Large-scale Vibration Monitoring of Aircraft Engines from Operational Data using Self-organized Models

Florent Forest[1], Quentin Cochard[2], Cecile Noyer[3], Adrien Cabut[4], Marc Joncour[5], Jérôme Lacaille[6], Mustapha Lebbah[7], and Hanene Azzag[8]

[1,2,3,4,5,6] *Safran Aircraft Engines, 77550 Réau, France*
first.last@safrangroup.com

[1,7,8] *LIPN (UMR CNRS 7030), Université Sorbonne Paris Nord, 93430 Villetaneuse, France*
last@lipn.univ-paris13.fr

ABSTRACT

Vibration analysis is an important component of industrial equipment health monitoring. Aircraft engines in particular are complex rotating machines where vibrations, mainly caused by unbalance, misalignment, or damaged bearings, put engine parts under dynamic structural stress. Thus, monitoring the vibratory behavior of engines is essential to detect anomalies and trends, avoid faults and improve availability. Intrinsic properties of parts can be described by the evolution of vibration as a function of rotation speed, called a vibration signature. This work presents a methodology for large-scale vibration monitoring of operating civil aircraft engines, based on unsupervised learning algorithms and a flight recorder database. Firstly, we present a pipeline for massive extraction of vibration signatures from raw flight data, consisting in time-domain medium-frequency sensor measurements. Then, signatures are classified and visualized using interpretable self-organized clustering algorithms, yielding a visual cartography of vibration profiles. Domain experts can then extract various insights from the resulting models. An abnormal temporal evolution of a signature gives early warning before failure of an engine. In a post-finding situation after an event has occurred, similar at-risk engines are detectable. The approach is global, end-to-end and scalable, which is yet uncommon in our industry, and has been tested on real flight data.

1. INTRODUCTION

Vibration analysis is an important component of condition monitoring of rotating industrial equipment (Randall, 2004, 2011). Condition monitoring (CM) of industrial assets is a set of techniques that aims at increasing machine availability and safety, while reducing maintenance costs (and thus the ownership cost). It is at the core of a predictive maintenance (PM) strategy (also called condition-based maintenance). Indeed, implementing a condition-based maintenance program requires in-depth knowledge of the machine's condition. Vibration analysis provides this knowledge by enabling to look inside a rotating machine. Its applications include the detection of unbalance, misalignment, or flutter, due for instance to gears, rollings or bearings damage or even cracks or loose parts.

Aircraft engines in particular are complex rotating machines where vibrations put engine parts under dynamic structural stress. In this work, we are interested in turbofan engines used in civil aircraft. PM for aircraft engines consists in adapting the maintenance plan to the actual state of each individual engine, unlike traditional time-based preventive maintenance, the state of each engine being the result of its actual use during its lifetime. This allows a more efficient scheduling of preventive and corrective actions (e.g. shop visits): time between actions can be increased if no maintenance is necessary (thus reducing costs), and actions can be taken earlier thanks to enhanced predictability of events (thus improving safety). Concretely, CM combines historical data and physical models to raise alerts, build models that evaluate wear of parts and their residual useful life, probability of failure, etc. These models can be based on thresholds, statistical models incorporating physical knowledge, or machine learning, i.e. statistical models whose parameters are learned from historical data. In this work, we tackle monitoring and raising alerts. Diagnosis and prognosis are then done by relevant experts.

Modern aircraft are equipped with thousands of sensors, generating huge amounts of data during each flight. At the same time, air traffic is growing exponentially. Due to this increasing volume and velocity, we clearly are in a Big Data context, which implies the use of scalable infrastructure and

software tools to effectively handle operational data for CM (Forest, Lacaille, Lebbah, & Azzag, 2018). In this work, we present a methodology for vibration monitoring of a fleet of civil aircraft engines using historical flight data and unsupervised learning algorithms. Every step, from ingestion to visualization, is made scalable through distributed processing on a cluster using the Spark framework.

We propose a method for vibration monitoring of a fleet of civil aircraft engines using historical flight data, based on distributed processing on a cluster and unsupervised learning algorithms. Such global and large-scale approaches are yet uncommon in aerospace industry. Our main contribution is two-fold:

- First, we present a pipeline to massively extract vibration signatures from time-domain medium-frequency flight recorder data, stored on a Big Data platform. Every step of the process is flexible, generic and scalable, and can be easily tuned by engineers to solve various use cases.
- Second, vibration signatures are classified and visualized using interpretable self-organized clustering algorithms, yielding a visual cartography of vibration profiles. The resulting models can be used by domain experts for monitoring, anomaly detection, giving early warnings and other insights. As an example, we show it can be used to detect anomalies, compute anomaly scores, or find similar engines (which is useful to identify at-risk engines in a post-finding situation after an event has occurred).

Our method has already been tested on real flight data from operating aircraft, and is intended to be part of the ground component of an EHM system (Bastard, Lacaille, Coupard, & Stouky, 2016).

2. RELATED WORK

In this section, we will first provide a brief review of vibration analysis techniques and how they are applied to aircraft engines. Then, we present applications of unsupervised learning algorithms, and in particular self-organized maps for clustering and visualization of high-dimensional data.

2.1. Vibration analysis on aircraft engines

A turbofan engine is composed of two main shafts, the low pressure (LP) shaft and high pressure (HP) shaft. The LP shaft is powered by the LP turbine and drives the fan (engine inlet) and LP compressor. The HP shaft is powered by the HP turbine (following the combustion chamber) and drives the HP compressor. See Figure 1 for a simplified diagram. Sensors are disposed to measure the rotation speed of each shaft (also called regime) and vibration amplitude. Vibration amplitude can be expressed in three different ways: displacement (unit: mm SI or $mils$), velocity (unit: mm/s SI or ips) or acceleration (unit: m/s^2 SI or g). In order to measure it on a machine, two possibilities exist. First, directly measuring displacement of moving parts, using eddy current (also known as Foucault's current) proximity sensors. This solutions is used for testing (e.g. tip-timing), but is unpractical in operating engines. Instead, the second solution is to measure the acceleration of non-moving parts (e.g. bearing or casing) using accelerometers (which are much easier to install on smaller parts), and integrate to obtain speed or displacement. In the following section, we will describe the acquisition process and properties of the sensor data that will be used in this work. As part of aircraft engine health monitoring (EHM) (Bastard et al., 2016), vibration analysis tackles following issues: rotor unbalance (fan, compressors or turbines), rotor/stator contact (Peng, Chu, & Tse, 2005), or defects due to wear affecting blades (Kharyton, 2009; Hazan, Verleysen, Cottrell, & Lacaille, 2010), bearings (Orsagh, Sheldon, & Klenke, 2003) or gears (Wang, Ismail, & Farid Golnaraghi, 2001).

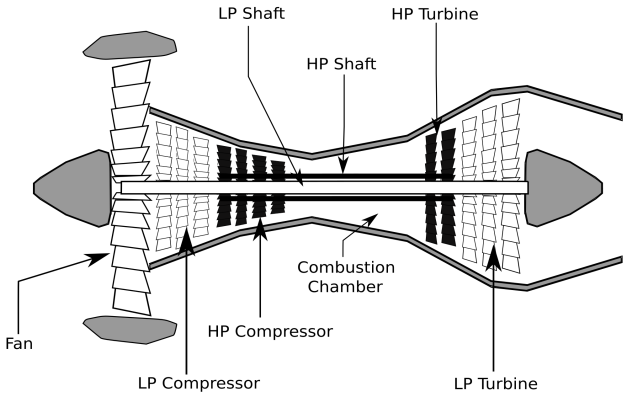

Figure 1. Simplified diagram of a turbofan engine with fan, low-pressure and high-pressure compressors and turbines attached to their respective shafts.

Frequency analysis Vibrations signals are usually processed not in the time-domain, but in the frequency or time-frequency domain. When signals are stationary, i.e. when the engine rotation speed is constant, the Fourier transform is traditionally used to analyze the spectrum (Randall, 2011). When rotation speed is varying, during an acceleration or deceleration, analysis takes place in the time-frequency domain and makes use of spectrograms. The works presented in (Hazan et al., 2010; Lacaille, 2013; Abdel-Sayed, Duclos, Faÿ, Lacaille, & Mougeot, 2015) tackle the problem of pattern recognition in high-frequency, high-bandwidth vibration data measured on aircraft engines on a test bench, as part of the production process. These data contain the complete spectral information on the engine and allow to prevent faults in new engines coming out of the production plant. Due to the high frequency of the measurements (51 kHz), the vibration data are represented as spectrograms. Traditionally,

experts perform a visual analysis of the spectrograms to detect anomalous patterns. The goal is to automate this process using algorithms and numerical methods. In (Lacaille, 2013), spectrogram patches are queried against a database of reference patterns, using dimensionality reduction through non-negative matrix factorization (NMF). (Abdel-Sayed et al., 2015) propose an automatic anomaly detection procedure also based on NMF. This line of work differs vastly from our contribution, firstly because we are interested in a fleet of operating engines, and not a test bench. The nature of our data is also different, as we have medium-frequency time-domain signals, already aggregated by the flight recorder, but measured during entire flights. Moreover, we are not interested in early detection of faults in young engines just coming out of the plant, but in the evolution of vibration signatures of operating engines, flight after flight.

Time-domain analysis In this work, we are not interested in the frequency information contained in the spectrum of the signal, but we will directly manipulate vibration amplitude signals already aggregated by the electronic flight recorder into medium-frequency time-domain signals. Amplitude is measured either by displacement, velocity or acceleration. Instead, we are interested in the vibratory response of specific parts of the engine as a function of regime, called a vibration signature (Randall, 2004). In rotor dynamics, a vibration signature can describe intrinsic properties of parts. It is generally measured during an acceleration (monotonic increase of the regime) or a deceleration of the engine (monotonic decrease of the regime). Vibration signatures can then be represented as Campbell diagrams as a function of time or equivalently as a function of regime.

2.2. Unsupervised learning for engine data analysis

As more and more data are collected on modern aircraft, data-driven approaches and machine learning have become useful tools for condition monitoring. Supervised learning allows to build predictive models when target values or labels are available. Unsupervised learning, on the other hand, can be used for data exploration, anomaly detection, monitoring, etc. We have seen previously that dimensionality reduction allows to compress and extract information from high-dimensional data (Abdel-Sayed et al., 2015). Another major tool is clustering, also known as unsupervised classification. Clustering is a family of unsupervised learning techniques that try to discover groups of similar elements in a dataset, providing information on the structure of the underlying data distribution. The approach used in (Hazan et al., 2010) uses clustering to detect signatures of orders in spectrograms. In this work, we focus on a family of clustering algorithms called self-organizing maps (SOM). SOM algorithms enforce neighborhood constraints on the cluster centers and have the advantage of producing smooth, interpretable visualizations. High-dimensional data are clustered and projected onto a low-dimensional manifold (usually two-dimensional) with a grid topology, called a map. Each unit of the map corresponds to a prototype vector in the original high-dimensional space, and new data points are projected on the map by finding the closest prototype vector w.r.t. euclidean distance. Originally introduced by Kohonen (Kohonen, 1982), there are many variants of SOM working with relational data defined by a distance matrix (Olteanu, Villa-Vialaneix, & Cottrell, 2013) or using unsupervised neural networks for joint representation learning (Forest, Lebbah, Azzag, & Lacaille, 2019b; Fortuin, Hüser, Locatello, Strathmann, & Rätsch, 2019).

Self-organizing maps were already used for aircraft engine fleet monitoring in (Cottrell et al., 2009; Côme, Cottrell, Verleysen, & Lacaille, 2010, 2011; Forest et al., 2018). These works focus on the performance health state of the engine and not the vibration aspects. In (Faure, Olteanu, Bardet, & Lacaille, 2017), SOM are used to classify transient flight phases.

3. DATA DESCRIPTION

This section describes the acquisition, ingestion and storage process of the sensor data used in this work. Continuous Engine Operational Data (CEOD) designates the data recorded by modern civil aircraft. It consists in a set of parameters and sensors that are recorded during entire flights, from engine start to landing. Due to the large number of parameters and their possibly high frequency (up to 66 Hz), the volume of such data is very important. Finally, the studied vibration signatures are presented.

3.1. Sensors and acquisition

Two types of signals are necessary to compute vibration signatures: rotation speed (or regime), and vibration amplitude. On the regime side, two variables are considered:

- N1: rotation speed of the LP shaft.
- N2: rotation speed of the HP shaft.

Rotations speeds are recorded by two phonic wheels at an initial frequency of 51 kHz, before being down-sampled onboard to 66 Hz. On the vibration side, vibration peak amplitudes (displacement, speed or acceleration) are measured by two accelerometers. One of the sensors (ACC1) is located near #1 bearing, placed on the static frame as close as possible to the LP shaft, whereas the second (ACC2) is located at the turbine rear frame. Vibration is also sampled at 51 kHz and then aggregated to a lower frequency of 4 Hz. Through filtering, we obtain vibrations corresponding to N1 and N2 regimes, producing a total of four vibration variables:

- LP-ACC1 and LP-ACC2: vibration amplitude at N1 speed (in terms of displacement in $milsda$).
- HP-ACC1 and HP-ACC2: vibration amplitude at N2 speed (in terms of speed in $ipspk$).

A cross-section schema of the engine with sensor positions is displayed in Figure 2. The signals are measured during en-

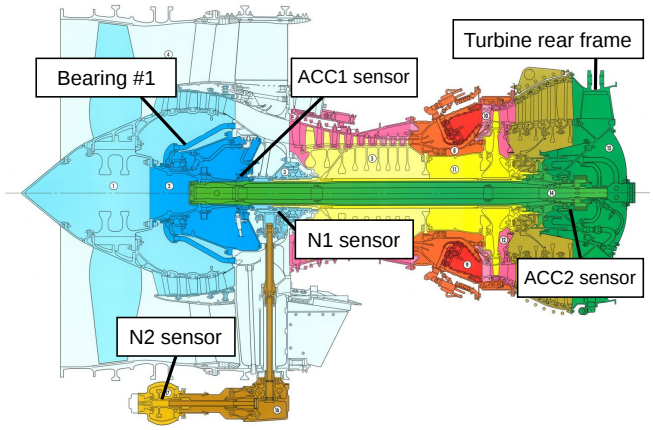

Figure 2. Engine cross-section with positions of the rotation (N1, N2) and vibration (ACC1, ACC2) sensors.

tire flights, from engine start to engine stop. Figure 3 shows an example of N1, LP-ACC1 and LP-ACC2 signals. The N1 rotation speed is directly controlled by the pilot pushing on the thrust lever, and corresponds to the engine thrust. It is expressed as a percentage of maximum thrust (this maximum depends on many factors and flight conditions). During a normal passenger flight, the N1 signal can be broadly divided into different phases: first, a strong acceleration during take-off and ascent, then a long stabilized phase during cruise, and finally a decrease during descent, with short peaks corresponding to maneuvers before landing. The LP-ACC1 signal follows N1 during the first part of the flight, increasing during acceleration, with a small mode at around 90% regime. However, the strongest vibrations are observed during deceleration, with several peaks showing an important mode at around 40% regime. The LP-ACC2 signal is interesting because it exhibits a very strong mode at low regimes. Signals N2, HP-ACC1 and HP-ACC2 for the same flight are displayed on Figure 4. The behavior of N2 is similar to N1, with an acceleration until it reaches a plateau just over 100% regime, where vibration is higher, before entering the stabilized cruise regime. Both signals contain a very sharp and high peak at low regime.

3.2. Data ingestion process

First, airline operators manually download the raw data from the aircraft flight recorder (in future, this data may be streamed in real-time). Depending on the time since last download, raw CEOD may contain the concatenated recordings of several flights, as the flight recorder writes into memory in a sequential manner. Then, raw data is decoded into a structured file format using a proprietary software. Finally, files are cut into distinct flights, by detecting flight start and end based on sensor values. CEOD are then in-

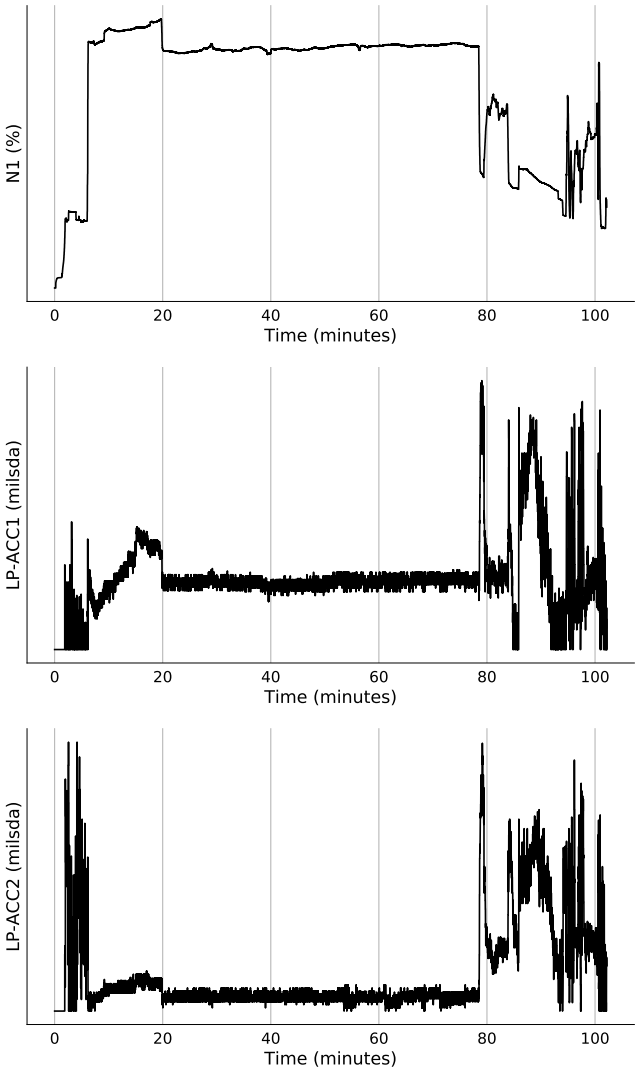

Figure 3. Example of rotation speed N1 and vibration amplitude signals LP-ACC1 and LP-ACC2 during a flight.

gested into a Hadoop cluster and stored on the Hadoop Distributed File System (HDFS), using the Hive data warehouse (Apache Hive, 2010), in order to benefit from scalability and fault-tolerance (every chunk of data is replicated several times across the cluster nodes). The properties of the data analyzed in this work are described in Table 1.

3.3. Vibration signatures

In order to represent the vibratory response of the engine, the raw time series are transformed into signatures that represent vibration as a function of regime. Thus, a signature can directly relate a given regime to a vibratory mode. The location and intensity of these modes are crucial to understand what happens inside the engine. Here are the four signatures studied in this work:

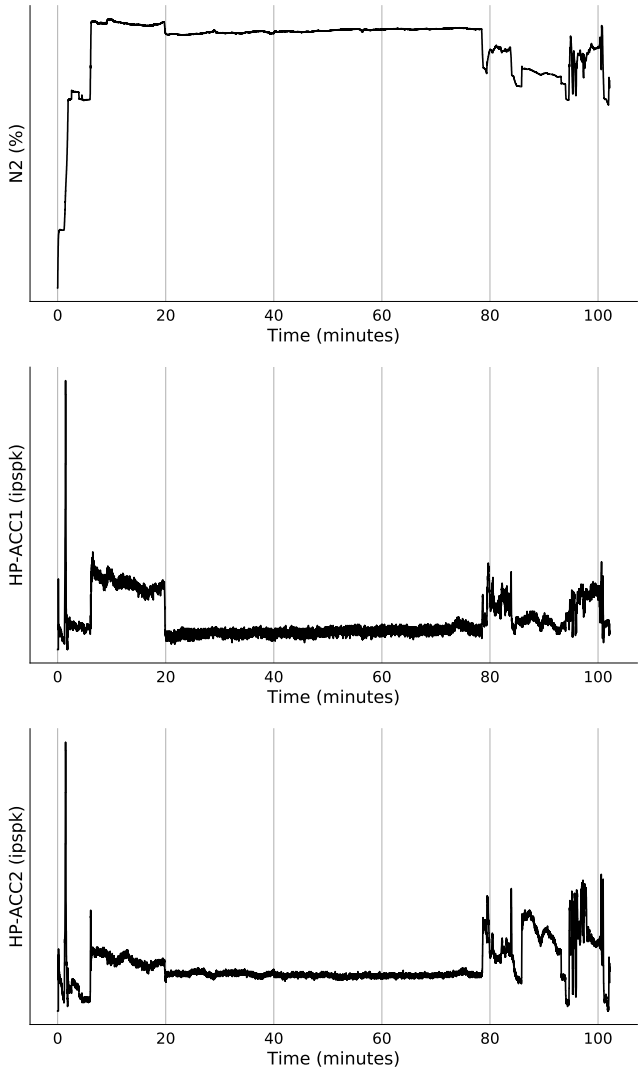

Figure 4. Example of rotation speed N2 and vibration amplitude signals HP-ACC1 and HP-ACC2 during a flight.

1. LP-ACC1 vs N1.
2. LP-ACC2 vs N1.
3. HP-ACC1 vs N2.
4. HP-ACC2 vs N2.

By observing these signatures, experts are able, for example, to detect unbalance at a specific location of the engine. As we define signatures in terms of entire flights, we are not in the standard setting of a monotonic acceleration or deceleration. Thus, a given regime is reached several times during a flight, and may correspond to different vibration amplitudes, producing a point cloud, as shown for signature 4 on Figure 5. To extract the modes and general shape, we cut the x-axis into bins of 5% regime, and aggregate values by taking the quantile at 75% (not the maximum because it is sensitive to outliers). It is clear that manual monitoring of these signatures is impossible, because of their variability and the huge number

Table 1. Data properties.

Property	Approximate value
Number of engines	1000
Number of flights	1 million
Number of parameters	6
Frequency of parameters	4 Hz or 66 Hz
Total HDFS storage volume	1 TB

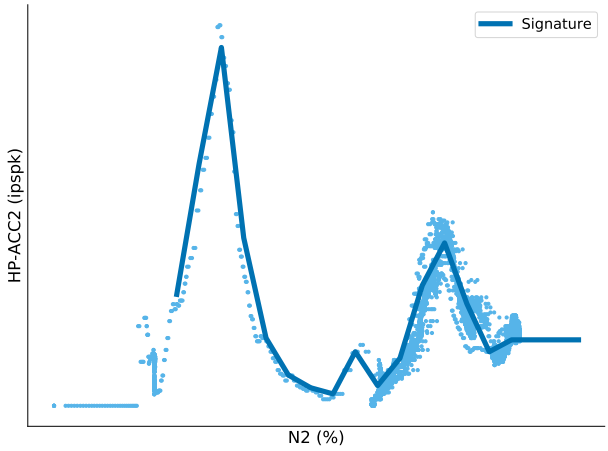

Figure 5. Signature 4 (HP-ACC2 vs N2) on an example flight. Each point is a measurement during the flight (after re-sampling).

of flights. The next sections present how we process data and use self-organized clustering models for efficient fleet CM.

4. GENERIC BIG DATA PROCESSING PIPELINE

For the large-scale computation of vibration signatures on a fleet of civil aircraft engines, we use the generic big data processing pipeline introduced in (Forest et al., 2018). This pipeline has been designed to analyze operational flight data on a Hadoop cluster and is based on the Apache Spark distributed computing engine (Apache Spark, 2014). It allows to deploy custom functions containing the engineers' business logic without knowledge of distributed programming, and is composed of several modules presented in the next paragraph.

4.1. Description

The first step in the pipeline is basic preprocessing and selection queries against the flight database. The second step is generic feature extraction, using predefined or user-provided functions to compute flight features. This is where domain knowledge is incorporated. Signature computation happens at this step. Finally, learning algorithms are trained to obtain models, and results are visualized on a visualization application to extract insights. Each step is configured using a flexible configuration file with JSON syntax. All the processing is based on the Spark framework.

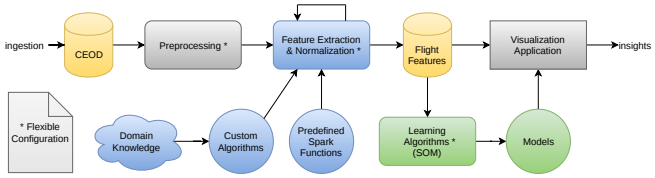

Figure 6. Big data processing pipeline.

4.2. Vibration signature computation

Signatures are computed by a custom function taking the parameters of a single flight as input, and outputs the resulting signature. The code is written in Python with standard numerical and data analysis libraries. This function has five parameters: the names of both input signals (x- and y-axis), the period and range of the x-axis (e.g. the regime range), and the type of operation used to aggregate points in each bin of the x-axis (e.g. average, max, quantiles, etc.). Parameters are set in the configuration file, allowing to extract various signatures using the same generic code. The feature extraction module applies this function on flights in parallel across the cluster, as illustrated in Figure 7.

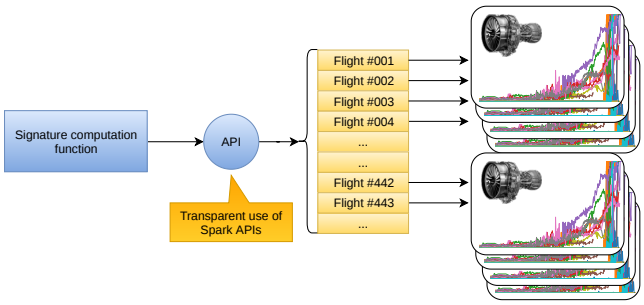

Figure 7. Data-parallel signature extraction on a collection of flights.

In this work, we use a period of 5% regime bins in the [25%, 100%] range, thus each signature can be viewed as a 15-dimensional vector, or a one-dimensional curve of length 15.

5. CLUSTERING AND VISUALIZATION WITH SELF-ORGANIZED MODELS

5.1. Self-Organizing Maps

Self-Organizing Map (SOM) (Kohonen, 1982) are clustering models enforcing a topological relationship between clusters. For a background on SOM algorithms, please refer to the appendix. For now, let us consider that the algorithm takes as input the set of vibration signatures $\mathbb{S} = \{\mathbf{s}_i\}_{1 \leq i \leq N}, \mathbf{s}_i \in \mathbb{R}^D$, with $D = 15$, and outputs a square map composed of $K = 8 \times 8$ units. Each unit is associated to a prototype signature $\{\mathbf{m}_k\}_{1 \leq k \leq K} \in \mathbb{R}^D$. A new flight is projected onto the map by finding its closest prototype signature w.r.t. euclidean distance. We call the corresponding map unit *best-matching signature* (BMS):

$$\text{BMS} := \underset{k}{\text{argmin}} ||\mathbf{s}_i - \mathbf{m}_k||_2^2$$

Before feeding into SOM, the data set is z-normalized to zero mean and unit variance to give each point of the signature an equal weight in euclidean distances. The resulting map for signature 4 (HP-ACC2 vs N2) is displayed Figure 8 and will be investigated further in the next paragraph. We use a distributed, data-parallel implementation of batch SOM using the map-reduce paradigm and the Apache Spark framework (Apache Spark, 2014). This allows to leverage the production cluster to train SOM models on very large datasets of several million flights. The code is open-source and available online at `https://github.com/FlorentF9/sparkml-som`.

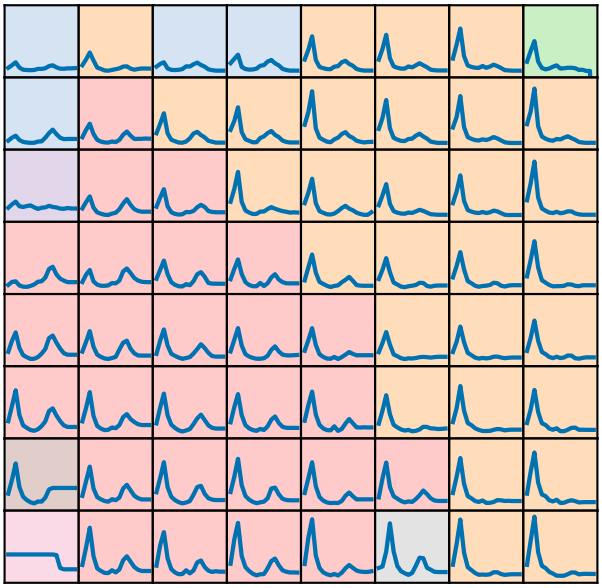

Figure 8. SOM map of signature 4 (HP-ACC2 vs N2). Each cell represents a vibration signature prototype. The background colors are higher-level profiles obtained by Ward hierarchical clustering (here with 8 clusters).

5.2. Analysis and Methodology

One year of historical flight data for 1000 engines, representing approximately 1 million flights and 1 TB of raw signal data, has been processed. After training a SOM for each signature, the resulting models are saved.

Vibration signatures exhibit several modes at particular regimes, visible on the visualizations provided in Figure 8 and appendix Figures 12, 13 and 14. The variability in locations and amplitudes of the modes translates into smooth transitions on the map. Experts in engine dynamics are able to identify these modes and link them, for example, to un-

balance at a specific part of the engine. Moreover, certain vibratory behaviors are normal, such as vibrations of the whole aircraft structure, or temporary unbalance due to thermal conditions. On the other side, certain behaviors are due to wear and must be monitored closely.

In order to classify map cells into higher-level vibration profiles, we perform hierarchical clustering (HC) on the prototype signatures. This classification is materialized by the cell's background color. These profiles may correspond to very well-balanced engines (see the very low-amplitude signatures on Figure 8 and appendix), rotor unbalance at fan, compressor or turbine, but also issues not related to the engine at all (e.g. flat signals, such as the bottom-left cell on Figure 8, are due to a sensor switched off or some issue during data decoding or ingestion). As a result from this analysis, each map region has been interpreted and labeled by experts.

For EHM of operating aircraft, new flights are projected onto their best-matching signature (BMS). The (euclidean) distance between a flight and its BMS is a proxy for an anomaly score: a large distance means that a flight is dissimilar to previously observed behaviors, thus it needs to be investigated carefully. Flights that are projected onto abnormal regions of the map raise alerts and can be immediately investigated by engineers. The sequence of projections of a single engine, flight after flight, is called a *trajectory*. Because a signature describes intrinsic properties of an engine, it should not change drastically from one flight to another. Thus, a trajectory should stay within the same region or higher-level profile. A sudden jump, or a progressive trend towards a different region, can be a warning for abnormal wear. However, changes in vibration profiles may also be due to maintenance operations or a folding of the SOM map (Kiviluoto, 1996). An engine trajectory is represented on Figure 9. The BMS of a flight is marked by a black circle, where the radius is proportional to the number of flights where the engine stayed on the same cell. The lower part of Figure 9 represents the sequence of higher-level profiles found by HC. Clearly, the vast majority of flights have similar vibration profiles. Out of 684 flights, only 16 fall outside the orange area, and most transitions occur within the higher-level profiles. For sake of readability, only transitions between non-neighboring cells were represented by arrows on the map. The fact that a signature is an individual property of engines is supported by a heatmap visualization of BMS counts, i.e. by representing the number of flights projected on each cell for different engines (Figure 10). Finally, in a post-finding situation, after an event has occurred, we can find *similar* engines that share the same vibration patterns or have similar temporal evolutions (e.g. with an edit distance on trajectories (Côme et al., 2011)). However, a map is only a snapshot of past flights. Periodically, models must be re-trained with up-to-date flight data, to account for new trends and the aging of the fleet. The complete methodology, summarized visually on Figure 11,

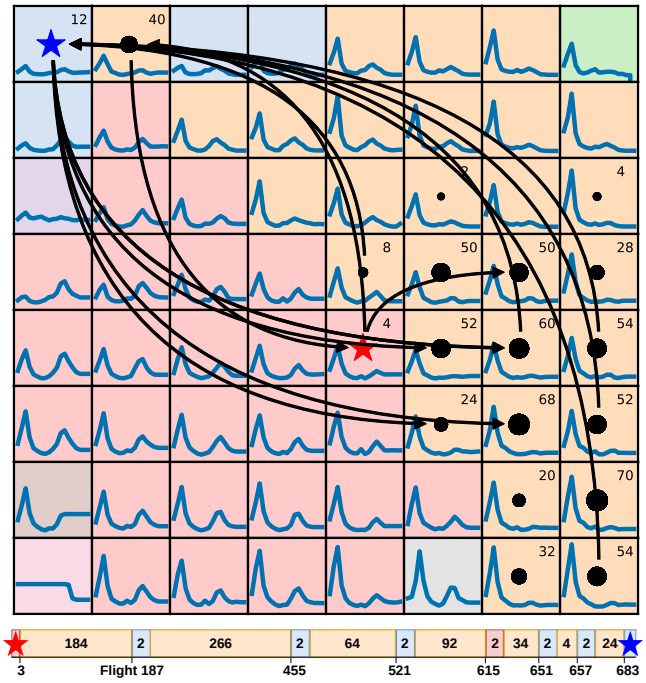

Figure 9. Trajectory of a single engine on the SOM of signature 4, for a total of 684 flights. Circles correspond to flight projections. The first and last flights are marked by red and blue stars. The radius of a circle is proportional to the number of flights the engine stayed on the same vibration profile (this number is also printed within each cell). Sudden jumps between non-neighboring cells (marked by arrows) indicate abrupt changes in vibration profiles, which may correspond to operational events or a SOM folding. This engine mostly stays within the orange region, as shown by the sequence of transitions between higher-level profiles (bottom diagram).

can be analyzed under the OSA-CBM framework for EHM (described in (Bastard et al., 2016)): (1) signature computation corresponds to Data Acquisition & Manipulation; (2) State Detection assigns flights to vibration profiles as well as distances to the map; (3) Health Assessment consists in the classification of the profiles and anomaly detection; (4) analysis, prediction and search of similar engine trajectories is part of Prognostics Assessment and finally (5) Advisory Generation encompasses visualization and alerts generation.

6. CONCLUSION

This work presents a methodology for large-scale vibration monitoring on thousands of operating civil aircraft engines, based on unsupervised learning algorithms and a database of flight recorder data. These signatures are classified and visualized using interpretable clustering models called self-organized maps, yielding a cartography of vibration profiles. As part of a CM strategy, these profiles are useful for experts who can quickly gain insights about the vibratory state of a fleet, and detect unbalance or other abnormal behaviors. Interpretable clustering and visualization for decision-making

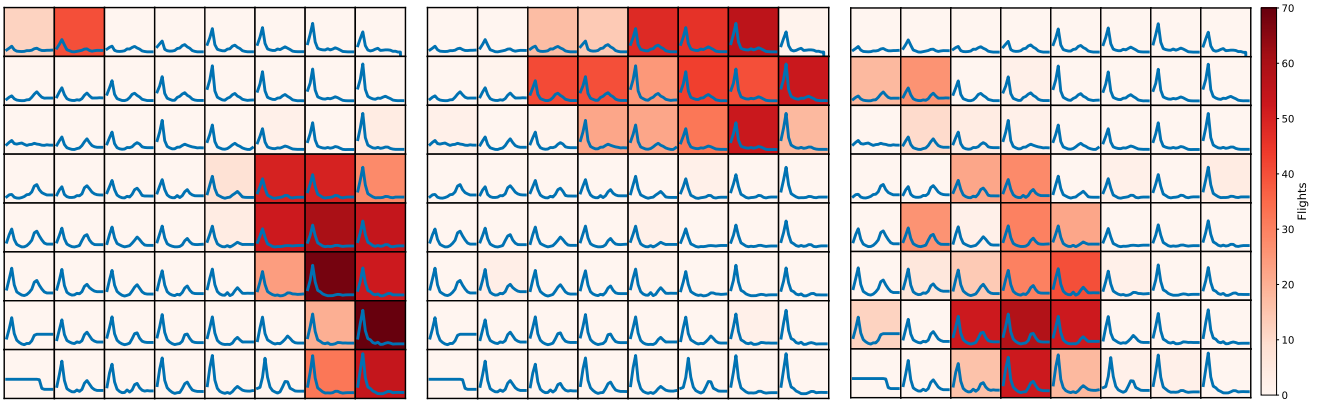

Figure 10. Heatmaps of projection counts on SOM map of signature 4, for three different engines. Each individual engine has its vibration signatures concentrated in a single region, because a signature is an intrinsic property of engines.

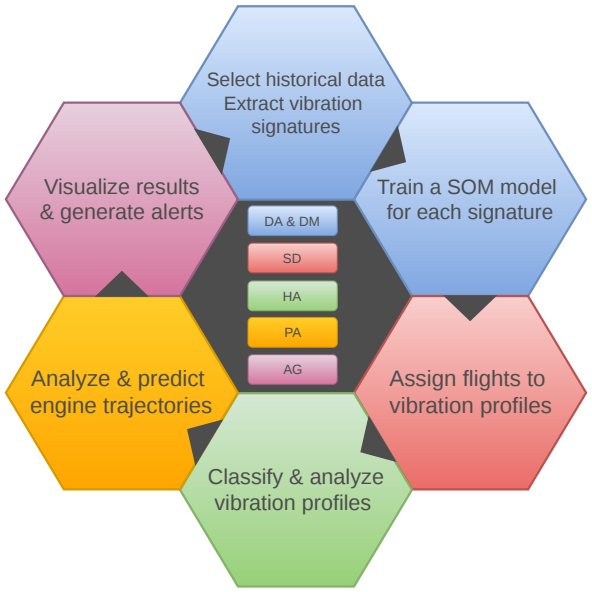

Figure 11. Vibration monitoring methodology. Colors represent the steps of the OSA-CBM standard: Data Acquisition & Manipulation (DA), State Detection (SD), Health Assessment (HA), Prognostics Assessment (PA), and Advisory Generation (AG).

is crucial in aerospace industry but also in other fields such as healthcare. In healthcare, probabilistic models have been used to make temporal predictions on the state of a patient using *disease trajectories* with Gaussian processes (Schulam & Arora, 2016), sometimes in combination with recurrent neural networks (Lim & van der Schaar, 2018). (Fortuin et al., 2019) apply their SOM-VAE model to time series from the intensive care unit. These ideas are considered for our use case in future work, in order to model and predict the future trajectory of an engine. In particular, the remaining number of flights before reaching a *risky* state (for example a state where an event has occurred in the past) could be estimated, a kind of remaining useful life (RUL). Other perspectives include analyzing vibration signatures from the radial drive shaft (RDS), not tackled in this work but whose behavior is of great interest. The RDS is linked to the HP shaft and provides power to the accessory gearbox. Instead of computing a one-dimensional curve from the point cloud shown in Figure 5, we could extract multidimensional vectors from the distribution (in particular, the standard deviation or envelope). This might require to use deeper SOM architectures (Forest et al., 2019b; Forest, Lebbah, Azzag, & Lacaille, 2019a) for dimensionality reduction. Finally, a single family of engines was considered here, but we plan to extend it to other families.

ACKNOWLEDGEMENTS

This work was supported by the French agency for research and technology (ANRT) through the CIFRE grant 2017/1279 and Safran Aircraft Engines (Safran group).

NOMENCLATURE

CEOD	Continuous Engine Operational Data
CM	Condition Monitoring
EHM	Engine Health Monitoring
HC	Hierarchical Clustering
HDFS	Hadoop Distributed FileSystem
HP/LP	High Pressure/Low Pressure
HP-ACC1/2	HP vibration (accelerometer 1/2) [$ipspk$]
LP-ACC1/2	LP vibration (accelerometer 1/2) [$milsda$]
N1	Rotation speed of LP shaft [%]
N2	Rotation speed of HP shaft [%]
OSA-CBM	Open Systems Architecture for Condition-based Maintenance
SOM	Self-Organizing Map
mil	milli-inch, length equal to $0.0254mm$ SI
$milsda$	milli-inches double amplitude
$ipspk$	inches per second peak, speed unit equal to $25.4\ mm/s$ SI
g	standard gravity, acceleration unit equal to $9.81\ m/s^2$ SI

REFERENCES

Abdel-Sayed, M., Duclos, D., Faÿ, G., Lacaille, J., & Mougeot, M. (2015). NMF-based decomposition for anomaly detection applied to vibration analysis. In *International conference on condition monitoring and machinery failure prevention technologies* (pp. 73–81). doi: 10.1784/204764216819708104

Apache Hive. (2010). *Hive Project.* Retrieved 2020-04-01, from `http://hive.apache.org/`

Apache Spark. (2014). *Spark Project.* Retrieved 2020-04-01, from `https://spark.apache.org/`

Bastard, G., Lacaille, J., Coupard, J., & Stouky, Y. (2016). Engine Health Management in Safran Aircraft Engines. In *Phm society.*

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010). Aircraft engine health monitoring using Self-Organizing Maps. In *Industrial conference on data mining.*

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2011). Aircraft engine fleet monitoring using Self-Organizing Maps and Edit Distance. *WSOM*, 298–307.

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., & Verleysen, M. (2009). Fault prediction in aircraft engines using Self-Organizing Maps. In *Wsom.*

Faure, C., Olteanu, M., Bardet, J.-M., & Lacaille, J. (2017). Using self-organizing maps for clustering and labelling aircraft engine data phases. In *Wsom.* doi: 10.1109/WSOM.2017.8020013

Forest, F., Lacaille, J., Lebbah, M., & Azzag, H. (2018). A Generic and Scalable Pipeline for Large-Scale Analytics of Continuous Aircraft Engine Data. In *Ieee international conference on big data* (pp. 1–7). doi: 10.1109/BigData.2018.8622297

Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2019a). Deep Architectures for Joint Clustering and Visualization with Self-Organizing Maps. In *Workshop on learning data representations for clustering (ldrc), pakdd* (pp. 1–12).

Forest, F., Lebbah, M., Azzag, H., & Lacaille, J. (2019b). Deep Embedded SOM: Joint Representation Learning and Self-Organization. In *European symposium on artificial neural networks, computational intelligence and machine learning (esann)* (pp. 1–6).

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., & Rätsch, G. (2019). SOM-VAE: Interpretable Discrete Representation Learning on Time Series. In *Iclr* (pp. 1–18).

Hazan, A., Verleysen, M., Cottrell, M., & Lacaille, J. (2010). Trajectory Clustering for Vibration Detection in Aircraft Engines. In *Industrial conference on data mining.*

Kharyton, V. (2009). *Faults Detection In Blades Of An Aviation Engine In Operation* (Unpublished doctoral dissertation). École Centrale Lyon.

Kiviluoto, K. (1996). Topology preservation in self-organizing maps. *IEEE International Conference on Neural Networks - Conference Proceedings*, *1*, 294–299. doi: 10.1016/b978-044450270-4/50022-x

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69. doi: 10.1007/BF00337288

Lacaille, J. (2013). Searching similar vibration patterns on turbofan engines. In *International conference on condition monitoring and machinery failure prevention technologies* (pp. 338–349).

Lim, B., & van der Schaar, M. (2018). *Disease-Atlas: Navigating Disease Trajectories with Deep Learning.* Retrieved from `http://arxiv.org/abs/1803.10254` doi: arXiv:1803.10254v3

Olteanu, M., Villa-Vialaneix, N., & Cottrell, M. (2013). Online relational SOM for dissimilarity data. *Advances in Intelligent Systems and Computing*, *198*, 13–22. doi: 10.1007/978-3-642-35230-0_2

Orsagh, R. F., Sheldon, J., & Klenke, C. J. (2003). Prognostics/diagnostics for Gas Turbine Engine Bearings. In *Asme turbo expo* (pp. 1–9).

Peng, Z. K., Chu, F. L., & Tse, P. W. (2005). Detection of the rubbing-caused impacts for rotor-stator fault diagnosis using reassigned scalogram. *Mechanical Systems and Signal Processing*, *19*(2), 391–409. doi: 10.1016/j.ymssp.2003.09.007

Randall, R. B. (2004). State of the art in monitoring rotating machinery - Part 1. *Sound and Vibration*(March), 14–20.

Randall, R. B. (2011). *Vibration-based condition monitoring*. Wiley.

Schulam, P., & Arora, R. (2016). Disease Trajectory Maps. In *Nips.* Retrieved from `http://arxiv.org/abs/1606.09184`

Wang, W. Q., Ismail, F., & Farid Golnaraghi, M. (2001). Assessment of gear damage monitoring techniques using vibration measurements. *Mechanical Systems and Signal Processing*, *15*(5), 905–922. doi: 10.1006/mssp.2001.1392

BIOGRAPHIES

Florent Forest is currently a PhD student in computer science at Université Sorbonne Paris Nord (Paris, France), in the machine learning team of the LIPN lab. Through an industry research contract, he also works on industrial applications at Safran Aircraft Engines since 2018, in the Datalab team. In 2017, he graduated from ISAE Supaero engineering school (aerospace institute in Toulouse, France), with a specialization in data science and aerospace engineering. His main research interests are unsupervised machine learning, clustering, big data analysis and applications in aerospace industry.

Quentin Cochard, Cecile Noyer and **Adrien Cabut** are engineers in vibration dynamics at Safran Aircraft Engines.

Marc Joncour is a data scientist in the Datalab team at Safran Aircraft Engines.

Jérôme Lacaille is an emeritus expert in algorithms for the Safran international aeronautics group. He joined the Safran Aircraft Engines company in 2007 with responsibility for developing a health monitoring solution for jet engines. Jérôme has a PhD from the Ecole Normale Supérieure (France) in Mathematics. He has held several positions including scientific consultant and professor. He has also cofounded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier, France) and PDF Solutions (San Jose, CA). He developed specific mathematic algorithms that were integrated in industrial processes. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling as well as some industrial patented applications.

Mustapha Lebbah is currently associate professor at Université Sorbonne Paris Nord and member of machine learning team A3, LIPN. His main researches are centered on machine learning (unsupervised learning, mixture models, cluster analysis, scalable machine learning big data and data science). Graduated from USTO University where he received his engineer diploma in 1998. Thereafter, he gained an MSC (DEA) in Artificial Intelligence from the Université Sorbonne Paris Nord in 1999. In 2003, after three years at Renault R&D, he received his PhD degree in Computer Science from the University of Versailles. He received the "Habilitation Diriger des Recherches" in Computer Science from USPN in 2012.

Hanane Azzag is currently associate professor at Université Sorbonne Paris Nord and a member of machine learning team A3 at LIPN computer science lab. Her main research topics are biomimetic algorithms, machine learning and visual data mining. Graduated from USTHB University where she received her engineering degree in 2001. Thereafter, in 2002 she gained an MSC (DEA) in Artificial Intelligence from Tours University. In 2005, after three years at a lab in Tours, she received her PhD degree in Computer Science from the University of Tours.

APPENDIX

Background on SOM

The Self-Organizing Map (SOM) (Kohonen, 1982) is a clustering model that introduces a topological relationship between clusters. It consists in a network of two layers: an input layer, and an output layer of interconnected nodes, often called *neurons* or *units*. Typically, the topology of this layer is chosen as a two-dimensional grid, because it can be easily visualized. This visualization capability characterizes SOM as an interpretable clustering method.

The set of input data samples is denoted $\mathbb{X} = \{\mathbf{x}_i\}_{1\leq i\leq N}, \mathbf{x}_i \in \mathbb{R}^D$. A self-organizing map is composed of K units, associated to the set of prototype vectors $\{\mathbf{m}_k\}_{1\leq k\leq K} \in \mathbb{R}^D$. A data point is projected on the map by finding its closest prototype vector according to euclidean distance. The corresponding map unit is called the *best-matching unit* (BMU). We introduce the notation b_i for the BMU of $\mathbf{x}_i$:

$$b_i = \underset{k}{\operatorname{argmin}} ||\mathbf{x}_i - \mathbf{m}_k||_2^2 \tag{1}$$

The grid topology allows to define an inter-node distance $\delta(k, l)$, which is the topographic distance between units k and l on the map, here the Manhattan distance (the length of the shortest path on the map between the two units). We then define the neighborhood function of the SOM and a temperature parameter T, controlling the radius of the neighborhood around a unit. In this work, we will use a Gaussian neighborhood function, expressed as follows:

$$\mathcal{K}^T(d) = e^{-d^2/T^2}$$

The temperature T is decreased at each training iteration, as in simulated annealing. A common choice is exponential decay, starting from an initial temperature T_{max} towards a final temperature T_{min}, i.e. at iteration i:

$$T(i) = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{i/\text{iterations}}$$

The original SOM learning algorithm, also called *stochastic algorithm* or *Kohonen algorithm*, takes each training sample $\mathbf{x}_i$ and updates every prototype vector by moving them closer to the point $\mathbf{x}_i$. The updates are weighted by the neighborhood around the best-matching unit, so that neighboring units receive a large update and very distant units are not updated at all. This expresses as the following update rule:

$$\mathbf{m}_k \leftarrow \mathbf{m}_k + \alpha\mathcal{K}^T\left(\delta(b_i, k)\right)(\mathbf{x}_i - \mathbf{m}_k) \tag{2}$$

where α is a learning rate that is decreased during training. The stochastic algorithm is detailed in algorithm 1.

Input: training set $\mathbb{X}$; SOM map size; temperatures T_{max}, T_{min}; $iterations$
Output: SOM code vectors $\{\mathbf{m}_k\}$
Initialize SOM parameters $\{\mathbf{m}_k\}$;
for $n = 1, \ldots, iterations$ **do**
  $T \leftarrow T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{n/iterations}$;
  Load next training sample $\mathbf{x}_i$;
  Compute BMU b_i ;
  for $k = 1, \ldots, K$ **do**
    Update prototype $\mathbf{m}_k$ (by equation 2) ;
  end
end

Algorithm 1: Stochastic SOM algorithm.

A disadvantage of this algorithm is that it converges slowly, is sequential and cannot be parallelized. Therefore, another algorithm was introduced: the batch SOM algorithm. It consists in minimizing following cost function, called *distortion*:

$$\mathcal{L}_{\text{SOM}}(\{\mathbf{m}_k\}, \mathbb{X}, b, T) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathcal{K}^T \left(\delta(b_i, k)\right) ||\mathbf{x}_i - \mathbf{m}_k||_2^2$$

Distortion is not directly differentiable because of the BMU assignments b. However, it can be empirically minimized by alternating between two steps:

1. Assignment of best-matching units using equation (1)
2. Minimization of distortion by fixing assignments, using following update rule:

$$\mathbf{m}_k \leftarrow \frac{\sum_{l=1}^{K} \mathcal{K}^T \left(\delta(k, l)\right) \sum_{i=1}^{N} \mathbb{1}_{[b_i=l]} \mathbf{x}_i}{\sum_{l=1}^{K} \mathcal{K}^T \left(\delta(k, l)\right) \sum_{i=1}^{N} \mathbb{1}_{[b_i=l]}} \quad (3)$$

The batch algorithm pseudo-code is detailed in algorithm 2.

Input: training set $\mathbb{X}$; SOM map size; temperatures T_{max}, T_{min}; $iterations$
Output: SOM code vectors $\{\mathbf{m}_k\}$
Initialize SOM parameters $\{\mathbf{m}_k\}$;
for $n = 1, \ldots, iterations$ **do**
  $T \leftarrow T_{max} \left(\frac{T_{min}}{T_{max}}\right)^{n/iterations}$;
  Compute all BMUs $\{b_i\}_{i=1\ldots N}$;
  for $k = 1, \ldots, K$ **do**
    Update prototype $\mathbf{m}_k$ (by equation (3)) ;
  end
end

Algorithm 2: Batch SOM algorithm.

Map visualizations

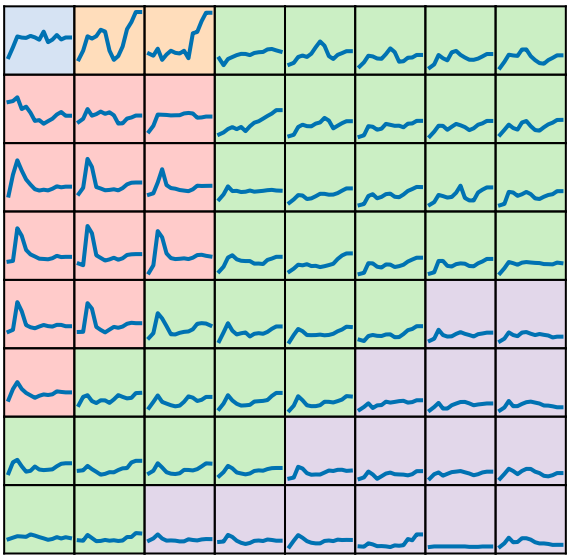

Figure 12. SOM map of signature 1 (LP-ACC1 vs N1). Each cell represents a vibration signature prototype. The background colors are higher-level profiles obtained by Ward hierarchical clustering (here with 5 clusters).

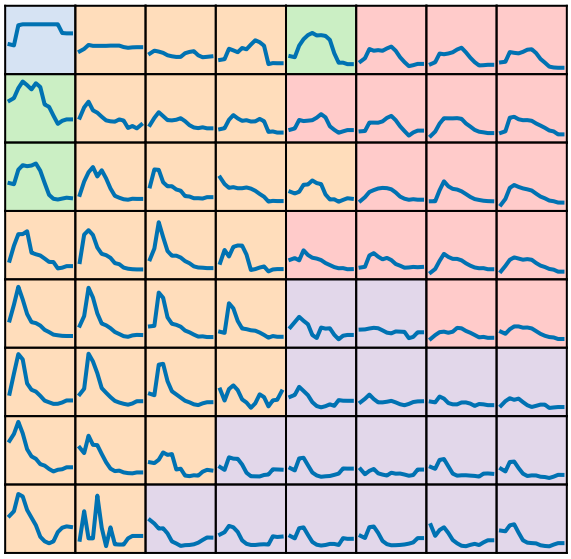

Figure 13. SOM map of signature 2 (LP-ACC2 vs N1). Each cell represents a vibration signature prototype. The background colors are higher-level profiles obtained by Ward hierarchical clustering (here with 5 clusters).

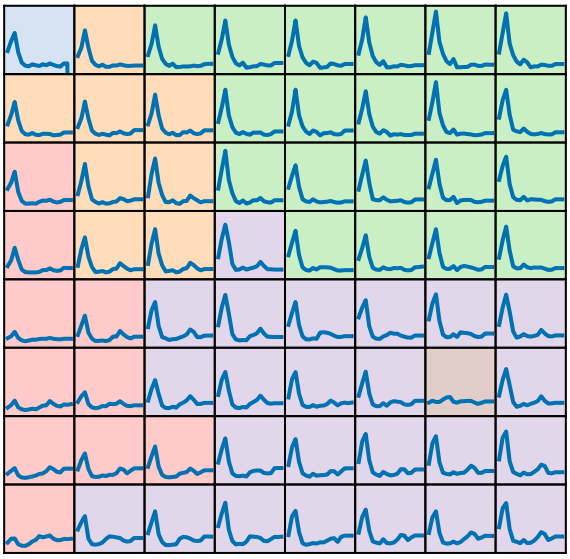

Figure 14. SOM map of signature 3 (HP-ACC1 vs N2). Each cell represents a vibration signature prototype. The background colors are higher-level profiles obtained by Ward hierarchical clustering (here with 6 clusters).