

Automatic Indexation of Turbofan Data to Identify Anomalous Behaviors

Jérôme Lacaille¹, Cynthia Faure², Madalina Olteanu³, and Marie Cottrell⁴

¹*Safran Aircraft Engines, 77550 Moissy Cramayel, France*
jerome.lacaille@safran-group.com

^{2,3,4}*SAMM, Université Paris 1, Panthéon-Sorbonne, 75013 Paris, France*
cynthia.eruaf@gmail.com
madalina.olteanu@univ-paris1.fr
cottrell@univ-paris1.fr

ABSTRACT

How can we tell if a flight is normal or abnormal? In Safran Aircraft Engines, we are interested in the engine behavior. Some data are collected at low frequency between 1Hz up to 66Hz. These data are mainly measurements acquired from engines sensors, information coming from the aircraft that are needed to control the propulsion system and results of online computations for monitoring and maintenance. Hence, a flight appears as a big multivariate temporal signal. But it is not just a simple temporal observation, this signal is structured and may be decomposed in standard phases like start, taxi, take-off, climb, cruise, descent, reverse and taxi again. Moreover, during each standard phase there may be stabilized regimes and transient phases, the stabilized parts are easy to understand and to model mathematically. There are mainly four stabilized regimes: slow ground speed, normal cruise speed, slow descent and climb. The transient regimes are more complex as they depend a lot on the command issued by the pilot, but we identify two classes of transient phases: accelerations and decelerations. Depending on the flight plan, the airport ground geography, the day time, season and meteorology those phases may appear randomly at different instants during the journey. All of this complexity makes the comparison of different flights very difficult. Our goal in this work is to give a definition of an abnormal flight based on a new kind of metric that we build to compare those multivariate temporal series two by two.

1. INTRODUCTION

Flight data are now flowing in our databases. We cannot analyze every single observation and we need a tool to Jérôme Lacaille et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

automatically alert in case of unusual behavior and another tool to find similarities between parts of real aircraft flights. Our proposal is to systematically index the databases replacing each multivariate numerical signal acquired by the aircraft and the engine sensors by a sequence of labels. Each label should characterize a specific part of the signal such as a stationary phase or a transient phase. Stationary phases are summarized by snapshots made of statistics on multivariate signal distribution parameters and are easy to characterize. Transient phases are a more complex in a multivariate environment. This work applies a specific fast change detection algorithm to identify transient phases and an adaptive classification neural network to label each temporal behavior. However, as it seems natural to automatically separate standard flight phases like engine start, taxi, take-off, climb, etc. our goal is to identify different behaviors among those main classes. For example, we detect engines with slow thermal stabilization during the take-off and separate them from engines with fast thermal stabilization. We also separate hot engines during the climb phase to cold ones. The same sort of analysis is done on mechanical transfer functions as we may identify fast or slow crossing of specific vibration modes, etc. At the end of this segmentation and classification process, each multivariate signal is replaced by sequences of classes corresponding to relations between contexts, rotation speed for example, and any endogenous observation like temperature or vibration. Then working on discrete data, it becomes easier to query this database for rare behaviors, usual behaviors, or to search some similarity with a specific engine observation. For example, looking at a specific temporal interval during a given flight it becomes possible to ask for flights and engines with similar behavior in the historic database.

This document has two main sections, the first one (section 2) shows some applications of a metric able to compare

flights, then section 3 explains the mathematic and algorithmic tools we used to build such metric.

2. DEFINITION OF AN ABNORMAL FLIGHT

2.1. Normal or usual behavior

With a metric to compare two flights it becomes possible to define the neighborhood of a given flight. In fact, we may identify the nearest neighbors of any flight.

A very common flight is one with a minimal distances to other flights; it minimizes the sum of all distances to its neighbors. Figure 1 shows such a flight seen from the low-pressure (LP) shaft speed N1 of one engine.

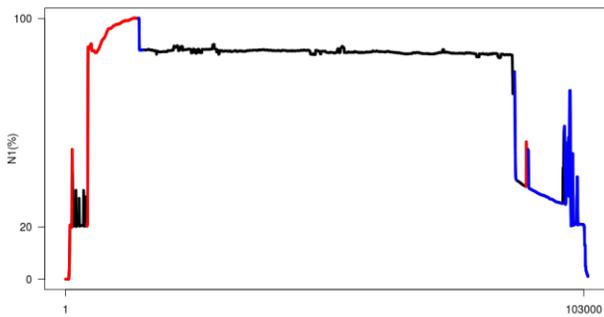


Figure 1. A good representative flight, one that minimizes the distance to its neighbors. The graphs shows the low-pressure shaft speed in percentages. This flight lasted approximately 3h30¹ (in red are the accelerations transients, in blue the deceleration and in black the stabilized phases).

The LP shaft drives the fan, which is mainly responsible of the thrust of the engine. In fact, the engine is controlled by thrust demand from the cockpit lever or the autopilot and looking at this N1 signal is a good method to analyze the context of the operation. On the figure, we colored the measurement according to different phases, in red are the accelerations, in blue the decelerations and in black the stabilized phases. We can see that stabilized phases may include some local scatter.

As N1 is the result of some controlled decision computed on the embedded computer, we may not always see a direct causality with other measurements. For example the lever angle computed by the autopilot changes before the N1. Moreover, as the engine is itself driven by a high-pressure (HP) gas generator, then the HP shaft speed N2 and measurements inside the engine core may be more reactive than the LP shaft speed N1. This should be taken into account when we will work on multivariate signals.

2.2. Unusual or abnormal behavior

Even with a metric to measure similarities between two flights, there are multiple ways to detect an unusual behavior. A flight with far neighbors is an outliers and should be investigated. Here, we are intersected in another kind of behavior: we may dispose of a plurality of different but similar ways to measure similarity. In this application, we use the stochastic characteristic of our algorithm to compute the similarity distance: here a 2D auto-adaptive Self-Organizing Map (SOM) (Kohonen, 1995). This algorithm is randomly initialized, and uses random selection of observations during its iterations. This allows us to build a set of similar metrics and according to (de Bodt, Cottrell & Verleysen, 2002) a couple of observations (flights) may not always be neighbors. A flight with almost never the same neighbors is clearly special.

Let $v_{i,j} = 1$ if two flights of positions i and j on the map are neighbors and 0 otherwise and let $V_{i,j}$ be the sum of this boolean value over R random maps. In case of random association and if we do not take into account the edges of a rectangular map, the mean value of being a neighbor is $9/U$ where U is the number of units on the map². Hence, in that case, $V_{i,j}$ follows a binomial distribution $B(R, 9/U)$.

If R is big enough (~ 100) and $9/U$ not too small (~ 0.1) then Gaussian approximation is allowed and random acceptance interval for $V_{i,j}/R$ becomes $[A - B, A + B]$ where $A = 9/U$ and $B = 1.96\sqrt{A(1 - A)/R}$. Hence

- if $V_{i,j}/R$ is less than $A - B$ we can say that the pair is repulsive,
- if $V_{i,j}/R$ is greater than $A + B$ on the contrary the pair is attractive,
- and if $V_{i,j}/R$ is between A and B , the pair is inconsistent because it behaves like if the relation between the two flights i and j is random.

Then it is possible to order the flights according to the number of inconsistent neighbors, we call the ones with big number of inconsistent neighbors fickle or inconsistent flights.

Figure 2 shows such a flight with a level increase in the middle of the cruise phase at very high altitude. In general we observe a level decrease and mostly at the end of the flight. This may be a particularity of our database because we select only one airline for our tests.

¹ This plot and the following use 8Hz acquisition data for some sensors, lower frequency measurements are interpolated.

² 8 neighbor cells and the center position.

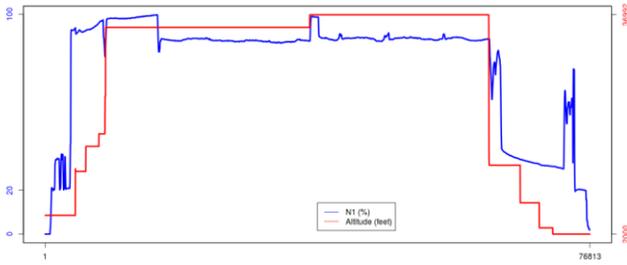


Figure 2. A fickle flight selected among the observations with the biggest number of inconsistent neighbors. The duration of this flight is around 3h30. The red line is the altitude, which is plotted along the usual low-pressure shaft speed N1.

Figure 3 is a clear case of abnormality because one see just a small take-off immediately followed by a landing. The whole trip lasted 35 minutes including taxi on the ground. This one also has almost no stable neighbor.

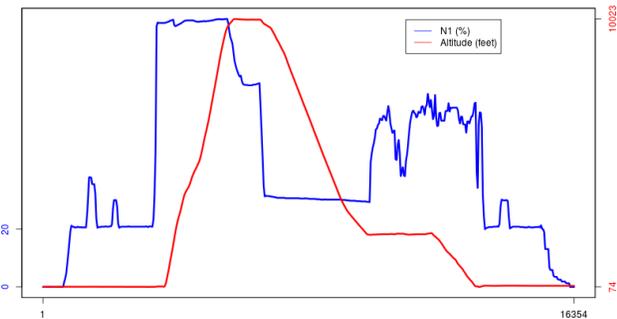


Figure 3. Another very short fickle flight. This one is probably an aborted flight because it last just 35 minutes and does not climb very high. (Remember it is a turbofan jet, small trips are very rare.)

2.3. First conclusions

- 1) With a similarity metric we are able to find representative flights, this is very nice if we expect to compute complex loads using very long finite element computations for example. This methodology helps identify the most representative flights and then it becomes possible to approximate life duration estimation by estimating a potential load counter after each flight.
- 2) As we found a stochastic methodology able to build not only one but a whole set of metrics sharing similar constraints about our interpretation of the multivariate signal, we also are able to identify very original flights (fickle flights). Forwarding this information to our system engineers will help maintenance logistic.

The next section explains our stochastic methodology to build similarity metrics³.

3. INDEXATION OF TEMPORAL MULTIVARIATE DATA

Analyzing multivariate time series created by sensors during a flight or a test bench represents a new challenge for engineers. Each signal can be decomposed into series of stabilized phases, well known by the expert, and transient phases that are hardly explored even if very informative when the engine is running. Our proposition aims at converting these time series into a succession of labels, designing transient and stabilized phases on a bivariate context to begin. This transformation of the data will allow several perspectives: tracking similar behaviors or bivariate patterns seen during a flight, detecting frequent or rare sequences of labels during a flight and discovering hidden multivariate structures.

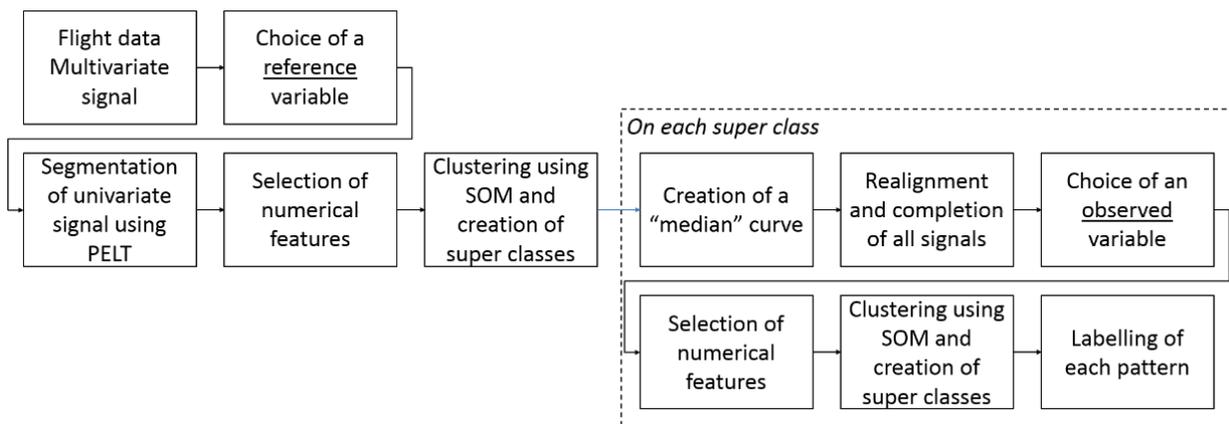


Figure 4. Schematic workflow of the segmentation process using a reference variable and one observation in the engine.

³ This is mostly a summary of Cinthia Faure PhD thesis (2018) in French and much development of intermediate results are given in her numerous publications.

(Faure, 2018) proposed an algorithm that builds a new database of transient patterns with a change-point detection method. Then, the bivariate transient patterns are clustered into a ranked number of typologies which will provide the labels. The clustering is implemented with self-organizing maps. All algorithms are applied on real flight measurements with a validation of the results from expert knowledge.

Figure 4 presents the process workflow to transform a temporal bivariate signal into a sequence of labelled patterns.

3.1. Segmentation of temporal data

Intuitively, a phase is stabilized when it does not contain any major variation, when it is not transient. The indexation of the database begins with an offline change-point detection method.

An algorithm computes the start and stop points of each phases. Different engine variables may be used as reference to segment the signal so the method has to be adaptable to the different properties each measurement might have. We mostly use the fan speed (N1) which is always acquired and stored on the data-flight recorders. A main characteristic of an engine during flight is its current thrust; our engines are regulated according to N1, which is directly proportional to the thrust. N1 is called a “reference” variable on which the clusterisation is based.

We select to represent N1 with a piecewise linear model. This is why we choose to detect changes in the slope and an algorithm that uses dynamic programming and penalty term (since the number of change-points is unknown). In order to detect potential crucial changes of characteristic variables, it is relevant to develop robust algorithms with accurate results.

Different segmentation algorithms were tested but only the PELT (Pruned Exact Linear Time) method (Killick, Fearnhead & Eckley, 2011) which achieved the best trade-off between results and computational time is used in this study (Faure, Bardet & Olteanu, 2016). This method indexes any monovariate signals into different phases: stabilized and transient. Of this partition will result a new database composed of patterns. A first simple classification into three classes is made; it creates three databases of stabilized, ascending and descending phases. They are classified with a 10% threshold on the rate of increase or decrease.

Figure 5 shows a result of the PELT algorithm. We applied this algorithm on 500 flights recorded from 8 different aircrafts of the same type and same engines type. The mean duration of one flight is around 2.8 hours. The PELT algorithm gathers around 8000 different transient phases from our N1 reference variable on which almost 4000 were ascending phases. All of the computed algorithms are done with language R and the clustering algorithm comes from the package SOMbrero (Villa-Vialaneix, Mariette, Olteanu, Rossi, Bendhaiba & Boelaert, 2017).

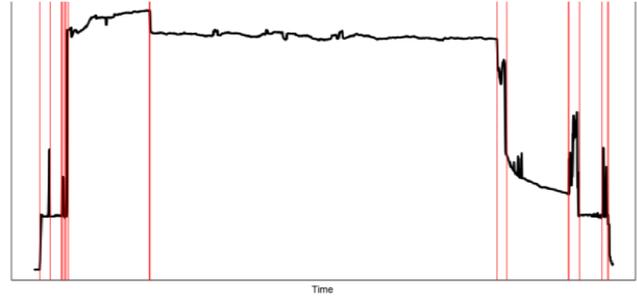


Figure 5. N1 profile in black and the result of the segmentation by the PELT algorithm in red.

3.2. Classification of transient patterns

A clustering method is applied to the transient phases. We select Self-Organizing Maps among many other segmentation tools (k-means, etc.) because this method has the best results concerning the visualization aspect and the computed errors (Faure, Olteanu, Bardet & Lacaille 2017). This technique is based on numerical features (mean, variance and a set of other statistics) extracted from the patterns.

After the pre-classification of the transient phases into two classes ascending and descending, we focused on the ascending phases first. Figure 6 shows the disparity of all ascending phases and the necessity to classify this database in relatively coherent clusters.

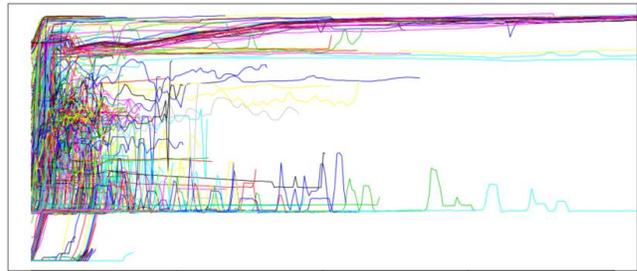


Figure 6. Plot of some ascending phases superposed on the same graph.

An 11x11 SOM is computed first with a super-classification realized by a hierarchic clustering (Figure 7). Finally, in order to validate the quality of the clustering, clusters were crossed checked with the flight mode variable (Table 1).

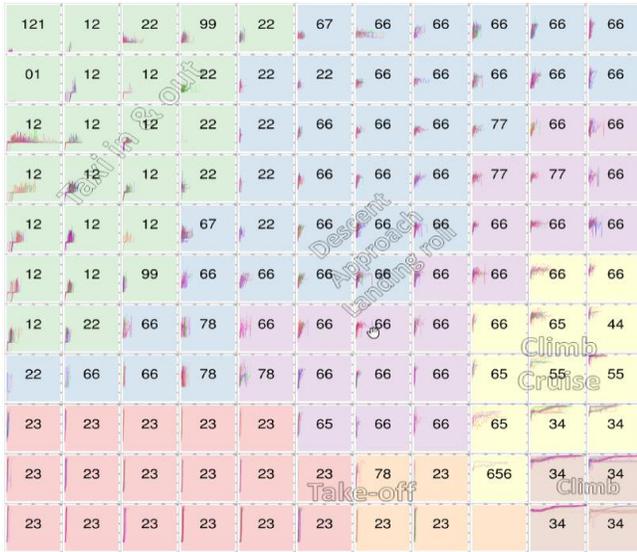


Figure 7. 11x11 SOM clustering of ascending transient phases. The numbers refers to the majority code⁴ in each cluster according to Table 1. The colored zones are computed by hierarchical clustering and were labelled by experts.

Table 1. Numeration of the official flight phases.

Preflight	0	Engine-Start	1
Taxi-Out	2	Take-Off	3
Climb	4	Cruise	5
Descent	6	Approach	7
Landing & Roll	8	Taxi-In	9

3.2.1. Synchronization

The extracted transients coming from multiple flights phases have unequal lengths and the clustering of these phases is not an easy task. The aim is to group the phases of very much alike shapes and identifies them. Dynamic Time Wrapping (DTW) (Berndt & Clifford, 1994) is not an option on our data because the time delays are of real physical importance. Pattern classification for multivariate time series frequently uses a distance measure and a hierarchical clustering method to compare and to group these time series. For this step, the unequal lengths of the phases make the definition of a similarity measure more complex. So, to transform all the phases of different lengths into patterns of equal lengths, a representative “median” curve of each clusters is selected. Each pattern of the cluster is compared and centered according to this median curve in a way that optimizes the convolution between observed and median curve (Figure 8). Depending on the pattern, the other phases must get bigger or be cut.

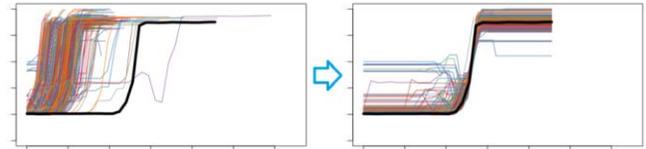


Figure 8. The synchronization of N1 patterns on a median curve.

3.2.2. Bivariate classification

Then a second “observed” variable (for example an engine temperature) is extracted on each signal according to the “reference clusters” identified previously (on referenced variable, here N1). The extraction process synchronizes all second variables according to the reference variable.

Figure 9 shows similar rotation speed increases that corresponds to different evolutions of the temperature. The second clustering step, implemented on each reference class, is based on numerical features of the new observation variable.

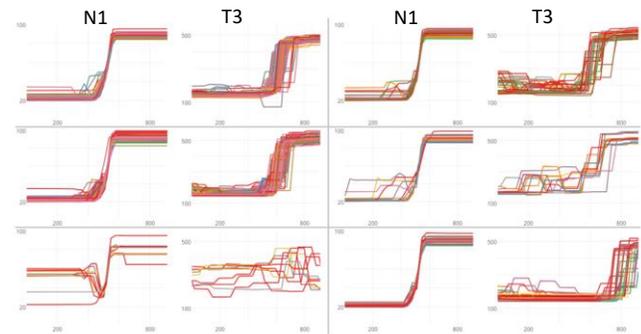


Figure 9. On this figure, each of the 6 cells corresponds to a couple of synchronized N1 (on the left) and compressor temperature (on the right) organized according to a second step of classification for the same N1 pattern. One can clearly identify fast (top left) or slow (bottom right) temperature adjustment after N1 increases.

Finally, for each couple of “referenced” and “observed” variable we get a specific behavior label. And this same process can be replayed for any couple of reference and observation (Cottrell, Faure, Lacaille & Olteanu, 2019).

3.3. Sequences of temporal patterns

At this point, a bivariate signal is transformed into a string of the type “A11 A12 D74 S1 A53 A44 D42 D44 D61 A65 D65 A65 S2 D63 A67 D27 D75” (a real flight). Each label is either an acceleration transient (A) a deceleration transient (D) or a stabilized phase (S). The following index corresponds to the first segmentation step on the reference variable (N1 here) and second step for transient phases (here on a compressor

⁴ Each label is composed of at least two digits because transient phases are across a couple of successive standard regimes.

temperature). One may see those sequences as a chromosome description and like in genetic we apply a comparison between sequences using an optimal matching distance (Wunsch & Needleman, 1970) also known as edit distance or Levenstein distance. Note that transformation of a temporal multivariate signal into a sequence of labels is a nice solution to index the data as such tools already exist for document search (Ukkonen, 1985).

Optimal matching distance is based on the definition of a cost function for each move that is used to transform one sequence into another (Studer & Ritschard, 2016). Clearly, in our case, we define such cost according to the distance between classes on the original map. We also have to deal with some constraints such that the cost of removing a label and inserting another is more expensive than exchanging both labels. Anyway, here again, the Kohonen map appears to be useful as it maintains a correspondence between distances in the original space and distance on the map (Lacaille, Bense, Berechet & Faure, 2016).

4. CONCLUSION

Temporal signals describing flights are structured into specific phases. Each phase is extracted from a univariate reference measurement. Each phase should also be modelled according to the particular physics involved in the engine. However, a big-data approach may be used with rough models, here we consider transients as piecewise linear evolution and stabilized phases as constant with some scatter.

A fast algorithm (e.g. PELT which works in linear time) is used to segment a whole database of measurements. Fast computation is really needed because we want to index a very big amount of data and we want to do it according to different reference measurements.

Once the signal is segmented, we classify the reference measurement to identify the category of phases, then inside each category, we synchronize other signals and build a second bi-variate classification for transient observations. The new signals are selected according to the specific behavior of interest like thermic or dynamic responses.

Both levels of classification use self-organizing maps to keep track of the distance between observations, followed by hierarchical clustering to limit the number of labels.

Once each signal is segmented and each segment is labelled, we have a new representation of the database into sequences of labels, which can be used to index the flight.

To compare sequences, we use an optimal matching distance which cost function is parametrized by the distances on the Kohonen maps.

This distance lets us find similar flights to a given one or identify representative flights, which may be of great help to limit computation by heavy physics finite element models.

Finally, the intrinsic random property of the Kohonen algorithm lets us build numerous solutions of indexation, hence identifying specific flights labelled as “fickle flight” or inconsistent flights by identification of specific sequences that are difficult to classify and should be observed in priority.

REFERENCES

- Kohonen, T., (1995), “Self-Organizing Maps”, Springer Series in Information Sciences (Vol. 30), Springer.
- de Bodt, E., Cottrell, M., Verleysen, M., (2002) “Statistical tools to assess the reliability of self-organizing maps”, *Neural Networks*, 15(8-9), pp 967–978.
- Faure, C., “Détection ruptures et identification des causes ou des symptômes dans le fonctionnement des turboréacteurs durant les vols et les essais”, PhD Thesis, Paris 1, Panthéon Sorbonne, 2018.
- Killick, R., Fearnhead, P., and Eckley, I.A., (2011) “Optimal detection of changepoints with a linear computational cost”, p. 37.
- Faure, C., Bardet, J.M., and Olteanu, M., (2016), “Comparison of three algorithms for parametric changepoint detection,” in *ESANN*, pp. 2–7.
- Villa-Vialaneix, N., Mariette, J.n Olteanu, M., Rossi, F., Bendhaiba, L., and Boelaert, J., (2017), “SOMbrero: SOM Bound to Realize Euclidean and Relational Outputs,” CRAN. [Online]. Available: <https://rdrr.io/cran/SOMbrero/>. [Accessed: 19-Sep-2017].
- Faure, C., Olteanu, M., Bardet, J.M., and Lacaille, J., (2017), “Unequal time series clustering applied on flight data,” in *WSOM*.
- Berndt, D., and Clifford, J., (1994), “Using dynamic time warping to find patterns in time series,” *Work. Knowl. Knowl. Discov. Databases*, vol. 398, pp. 359–370.
- Cottrell, M., Faure, C., Lacaille, J., Olteanu, M., (2019), “Anomaly detection for bivariate signals”, in *IWANN*.
- Wunsch, C.D. and Needleman, S.B., (1970), “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, *Journal of Molecular Biology*, 48(3), 443–453.
- Ukkonen, E., (1985), “Algorithms for Approximate String Matching”, *Information and Control*, 64(1–3), 100–118.
- Studer, M., and Ritschard, G., (2016), “What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures”, *Journal of the Royal Statistical Society. Series A: Statistics in Society*, 179(2), 481–511.
- Lacaille, J., Bense, W., Berechet, I. & S., Faure, C., (2016), “Indexation of Numeric Bench Test Records, a Big Data Vision”, in *IEEE Aerospace Conference*.

BIOGRAPHIES

Jérôme Lacaille is a Safran Emeritus Expert whose mission is to help in the development of mathematical algorithms used for the engine health monitoring and statistical analysis of company data. Jérôme has a PhD in Mathematics on “Neural Computation” and habilitation thesis on “Algorithms Industrialization” from the Ecole Normale Supérieure (France). He is at the origin of Safran aircraft Engine DataLab.

Cynthia Faure is a PhD doctor in statistics from University Paris 1, Pantheon-Sorbonne and Safran Aircraft Engines. She is currently working on aeronautic data for Aosis Consulting.

Madalina Olteanu graduated the University of Bucharest in 2001, and obtained her Ph.D. in Applied Mathematics in December 2006, from University Pantheon-Sorbonne. Since Sept. 2007 she is an Assistant Professor at University Paris 1 Pantheon-Sorbonne and member of the SAMM research team (Statistics, Analysis and Multidisciplinary Modelization). Her main research interests are nonlinear time series, model selection, machine learning and applications to humanities and social sciences. She is author or co-author of more than 30 peer reviewed papers in international journals or conference proceedings.

Marie Cottrell is Emeritus Professor at the SAMM laboratory of University Paris 1, Panthéon-Sorbonne.