ENHANCING OPERATIONAL FAULT DIAGNOSIS BY ASSESSING MULTIPLE OPERATIONAL MODES

Juan José MONTERO JIMENEZ

TEC-Costa Rica Institute of Technology
juan.montero@itcr.ac.cr
ISAE-SUPAERO – University of Toulouse
juan-jose.montero-jimenez@student.isae-supaero.fr

Rob VINGERHOEDS

ISAE-SUPAERO – University of Toulouse
rob.vingerhoeds@isae-supaero.fr

ABSTRACT: *Operating a modern technical system, such as a train or aircraft, calls for good organised engineering, operation and maintenance to keep the system in an optimal operational condition. Predictive maintenance is being studied and has as aim to identify errors early enough to still be able to propose a suitable solution before a real incident occurs. After all, technical problems in service may lead to delays or even interruptions of service due to extensive repair actions, such as the replacement of components. Often, predictive maintenance aims at recognising patterns in time series of monitored data and classifying these patterns as known conditions (faulty or correct). As such it provides a vital source of information for maintaining a healthy operational status. However, these approaches are still in their early phases and rely still heavily on skill and experience from the expert. In this paper, the use of self-organising maps for predictive maintenance is being discussed, applied to data of a jet engine. The aim of the study was to assess the usability of such approaches to real-life situations, assessing the learning and validation phases.*

KEYWORDS: *Maintenance, condition monitoring, fault diagnosis, neural networks.*

1 INTRODUCTION

Safe and efficient operation of modem technical systems involves several online and off-line diagnostic tasks as well as the incorporation of the diagnostic results in timely maintenance. Off-line diagnostic tasks involve the handling of recorded data, analysis and interpretation, and organised maintenance. The information gathered during operation, such as that from crew reports and data recordings form vital sources of input for such off-line diagnostic tasks.

Maintenance is vital during a systems life cycle to ensure their functionality. Several types of maintenance can be distinguished; corrective, preventive and predictive maintenance. A good combination of the three is vital to make a system's operation reliable. Even though corrective and preventive maintenance remain today important research topics, a lot of attention is currently being given towards predictive maintenance that aims for defining the best possible moment to trigger maintenance actions. Triggering too late may lead to failure occurrence, causing financial losses, sometimes image damage and may even lead to casualties and/or losses of human lives. Triggering too early may lead to replacing components that are not faulty through costly interventions. Predictive maintenance aims at proposing a solution by monitoring the health status of the systems and components, identifying incipient faults and forecasting the exact moment of failure.

In this paper, the use of Self-Organising Maps (SOM) is discussed to detect degradation patterns on aircraft jet engine data as part of a fault diagnosis approach within the framework of predictive maintenance.

The paper is organised as follows. Section 2 describes the diagnostic problem as studied within the framework of this project. Section 3 describes the case study, an aircraft jet engine. Sections 4 and 5 then explain the use of Self-Organising Maps for this case study, first the more regular case as for SOM for a given condition and then the case for bringing together several operational modes into one SOM. Section 6 draws conclusions and indicates some paths for future research on this topic.

2 DIAGNOSTICS FOR PREDICTIVE MAINTENANCE

Diagnostics approaches address the data acquisition, processing and fault identification using explicit methods and implicit methods. Prognostics approaches are oriented to determine the Remaining Useful Life (RUL) of system components using reliability and availability criteria. Both approaches intervene and contribute to the overall idea to better support the decision-making process for operations engineers when recommending or not a maintenance intervention.

Explicit methods group the techniques in which the diagnostic of faults is made "explicitly" from data by experts linking the symptoms to the failures and faults and pointing to the root cause. Here one can think of techniques such as Rule-Based Reasoning (RBR), Case-Based-Reasoning (CBR), decision trees, fuzzy logics, etc. Implicit methods, on the other hand, involve the use of algorithms to transform complex data sets into useful data that can be interpreted afterwards by experts in form of reports or alarms. One can think of Neural Networks (NN) and other learning algorithms.

From a top-level point of view, one can see that predictive maintenance is mainly aimed to trigger maintenance actions at the suitable time, defining an analysis strategy to address complex data, by means of specialized methods and tools in order to identify symptoms related to known failures that will lead to determine the actual state (diagnosis approach) and the remaining useful life (prognosis approach) of the component or system (see Figure 1).

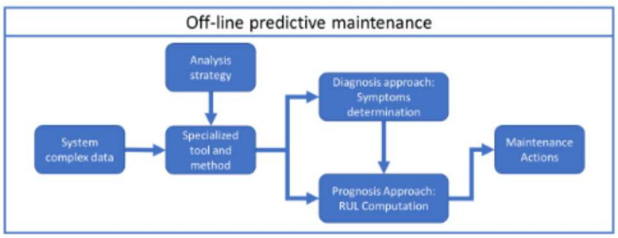

Figure 1: Predictive maintenance model

3 CASE-STUDY ON JET ENGINE

Keeping aircraft engines in an optimal operational condition is a vital aid for operating modern airlines. Engine Condition Monitoring (ECM) is used to have a regular overview of the proper functioning, a good health, of the aircraft engines, by analysing in-flight measured aircraft and engine parameters. Modern ECM is built around performance trend analysis approaches but at the same time also relies heavily on human expertise. The aircraft and engine parameters are visualised and the evolution over time (trends) of the most important parameters is analysed. Specific parameter trend evolutions often precede the occurrence of incidents and allow for early detection of deterioration of engine modules, failures, and/or malfunctions. The application of modern artificial intelligence so to enhance trend analysis to anticipate incidents by determining the need for maintenance and detecting the components or systems where maintenance is required has been part of several studies (e.g. Vingerhoeds, 1995). Trend analysis can offer a good support for performance engineers in airlines engineering departments in detecting problems early. In this case-study, a neural network approach (in this case a self-organising map) will be used to assess the engines health state. (Vingerhoeds, 1995) gives more information on the general conditions for such predictive maintenance approaches, and in particular the operational conditions for diagnosing jet engines.

Performance engineers use a variety of different techniques to assess the engine's health state. In this study, measured in-flight behaviour is used for assessing the engine's health state. It provides early warning on ongoing or imminent problems prior to serious malfunction. Effects of for example bird strikes can be seen and performance engineers can use witnessed changes in the engine's prime parameters to determine the eventual need for corrective maintenance. Other sources of information (flight trouble reports, etc.) offer complementary information that can be used to enhance the overall diagnostic approach.

The complete view for a performance engineer is complex, many parameters come into play and eventual wrong decisions on corrective maintenance (if in the end there was no need) would have a strong financial impact. The overall goal of this work is to improve condition monitoring to a point where performance engineers can rely on these techniques for their decisions.

In this paper, use is made of the database made available for the PHM08 benchmark for the International Conference on Prognostics and Health Management in 2008, as described in (Saxena, 2008). The data was created using a commercial software tool, C-MAPSS, that allows for simulating a realistic large commercial turbofan engine.

The data set consists of multivariable cycle series, divided into training, test and validation subsets with multiple jet-engines of the same type and under comparable nominal operational conditions. Each engine has some unknown initial wear and manufacturing variability that should not be confused with faults. The engines degrade along the operational cycles and upon reaching a specific limit are to be taken out of service. The definition of these limits is not provided and the predictive maintenance approach needs to be able to identify such limits.

The data set contains three input variables representing the engine control settings: the flying altitude, the Mach number and the throttle resolver angle. 6 different combinations of these control settings are used, defining the operational modes in the data sets. The rest of the data correspond to 21 out of 58 different outputs available on the C-MAPSS tool (see Table 1). These 21 variables correspond to sensor information from different engine parts (see a simplified engine diagram in Figure 2).

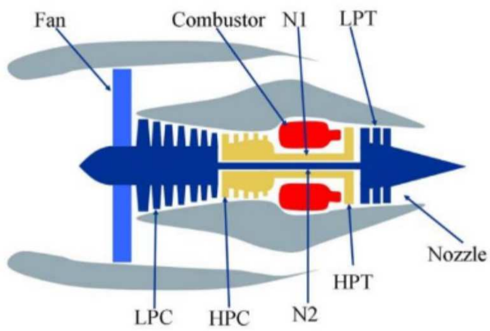

Figure 2: Simplified diagram of engine simulated in C-MAPSS (Saxena 2008)

HPT: High pressure turbine.
LPT: Low pressure turbine.
HPC: High pressure compressor.
LPC: Low pressure compressor.
N1: Outer shaft.
N2: Inner shaft.

Table 1. Sensors list present in the PHM08 Challenge (Saxena 2008).

Number of sensor	Symbol	Description	Units	Column in data set
1	T2	Total temperature at fan inlet	° R	6
2	T24	Total temperature at LPC outlet	° R	7
3	T30	Total temperature at HPC outlet	° R	8
4	T50	Total temperature at LPT outlet	° R	9
5	P2	Pressure at fan inlet	psia	10
6	P15	Total pressure in bypass-duct	psia	11
7	P30	Total Pressure at HPC outlet	psia	12
8	Nf	Physical fan speed	rpm	13
9	Nc	Physical core speed	rpm	14
10	epr	Engine pressure ratio (P50/P2)	**	15
11	Ps30	Static pressure ratio at HPC outlet	psia	16
12	phi	Ratio of fuel flow to Ps30	pps/psi	17
13	NRf	Corrected fan speed	rpm	18
14	Nrc	Corrected core speed	rpm	19
15	BPR	Bypass Ratio	**	20
16	farB	Burner fuel-air ratio	**	21
17	htBleed	Bleed Enthalpy	**	22
18	Nf_dmd	Demanded fan speed	rpm	23
19	PCNfR_dmd	Demanded corrected fan speed	rpm	24
20	W31	HPT coolant bleed	lbm/s	25
21	W32	LPT coolant bleed	lbm/s	26

Analysis on the data showed that only 7 out of the 21 data deliver significant input. Others are either constant or binary, or redundant information with other information. This discrimination process is important for identifying which parameters really come into play for diagnostics. At the same time, as the computation time for the SOM approach is a function of the data dimension, a reduction of the number of output data will allow for a reduction in this computation time.

4 SELF-ORGANISING MAPS FOR DIAGNOSTICS

Self-Organising Maps (SOM) are Neural Networks with unsupervised learning inspired by the way the human brain functions (Kohonen, 1990). The development of a SOM consists of two main phases; the training phase and the labelling phase. Those phases are strongly linked and both must be carefully developed to obtain a reliable tool. The training phase is oriented to teach the algorithm with the dataset in such a way that the neural network clusters similar data features on specific regions on the map. The labelling phase, once the training phase has been finished, links these regions to identified features of the dataset (labelling). This allows the algorithm to identify similar data patterns in complex data. When similar data patterns are presented to the SOM, the right region on the map is identified, hinting, for the current study, a degradation status of the engine.

Usually a two-dimensional, rectangular architecture is used for the SOM's. Other architectures include 2D hexagonal maps, but also 1D and 3D architectures. In this paper a rectangular map is selected for the health monitoring (see Figure 3).

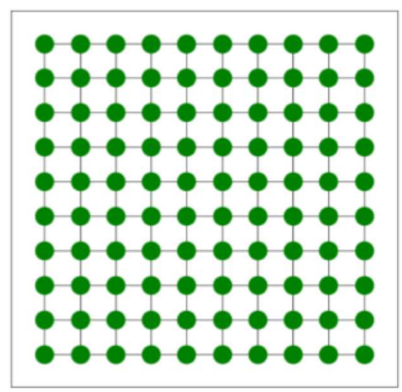

Figure 3: Rectangular SOM architecture.

Taking a closer look at the SOM training phase shows that this phase consists two different main steps; competitive and convergence. During the competitive step, a set of input vectors x_p is presented to the 2D map (p denotes the dimension of the input and output data, for the jet-engines database it is 7). The map is constituted of "n" neuron with coordinates i and j on the map and which have a weight w_{ij}. The weights are vectors with the same dimension as input vectors. At the start of the training process the weights values are set randomly in the intervals [0,1]. A normalisation of the input vectors is performed prior to the training phase so to avoid algorithmic problems. Such normalization allows to bring variable values of very different orders of magnitude to a similar range. This requires expert knowledge on the application domain, on the viable range of variable values so to identify the actual limits.

During this competitive process, the Euclidean distance is computed between one input and each neuron. When the lowest distance among all of them is found, the neuron related to this distance is declared as the winner neuron (see equation (1)), or in more formal words, the Best Matching Unit (BMU). The BMU is surrounded by its neighbourhood constituted by neurons within a certain radius from the BMU.

Now, the Euclidean distance is computed between the coordinates of the BMU and the coordinates of all the rest of the map, if the result is lower than the threshold, the neuron become part of the neighbourhood. The radius determining the neighbourhood starts large at the beginning of the training phase (around the half the size of the map) and will decrease gradually using decay function (2) along the learning process. When the neighbourhood is constituted only by the BMU it-self (equations (3)-(4)), the training phase ends.

$$BMU = \arg\min_{\forall ij}\{\|x_p(t) - w_{ij}(t)\|\} \quad (1)$$

$$decay(t) = exp\frac{-t}{\lambda} \quad (2)$$

with t being the current iteration and λ a time constant.

$$\lambda = \frac{number\ of\ iterations}{\log r_0} \quad (3)$$

with r_0 being the initial neighbourhood radius (half of the map size), then

$$r = r_0 \times decay(t) \quad (4)$$

After finding the BMU, the weights of the winner neuron and the neighbourhood are updated using the following equation:

$$w(t+1) = \begin{cases} w(t) + \alpha(t)\beta(t,r)(x(t) - w(t) & if\ \ n_{ij} \in \Lambda_{BMU}(t) \\ w(t) & if\ \ n_{ij} \notin \Lambda_{BMU}(t) \end{cases} \quad (5)$$

where:

$\alpha(t)$ is the learning rate depending on the iteration.

$\beta(t,r)$ the influence rate depending on the iteration and the radius from the BMU.

n_{ij} a neuron with its i and j coordinates on the map.

$\Lambda_{BMU}(t)$ Neighbourhood of the BMU for the current iteration.

The weights updating is strongly linked with the influence rate $\beta(t,r)$. It defines the magnitude of the change when updating the weights of the BMU and its neighbours, depending on the distance from the winner neuron. The further away the neighbour is, the lesser the influence. There are some influence functions which affect negatively the furthest neighbours, this means, there is an inhibitory effect on them (e.g. the "Mexican hat" function). In this study the well-known Gaussian Function, is the one used.

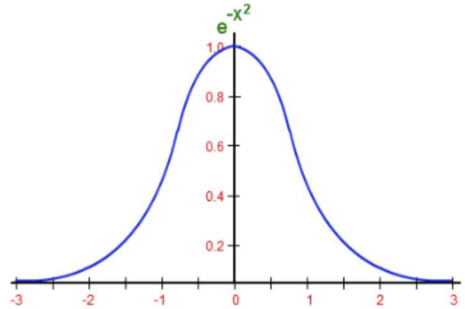

Figure 4: Gaussian function as influence rate.

Once the weights are updated, a second input is shown to the map, the BMU is found and the weight are updated again. This process is repeated with all the inputs and several times (iterations) to ensure a good learning process. In this study, the maximum number of iterations was set to 500 times the number of nodes in the map, described in literature as a reasonable estimate for good convergence (Kohonen, 1995).

Another important factor in the training phase is the learning rate, the speed with which the neural network learns. At the beginning of the learning process, the learning rate is rather large (in this study 0.9) and it will decrease exponentially along the learning process When the learning rate is high, neurons with similar features are rapidly on a map region, this is known as the topological step in the training phase. As the learning rate is decreasing, the convergence step begins when the weights of the neurons are slightly refined. Equation (6) has been used to determine the learning rate along the training process.

$$\alpha(t) = \alpha(t)_0 \times decay\ (t) \quad (6)$$

To summarize the SOM algorithm, the following steps are proposed:

1. Define the map architecture
2. Initialize the neuron weights randomly.
3. Show the first input vector to the SOM.
4. Find the BMU.
5. Define the neighbourhood.
6. Update the weights of the winner neuron and its neighbourhood
7. Repeat steps 3 to 6 with all the inputs vectors.
8. Show all the input vectors several times to the neural network to facilitate the SOM convergence.

If the algorithm converges properly, some defined regions, or trends, will appear on the map. These regions concern areas where comparable inputs are recognised This allows to see what data is considered, by the algorithm, to be similar. It is the task of the human expert to assess these results, so to ensure that the data concerns indeed similar patterns. Knowing now these regions on the map, allows for labelling these regions with clear identification on what the cluster is about. It concerns therefore a manual identification of the different regions / clusters by SOM algorithm. A name, numeric value or any other identification character will be assigned to the map regions or neurons so that new data could be assessed and identified automatically by the trained SOM.

5 APPLICTION OF SOM'S FOR DIAGNOSTICS

SOM have been applied in several approaches for diagnosis and prognosis in predictive maintenance precisely due to its identification abilities in complex data. For example, (Germen et al. 2014) presents a SOM application to determine different types of faults in induction motors using sound based data. (Côme et al. 2010) use self-organising maps to monitor different parameters of aircraft engines for health diagnosis purposes. (Hu et al. 2013) determines the dynamic degradation on bearings using self-organising algorithms from incipient faulty conditions to the failure state.

This paper proposes a different approach of analysis with SOM on the jet-engine health diagnosis, assessing the engine degradation trend on different operational modes scenarios at the same time. The degradation trend is quantified by a degradation index obtained from the trained SOM for each operational mode, allowing to determine the engine health status in a more integrated way, assessing multiple operational modes at the same time to avoid false positives (suggesting a fault where it is still good), or false negatives (suggesting no faults while a fault has occurred). The rationale behind this approach is to be able to compare diagnostics from different operational modes so to build up "evidence" for a suspected fault as time goes along and different control setting are encountered.

The study case is built on the database described in paragraph 3. For each engine cycle in the database, only one operational mode is used to record discrete data. For each operational mode, the data patterns are very different in terms of range so that an unambiguous determination of a degradation trend is not possible over all operational modes at the same time. A first approach therefore has been to develop an SOM for each operational condition, leading to 6 different SOMs, one for each operational mode. This allows to follow the degradation evolution through the lifecycle on each single scenario for the given operational condition.

The results on the 6 SOMs were similar. On each map two main regions were clustered: a white region representing the good operational conditions of the engine and a black region representing the faulty conditions. Besides these regions, a grey scale witnessing the transition over engine lifetime between both regions appeared on the rest of the map, therefore representing the engine degradation in the intermediate states. Figure 5 represent one of these single operational mode SOMs. The red circles within some neurons connected by arrows, represent the lifecycle of one extra engine tested with the trained SOM. It is possible to observe the degradation evolution on the tested engine from the white to the black region, going through the grey neurons along its lifecycle.

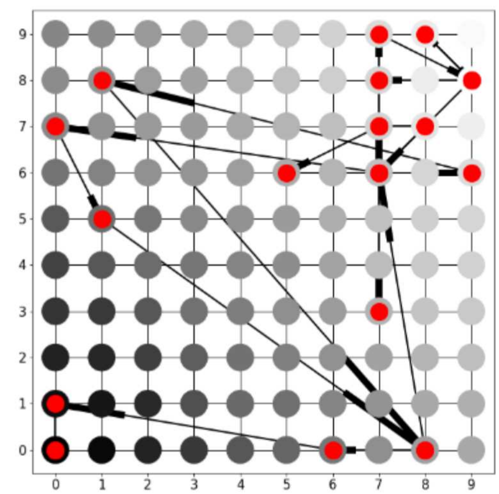

Figure 5: Engine degradation trend observed on a SOM trained with one control setting data.

Even when consistent degradation trends were obtained on individual operational modes, showing the evolution from the good to the faulty operational conditions, it is possible to obtain incomplete degradation trends on these individual SOM. Considering the data format, only one operational mode is encounter for each engine cycle (changing randomly from one cycle to another among six different options) and this could yield a lack of data for a specific operational mode at some stages of the engine lifecycle. This condition is more critical at the end of the lifecycle, where a lack of information on a specific operational mode might hide a faulty engine condition.

Besides, in real life, an engine passes via multiple operational conditions during one single flight. As such, information from these different operational modes could be gathered and used to improve the confidence in the diagnostic result. If information is used only from one operational condition, many external factors, like bad weather conditions, avoidance manoeuvres or gusts might induce a false result in the degradation indicator and this could lead to trigger maintenance actions in the wrong moment.

To address this situation, a degradation assessment on several different operational modes is proposed to enhance the decision-making process. A new SOM was developed, this time using the information from the 6 operational modes available on the dataset. The SOM size for this case was determined following the method suggested by (Delgado et al. 2017), resulting a 15x15 map the best option. Verification has shown that no significant improvements were obtained using SOMs with a higher dimension maps. The result of the trained SOM with the 6 operational modes is shown in Figure 6. Each bounded region on the trained SOM represents an operational mode that can be easily labelled using the known features of each mode in the dataset.

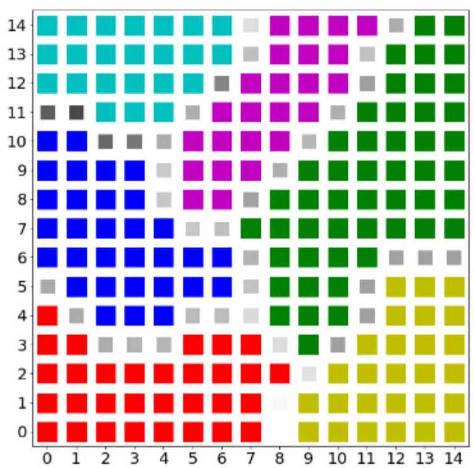

Figure 6: 15x15 SOM trained with full jet-engine dataset.

Once the regions were identified a test was carried out to prove SOM accuracy at clustering the engines cycles to the corresponding operational mode. All engines available in the data set were shown to the SOM and it succeeded allocating each cycle to the correct map region. This shows that each operational mode is well bounded to a single map region allowing analysing the degradation trend on each operational mode at the same time for a single engine. Figure 7 shows a single engine tested with the 15x15 SOM. The black point within the coloured neurons represent the exited neurons of each region and arrows represent the degradation evolution on each operational mode.

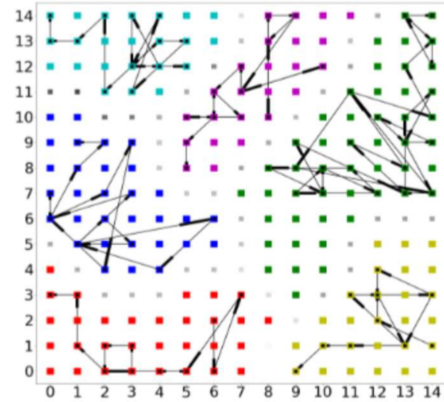

Figure 7: Testing engine sample from training subset on the 15x15 SOM.

Since the difference between each operational mode is large, the small differences within each region are not easily observable as it was on SOM trained with one single operational mode. This poses a serious problem to follow the engine degradation trends. Yet, even when these variations between the neurons on a map region are too small to be observed by the colours of the map, variation on the neurons weights (numerical values) exist and this information can be used to describe the engine degradation process.

The norm of each neuron weight vector is computed, this value is normalized between 0 and 1 for each operational mode. Let be $\mathrm{OP_m}$ a set of neurons that constitute an operational mode $'\mathrm{m}'$, then:

$$\forall\, n_{ij} \in OP_m$$

$$d_m = \frac{\left\|W_{i\prime j\prime}\right\|_2 - \min(\left\|W_{i\prime j\prime}\right\|_2)}{\max\left(\left\|W_{i\prime j\prime}\right\|_2\right) - \min(\left\|W_{i\prime j\prime}\right\|_2)} \tag{7}$$

with d_m being the degradation index on each neuron n_{ij} and, $w_{i\prime j\prime}$ being each neuron weight within the same operational mode.

The obtained values are used as labels for each neuron and once the labelling task was finished, a clear evolution from 0 to 1 is easily observed on each map region. This value has been called “degradation index” and its use is key for diagnosis strategy. Figure 8 shows the degradation index (d_m) as labels on the 15x15 SOM, being the 0 the best operational conditions, the 1 the faulty conditions and the values between them the degradation process on the engine lifecycle.

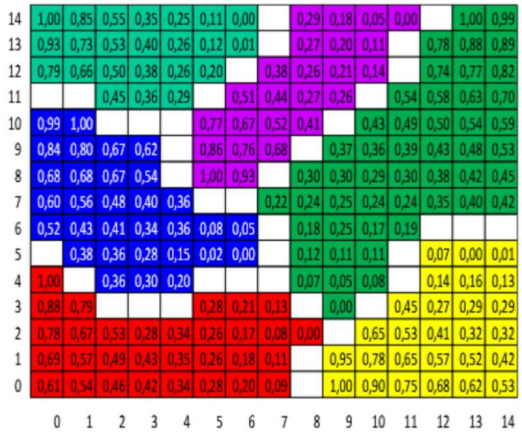

Figure 8: 15x15 SOM labelled with degradation values on each region.

As the goal is to follow the current health state of the engines, the approach is to determine whether the engines condition is faulty or nominal. First, individual degradation indexes d_m are obtained for all operational modes in one single flight. Once the individual degradation indexes d_m are obtained it is possible to combine them in a consolidated degradation index D_C. This consolidated value allows to observe a general degradation evolution considering as many operational modes as possible (equation 8).

$$D_C = \sum_{m=1}^{l} \frac{d_m}{l} \tag{8}$$

D_C: consolidated degradation index at any flight.

d_m: individual degradation indexes for each m operational mode.

l: number of operational modes registered during the flight.

To compute D_C, the ideal case is when all possible operational modes are registered in the same flight (six operational modes for the current study). However, the engine might not register one or several operational modes during a flight, if this happens, d_m might be obtained from the previous flights, as long the data is still considered as valid. A first experimental approach in the current study established the previous 5 flights as valid to take missing d_m for the latest flight D_C computation.

The consolidated degradation index D_c encounters different perspectives of the degradation evolution using a single value for each engine flight. Plotting this value versus the number of flights (discrete cycles for the simulated data), an increasing evolution of the degradation is observed until a certain threshold, close to value 1. Figure 9 shows the degradation evolution along the lifecycle of one engine, using D_c approach.

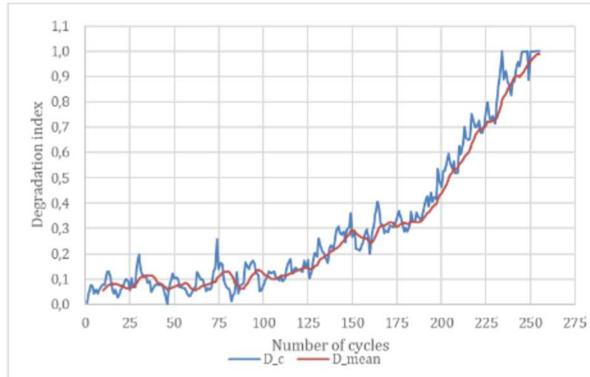

Figure 9: Degradation evolution of one cycle

The fluctuations of curve D_C pose a problem to study the degradation trends and determine an experimental degradation index threshold. To soften D_C curve, the arithmetic moving average is applied considering the measurements of the last 10 flights. These 10 flights represented the best experimental compromise between softening the curve and keeping the variation of the degradation index as wide as possible. This produce the curve D_{mean} versus the number of flights (cycles), which facilitate the degradation trends study. Figure 9 also shows the D_{mean} curve for the current engine under study.

Obtaining D_{mean} curve for multiple failed engines and plotting all of them in the same graphic, it is possible to determine an experimental threshold for the proposed method of degradation index. For the current jet-engines analysis a threshold of 0.93 for D_{mean} is proposed. Engines below of such threshold are considered in nominal conditions and those overcoming the threshold are considered as failed. In real life applications, such thresholds must be supported by knowledge of experts, regulations and manufacturers specifications.

Once the threshold is known other engines were evaluated with the proposed method to confirm the current health status of the engine. Table 2 summarize the data for the last measured flight of some of the assessed engines. Columns 2 to 7 show the individual degradation indexes for the last measurement of the engines Columns 8 and 9 present D_C and D_{mean} correspondingly and last column shows the confirmed health status of the engine.

Table 2. Engine condition diagnosis from the degradation index.

Engine	d_1	d_2	d_3	d_4	d_5	d_6	D_C	D_{mean}	Condition
Engine1	0,99			1,00		1,00	1,00	0,99	Failed
Engine2	1,00	1,00			1,00	1,00	1,00	0,98	Failed
Engine3			0,99	1,00		0,93	0,97	0,93	Failed
Engine4		0,78	0,78		1,00	0,73	0,82	0,77	Nominal
Engine5	0,41	0,42	0,35	0,44	0,14		0,35	0,38	Nominal
Engine6	0,38		0,24	0,18		0,38	0,30	0,31	Nominal

As explained in section 3, there are two subsets in the data base; the training subset and the testing subset. Engines from 1 to 3 in table 2 were taken randomly from the training subset, in which all engines reached the faulty threshold. For such engines, the degradation index value confirms the reaching of a faulty condition since the value is greater than the experimental threshold. Engines from 4 to 6 were selected randomly from the testing subset and in this case, it concerns engines that remained in good operational conditions (nominal condition). The degradation index confirms this, as the value remains under the degradation threshold.

6 CONCLUSION

The correct diagnosis of health state allows to improve the decision-making process in maintenance departments, bringing benefits to the companies. When a system faces a normal lifecycle, it is summited to degradation thought the time and sometimes is not easy to identify the degradation evolution from complex data so that specialized methods such as SOM should be applied to facilitate the data analysis.

Using SOM is has been possible to identify the degradation trend on simulated aircraft jet-engines, considering first, a single operational mode of the engines at the time. This yielded a SOM in which was possible to observe the evolution of the degradation by means of a colours transitions on the map neurons, from a white region (optimal conditions) to a black region (faulty conditions) on the trained map.

Since an Aircraft faces several operational modes during a flight, a diagnosis based on one single operational mode is not enough, since the data might be altered for external reasons. A multiple operational mode approach is proposed, training another SOM with all operational modes available on the jet-engines dataset. To facilitate the analysis, a degradation index has been computed from each neuron weight vector and normalized from 0 to 1 on each operational mode. These degradation indexes allowed to determine if the tested engines are in the faulty condition or not.

Future work will be dedicated to the computation of Remaining Useful Life of the jet-engines considering the degradation indexes obtained from the trained SOM, this includes further analysis among the degradation index threshold. Finally, based on the obtained results, different SOM architectures will be studied so to assess their suitability to the problem at hand, and looking at the applicability in terms of computation time for training, convergence, etc.

REFERENCES

Côme, E., Cottrell, M., Verleysen, M., and Lacaille, J., 2010, Self-Organising Star (SOS) for health monitoring. Michel Verleysen Ed. European conference on artificial neural networks, pp. 99-104.

Côme, E., Cottrell, M., Verleysen, M., and Lacaille, J., 2010, Aircraft engine health monitoring using Self-Organising Maps, Petra Perner Ed. 10th Indutrial Conference, ICDM 2010, Berlin Springer 6171, pp. 405-417.

Delgado, S., Higuera, C., Calle-Espinoza, J., Morán, F., and Montero, F., 2017, A SOM prototype-based cluster analysis methodology, Expert Systems with Applications, vol. 88, pp. 14-28.

Germen, E., Basaran, M., & Fidan, M. 2014. Sound based induction motor fault diagnosis using Kohonen self-organising map. *Mechanical Systems and Signal Processing, 46*, 45-58.

Hu, J., Zhang, L., and Liang, W., 2013, Dynamic degradation observer for bearing fault by MTS-SOM system, Mechanical Systems and Signal Processing, vol. 36, pp. 385-400.

Jardine, A.K.S., Lin, D., and Banjevic, D., 2006, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing 20, pp. 1483-1510.

Kohonen, T., 1990, The self-organising map, Proceedings of the IEEE, vol. 78, nº 9, pp. 1464 -1480.

Kohonen, T., 1995, Self-organising map. Second Edition. Springer Series in Information Sciences. Ed. T.S. Huang, T. Kohonen and M.R Schroeder. Berlin Heidelberg: Springer-Verlag.

Nguyen, K.-A., Do, P., and Grall, A., 2015, « Multilevel predictive maintenance for multi-component system », Reliability Engineering and System Safety 144, pp. 83-94,.

Saxena, A., Goebel, K., Simon, D., and Eklund, N., 2008, Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation, NASA, Integrated Vehicle Health Management.

Vingerhoeds, R.A., Janssens, P., Netten, B.D., and Aznar Fernández-Montesinos, M., 1995, Enhancing off-line and on-line condition monitoring and fault diagnosis, Control Engineering Practice, Vol.3, pp. 1515-1528.