A Batch Detection Algorithm Installed on a Test Bench

Jérôme Lacaille[1], Valerio Gerez[1]

[1]*Snecma, 77550 Moissy-Cramayel, France*

jerome.lacaille@snecma.fr
valerio.gerez@snecma.fr

ABSTRACT

Test benches are used to evaluate the performance of new turbofan engine parts during development phases. This can be especially risky for the bench itself because no one can predict in advance whether the component will behave properly. Moreover, a broken bench is often much more expensive than the deterioration of the component under test. Therefore, monitoring this environment is appropriate, but as the system is new, the algorithms must automatically adapt to the component and to the driver's behavior who wants to experience the system at the edge of its normal domain.

In this paper we present a novelty detection algorithm used in batch mode at the end of each cycle. During a test cycle, the pilot increases the shaft speed by successive steps then finally ends the cycle by an equivalent slow descent. The algorithm takes a summary of the cycle and works at a cycle frequency producing only one result at the end of each cycle. Its goal is to provide an indication to the pilot on the reliability of the bench's use for a next cycle.

1. INTRODUCTION

This document follows two previous articles published in 2010 and 2011 in the PHM Society. The first one (Lacaille & Gerez & Zouari, 2010b) presents the health-monitoring architecture we deploy on one of our test benches and gives clues about adaptation to context changes in the use of the machine. We proposed an algorithmic solution using simultaneously an autoadaptive clustering algorithm and local detection tools calibrated on each cluster. In the second paper (Lacaille & Gerez, 2011c) a lighter solution based on similitude computations and nearest neighbor algorithms was given. This implementation was essentially given to be embedded in the FADEC computer of the engine. In fact the algorithms used on test benches are also good prototypes for online solutions. It's why a fast solution needed to be developed to check if it could also work on dedicated hardware when the engine is installed under an aircraft wing.

Those two previous propositions deal with online abnormality detection: during the execution of the test. They essentially detect fleeting events that suddenly appears without more warning. This paper presents a solution for an off line analysis. The algorithm was already implemented in a lighter form on operational data broadcasted via SatCom (as ACARS messages) to the ground. This limited version of the algorithms was partly described in (Lacaille, 2009c), the current proposition deals with automatic detection of stationary levels, building of temporal snapshots, analysis of the ground database of such snapshots with a clustering algorithm to detect recurrent configurations, and the novelty detection algorithm. Figure 1 shows the OSA-CBM decomposition of each layer of the algorithm.

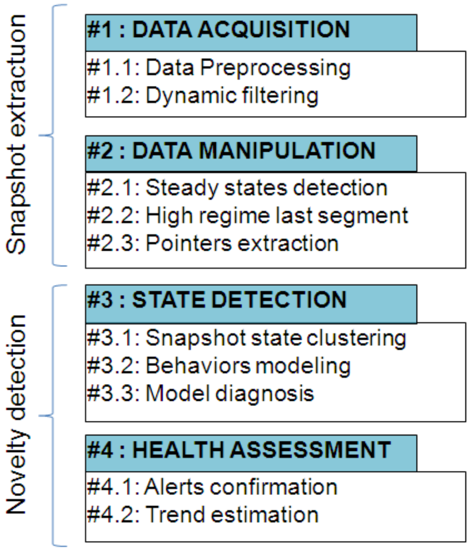

Figure 1: OSA-CBM architecture of the algorithm.

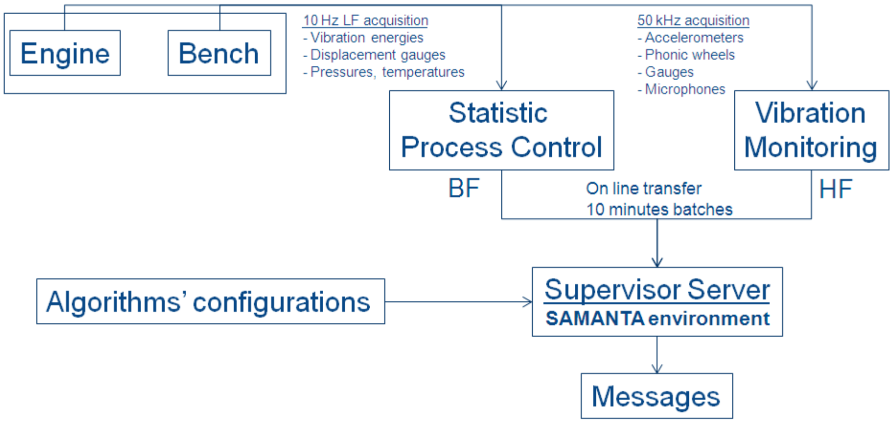

Figure 2: Deployment of the PHM system on a distinct server.

The health-monitoring algorithms are developed by Snecma under the SAMANTA platform which was previously described in (Lacaille, 2009b). This environment industrializes blocs of mathematic processing tools in graphical units. Aeronautic engineers are able to exploit each mathematic module to build their own specific solutions. In our case this method uses signal filters to prepare the data, a stationary process detector, a clustering tool, some regression and dimension reduction algorithms to help normalize observations and make them as much independent of the acquisition context (bench cell driving) as possible. For the novelty detection part, it uses a score computation, a threshold based configurable statistic test and a diagnosis confirmation tool. The two first OSA-CBM layers (mostly signal filtering) are assembled apart from the rest of the algorithm. They produce an off line database of snapshots which is processed by the learning phase of the analysis layer (#3). The results from the data-driven part and the statistic test in last layer (#4) produce the diagnosis.

SAMANTA platform embed also some automatic validation layers (see Lacaille, 2010a and 2012) that helps compute key performance indicators (with precision) using cross-validation schemes.

2. Context of application

A turbofan development test bench is subject to lots of changes in behavior. The interaction between pilot and engineers is really tense and the system may be stopped at any instant if some analyst finds an abnormality in the observations. The sensors are directly broadcasted to observations consoles and validated numeric solutions may launch alarms. The health-monitoring goal is not to stop the process but to provide information about the health of the test bench itself (or the tested engine part, but we give a lot more attention to the bench which is more expensive and less damageable than the tested prototypes).

2.1. Implementation in the test cell

To minimize interactions between the driving of the system and the PHM algorithms we implement an execution driver of our SAMANTA platform on a separate server with a local memory buffer able to deal with days of high frequency acquisition data (50kHz) and weeks of low frequency acquisition (10Hz) and enough storage space to manage a big database of snapshots (some data vectors per cycle – with one or two cycles each day).

2.2. Reliability computation

The PHM algorithms should present computation results with a minimum of reliability because we don't want to interrupt an expensive test experiment scheduled for weeks or months with bad reasons. Hence a very important attention is given to the false alarm performance indicator (PFA). The other indicator we follow is the probability of detection (POD). It is a lot easier to compute because we have some past logbooks on which all historical events where recorded. The main job in that case was to label those (handwritten) data and to compute the detection rate on past tests.

The PFA indicator is given by Eq. (1). If one writes $\mathrm{P}(Detected)$ the probability that an abnormality is detected by the algorithm, $\mathrm{P}(Healthy)$ the probability that the system is healthy. Then the false alarm rate is just the probability that the system is healthy but that an abnormality is detected. It is represented as the following conditional probability:

$$PFA = \mathrm{P}(Healthy|Detected) \qquad (1)$$

This is clearly different from the usual $\alpha = \mathrm{P}(Detected|Healthy)$ which is the first species statistic error that one needs to calibrate to define the test rejection domain. This PFA value really represents the inconvenience of stopping a test for no reason.

The probability of detection is simply given by Eq. (2):

$$POD = \mathrm{P}(Detected|Faulty) \tag{2}$$

It is the standard $1-\beta$ value, usually called “test power” in statistical background.

PFA can also be rewritten according to Bayes’ rule and a computation of α in Eq. (3) shows that the test threshold should be chosen very far from normal behavior when one intends to respect a small boundary constraint on the false alarm rate:

$$\alpha = POD \cdot \frac{PFA}{1-PFA} \cdot \frac{\rho}{1-\rho} \simeq POD \times PFA \times \rho \tag{3}$$

where $\rho = \mathrm{P}(Faulty)$ is usually very small for aircraft engine parts (less than 10^{-6} per hour). A very careful attention is needed for the choice of this detection threshold. It is why the decision part of the algorithm has two additional modules: one for confirmation by several successive detections and another for the optimization of the threshold, using a model of the score distribution queue with Parzen windows.

Figure 3 shows the meaning of the rejection threshold computed from a choice of α and the power $1-\beta$ of a statistical test.

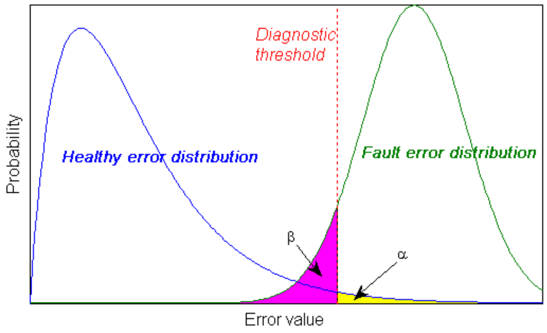

Figure 3: Threshold selection for decision test.

2.3. Description of the bench data

The main element we have to monitor is the rotating shaft and the principal bearing (called #4 here). One of the exogenous information we have to deal with is the external loading applied on the right of this shaft. This is a longitudinal force which have a lot of influence on the system behavior because it may change the dynamic mode positions.

Most measurements come from dynamic high frequency acquisitions. The corresponding low-frequency observations are filtered energy computations which may be either piloted by the shaft speed or be total vibration energy, eventually quantified according to given bandwidths.

The table below (Table 1) gives the complete list of used sensors.

	Index		
	ESN	Engine serial number	
	CYCL	Engine cycle reference	
	DATE	Date of cycle (start to off)	
N°	Sensor	Label	Unit
01	XN	Shaft rotation speed	tr/min
02	XN_DERIV	Accel. of the rotation	-
03	TORQUE	UI/w	kg/h
04	P4	Pressure Piston #4	bar
05	PORSDE	Position rectifier	deg
06	VANPRIM	Position primary vane	%
07	P1	Pressure Piston #1	bar
08	K_0N	Disp. Up Pilot	mmDA
09	K_0T	Disp. Up RMS	mmDA
10	K_1N	Disp. Down Pilot	mmDA
11	K_1T	Disp. Down RMS	mmDA
12	ACC_4RN	Accel. #4 Rad Pilot	cm/s eff
13	ACC_4RT	Accel. #4 Rad RMS	cm/s eff
14	T4	Temp. #4	degC
15	ACC_1HN	Accel. #1 Horiz Pilot	cm/s eff
16	ACC_2VN	Accel. #2 Vert Pilot	cm/s eff
17	ACC_3VN	Accel. #3 Vert Pilot	cm/s eff
18	ACC_MN	Accel. Engine Pilot	cm/s eff
19	ACC_1HT	Accel. #1 Horiz RMS	cm/s eff
20	ACC_2VT	Accel. #2 Vert RMS	cm/s eff
21	ACC_3VT	Accel. #3 Vert RMS	cm/s eff
22	T1	Temp. #1	degC
23	ACC_MT	Accel. Engine RMS	cm/s eff
24	T2	Temp. #2	degC
25	T3	Temp. #3	degC

Table 1: List of sensors and corresponding units, blue and green backgrounds identify respectively a selection for the exogenous and endogenous variables.

The #1 to #4 numbers refer to the different bearings where accelerometers are measuring vibration data. Those vibration values are summarized as local energies for a frequency band that corresponds to the shaft speed (N) or a total amount of energy (T). The first sensors (blue background) are used as context information or exogenous variables. The corresponding data vector is used to identify the context of the measurement. They are used to select stationary snapshots and to classify the snapshots into clusters. The others variables (endogenous) are used to monitor the bench when a context is clearly identified.

Other variables such as microphone band energies are not displayed in Table 1. Such selection of endogenous and exogenous variables defines an instance of the algorithm. It is possible to build different kind of instances (with corresponding algorithmic parameters) for any part of the test bench one selects to monitor.

The abnormalities may be very tricky to detect. For example on Figure 4 one can see measurements taken during a test cycle that contains such anomaly. Just looking the data is rarely sufficient to find the abnormal behavior. A mathematic comparative analysis is definitely needed.

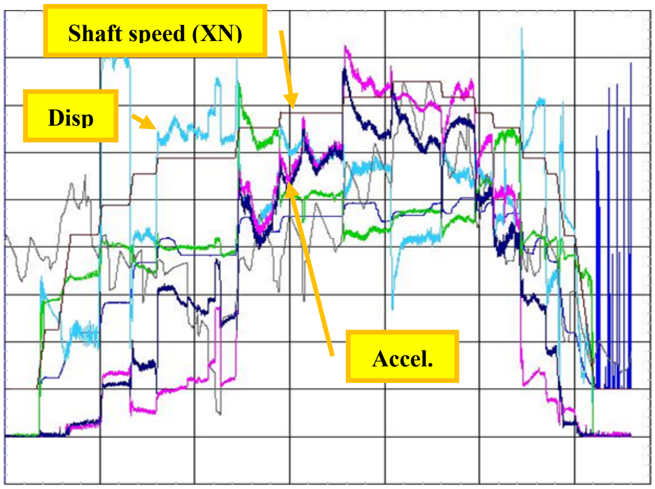

Figure 4: Example of measurements acquired during a whole cycle. The black stepwise line represents the shaft speed. Displacement (light blue and green) and accelerometers (dark blue and magenta) are highlighted. The other blue sensor is the load (the last data have no meaning since the shaft stopped). An abnormality is hidden in those data.

3. ALGORITHMS DESCRIPTION

The algorithm is made of two parts. The first one identifies stationary measurement intervals (in context data) and builds a snapshot of the endogenous measurements. The second part loads the database of snapshots, builds clusters, and for each cluster search for abnormalities.

3.1. Snapshots extraction

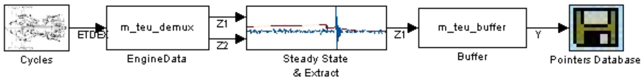

Figure 5: Graph of SAMANTA modules to extract snapshots and build a database.

The first step of snapshot extraction is the selection of measurements to identify stationary data. The stationary measurement detector waits for a main control value to be almost constant and tests a vector of endogenous measurements for second order statistic stationarity. In our case we use the shaft speed as main control and test other endogenous data for stationarity.

Once a stable point detected, a buffer of observations is recorded and defines the snapshot. Figure 6 shows a list of snapshots detected on a symbolic cycle that may represent a real flight.

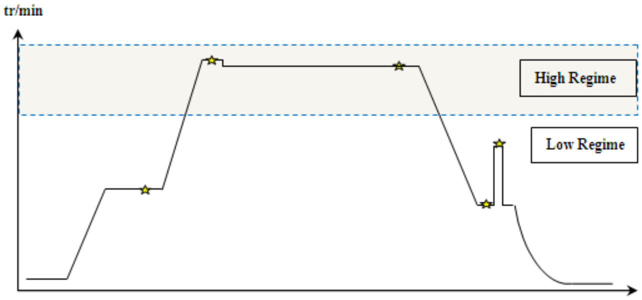

Figure 6: Example of snapshots identification, each star represents a point detected as a possible snapshot for the test cycle.

3.2. Novelty detection

The detection part uses three mathematic models: the clustering algorithm, the score algorithm and the decision algorithm. Each one needs a specific learning phase to calibrate.

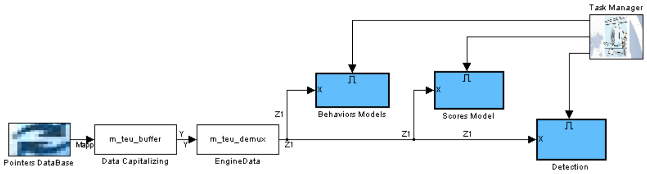

Figure 7: The novelty detection graph of SAMANTA modules.

The **clustering learning phase** uses the whole snapshot database (eventually obtained from a sub-sampling of the snapshot buffers) but only exogenous vectors of values to isolate homogeneous clusters with an EM algorithm. This algorithm, as described in (Lacaille et al. 2010b), is a generative statistic model from a mixture of Gaussian distributions. Each Gaussian identifies a different set of snapshots. The number of classes is estimated by a BIC criterion and the unclassed snapshots are not used.

During the learning phase a database of snapshot buffers is used to define the individual classes which can eventually further be labeled as flight regimes or operating modes. To make this possible, each buffer signal curve is compressed into a set of shape indicators hence replacing the multivariate temporal signal by a vector of indicators *U*. The compression scheme (Figure 8) uses specific algorithms to enlighten changes in the data: for examples an algorithm computes the trend of the signal, another looks for jumps and a generic compression uses automatic templates built from a principal component analysis (PCA).

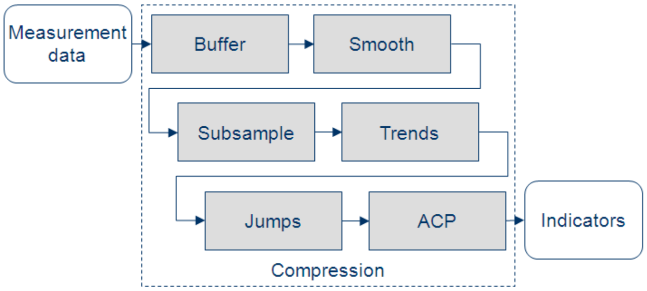

Figure 8: Compression process that builds indicators from a multivariate temporal buffer.

The set of exogenous indicator vector U is then used by the classification algorithm to build classes as a mixture of Gaussians. The number of classes to build is controlled by a BIC criterion but may be also given by expert bounds as the snapshots are essentially identified as standard layers.

Each set of classed snapshots (the ones that belong to a cluster) are used to **calibrate a score model**. Once indentified in a specific class, each multivariate temporal signal of endogenous data is compressed locally in another indicator vector Y (Figure 9). The score process has two steps; the first one normalizes the endogenous data suppressing disparities due to little variations in the context. This is done by a regression algorithm controlled by a L_1 criterion (LASSO algorithm) as described in (Lacaille & Côme, 2011b). The second step is a model of the residual of this regression by a Gaussian score (a Mahalanobis distance) see (Lacaille, 2009c).

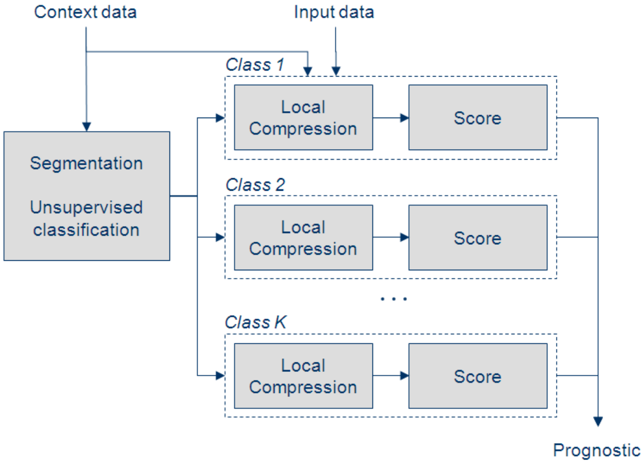

Figure 9: Score analysis after snapshot identification.

Equation 4 explains the mathematics of the regression model. Each parameter $X_{i,j}$ correspond to a combination j, non necessary linear, of exogenous variables in U used to predict endogenous observation $Y_{i,k}$ of parameter k (or a function of endogenous parameters) for all snapshots i of a given cluster.

$$\operatorname{Arg\,min} \sum_i \left(Y_{i,k} - \sum_j \lambda_j X_{i,j}\right)^2 \quad \text{subject to } \sum_j |\lambda_j| < C \qquad (4)$$

The final and optimal constant C is chosen such that the generalization error of the regression is the smallest. The generalization error is computed by a cross validation scheme. The next graph (Figure 10) explains in 2 dimensions why the constraint is to be chosen in absolute value instead of Euclidian norm. The figure schematizes the mean square regression coefficient as the point β_0 and iso-square errors as red ellipses (first part of equation 4). The blue shapes (disk for L_2 constrain and square for L_1 constraint) represents the value of the second part of equation 4. The value β to select is on a tangent intersection of an ellipse (ellipsoid in higher dimension) and the surface of the blue shape. The radius of the ellipse is the square error and the radius of the blue shape is the constraint. It appears clearly that with a square (cube, hypercube) the chance to find an edge point of the surface is important. As soon as those points are on the main axis, most of the coefficients of β should be zero.

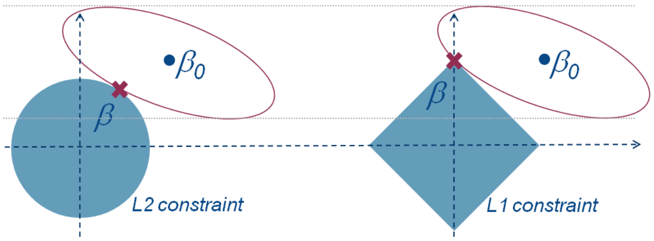

Figure 10: Minimization subject to L_1 constraint (absolute value) instead of a L_2 constraint (Euclidian norm) ensures that most coefficients are set to zero, hence improving the regression robustness with a small loss in mean square error.

As this model automatically selects variables to ensure a good robustness, we may use a big set of computations as inputs. In fact the aeronautic experts give clues about the physic process and help the conception of a big vector X of indicator functions that makes a great uses of formal physical equations.

The final score s of a single observation is given by (Eq. 5) where $r_k = Y_k - \hat{Y}_k$ is the residual of the preceding regression for a variable k and a current snapshot and Σ is the matrix of covariance of the vector $r = (r_k)$.

$$s = r'\Sigma^{-1}r \qquad (5)$$

The score should follow a standard χ^2 distribution under very restrictive constraints on the model and residual distribution. In general those constraints are not completely respected, and in any cases as the real dimensionality of the problem stays approximate, an indetermination of the freedom level is possible. We use a last module of algorithm to establish a more precise **decision rule**. This decision rule is not only based on one score computation but on a small list of successive scores. Each individual score is compared to a threshold α and the Boolean results are combined together by a vote process. The law of the scores is modeled with Parzen windows on healthy observations. Then artificial abnormal behaviors are produced with the help of aeronautic experts and a mechanical physic simulator. Using the empirical distribution of scores and the pseudo abnormal observations one is able to determine a good choice for the threshold for a given α computed from equation 3.

4. Results and Conclusion

Two campaigns of measurements where done on the same bench test cell. The first one was to challenge a civilian compressor and lasted almost 3 months. It was a bench calibration test. The second campaign uses a military compressor as an extractor for another development bench, we get also around 3 months of data. In each case, when the bench was working we may have one to three runs per day. Results cannot be presented in this article; the main goal for the PHM team was to validate the algorithm and the monitoring process.

4.1. Examples of detected abnormalities

Example 1. Normally, during a stabilized step, if the axial pressure on the shaft increases, the vibration level should decrease. It was not the case on Figure 11 and this was detected as an abnormal feature.

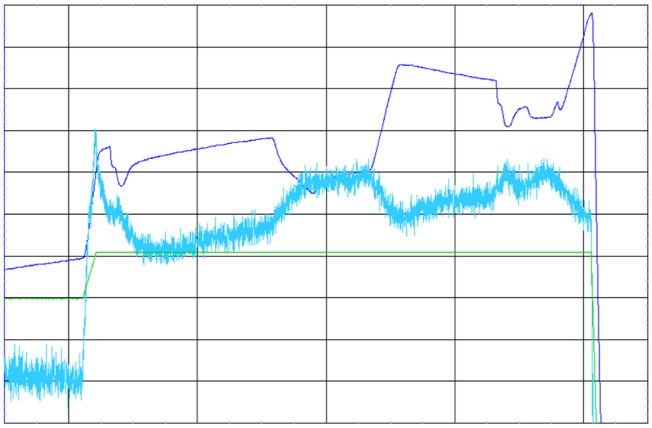

Figure 11: On that test run the vibration level, measured by an accelerometer (light blue), decreases when the pressure augments (dark blue) and during a stabilized level (XN in green).

Example 2. Figure 12 shows a sudden increase of the total vibration levels during a deceleration phase. This is also another kind of abnormal behavior which was not detected at first because the system was not able to differentiate between main accelerations and deceleration (begin or end of the run), but as soon as we enter this information as a context observation the detection is fixed.

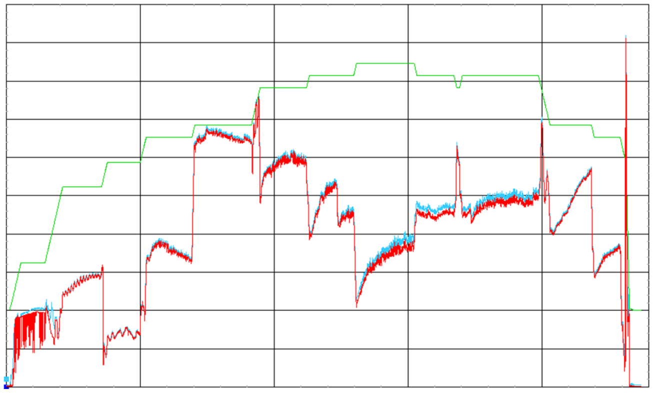

Figure 12: A sudden augmentation of vibration was detected by vibration sensors (red and blue) during a deceleration (shaft speed in green).

4.2. Conclusion

The work was tested on data obtained during the 6 months of experimentation. The PFA indicators were computed on observed data with the prerequisite identification of all the real abnormalities referenced in the logbook. Then new simulated defects compatible with the real observations were artificially added to the data at known random positions on the signals. A false alarm rate of PFA=1.3% and a detection rate of POD=78% were obtained on high speed rotation clusters. On low speed clusters, too much variability of the endogenous variables was observed to give conclusive results. More work will be done on the identification of specific recurrent clusters, but in any cases the algorithm may still be used if some standard states are defined, and if one asks the pilot to reach those states at the beginning of each run.

Another model should also be defined for non-stationary measurements because some known difficulties may arise when the bench crosses a vibration mode during a transient phase such as acceleration or deceleration.

Nomenclature

BIC	Bayesian Information Criterion
LASSO	Least Absolute Shrinkage and Selection Operator
FADEC	Full Authority Digital Engine Control
OSA-CBM	Open Systems Architecture for Condition-based Maintenance
PFA	Probability of False Alarm
POD	Probability Of Detection
SAMANTA	Snecma Algorithm Maturation And Test Application

References

Bellas, A. & Bouveyron, Ch. & Cottrell, M. & Lacaille, J. (2012), *Robust Clustering of High Dimensional Data*. In proceedings of ESANN, Bruges, Belgium.

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

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

Cômes, E. & Cottrell, M. & Verleysen, M. & Lacaille, J. (2010b), *Self organizing star (sos) for health monitoring*. In proceedings of ESANN, Bruges, Belgium.

Cômes, E. & Cottrell, M. & Verleysen, M. & Lacaille. J. (2011), *Aircraft engine fleet monitoring using Self-Organizing Maps and Edit Distance*. In WSOM proceedings, Espoo, Finland.

Cottrell, M. & Gaubert, P. & Eloy, C. & François, D. & Hallaux, G. & Lacaille, J. & Verleysen, M. (2009), *Fault prediction in aircraft engines using self-organizing maps*. In proceedings of WSOM, Miami, FL.

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

Hazan, A. & Verleysen, M. & Cottrell, M. & Lacaille, J. (2010a), *Trajectory clustering for vibration detection in aircraft engines*. In proceedings of ICDM, Berlin, Germany.

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

Hazan, A. & Lacaille, J. & Madani, K. (2012), *Extreme value statistics for vibration spectra outlier detection*. In proceedings of CM-MFPT, London, UK.

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

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

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

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

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

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

Lacaille, J. & Gerez, V. & Zouari, R. (2010b). *An Adaptive Anomaly Detector used in Turbofan Test Cells*. In proceedings of PHM Society Conference, Portland, OR.

Lacaille, J. & Côme, E. (2011a), *Visual Mining and Statistics for a Turbofan Engine Fleet*. In Proceedings of IEEE Aerospace Conference, Big Sky, MT.

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

Lacaille, J. & Gerez, V. (2011c), *Online Abnormality Diagnosis for real-time Implementation on Turbofan Engines and Test Cells*. In proceedings of PHM Society Conference, Montreal, Canada.

Lacaille, J. (2012), *Validation Environment of Engine Health Monitoring Algorithms*. In proceedings of IEEE Aerospace Conference, Big Sky, MT.

Nya-Djiki, R. & Hezard, E. & Lacaille, J. (2012), *Enhanced Massive Visualization of Engines Performance*. In proceedings of COMADEM, Hudderfield, UK.

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

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. 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.

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.