. Verleysen, and J. Lacaille. Self organizing star (sos) for health monitoring. In *ESANN*, Bruges, 2010.

(Cottrell and all., 2009) M. Cottrell and all. Fault prediction in aircraft engines using self-organizing maps. In *WSOM*, Miami, FL, 2009.

(De Troyer *et al.*, 2008) Tim De Troyer, Rafik Zouari, Patrick Guillaume, and Laurent Mevel. A new frequency-domain flutter speed prediction algorithm. In *Proceedings of ISMA2008 - Noise and Vibration Engineering Conference*, Leuven, BE, September 2008.

(Dempster *et al.*, 1977) A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical society*, 39(1):1–38, 1977.

(Flandrois and Lacaille, 2009) X. Flandrois and J. Lacaille. Expertise transfer and automatic failure classification for the engine start capability system. In *AIAA Infotech*, Seattle, WA, 2009.

(Hartmann, 1998) W. M. Hartmann. *Signals, Sound, and Sensation*. Springer-Verlag, New York, 1998.

(Haykin, 1994) S. Haykin. *Neural Networks, a Comprehensive Fondation*. MacMillan, IEEE Press, 1994.

(Hazan *et al.*, 2010) A. Hazan, M. Verleysen, M. Cottrell, and J. Lacaille. Trajectory clustering for vibration detection in aircraft engines. In *ICDM*, Berlin, Germany, 2010.

(Klein, 2009) R. Klein. Model based approach for identification of gears and bearings failure modes. In *PHM Conference*, San Diego, CA, 2009.

(Lacaille and Dubus, 2005) J. Lacaille and H. Dubus. Defectivity analysis by a swarm of intelligent distributed agents. In *AEC/APC*, Palm Spring, CA, 2005.

(Lacaille and Nya Djiki, 2009) J. Lacaille and R. Nya Djiki. Model based actuator control loop fault detection. In *Euroturbo Conference*, Graz, Austria, 2009.

(Lacaille and Zagrebnov, 2006a) J. Lacaille and M. Zagrebnov. Building a robust model for process control using advanced mathematical techniques. In *AEC/APC tutorial*, Aix en Provence, France, 2006.

(Lacaille and Zagrebnov, 2006b) J. Lacaille and M. Zagrebnov. A statistical approach of abnormality detection and its applications. In *AEC/APC*, Denver, CO, 2006.

(Lacaille and Zagrebnov, 2007) J. Lacaille and M. Zagrebnov. An unsupervised diagnosis for process tool fault detection: the flexible golden pattern. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):355–363, 2007.

(Lacaille, 2004) J. Lacaille. *Industrialisation d'algorithmes mathématique*. Habilitation à driger des recherches, Université Paris 1, Sorbonne, 2004.

(Lacaille, 2005) J. Lacaille. Mathematical solution to identify the causes of yield deterioration - a defectivity data based solution with an emergent computing technology. In *ISMI*, Austin, TX, 2005.

(Lacaille, 2006) J. Lacaille. Advanced fault detection. In *AEC/APC tutorial*, Denver, CO, 2006.

(Lacaille, 2007) J. Lacaille. How to automatically build meaningful indicators from raw data. In *AEC/APC*, Palm Spring, CA, 2007.

(Lacaille, 2008) J. Lacaille. Global predictive monitoring system for a manufacturing facility, 2008. U.S. Patent 20080082197A1.

(Lacaille, 2009a) J. Lacaille. An automatic sensor fault detection and correction algorithm. In *AIAA ATIO*, Hilton Beach, SC, 2009.

(Lacaille, 2009b) J. Lacaille. A maturation environment to develop and manage health monitoring algorithms. In *PHM*, San Diego, CA, 2009.

(Lacaille, 2009c) J. Lacaille. Standardized failure signature for a turbofan engine. In *IEEE Aerospace Conference*, Big Sky, MT, 2009.

(Lacaille, 2010) J. Lacaille. Validation of health-monitoring algorithms for civil aircraft engines. In *IEEE Aerospace Conference*, Big Sky, MT, 2010.

(Mardia *et al.*, 1979) K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

(Vapnik, 1995) V. N. Vapnik. *The Nature of Statistical Learning*. Springer Verlag, NY, 1995.

(Zouari *et al.*, 2008a) Rafik Zouari, Tim De Troyer, Laurent Mevel, Michèle Basseville, and Patrick Guillaume. Flutter monitoring using a mixed model-based and data-based approach. In *Proceedings of ISMA2008 - Noise and Vibration Engineering Conference*, Leuven, BE, September 2008.

(Zouari *et al.*, 2008b) Rafik Zouari, Laurent Mevel, and Michèle Basseville. An adaptive statistical approach to flutter detection. In *Proceedings of the 17th IFAC World Congress*, pages 12024–12029, Seoul, KR, July 2008.

(Zouari *et al.*, 2008c) Rafik Zouari, Laurent Mevel, Andrzej Klepka, and Michèle Basseville. Adaptive flutter monitoring using wavelet filtering and recursive subspace-based detection. In *Proceedings of the 4th European Workshop on Structural Health Monitoring*, pages 1137–1144, Cracow, PL, July 2008.

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 École 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.

Rafik Zouari is a method engineer in Safran Engineering Services. He is on charge since 2008 of the design and validation of statistical and signal processing based algorithms for engine health monitoring. He received the engineer diploma in Signal Processing, from the Tunisia Polytechnic School, in 2004 and the Master degree in Mechanics from the University of Technology of Compiegne, France, in 2005. He has a PhD in 2008 from the Institute for Research in Computer Science and Random Systems in Rennes, France, entitled "Early detection of aeroelastic instability on aeronautical structures". On his research activities, Rafik has published several papers on statistical detection algorithms and modal identification with industrial applications on aeronautics.

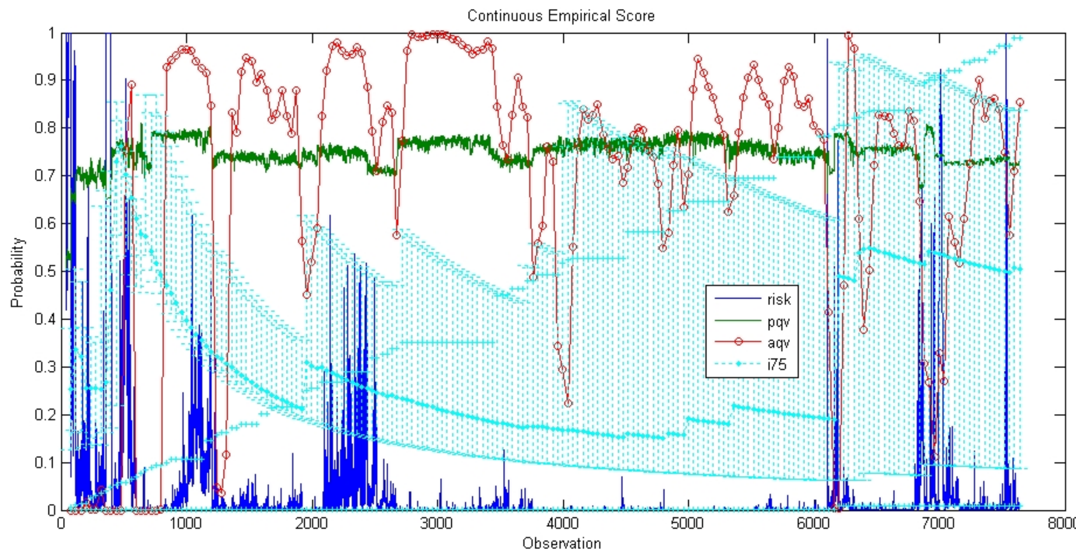

Figure 10: Beginning of some experimentation results. The time unit is in seconds. The red curve (-o-) corresponds to the adequacy. It is low at the beginning then increases progressively when the system learns. The risk (bottom blue line) shows some variation, but mostly because the system is not completely functional. The precision (top green line) is almost stable (around 0.75). Its main variations identify the changes of operational conditions. Finally the cyan vertical lines represent the interquartile range (IQR) of the distribution of time index of the learning set (0 are oldest data and 1 are the latest). The supplementary points for each IQR represent the position of the first and last measurements included in the database.

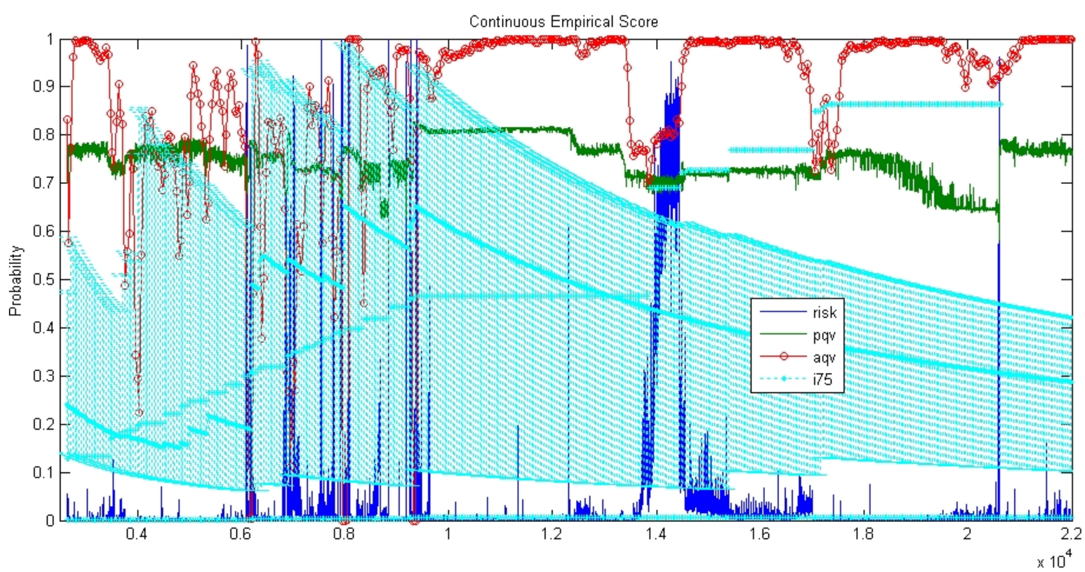

Figure 11: Six hours of test. This figure is similar to the preceding one (fig. 10) but the time axis spanned to a little more than 6h. When the process stabilizes after less than 3 hours (9500s), the adequacy stays up and the system mostly detects unusual behavior at time 14000s (4h) and before 21000s (5h30).