Enhanced Massive Visualization of Engines Performance

N D Rostand¹, H Eglantine² and L Jerôme³

¹ R&D Monitoring System Technical Manager, Safran Engineering Services, 10 Rue du Fort de Saint Cyr, 78180, Montigny-Le-Bretonneux, France.
² Senior Expert Algorithm, SNECMA, Rond-point René Ravaud, 77550, Moissy-Cramayel, France.

E-mail: ¹rostand nga-djiki@safran-engineering.com; ²eglantine.chezard@safran-engineering.com; ³jerome.lacaille@snecma.com

Abstract. Today, we are witnessing an increasing complexity of transport in order to deal with requirements of safety, security, reliability and efficiency. Such transport is generally equipped with drive systems; it is nevertheless for engine manufacturers to overcome the performance requirements of energy efficiency throughout their operations. To this end, this article proposes a performance monitoring solution for a large fleet of engines in operation. It uses a pre-calibrated physical model developed by the engine manufacturer regarding the performance objectives as reference. The physical model is firstly decomposed into critical performance modules, and is secondly updated on current observations extracted at specific predefined operating conditions in order to derive residual errors status of each engine tested. Through a process of standardization of those contextual differences remaining, the solution offers a synthesis mapping to visualize the evolution of performance of each engine throughout its operations. This article describes the theoretical methodology of implementation mainly based on universal mathematical foundations, and vindicates the interests of its industrialization in the light of the proactive findings.

1. Introduction

This paper describes an algorithm to help oversee the graphic of an engines’ fleet in operation based supervised classification of built-in performance health indicators. Through a synthesis map, it allows making mark subsets of engines organized by serial number or by reference family. Several definitions of marking are possible, but we will limit ourselves by the following two types of tagging-related performance objectives engine:

- Class A: Performance acceptable
- Class B: Performance down

1.1. Industrial need

Indeed, engine manufacturers must deal with the problem of online/offline diagnosis of their system which is more and more complex in order to anticipate the cost of operability while ensuring the objectives of efficiency and performance of their system. The diagnosis by this algorithm may be...
optionally fused to a collection system malfunctions or anomalies reported to produce a more accurate diagnostic report related to the actual health of each reference scan engine.

1.2. Context development
Throughout their operation, specific engines health parameters are capitalized in order to achieve a follow-up of their performance within the cycle to cycle of their operation. To do this, the embedded part of the present algorithm should perform records of the parameters with the terms of the optimal point of operation that ensure a meaningful level of performance expected in the design of the motor; ACARS snapshots for example. The analysis of yield performance of an engine is based on decomposition into modules of which at least one health parameter is developed. Once these parameters are capitalized for several references and for several engine operating cycles, the "on ground" part of the algorithm firstly, computes estimates of the physical measurements of on-board sensors as current characteristics of the engines states; and secondly, proceeds a contextual removal normalization of observed measurements of those sensors, and finally performs a projection of built-in standards residuals on a large space developed by calibration of the algorithm on data from expected performance engines.

1.3. Global synoptic
Three main scientific methodologies are implemented in the solution proposed:

- Kalman Filtering ([1],[2],[3],[31],[33])
- Contextual Standardization ([32])
- Adaptative Visual Mining ([13],[11])

The diagram below shows an illustrated OSA-CBM scheme of the algorithm, and helps to highlight some practical industrialization of mathematical algorithms.

---

**Figure 1.** Algorithm global synoptic.
The scientific methods used are universal and have the distinction of being adapted in response to industrial needs diagnosis.

2. Inputs Data

As introduced in chapter §1.1, the algorithm inputs are set in two main parts: Extracted from fleet database ACARS report, and Computed from physicals simulations. According to the need of engine physical model and yield performance reference information, inputs per engine could be distinguished into three main categories:

- Modular performance parameters (P,U)
- State parameters (E)
- Environment parameters (C)

Capitalized for a huge quantity of engines references regarding their corresponding operations, inputs could be organized as follow in the table below. It shows an example of architecture of the database entry for a fleet of engines.

| Performance (P,U) | State (E) | Context (C) |
|-------------------|-----------|-------------|
| \( u_1 \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( u_j \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( p_1 \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( p_k \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( e_1 \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( e_k \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( c_1 \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |
| \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( c_k \) - - - - - - - - - - | - - - - - - - - - - | - - - - - - - - - - |

The following figure illustrates a data acquisition process which could be performed by a simple system data access object for industrialization robustness needs.
2.1. Modular performance parameters
Those parameters are designed according to the execution needs of the engine physical model set by the manufacturer. This would mean that the used physical model must be representative of local modular performance to monitor and should be pre-calibrated and formatted for expected yield performance as reference. Generally, those parameters are designed as a couple \((P, U)\) where \(P\) gives the information of performance or performance to be achieved (objective), and \(U\) refers to the information package or dedicated control. So, the fixed physical model will automatically execute on current observations of \((P, U)\) in order to estimate engine state parameters. This proceeding and their interest are described below. The table below shows an example relating to a turbojet engine, wherein five modules can be identified:

| Modules | Fan | Booster | CoHP | TuHP | TuBP |
|---------|-----|---------|------|------|------|
| Performance Objective \((P)\) | \(P_1\) | \(P_2\) | \(P_3\) | \(P_4\) | \(P_5\) |
| Command Objective \((U)\) | \(U_1\) | \(U_2\) | \(U_3\) | \(U_4\) | \(U_5\) |

2.2. State parameters
Those parameters \((E)\) refer to physical measurements from engine on-board sensors. They provide information about the current state of operation corresponding to the observed measures of the performance modular. That information corresponds to extracts that can also be reported by the embedded part of the algorithm. They therefore rely on previous parameters \((P, U)\) which refer to each engine type and operation according to the specific operation condition studied. It is important that all state parameters outputted by the fixed physical model must be available among extracts reported by the embedded part of the algorithm. The following table shows an example of turbojet engine state parameters that could be used to deal with modular performance analysis.

| Index | Details |
|-------|---------|
| ESN   | Engine serial number |
| CYCL  | Engine cycle reference |
| DATE  | Date of cycle |

| Sensors | Details |
|---------|---------|
| WOW     | A/C weight on wheel |
| N1SEL   | Core LP rotation speed |
| N2SEL   | Core HP rotation speed |
| P23     | Air pressure in combustion chamber |
| TLA     | Throttle lever angle |
| EGT     | Exhaust gas temperature |
| IVV     | Inertial vertical velocity |
| VSV     | Variable stator vane position |
| VBV     | Variable boost vane position |
| PS3     | HP Compressor discharge static pressure |
| P25     | HP Compressor inlet total air temperature |
| T25     | Air temperature in combustion chamber |

Subsequently, we will see that the algorithm described in this article includes a method of classifying data in a space of high dimension; thus, there is no limitation of sensors data in the report of input data.
2.3. Environment parameters
Also known as context parameters, environmental parameters (C) are necessary to extract by the embedded part report, in order to take into account specific context of the engine when operating. They notify of any particular information on the operation condition which could justify abnormal behaviours of engine. So, the algorithm described in this paper might deal with problem of bias introduced by impacts of the environment context on performance reliability. The proceeding implemented as solution of that problem, is described below. For any type of engine, the following table lists some generics environment parameters:

| Sensors | Description               |
|---------|---------------------------|
| PTAPU   | Air pressure provider unit|
| WFM     | Weighted fuel metering    |
| TFUEL   | Fuel temperature          |
| TOIL    | Oil temperature           |
| POIL    | Oil pressure              |
| PAMB    | Ambient air pressure      |
| TAMB    | Ambient air temperature   |
| ALT     | Altitude                  |
| MACH    | A/C air velocity          |
| GSPD    | Ground speed              |

2.4. Data validation
Once a huge quantity of inputs data is available, it is more convenient to proceed for a validation process of the two types of parameters (E and C) listed above. Considering that each input sample measurement derives from a sensor model, an input sample can be modelled as follow:

\[ x_{i,m} = f_m(z_i) \]  

Where:
- \( z_i \) refers to a measurement of sensor (m)
- \( f_m(.) \) refers to sensor (m) model function

By implementing a Forester diagram through loopback tests between the balance equations of the model sensors and their possible functions of learning, the algorithm manages to eliminate systematic and random errors of measurements.

For example, the following statement designs processing of validation of “ALT” signal:

**Step #1:** An estimated dynamic model of accelerometer can be design as follow:

\[
\frac{dz_a}{dt} = v_z \\
\frac{dv_z}{dt} = b_z + a_z \quad \text{where} \quad a_z = \frac{d^2h}{dt^2} \\
\frac{db_z}{dt} = w_a
\]

Where:  
- \( t \): continuous time; \( a_z \): Inertial vertical acceleration; \( z_a \): Measured altitude; \( v_z \): Measured inertial vertical velocity; \( w_a \): random noise; \( b_z \): bias of accelerometer; \( h \): Altitude
Step #2: The discrete form of the model above gives the parametric model function:

\[ z_{a}(t+1) = f_{ALT}(z_{a}(t), b_{a}, w_{a}) \]  

Where: \( \tau = t*p \): \( p \) equals to sampling period; \( f_{ALT} \): Calibrated sensor model giving optimal parameter \((b_{a}, w_{a})\).

Step #3: “ALT” signal process validation involves computation of corresponding accuracy signal cycle by cycle, as square errors between observed and computed:

\[ \Delta ALT = (ALT - \tilde{z}_{a})^2 \]  

This proceeding improves clearance on algorithm inputs before entering the first stage of processing: Kalman filtering process.

3. Kalman Filtering (KF) Application

The present algorithm implements a Kalman filtering process in order to perform computation of a temporal behavior of an engine performance module using established sensors physical model. Indeed, this fixed model, essentially based on dynamic modeling, therefore introduces synchronous and delayed effects when computing expected engine states parameters corresponding to the expected reference engine yield performance. Thus, that filtering process allows reducing those effects and keeping track of real expected engines states parameters related to reference performance.

The diagram below describes the first step of data manipulation according to the OSA-CBM algorithm architecture.

The following steps describe the use in monitoring of high pressure compressor (CoHP) aircraft engine performance module. The main objective is to monitor energy efficiency parameters (P, U) regarding expected one computed from physical simulation.
3.1. State parameters based fixed physical model simulation

Once the inputs database is made sufficiently representative and valid, the algorithm applies the
discrete balance equation established from predefined sensors (endogenous) physical models, in order
to compute estimated module state parameters.

For example, the following expressions can illustrate models of endogenous parameters necessary
for CoHP efficiency monitoring (focusing stall monitoring):

**Step #1:** Development of models showing the measuring points of the mass flow rate, fuel pressure,
core rotation speed and theoretical energy density. Those models shall be adjusted by experts
knowledge’s in terms of thermodynamic and mechanic consistency:

- Fuel mass flow can be perform through dynamic fuel pump model (a) and reduced equivalent
  model (b):
  
  \[ \eta_{rel} (N, \Delta P) = 1 - B \left( 1 - 0.99 - \frac{A_1 \Delta P^a}{N} \right) \]
  
  \[ WFM_{rel} = \frac{WFM}{PT_i} + \epsilon_{WFM} \]

- Pressure rate model estimated by:
  
  \[ \pi = \frac{PT_{23}}{PT_1} + \epsilon_{\pi} \]

- Core rotation speed dynamic model estimated from theoretical energy density thermodynamic
  expression:
  
  \[ \frac{dN}{dWFM} = \Delta_1 \left( \dot{U} \cdot \ddot{V} \right) + \Delta_2 WFM \]

**Step #2:** Establishment of discrete balance equations from models described above [(5),(6),(7),(8)]
which are respectively express estimation of temporal mass flow (WFM(k)) and pressure rate (\( \pi(k) \)).
Assuming that all edited models are linear, we can easily design a global discrete state model of the
module:

\[
\begin{align*}
X(t+1) &= FX(t) + V(t) \\
Y(t) &= HX(t) + W(t)
\end{align*}
\]

Where:

\( X = (WFM, PT12, TFUEL, N2, PT22)^T \) designs the unknown state parameters vector.
\( Y = (m, \pi)^T \) designs observations vector of mass flow (\( m = \frac{WFM}{\rho} \)) and pressure rate (\( \pi = \frac{PT_{23}}{PT_1} \)).

Covariance’s matrices \( V \) and \( W \) give established Gaussian noise of state parameters regarding balance
equation.
State matrix \( F \) and command matrix \( H \) coefficients are also established according to the balance
equation.

**Step #3:** Applying the built Kalman filter, the algorithm performs estimation of \( (m, \pi)_{KF} \) module
performance characteristics, cycle by cycle.
The next step involves comparing that estimation to expected get from physical simulation.
3.2. Residuals module characteristics computing
The algorithm computes current residuals $\hat{R}$ of engine module characteristics as differences between observed ones deduced from allocated ($P,U$) parameters extracted from physical simulations; and those resulting from crude estimation by the previous established Kalman filter. Residuals expression of example treated above:

$$
\begin{align*}
  r_m &= m(P,U)_{SIM} - \bar{m}_{KF} \\
  r_\pi &= \pi(P,U)_{SIM} - \bar{\pi}_{KF}
\end{align*}
$$

(10)

The next step of the algorithm is related to the consideration of environment parameters. It might deal with another form of process denoising by suppressing effects of context (or environment) of engines operations, on health indicators. This procedure is formalized as contextual normalization of performance health indicators.

4. Contextual Normalization
The sake of uniformity of the residuals $\hat{R}$ defined above, the algorithm proceeds to an implementation of a standard called “Contextual Removal Normalization (CRN)” ([32]) regarding the entire fleet of engines and analysis on several operating cycles. It is the aim of the second step of data manipulation regarding the algorithm OSA-CBM architecture. The diagram below illustrates the dynamics of the process of standardization.

The process applies standard normalization by centering and reducing the variable $Y$ which represent the normalized previous residuals module characteristics $R$. The normalization of $R$ is based on context parameters $C$ removal effects generally introduced by extracted raw pointers. In order to keep track of abnormality regarding a predefined reference statistical model, the process performs computation Z-scores ($Z$) of normalized residuals $Y$. 

---

8
4.1. CRN Algorithm application
CRN (Contextual Removal Normalization) is a method described in [32]. The method describes a consistent way of suppressing data acquisition context effects on feature signatures pointers. For the present use, the CRN method allows to deal with suppression of environment (context C parameters) effects on residuals R of engine studied module performance indicators. For example, the aircraft altitude, the temperature and ambient pressure, the fuel temperature and other environment parameters are used to describe the acquisition of the exhaust gas temperature of the engine during the cruise phase. Some differences appear depending on if the flight day is cold or hot, depending also on the flight phase studied, and or whether the pilot is not the same from one day to another. CRN proposes a solution to set a clearance on main indicators considered as endogenous or more informative data.

4.1.1. Data calibration. Data calibration phase involves statistical model computing. The following illustrations show how online calibration could be approached.

**Inputs data**
R: Endogenous as computed residuals; and C = [X, W]: Where X defines inputs exogenous and W refers to environment information. [X, W] are essentially composed of some predefined extracted raw pointers (§2.2 and §2.3).

**1st Step Model learning: Robust regression models**
First step model learning is to build a robust regression model of each local endogenous on a given representative database. For example referring to §3.2, a robust model regression is for each residual module performance indicator set as endogenous, considering C parameters as exogenous:

$$\tilde{r}_i = \sum_{k=1}^{n} f_k(\theta_k, C)$$  \hspace{1cm} (11)

Where:

- $i$: indexes components of R; for example $i = [m, n]$
- $f_k(\theta_k, C)$: derived mathematic function with coefficients $\theta_k$ which expresses residuals knowing C.

So, the first step model learning involves computing list of mathematic functions $f_k(\theta_k, C)$ of each component of R.

**2nd Step Model learning: Means models**
Second step model learning is to compute means of R component global learning database.

$$\text{Moy}_R = \frac{1}{L} \sum_{k=1}^{L} \tilde{r}_i(k)$$  \hspace{1cm} (12)

**3rd Step Model learning: Z-scores models**
Last step model learning is to compute Z-scores models of normalized R components, essentially based on monovariate Gaussian model given local mean and standard deviation of each normalized ($Y$) components:

- Normalized $r_i$ component: $$Y_i = \text{Moy}_R(i) + (r_i - \tilde{r}_i)$$  \hspace{1cm} (13)
- Z-score model of normalized $r_i$ component: $$\mathbb{Z}(m_i, \sigma_i)$$  \hspace{1cm} (14)

Where: $m_i = \mathbb{E}(Y_i)$ and $\sigma_i = \mathbb{Z}(Y_i)$
4.1.2. Z-scores computation. Once normalized (Y) residuals of module performance indicators, the proceeding computes main indicators $Z$ of engines performance as local Z-scores. This involves online centering and reduction of normalized residuals:

$$Z_{ij} = \frac{Y_{ij} - m_j}{\sigma_j}$$  \hspace{1cm} (15)

As illustrated, we used 8 turbojet engine state parameters (blue) on amount of data referring to cruise operation. The corresponding Z-data indicators (red) show the normalized in terms of suppressing effects of air pressure and ambient temperature, fuel and oil temperature.

![Figure 5. Illustration of contextual normalization.](image)

5. Module Performance Visualization

Knowing normalized (Y) residuals of a studied engine module performance indicator, the proceeding can easily offers a visualization tool of performance trend according to predefined grid giving acceptable engine operating points.

For example treated in §3.1 and §3.2, current computed samples of $Y_{r_m}$ (Normalized flow mass) and $Y_{r_c}$ (Normalized compression rate) are projected on a predefined 2D card in order to visualize current engine operating point. Holding grid defining zones of nominal operating points, we can therefore mark derived engines and editing and automatic alert message through a simple SPC (Statistical Process Control).

The following cartography shows an example of grid map used for an engine aircraft compressor performance monitoring purpose. It is a 2D card where X-axis gives corrected air mass flow ($r_m$) in [lb/min] and Y-axis gives pressure rate ($r_c$).
We introduce an autoadaptative process in order to update border built in the initial space. This automation process is implemented through an iterative algorithm which helps to better represent the data in the selected engine modular card (2D space).

6. Unsupervised Classification based EM approach
The proposed visualization tool allows showing performance trend cycle by cycle; but doesn’t give explanation on abrupt changes relating to abnormalities detection. Applying an unsupervised classification, the algorithm proceeds with a Gaussian mixture Z-data (§4.1.2) model based on expectation maximization (EM) method. Results expected by applying EM method could be well adapted for a physical interpretation, according to several types of abnormalities and events.

We describe how EM method is implemented focusing abnormalities detection.
6.1. Unsupervised Classification Learning

6.1.1. Reference behaviors map. The learning database $Z$ is essentially formed of $Z$-data capitalized for a specific engine condition operation, and refers to several operations and several engine references. So, the calibration phase consists in minimization of the computed-in Bayesian Information Criterion (BIC) designed by Schwartz (1979) in order to set optimal number of engines behavior classes as reference ones:

$$\kappa^* = \text{argmin} \left( -2L_w(Z) + d_w \log(n) \right)$$

(16)

Where:

$$L_w(Z) = \sum_{j=1}^{n} \log \left( \sum_{k=1}^{m} \pi_k p_{u_j} Z_{j,k} \right)$$

The loglikelihood function valuated on a learning database $Z$.

$$w_k = (\pi_k, p_{u_j})$$

A built classification model of $m$ classes indexed by wear indicator $k$ giving engine behaviour occurrence probability $\pi_k$ and density function $p_{u_j}$.

$k^*$ gives optimal groups per wear indicator corresponding to the predefined optimization criterion and the set learning database. Those groups involve reference map of studied engines performance referring to the analyzed operation condition.

6.1.2. Modular performance map. Once groups per indicator is computed, the algorithm could automatically perform a cartography of engines modular performance part by merging some predefined wear indicators (as map components). This need involves the problem of merging statistical components. Several scientific approaches exist such as the method based on the mean of Mahalanobis distance; based on L1 distance; based on the Kolmogorov Smirnov test and the Monte Carlo test.
Carlo approach... However, we choose to implement the method based on Kullback-Leibler divergence which performs comparison of indicators statistical law:

\[
KL(l_1 \| l_2) = \int_{-\infty}^{\infty} p_{l_1}(z) \log \left( \frac{p_{l_1}(z)}{p_{l_2}(z)} \right) dx
\]  

(17)

6.2. Adaptative Visual Mining

The final step of the algorithm provides a synthetic visualization of performance classes using the previous learned reference map. It therefore allows seeing the trend of engines modular performance for each new sample of Z-data treated as novelties, and referred to the learnt operation condition. An automatic recalibration performs an internal learning process of the algorithm for the purpose of map adjustment according to novelties. The method used is well described in [13].

The following figure illustrates the projection of indicators designed in chapter §4.1.2 (fig5) on a cartography based EM approach:

![Figure 8. Illustration of an EM monitoring card of engine modular MOD25-A designed by 2 components (T3, EGT).](image)

The 2D plot of corresponding input Z-data show observations of several engines operations for a specific operation condition. According to the 3 selected clusters or groups (red, green, and blue color), we can observe some disparity of engines around the focus reference point (0,0). Thus, with an outline envelope defined as border of separability, the algorithm allows classifying engines modules according to their operations behavioural drift.
7. Conclusion

The solution of engines fleet modular performance monitoring, described in this paper, is to date in a study phase as it is essentially based on the theoretical mathematical approaches listed above. However, a practical approach is feasible in light of the results rather convincing theoretical demonstration.

By projecting on predefined module performance cartography, it is more natural to monitor the trend on the studied operating point. The solution allows a SPC tool which automatically sets alert messages according to predefined acceptable bounds limit.

However, EM approach for the case of unsupervised processing seems well adapted for our application case relating to engines fleet performance modular analysis. Classification of Z-scores indicators allows explaining eventual drift of module performance.

The expected future goal of the proposed application is to correlate detected abrupt changes (as abnormalities) to events traced on fleet manager logbooks, for validation purpose.

Acknowledgments

[A] The physical turbojet engine model modular analysis as well as the identification of influent input data and the definition of failure modes was done by Snecma TMM cell members. This work is well described in paper related to the industrialized AMP (Analyse Modulaire de Performance) Algorithm, and it is a result of Snecma TMM cell projects.

[B] A datamining tool dedicated to civil turbojets fleet maintenance is developed by Snecma TMM cell members. The main goal is to label the hierarchical clustering and add this information to the some predefined statistic queries. This work is well described in “Datamining and Statistics for a Turbofan Engine Fleet”, written by J LACAILLE and R COME.

References

[1] H W SORENSON (Ed.), « Kalman Filtering: Theory and Application », IEEE Press, New York, 1985.
[2] T KAILATH (Ed.), « Linear Least-Squares Estimation », Dowden, Hutchinson and Ross, Inc. Stroudsburg, PA, 1977.
[3] B D O ANDERSON, J B MOORE, « Optimal Filtering », Prentice Hall, Englewood Cliffs, 1979.
[4] M BASSEVILLE, I V NIKIFOROV, « Detection of abrupt changes: Theory and Application », Prentice Hall, Information and System Sciences Series, 1993.
[5] C CORTES, V VAPNIK, « Support-Vector Network », AT&T Labs-Research, USA, 1994.
[6] V VAPNIK, « Statiscal Learning Theory », John Wiley, New York, 1998.
[7] V VAPNIK, « The Nature of Statistical Learning », Springer Verlag, New York, 1995.
[8] J A K SUYKENS, J VANDEWALLE, « Least Squares Support Vector Machine Classifiers », Academic Press, Belgium, 1998.
[9] T KOHONEN, « Self-Organizing Maps », Springer Series in Information Sciences, Vol 30, Springer, 1995.
[10] J LACAILLE, « Validation of Health Monitoring Algorithms for Civil Aircraft Engines », In IEEE Aerospace Conference, Big Sky, MT, 2010.
[11] J LACAILLE, R COME, « Datamining and Statistics for a Turbofan Engine Fleet », In Proceedings of IEEE Aerospace Conference, AC, 2011.
[12] J LACAILLE, R N DJIKI, « Model Based Actuator Control Loop Fault Detection », In Proceedings of Euroturbo Conference, Graz, Austria, 2009.
[13] J LACAILLE, V GEREZ, R ZOUARI, « An Adaptive Anomaly Detector used in Turbofan Test Cells », PHM 2010, Portland, OR.
[14] J LACAILLE, « Standardized Failure Signature for a Turbofan Engine », In Proceedings of IEEE Aerospace Conference, Big Sky, MO, 2009.
[15] J LACAILLE, « An Automatic Sensor Fault Detection and Correction Algorithm », Hilton Beach, SC, AIAA ATIO 2009.
[16] J LACAILLE, « Validation of Health Monitoring Algorithms for Civil Aircraft Engines », In IEEE Aerospace Conference, Big Sky, MT, 2010.
[17] A SMOLA, B SCHOLKOPF, « A Tutorial on Support Vector Regression », 30 Sept. 2003.
[18] A SMOLA, B SCHOLKOPF, « On a kernel-based method for pattern recognition, regression, approximation and operator inversion », Algorithmica, 22:211.231, 1998a.
[19] V VAPNIK, A CHERVONENKIS « Theory of Pattern Recognition », [In Russian] Nauka, Moscow, 1974.
[20] M CONTREL, G GAUBERT, C ELOY, D FRANCOIS, G HALLAUX, J LACAILLE, M VERLEYSEN, « Fault prediction in aircraft engines using self-organizing maps », In Advances in Self-Organizing Maps, Vol. 5629, Springer 37-44, 2009.
[21] R HERBRICH, « Learning Kernel Classifiers: Theory and Algorithms », MIT Press, 2002.
[22] O L MANGASARIAN, « Linear and Nonlinear separation of patterns by linear programming », Operations Research, 13:444.452, 1965.
[23] S SAI TOH, « Theory of Reproducing Kernels and its Applications », Longman Scientific & Technical, Harlow, England, 1988.
[24] V VAPNIK, « Estimation of Dependences based on Empirical Data », Springer, Berlin, 1982.
[25] T W ANDERSON, R R BAHADUR, « Classification into two multivariate normal distributions with different covariance matrices », Ann. Math. Stat., 33:420-431, 1966.
[26] D R COX, H D MILLER, « The Theory of Stochastic Processes », Wiley, New York, 1965.
[27] I V NIKIFOROV, « A generalized change detection problem », IEEE Transactions on Information Theory, IT-41(1); 171-187, 1995.
[28] I V NIKIFOROV, « Two strategies in the problem of change detection and isolation », IEEE Transactions on Information Theory, IT-43(2); 770-776, 1997.
[29] K FUKUNAGA, « Statistical pattern recognition », Academic Press, San Diego, New York, Boston, 1990.
[30] B K GHOSH, « Sequential Tests of Statistical Hypotheses », Addison-Wesley, Cambridge, Mass, 1970.
[31] B D O ANDERSON AND J B MOORE, « Optimal Filtering », Prentice Hall, Englewood Cliffsddison-Wesley, Cambridge, Mass, 1970.
[32] J LACAILLE AND N HENRY, « Standardisation de données utilisées pour la surveillance d’un moteur d’aéronef », SNECMA, Groupe Safran, Patent N°C FR 2,939,928-A1, 2008.
[33] D SIMON AND D L SIMON, « Aircraft turbofan engine health estimation using constrained Kalman Filtering », NASA/TM-2003-212528, In Turbo Expo 2003, Atlanta, Georgia, 2003.

Appendix A. The specific use of Kalman filter
Considering a generalized discrete dynamical system state model designed as follow:

\[
\begin{align*}
X(t+1) &= FX(t) + GU(t) + V(t) \\
Y(t) &=HX(t) + W(t)
\end{align*}
\]

Where \(t\) designs the discrete time \((t=0,1,2,3,...);\) \(X \in \mathbb{R}^n\) is the unknown state vector, \(Y \in \mathbb{R}^m\) is measurements vector, \(U \in \mathbb{R}^l\) is known command vector; \(F \in M_m (\mathbb{R})\), \(G \in M_m (\mathbb{R})\), \(H \in M_m (\mathbb{R})\) are known matrices; \(V \in \mathbb{R}^n\) and \(W \in \mathbb{R}^m\) are independent random vectors distributed by a Gaussian law as follow: \(V \sim \mathcal{N}(0, R_v)\) and \(W \sim \mathcal{N}(0, R_w)\).
The specific challenge involved by the use of Kalman filter in our study case, is to temporarily estimate the state parameters vector $X(k)$ dealing with synchronous and delay effects of the system. We use the Bayesian estimator of conditional mean of $X$ in order to perform his estimation.

From references ([1], [2], [3]), we can conclude that the conditional distribution of $X(t)$ or $X(t+1)$ knowing $Y(t)$, is given by:

$$X(t)|Y^t - \mathbb{E}\left\{\hat{X}(t|t), P(t|t)\right\}$$

$$X(t+1)|Y^t - \mathbb{E}\left\{\hat{X}(t+1|t), P(t+1|t)\right\}$$

Where $\hat{X}(t|t)$ and $\hat{X}(t+1|t)$ are conditional means of $X(t)$ (or $X(t+1)$) with respect to $Y^t$.

We therefore apply the Kalman filter which helps to compute best estimation of conditional means recursively. Here, brief demonstration of the Kalman filter:

$$\hat{X}(t+1|t) = F\hat{X}(t|t-1) + GU(t) + K_p(t)\left[Y(t) - H\hat{X}(t|t-1)\right]$$

$$\hat{X}(t+1|t+1) = \hat{X}(t+1|t) + K_f(t+1)\left[Y(t+1) - H\hat{X}(t+1|t)\right]$$

$$K_p(t) = FP(t|t-1)H^T\left[HP(t|t-1)H^T + R_z\right]^{-1}$$

$$K_f(t) = P(t|t-1)H^T\left[HP(t|t-1)H^T + R_z\right]^{-1}$$

$$P(t+1|t) = FP(t|t)F^T + R_i$$

$$P(t|t) = P(t|t-1) - P(t|t-1)H^T\left[HP(t|t-1)H^T + R_z\right]^{-1}HP(t|t-1)$$

With initial values: $\hat{X}(1|0) = \theta_0$ and $P(1|0) = R_0$

**Appendix B. Glossary**

**A/C** Aircraft
**ACARS** Aircraft Communication Addressing and Reporting System
**CoHP** High-pressure compressor
**TuHP** High-pressure turbine
**TuBP** Low-pressure turbine
**CES** Continuous Empirical Scores
**CRN** Context Removal Normalization
**CUSUM** CUmulative SUM
**RLS** Recursive Least Square
**OSA-CBM** Open Systems Architecture for Condition-Based Maintenance