Design Aircraft Engine Bivariate Data Phases using Change-Point Detection Method and Self-Organizing Maps

Jean-Marc Bardet, Cynthia Faure, Jérôme Lacaille, Madalina Olteanu

Design Aircraft Engine Bivariate Data Phases using Change-Point Detection Method and Self-Organizing Maps

Jean-Marc Bardet, Cynthia Faure, Jérôme Lacaille, and Madalina Olteanu

SAMM, EA 4543
Panthéon-Sorbonne University
90 rue de Tolbiac, 75013 Paris, France
http://samm.univ-paris1.fr
&
Safran Aircraft Engines,
Rond Point René Ravaud, Réau, 77550 Moissy Cramayel, France
https://www.safran-aircraft-engines.com

Abstract. Analysing multivariate time series created by sensors during a flight or a bench test represents a new challenge for aircraft engineers. Each time series can be decomposed univariately into a series of stabilised phases, well known by the expert, and transient phases that are merely explored but very informative when the engine is running. Our project aims at converting these time series into a succession of labels, designing transient and stabilised phases in a bivariate context. This transformation of the data will allow several perspectives: tracking similar behaviours or bivariate patterns seen during a flight, detecting frequent or rare sequences of labels during a flight and, discovering hidden multivariate structures. This manuscript proposes a methodology to automatically cluster all engine transient phases. First, the algorithm builds a new database of transient patterns with a change-point detection method. Second, 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 [SOM]. All algorithms are applied on real flight measurements with a validation of the results from expert knowledge.

1 Introduction

Multiple sensors placed on aircraft engines are daily generating important amounts of data, which are a big concern for engineers. Indeed, experts are interested in extracting patterns, such as, for instance, unusual behaviors or response delays between engine variables. These patterns are almost impossible to detect manually, due to the size and the complexity of the data, hence automatic tools, mixing time series and statistical learning techniques, are needed by the experts so that they may analyse and quickly compare interesting patterns.

Usually, aircraft engine data is provided as a multivariate time series, and it characterizes the behavior of the engine during the flight. One approach for

dealing with this data is to segment it into a succession of stabilized and transient phases. Whereas several methodologies and algorithms (health diagnosis, anomaly detection, and wear patterns) were designed under the stationarity assumption for stabilized phases, [12] and [9], transient phases are poorly mentioned in the literature, although they are of real interest for the engineers. Most of the interesting patterns, for example, occur during transient phases: take-off, landing, ... The aim of the present manuscript is to focus on transient phases, while attempting to identify, isolate, and characterize them. In a previous work, [5], the data was analysed as a univariate time series only, after having extracting one of the signals, suggested by the experts as being of particular interest. In this paper, the adopted point of view is a multivariate one: the time series will be first split into patterns, then the patters will be clustered into meaningful groups. The eventual goal is to provide a new way of expressing the complex multivariate data as a sequence of labels, making it easier for experts analysis.

The first step of the methodology proposed here is to split the data into stabilized and transient phases. Intuitively, a phase can be considered as stabilized whenever it does not contain any major variation. Using expert knowledge, a major variation for this kind of data is given by a fixed threshold measuring the absolute difference between the first and the last value of the phase (10% in our case). The partitioning is carried out in a univariate framework, similarly to [5], on a signal qualified as "reference variable" by the experts, such as the fan speed. Taking into account the characteristics of the extracted univariate time series which has a piecewise linear behavior, and also the above definition of a transient phase, the partitioning is done by detecting change-points in the slope.

In a previous work devoted to finding the most adequate strategy for segmenting the data [4], several algorithms for change-point detection in offline fashion were tested. Among them, the best trade-off between the computational complexity and the performances in correctly identifying the right number and the right positions of the change-points was achieved by the PELT method [7]. In the present manuscript, this algorithm only was used, with a cost function derived from the sum of squared errors in linear regression and a penalty term similar to that of the BIC criterion.

Once the univariate "reference time series" has been partitioned, the resulting phases or patterns are grouped, according to the definition of transient phases, into stabilized, ascending transient and descending transient. The second step of our methodology is to separately cluster ascending patterns and descending patterns, while also introducing a multivariate aspect in the clustering, as it will be subsequently explained. The clustering step may in turn be divided into several substeps.

First, the univariate transient phases previously identified are clustered. This task is not immediate, since the lengths of the patterns are not identical. Two strategies may at this point be considered: either use specific similarities and/or dissimilarities for time series and a clustering method suited for relational data [2] [10], or extract a fixed number of numerical features characterizing the patterns and use clustering algorithms designed for vector data. As already described in

[5], after having tested several clustering algorithms and several ways to resume the data, the self-organizing maps (SOM) algorithm [8], trained on a set of numerical features, provides good clustering results, as well as graphical tools for visualizing the groups of patterns.

Second, the clusters obtained on the patterns of the "reference" univariate time series are next divided into subclusters, which take into account a second signal or time series. This step is the main contribution of the present manuscript with respect to our previous work, since we refine the initial univariate clustering proposed in [5] by considering supplementary information. The bivariate framework may allow the observation of interesting phenomena which was not possible in the univariate context such as response time between the signals. The resulting final clustering is eventually validated, using either expert knowledge, or additional variables in the data set and resuming the states of the engine.

The rest of the manuscript is organized as follows: Section 2 contains the detailed methodology on segmenting and clustering bivariate time series, while Section 3 overviews the experimental results on real data. The conclusion as well as the future work tracks are given in Section 4.

2 Segmenting and clustering bivariate time series

Let us now describe the methodology proposed here for extracting and clustering patterns from bivariate time series. Let $(Y^l_{1:N_l,i=1,2})_{l=1,...,L}$ be the available data which corresponds to L flights, each flight l being of length N_l. For each flight, a bivariate time series, corresponding to two signals of interest for the engineers (the "reference variable" and an additional one) is considered.

The first step of the analysis consists in partitioning a univariate time series, the "reference" signal, in order to identify the patterns, while the second step is the bivariate clustering of the patterns. The complete procedure is illustrated in Figure 1.

2.1 Pattern extraction

For each flight l, the univariate "reference" time series $(Y^l_{1:N_l,1}) = (Y^l_{1,1}, ..., Y^l_{N_l,1})$ is partitioned into patterns or phases. As mentioned in the introduction, due to the piecewise linear behavior of the signal, partitioning is achieved through offline change-point detection in the slope.

During this step, for a flight l we estimate the optimal number of change-points K^*_l and their positions $(\tau^*_j)_{j=1,...,K_l}$, by minimizing the penalised cost function:

$$\mathcal{C}(Y^l_{1:N_l,1}, ({\theta^*}^l_{{\tau^*}^l_j+1,{\tau^*}^l_{j+1}})_{j=1,...,K^*_l}) = \sum_{j=1}^{K^*_l} \Big(C(Y^l_{{\tau^*}^l_j+1:{\tau^*}^l_{j+1},1}, {\theta^*}^l_{{\tau^*}^l_j+1,{\tau^*}^l_{j+1}}) + \beta \Big)$$

in $K^*_l, ({\tau^*}^l_j)$ and ${\theta^*}^l_{{\tau^*}^l_j+1,{\tau^*}^l_{j+1}}$.

where $C(Y^l_{\tau^{*l}_j+1:\tau^{*l}_{j+1},1}, \theta^{*l}_{\tau^{*l}_j+1,\tau^{*l}_{j+1}})$ is the minimum description length [3] associated to a linear regression model and β the penalty term (BIC). Let θ^l be the estimate of θ^{*l}. The problem becomes:

$$(K_l, \tau^l_1, \dots, \tau^l_{K_l}) = \underset{K^*_l;\tau^{*l}_1<\tau^{*l}_2<\dots<\tau^{*l}_{K^*_l}}{\arg\min} \left\{ \sum_{j=0}^{K^*_l} \sum_{t=\tau^{*l}_j+1}^{\tau^{*l}_{j+1}} \left(C(Y^l_{\tau^{*l}_j+1:\tau^{*l}_{j+1},1}, \theta^l_{\tau^{*l}_j+1,\tau^{*l}_{j+1}}) + \beta \right) \right\} \tag{1}$$

where $\beta = 2 \times \ln(K_l)$ and also for $t = \tau^l_j + 1, ..., \tau^l_{j+1}$, $Y^{l,j}_t = (\theta^{(1)}_{\tau^l_j+1,\tau^l_{j+1}})^l\, t + (\theta^{(2)}_{\tau^l_j+1,\tau^l_{j+1}})^l + \epsilon_t$ where $\epsilon_t \sim N(0, \sigma^2)$.

$$C(Y^l_{u:v,1}, \theta^l_{u,v}) = 3\ln(v-u) + (v-u)\log(2\pi\hat{\sigma}^2)$$
$$\text{where } \hat{\sigma}^2 = \frac{1}{v-u} \sum_{t=u+1}^{v} \left(Y^l_{t,1} - (\theta^{(1)}_{u,v})^l\, t - (\theta^{(2)}_{u,v})^l\right)^2$$

The optimization step is implemented using PELT algorithm which provides a good tradeoff in term of complexity and performances [4]. Once the PELT procedure has been trained on all the flights in the data set, one gets the set of all detected phases or patterns, $(Y^{l,j}_{\tau^l_j+1:\tau^l_{j+1}})_{l=1,\dots,L\ ;\ j=1:K^l-1}$ where l is the flight, j is the index of the pattern within the flight l. K_l is the number of the detected change-points $(\tau^l_j)_{j=1,\dots,K_l}$ of the flight l.

2.2 Pattern clustering

Next, using the empirical definition of transient phases and a fixed threshold of 10%, the extracted patterns are classified into stabilized, ascending transient and descending transient. Only the results on the ascending phases will be illustrated in the present manuscript, but the same framework could be used for descending phases.

In order to simplify the notations, suppose that $(K'_l)_{l=1:L}$ is the number of ascending transient phases detected for the l-th flight. Let $P = K'_1 + ... + K'_L$ be the number of all ascending transient patterns previously detected. By identification, only start points τ^s and end points τ^e of transient phases will be considered. Let $(\tau^l_q)^s_{q=1,\dots,K'_l}$ and $(\tau^l_q)^e_{q=1,\dots,K'_l}$ be all start and end points of the q-th ascending transient pattern of the flight l.

Once the patterns are identified, Self-Organizing Maps [SOM] method is applied for the clustering part. First the univariate methodology is described then the bivariate clustering is developed.

Univariate clustering Let $(\bar{Y}_{k:(k+u_k)})_{k=1,\dots,P}$ be all ascending transient patterns. $(\bar{Y}_{k:(k+u_k)})$ corresponds to the q-th ascending phase within the flight l, so $k = \sum_{l'=0}^{l-1} K'_{l'} + q$ where $K'_0 = 0$ and $u_k = (\tau^l_q)^e - (\tau^l_q)^s$.

Before training the clustering algorithm, online numerical SOM in this case, each pattern $(\bar{Y}_{k:(k+u_k)})$ is summarized by a vector of M_1 numerical features $(X_m^k)_{m=1,...,M_1}^{k=1,...,P}$. Then, these numerical vectors are clustered onto a sufficiently large SOM map, which ensures a good quality of the mapping and provides a first visualization tool for the experts.

Since the number of clusters issued from the SOM is too large for a meaningful typology of the patterns, it is reduced by alternatively testing either hierarchical clustering (AHC) or K-means. The criterion used for selecting the final (smaller) number of clusters ϕ (which will be the superclasses) is a fixed threshold on the within-class variance. A superclustering based on an optimal number of superclasses computed with the explained variance is applied: Let $(SC_1^1, ..., SC_\phi^1)$ be the new clusters [5]. For $v = 1, ..., \phi$, let a superclass $SC_v^1 = (\bar{Y}_{k:(k+u_k)})_{k \in I_v}$ where I_v is the set of all ascending transient phases that belong to the cluster v, so $\bigcup I_v = \{1, ..., P\}$.

Topographic and quantization errors are computed to validate the SOM. Experts can also validate it thanks to the easier interpretation.

Bivariate clustering Once $(SC_1^1, ..., SC_\phi^1)$ are computed, one wants to investigate these patterns in a bivariate framework. At this point, each cluster $(SC_v^1)_{1,...,\phi}$ will be divided into subclusters using a second signal or variable.

As mentioned before the next clustering allows a more in-depth study of the transient patterns already grouped with similar characteristics : comparison between bivariate patterns, analysis of time response between the signal without missing information... For example, when a variation occurs in a transient phase on signal (temperature, pressure, speed...), a variation might respond to this change in another variable but with a certain delay. For such analysis, transient phases with unequal length are not suitable. For this reason, all patterns in a cluster SC_v^1 must be re-aligned and if necessary lengthened or shortened so that they all have the same size (for example the take-off phases will be re-aligned based on their ascending shapes).

In order to carry out the alignment step, a reference curve $\breve{Y}_v$ is picked among $(\bar{Y}_{k:(k+u_k)})_{k \in I_v}$. For each superclass v a distance is defined to compare the patterns. After computing all distances by pairs (distance matrix), the minimum of the sum of each line of the distance matrix gives $\breve{Y}$. Let d be a distance between two patterns:

$$\breve{Y}_v = \underset{(\bar{Y}_{k_1,u_{k_1}})_{k_1=\{1,...,P\}}}{\arg\min} \left(\sum_{k_2=1;k_1 \neq k_2}^{P} d(\bar{Y}_{k_1,u_{k_1}}, \bar{Y}_{k_2,u_{k_2}}) \right) \tag{2}$$

Then all the phases are aligned and completed for each $(SC_v^1)_{v=1,...,\phi}$ according to $\breve{Y}_v$ to set all patterns at the same length. All phases of SC_v^1 are shifted (or not) to the right to be aligned with $\breve{Y}_v$. If there is a move to the right, information are missing in the left of the pattern and must be completed. Then if the phase is longer or smaller than $\breve{Y}_v$, the pattern will be shortened or lengthened. The completing part is possible because we have all the records.

All transient phases have now new start and end points: $(\tilde{\tau}_q^l)^s = (\tau_q^l)^s + \alpha_1$ and $(\tilde{\tau}_q^l)^e = (\tau_q^l)^e + \alpha_2$ where α_1 and α_2 may be different for each phase with $\alpha_1 \geq 1 - (\tau_q^l)^s$ and $\alpha_2 \leq N_l - (\tau_q^l)^e$. α_1 can never be negative with the distance we chose because there we assume there is no shifting to the left. If $\alpha_1 > 0$ then the new phase start at the index α_1 (same if $\alpha_2 < 0$). If $\alpha_2 > 0$ then we have to add α_2 elements on the right of the signal. Let $N_{\breve{Y}}$ be the length of $\breve{Y}$, the aligned and completed phases are $(\bar{Y}_{k':(k'+N_{\breve{Y}})})_{k' \in I_v}$ where $(\bar{Y}_{k':(k'+N_{\breve{Y}})})$ is the k'-th ascending pattern with the uploaded start and end points.

In each $(SC_v^1)_{v=1,\dots,\phi}$ all transient patterns will be completed with a new signal pattern to obtain bivariate phases. This added pattern will be extracted directly from the entire signal with given start and end points. These will be the same start points $(\tilde{\tau}_q^l)^s_{q=1,\dots,K_l'}$ and end points $(\tilde{\tau}_q^l)^e_{q=1,\dots,K_l'}$ as the first signal.

A second SOM is computed on each SC_v^1. The input data is $(Z_m^{k'})_{m=1,\dots,M_2}^{k'=1,\dots,P}$ where M_2 is the number of numerical features extracted from the second signal $(\bar{Y}^2_{k':(k'+N_{\breve{Y}})})_{k' \in I_v}$. The same methodology of clustering is applied on the second variable. The optimal number of clusters is crucial at this point and must be automatic for each SC_v^1 with a fixed threshold on the percentage of explained variance: $((SC_w^2)_{(w=1,\dots,\psi_v)})_{v=1,\dots,\phi}$ clusters are created where ψ_v is the number of superclasses of the superclass SC_v^1: $SC_w^2 = (\bar{Y}^2_{k':(k'+N_{\breve{Y}})})_{k' \in J_w}$ where J_w is the set of all ascending transient phases that belong to the cluster w and $\bigcup J_w = I_v$.

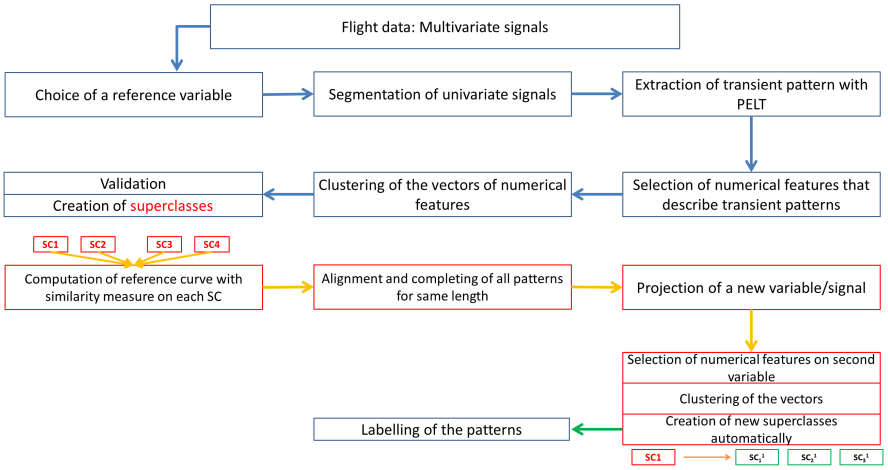

Fig. 1: Procedure of the methodology described step by step

3 Applications on flight data

The previous methodology is applied on flight data. The segmentation of the monovariate signals is briefly described. Then the results of pattern clustering are detailed.

3.1 Data information

About 500 flights coming from 8 different engines (with different take-off places and different landing positions) are analysed. In this database, the sensors present on the engine recorded around 50 variables. Each flight is described by multiple variables (for example the fan speed, the lever of the pilot, temperatures,...) with the same frequency. The mean duration of one flight is around 2,8 hours.

There are two main speeds in the engine: fan speed and core speed. Among all the variables, the fan speed is picked as a key variable (for examples see Figure 2). The control system is driven by the fan speed which is expressed as a percentage (ratio w.r.t the maximum). This variable is controlled by the pilot with the lever for the rev up. Fan speed is considered as a key variable because of its main impact on the functioning of the engine. The shape of the fan speed's measurements (and also of other variables) can be defined as piecewise linear. Therefore fan speed will the reference signal. Some features of the fan speed are enumerated in Table 1.

Mean of minima [SD]	0% [±0]
Mean of maxima [SD]	99% [±1, 2]
Mean of medians [SD]	76% [±21]
Mean of interquartile ranges [SD]	23% [±22]

Table 1: Information about the fan speed [SD=Standard deviation]

The Flight Mode [FM] variable is introduced for the validation part of the methods. Different well-known flight modes have been already defined for aircraft data. In Figure 2, the altitude and the fan speed are projected on the same graph with the flight modes. The tool has to detect transient patterns but not necessarily the transition between flight modes (for test data, flight mode variable is not available).

3.2 Pattern extraction

The change-points detected with the PELT criterion are illustrated in Figure 2. Globally, the change-points are well detected, even the small ones. The PELT algorithm gathered around 8000 transient phases of the reference variable including around 4000 ascendant transient phases (see Figure 3 for visualization of one engine). This algorithm is implemented with R [11] and the clustering algorithms come from the package SOMbrero [1]. Another package already exists for the PELT method [6] but the cost function can not be an argument.

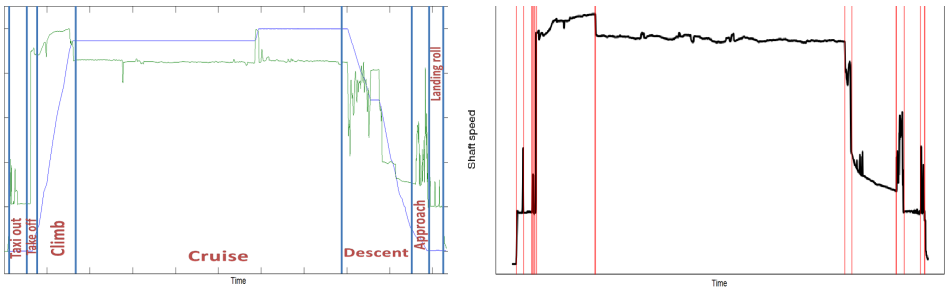

Fig. 2: Fan speed (green) and altitude (blue) during flight [left] & result of PELT on the fan speed [right].

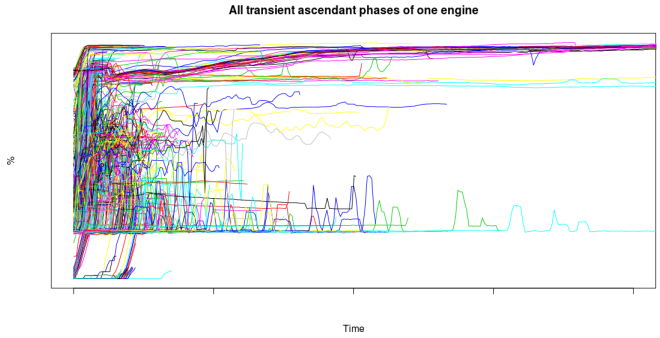

Fig. 3: Visualization of all transient phases for only one engine. The colour is randomly distributed for a better distinction between the curves.

3.3 Pattern clustering

Univariate clustering As previously mentioned, the numeric SOM method will be applied after the pre-classification of the transient phases into two classes "Ascending" and "Descending". We focused on the ascending phases first. In [5], a 11x11 SOM is computed and the 3x3 super-clustering is shown in Figure 4. The optimal size based on the explained variance is 7 clusters but we choose 9 clusters since the computed errors are still small. Also there would be less data in each cluster in this case which is easier for an exploratory analysis. The numerical features extracted for each phases are: start point, end point, length, mid point, median, variance, variance of first half, variance of second half, mean of first half, mean of second half.

Finally, in order to validate the quality of the clustering, clusters were crossed with the FM variable, which is computed by the manufacturer. This variable is computed based on the fan speed, and other variables (the valve, the pilot's lever, the altitude,...). According to the value of different variable, a state is reached. This variable is available but it will not be used in our methods (because it is only available for this type of data). The following modes are enumerated with

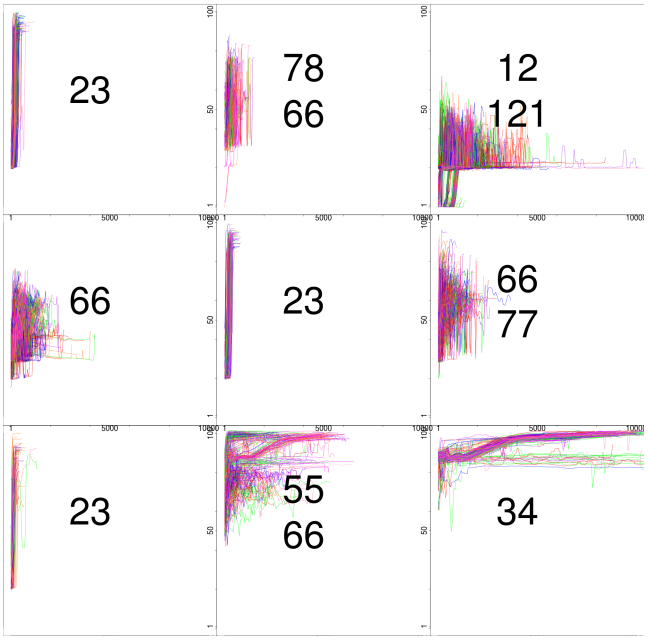

Fig. 4: Numeric super-custering 3x3 with flight modes of ascending phases

their digits labels : Pre-Flight [00], Engine Start [11], Taxi-Out [22], Take-off [33], Climb [44], Cruise [55], Descent [66], Approach [77], Landing Roll [88] and Taxi-In [99]. A variable present in the database can tell in which state an action is occuring. The detection of transient phases is done without considering the existence of the states of the flight but they help in the a posteriori validation of the quality of this map by experts. We assign a two or three digits label for each pattern, for example, a transient phase can have two consecutive flight modes: take-off (33) and climb (44). So the corresponding FM for this pattern is '34' which represents the transition state between take-off and climb.

For every detected pattern, we search its equivalent in the FM variable. For one transient phase, it is possible (and even expected) that it goes through multiples modes. On Figure 4, the FM is assigned to each cluster. The displayed figures represent the majority of states in the cluster and one can see that FM are well displayed on the map. Similar curve's shapes are grouped in the same area of the map. The projected flight modes mostly show that these phases occur during at least one transient state. So the validation with the FM variable shows a good clustering of transient patterns.

In Figure 4, there are 3 clusters of mostly take-off phases (clusters (1,1) [1st row and 1st column], (2,2) and (3,1)) but in clusters (1,2) and (2,3), there are some transient phases belonging to the descent and the landing phases. For the bivariate clustering, we will use one of the take-off superclasses.

Bivariate clustering For each superclass of Figure 4, bivariate phases are created based on a reference curve. In Figure 5 one can see that the superclass has phases of unequal lengths. The reference curve (in black) has to be close in term of distance and shape with all the other curves. The distance used in equation (2) is the Euclidean distance. Each element of the distance matrix is computed as follows:

- Take the smallest pattern between the two transient phases: $(\bar{Y}_{k^*,k^*+u_k^*}) = \bar{Y}^*$ and $(\bar{Y}_{k^{**},k^{**}+u_k^{**}}) = \bar{Y}^{**}$ of lengths n^* and n^{**} (let's assume that $n^* < n^{**}$ for this example).
- Slide it on the biggest phase: Create a vector V of length $|n^{**} - n^*|$. at each step $(i)_{i=0:(|n^{**}-n^*|-1)}$, compute $V(i) = \frac{1}{n^*} \sum_{s=i}^{i+n^*} |\bar{Y}^*_{1:n^*} - \bar{Y}^{**}_{(1+s):(s+n^*)}|$.
- Take the minimum of V as the result of the distance between two phases.

The methodology described with equation (2) is applied on the distance matrix to define the reference curve for each $SC^1_{1:\phi}$. All patterns are then aligned based on the reference curve. If after the alignment some values are missing or exceeding, the patterns are respectively lengthened with the rest of the signal (the indexes of the initial signal are saved) or shortened. After projecting a new signal, bivariate transient phases with equal length are created. With the recommendation of the experts, all curves will be enlarged by 25 seconds to the right so no delays effect will affect the clustering. This delay might depend on the conditions of the flight (difference of temperatures, climate of the city...).

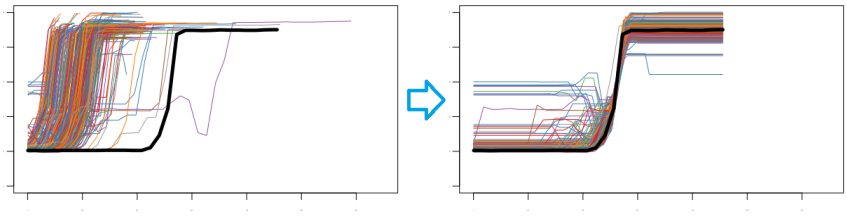

Fig. 5: Tranformation of phases of different lengths to equal length phases.

The same methodology is applied on the temperature data. The same numerical features are chosen (except the length). These input data are used as argument for a large numeric SOM 10x10 and based on the analysis of the explained variance, an optimal number of superclasses is computed. However this number needs to be automatic for each superclass from Figure 4. We chose a threshold of 80% of the information to select the number of superclasses $(\psi_i)_{i=1,\dots,\phi}$. K-means method happens to have more accurate result and 80% of the information is reached faster than the AHC method, so we used this for the optimal number of clusters. Then again with the K-means method, new superclasses are computed and the result is displayed in Figure 6.

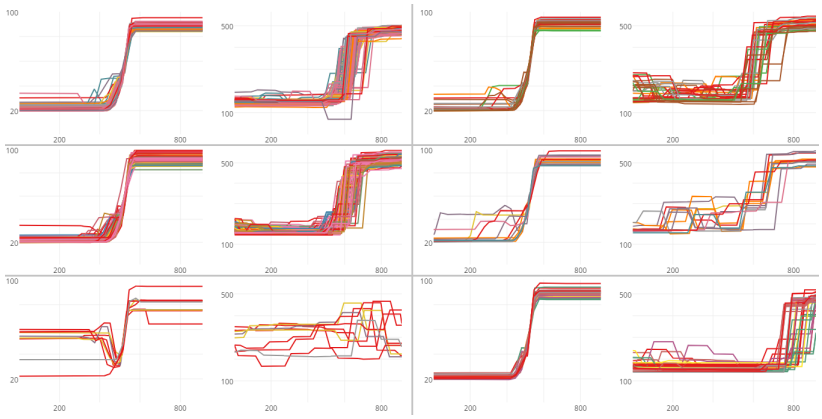

Fig. 6: Bivariate representation fan speed (left) and temperature (right) of the numeric super-custering 3x2 based on the temperature

In Figure 6, on each cluster one can see fan speed (left of the cluster) and the temperature (right). The SOM method was applied only on numerical features of the temperature. The rise of the temperature is expected after the take-off of the fan speed but with a certain delay. It is really interesting for the expert to see different delays even if the fan speed's take-off are similar. For example in the cluster (3,1) the rise of the temperature is not visible yet but it will later. In the cluster (2,2) a big variation of the temperature occurs before the rise. All these phenomena are the ones we wanted to detect through this work.

4 Conclusion

For each bivariate time series, the extracted patterns with the PELT algorithm are split into three types of phases (ascendant, descendant and stabilised). We used numeric SOM to achieve the clustering of patterns. Based on a synthesis of this classification (with K-means and Ascending Hierarchical Classification methods), we create a 3x3 super-clustering that shows the most frequent patterns.

On each superclass, after picking a reference curve using sliding curve method and the Euclidean distance, we created a methodology to transform the monovariate transient phases into bivariate transient phases. The bivariate phases have the same length and are aligned based on their shapes. This new representation is needed for the next clustering. The same clustering method is applied on the second selected variable phases (here temperature). This new representation allow the experts to discover new visualizations and behaviours on the engines never observed so far.

The last step is the labelling of the clusters so the phases can be replaced by characters. One way is to assign a letter to each cluster SC_v^1 and the label

of the transient phases that belong to this cluster will start with this letter. For the second clustering, instead of letters, numbers will be assigned.

Many possibilities are now available with this new representation of flight measurements: statistical and scoring approaches can be computed on each cluster (for example the repartition of each engine in each cluster, understand why some patterns are isolated, etc...) Also, the tracking of a specific pattern is possible (which was more difficult with the initial storage of the signals).

References

1. Bendhaiba, L., Boelaert, J., Mariette, J., Olteanu, M., Rossi, F., Villa-Vialaneix, N.: SOMbrero: SOM Bound to Realize Euclidean and Relational Outputs (2016), r package version 1.2
2. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. Workshop on Knowledge Knowledge Discovery in Databases 398, 359–370 (1994), http://www.aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-031.pdf
3. Davis, R.A., Lee, T.C., Rodriguez-Yam, G.A.: Structural break estimation for nonstationary time series models. Journal of the American Statistical Association 101, 223–239 (2006)
4. Faure, C., Bardet, J.M., Olteanu, M., Lacaille: Comparison of three algorithms for parametric change-point detection. In: ESANN. pp. 2–7 (2016)
5. Faure, C., Bardet, J.M., Olteanu, M., Lacaille: Using self-organizing maps for clustering and labelling aircraft engine data phases. In: WSOM (2017)
6. Killick, R., Eckley, I.A.: changepoint: An R package for changepoint analysis. Journal of Statistical Software 58(3), 1–19 (2014), http://www.jstatsoft.org/v58/i03/
7. Killick R., F.P., Eckley, I.: Optimal detection of changepoints with a linear computational cost. JASA 107(500), 1590–1598 (2012), http://arxiv.org/abs/1101.1438
8. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
9. Lacaille, J., Gerez, V.: Online Abnormality Diagnosis for real-time Implementation on Turbofan Engines and Test Cells. Phm pp. 1–9 (2011)
10. Olteanu, M., Villa-Vialaneix, N.: On-line relational and multiple relational SOM. Neurocomputing 147(1), 15–30 (2015)
11. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2016), https://www.R-project.org/
12. Rabenoro, T., Lacaille, J., Cottrell, M., Rossi, F.: Anomaly detection based on indicators aggregation. In: Proceedings of the International Joint Conference on Neural Networks. pp. 2548–2555 (2014)