Fault Detection for Aircraft Piston Engine Using Self-Organizing Map

Dubravko Miljković
Hrvatska elektroprivreda, Zagreb, Croatia
dubravko.miljkovic@hep.hr

Abstract - Aircraft piston engine can be monitored using an advanced graphic engine monitor. Such engine monitor can supply a large amount of data containing evolution of engine parameters through the time. Analysis of a vast amount of multidimensional temporal data by a self-organizing map may aid in data visualization, but also in detection of engine parameter deviations from normality indicating potential problems in operation of aircraft piston engine. For determination of engine parameter space that corresponds to normal engine operation quantization error of the self-organizing map is used.

I. INTRODUCTION

Piston engine is a heat engine designed to convert energy into rotational mechanical motion. It uses reciprocating pistons to convert pressure into a rotating motion, [1]. The chances for an engine failure are pretty remote, but do happen. Piston engine reliability depends on the complexity of the engine (number of cylinders, turbocharging), use (private or club aircraft) and maintenance. Piston engine has a lot of moving parts and in comparison to a turbine engine its reliability is significantly lower (about seven times), yet low cost of such engines makes them a popular choice among most general aviation airplanes. An engine failure is a serious situation, both in single and twin engine aircrafts. Single engine aircraft can attempt to glide to a nearest airport (if altitude and winds permit) or perform an off airport landing (challenging situation if over inhospitable terrain, at night, low cloud ceiling and low visibility). Twin engine (particularly piston engine) aircrafts encounter an asymmetric thrust situation that pose increased risk during take-off, initial climb and possible go-around (that should be avoided altogether). Single engine climb rate is severely reduced and is only about 20% of climb rate available when operating on both engines (not 50% as one would expect). Failures further complicate the fact that it can be partial power loss instead of full power loss.

II. ENGINE MONITOR

Engine monitors (also called engine analyzers or engine management system) provide monitoring of vital engine parameters, [1]. These parameters are measured, by engine probes, recorded and presented on a graphic display with parameters usually shown as vertical bars. Cylinder head temperatures (CHTs), Exhaust gas Temperatures (EGTs) for each cylinder and Turbine inlet temperatures (TIT) are shown graphically as bars on the display of an engine monitor, [2,3], as shown in Fig. 1. Some additional engine parameters like engine rotational speed (RPM), calculated % of maximal horsepower (% HP), etc. are also shown. All parameters are logged and can later be analyzed on the ground. Such an advanced piston engine-monitoring instrument helps pilots to better manage engine operation and detect engine problems in real time (while the engine is running). It is also of great value to maintenance personnel. Logged data are available for post-flight analysis helping to detect impeding problems and suggest appropriate preventive actions. Engine parameters that are most commonly monitored in engine monitors are listed in Table I, [2,3].

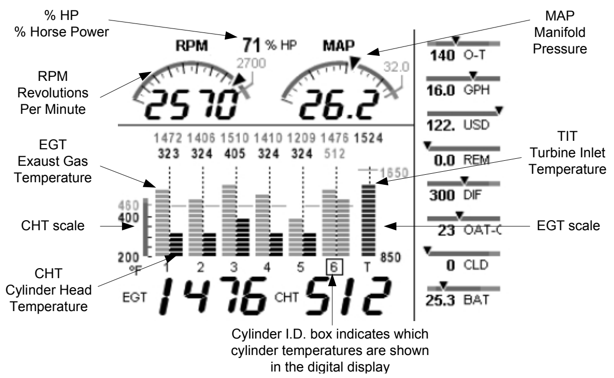

Figure 1. Engine monitor with separate bars for EGT and CHT (JPI EDM 830)

TABLE I. MONITORED ENGINE PARAMETERS

Parameter	Description
EGT	Exhaust Gas Temperature
CHT	Cylinder Head temperature
OIL TEMP	Oil Temperature [1]
OIL PRES	Oil Pressure [1]
TIT1	Turbine Inlet Temperature 1[1]
TIT2	Turbine Inlet Temperature 2 [1]
OAT	Outside Air Temperature
CDT	Compressor Discharge Temperature [1]
IAT	Intercooler Air Temperature [1]
CRB	Carburetor Air Temperature [1]
CDT - IAT	Intercooler cooling
RPM	Rotations Per Minute
MAP	Manifold Pressure
%HP	% Horse Power
CLD	CHT cooling rate [2]
DIF	EGT span [3]
FF	Fuel Flow [1]

[1]optional, [2]fastest cooling cylinder, [3]difference between the hottest and coolest EGT

An engine monitor typically has two modes: monitoring and lean operation mode (for accurate adjustment of fuel mixture). Various engine problems can be detected using an engine monitor by spotting characteristic EGT/CHT bar patterns, [4]. Such patterns are catalogued and included in a technical documentation accompanying an engine monitor, [2].

III. Visualization of Engine Parameters

Engine parameters recorded during each flight can later be visually presented using specialized plotting software, e.g. the EzTrends2, [5]. Graphical representation of engine parameters (multidimensional data) through the duration of the whole flight as presented by JPI EZTrends2 is shown in Fig. 2 (upper curves represent EGTs and lower CHTs, inverted colors and background).

Figure 2. Engine parameters plot

As can be seen form the figure, EGT and CHT curves from the engine with no faults present group together. Another, simpler, representation of engine operation that doesn’t include evolution of parameters during a time is use of flight summaries, Fig. 3. Temperature ranges and average temperatures are shown in summary table. Discrepancies from the symmetry between temperatures of various cylinders can also be easily spotted in accompanying graphical representation. Engine monitor data can also be analyzed statistically, one such analysis is given in [6].

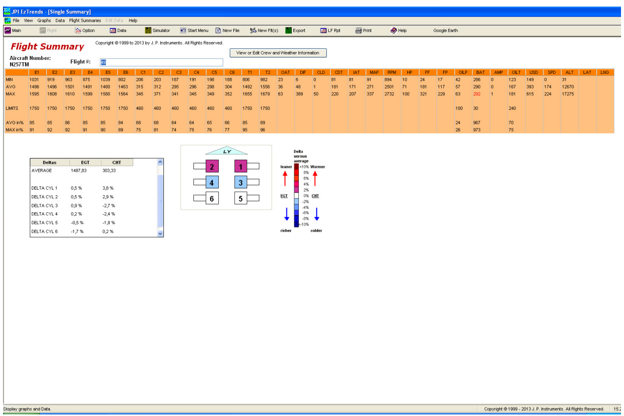

Figure 3. Flight summary

Despite interesting graphical representations of engine parameters, both in real-time on engine monitor display during a flight and later on computer using specialized accompanying software, more subtle engine problems are not so easy to discern and process require trained maintenance personnel with the experience in engine diagnostics from available recorded engine parameters. Catalogued patterns of various engine problems are commonly supplied with the engine monitor documentation, [2].

IV. Self-Organiznig Maps

Self-Organizing Map (SOM) is a type of neural network architecture that is trained using unsupervised training and produces a low-dimensional (most common 2D) discretized representation of the input space of the presented training samples. It provides dimensionality reduction of available multidimensional training data. The map is used in two modes: training mode and mapping mode. During the training mode a two-dimensional discrete representation of input space is formed. In the mapping mode input sample is assigned to the closest member of a map, thus classifying newly presented input vector. The most common two dimensional SOM lattice with neuron neighborhood is shown in Fig. 4. SOM uses competitive learning with lateral inhibition function. The most often used neighborhood function is the Mexican hat. An engine monitor with its parameters can form an input vector that is simultaneously fed to all SOM neuronal units. The input vector is mapped to the winner node (node with highest activation). More on SOM and learning algorithm could be found in [7,8].

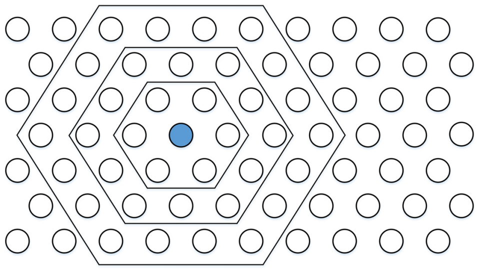

Figure 4. Self-organizing map (SOM) with hexagonal lattice and the local neighborhood of the one particular neuron

V. Fault Detection Using SOM

SOMs could be used for the visualization of engine and equipment parameters, [9,10]. Sometimes further step is taken with the application of the SOM to fault detection and condition monitoring, [11,12]. Fault detection using SOM is based on the assumption that a SOM (or its parameters like quantization error or SOMs hits map) that belongs to the normal engine operation differs from one that belongs to a faulty engine. The main problem is that SOMs trained on similar data may not always give quite similar results (e.g. occurrence of data shift in a map), [13]. Resulting self-organization is influenced by the various initial conditions: SOM initial weights, [14], the choice of the neighborhood function, data normalization, the learning rate and the order of presentation sequence of training vectors. One analysis of statistical measures to assess the stability of the results of SOM training is given in [15].

A. *Methods for Fault Detection*

Some ideas how to compare SOMs are listed below:

1) *Comparison of SOM maps*

Comparison of SOM maps (test and reference) is done by comparing weights belonging to SOM planes, Fig. 5. These maps can be compared visually, an analytical approach is, however, more difficult, [13,16].

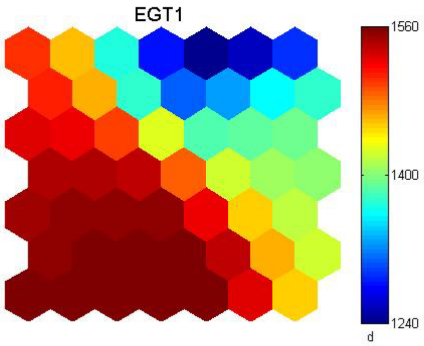

Figure 5. SOM plane for EGT1 (7x7 lattice example)

2) *Comparison of Hit Maps*

Comparison of SOM maps is done by comparing hits count belonging to SOM (depicted as size of hexagon line), Fig. 6. The method is based on the assumption that particular region of the map belongs to normal engine operation. Any visit to SOM units outside this region, particularly if held during a longer period, should be regarded as a suspicious engine condition.

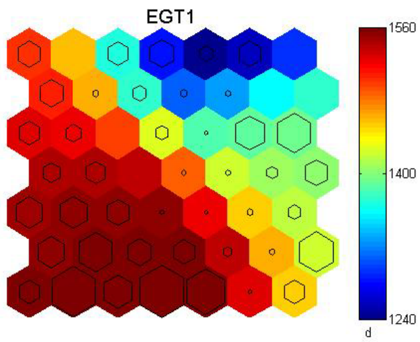

Figure 6. SOM plane for EGT1 with hits count

3) *Use of Mean Quantization Error (MQE)*

Quantization error e_{QE} is the distance between the input pattern vector $\boldsymbol{z}$ and the weight vector $\boldsymbol{w}_{bmu}$ of the Best Matching Unit (BMU), $\boldsymbol{w}_k$ is the weight vector of k^{th} SOM unit and K is the number of SOM units, (1), (2):

$$e_{QE} = \left\| \boldsymbol{z} - \boldsymbol{w}_{bmu} \right\| \tag{1}$$

$$e_{QE} = \min_{1 \le k \le K} \left\| \boldsymbol{z} - \boldsymbol{w}_k \right\| \tag{2}$$

Mean Quantization Error (MQE), e_{MQE} is given by (3):

$$e_{MQE} = \frac{1}{N} \left(\sum_{i=1}^{N} \min_{1 \le k \le K} \left\| \boldsymbol{z}_i - \boldsymbol{w}_k \right\| \right) \tag{3}$$

where N is the number of patterns in a dataset, and $\boldsymbol{z}_i$ is the i^{th} pattern vector in a dataset.

The accuracy of a SOM in representing its inputs can be validated using the mean quantization error (MQE). It quantifies how well a previously trained SOM approximates the presented data items. Low values for MQE show that the data set is well represented by the SOM.

B. *Method Used in This Paper*

Method with MQE is applied in this paper. MQE is a more robust measure than a comparison of SOMs as it is resistant to data shift in a map.

1) *Training Phase*

During the training phase, SOM captures the statistical distribution of the data from a non-faulty engine, Fig. 7.

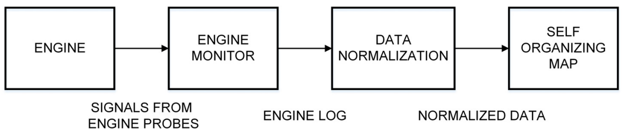

Figure 7. Training phase

2) *Monitoring Phase*

During the monitoring phase, input vectors are classified to the Best Matching Unit (BMU) and Mean Quantization Error (MQE) is determined, Fig. 8.

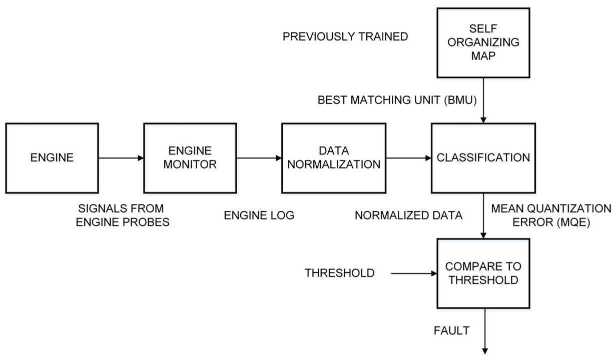

Figure 8. Monitoring phase

C. *Off-Line and On-Line Detection*

Method for fault detection can be performed as off-line detection (compares complete engine log with the SOM after the flight) and on-line detection (compares each new vector with the SOM).

1) *Off-Line Detection*

- Normal condition

The new engine log is classified as normal if e_{MQE} (mean quantization error) is less than the threshold L, (4):

$$e_{MQE} < L \tag{4}$$

- Suspicious condition
  The new engine log is classified as suspicious if e_{MQE} is greater than the threshold L, (5):

$$e_{MQE} \geq L \quad (5)$$

2) On-Line Detection

- Normal condition
  The new vector is classified as normal if e_{QE} (quantization error) is less than the threshold L, (6).

$$e_{QE} < L \quad (6)$$

- Suspicious condition
  The new vector is classified as suspicious if e_{QE} is greater than the threshold L, (7).

$$e_{QE} \geq L \quad (7)$$

In real applications additional filtering will be required (e.g. minimal number of threshold violations in a specific period of time).

VI. EXPERIMENT

The SOM was trained using engine parameters from available engine monitor logs.

A. Available Data

An experiment was performed using engine logs accompanying the EzTrends2 software, belonging to a six cylinder engine, with no known faults present, [5]. Available experimental data were flight logs Flt#192 (duration 1.49h) and Flt#193 (duration 1.23h). As these files belong to twin engine aircraft, only one (left) engine was selected for analysis. Engine parameters are logged every 6 seconds. Part of engine log Flt#192 is shown in Fig. 9, (left engine EGTs: LE1-6 and CHTs: LC1-6).

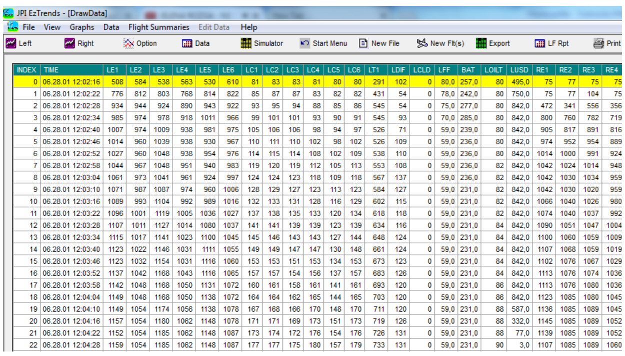

Figure 9. Example of engine log

B. Synthetic Faults

Because it is difficult to obtain data logs from faulty engines (as they are still quite reliable) and one has rarely access to large maintenance facility, testing datasets were created artificially by modifying existing engine data log (Flt#193) according to common descriptions of engine problems that include EGT and CHT temperature deviations. Examples of two patterns corresponding to problems in engine operation are shown in Fig. 10 and Fig. 11. Such patterns are often catalogued in documentation belonging to an engine monitor. Detailed description of fault indications as EGT/CHT bars on the engine monitor with temperature differences is given in [2]. This gives following datasets, as shown in Table II.

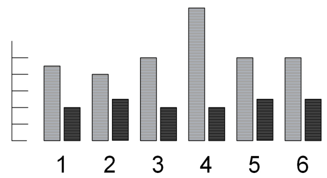

Figure 10. Example of fault pattern, failure 1, EGT rise for one cylinder

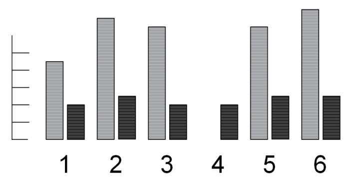

Figure 11. Another example of fault pattern, failure 2, loss of EGT for one cylinder

TABLE II. USED FILES

File	Duration	Frames	Description
Training Flt#192	1.49h	896	Training data
Test (normal) Flt#193	1.23h	736	Test (normal) data
Test (Failure 1) [1]	1.23h	736	Fouling, faulty plug, wire or distributor
Test (Failure 2) [1]	1.23h	736	Stuck valve
Test (Failure 3) [1]	1.23h	736	Faulty valve lifter
Test (Failure 4) [1]	1.23h	736	Dirty fuel injectors or fouled plugs
Test (Failure 5) [1]	1.23h	736	Burned exhaust valve
Test (Failure 6) [1]	1.23h	736	Detonation
Test (Failure 7) [1]	1.23h	736	Pre-ignition
Test (Failure 8) [1]	1.23h	736	Leaking exhaust gasket

[1] Flt#193 modified according to the particular fault description

C. Selection of Input Parameters

Engine monitors are primarily designed to monitor engine temperatures EGTs and CHTs as they reflect a combustion process that is happening in engine cylinders. The other parameters were added to most monitors later as additional information about engine operation. Parameters, not directly related to a combustion process, are not considered in this experiment. Selected input parameters are:

- Exhaust Gas Temperatures (cylinders 1-6): EGT1, EGT2, EGT3, EGT4, EGT5 and EGT6
- Cylinder Head Temperatures (cylinders 1-6): CHT1, CHT2, CHT3, CHT4, CHT5 and CHT6

Input vector $\boldsymbol{z}$ to SOM that consists of EGT and CHT temperatures is given in (8), (static, current state, neighborhood for time evolution analysis is not included):

$$\boldsymbol{z} = [\,T_{EGT,1}, \dots, T_{EGT,6}, T_{CHT,1}, \dots, T_{CHT,6}] \quad (8)$$

Software used for analysis is the SOM Toolbox for MATLAB 5, [17]. Initialization was linear. Training was performed using a batch algorithm.

D. Sizing of SOM

The size of the SOM was determined according to (9), where M denotes number of SOM units and N is the number of samples, [12]:

$$M \approx 5\sqrt{N} \tag{9}$$

Most flights last about an hour. One hour flight consists of 600 frames. For 600 frames $M \approx 123$. In case of SOM lattice with equal number of rows and columns it is a 11x11 lattice.

E. Normalization of Parameters

Due to various ranges of values for EGT and CHT parameters should be normalized (as parameters are features it is also feature scaling) before applying to SOM. One interesting analysis of data normalization applied to SOMs is given in [18].

Two methods for feature scaling are popular:

- Rescaling (normalization): rescales the values into to a [0, 1] range. This is useful in cases where all parameters must belong to the same range. The disadvantage of this method is that the outliers from the data set are lost, (10):

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{10}$$

- Standardization (variance method): rescales data to have a zero mean (μ) and standard deviation (σ) of 1 (unit variance), (11):

$$x' = \frac{x - \bar{x}}{\sigma} \tag{11}$$

Standardization method is recommended for most applications. It was also chosen for SOM visualizations, partially because it was already included in the SOM Toolbox.

Previous normalization methods are single-variable operations suitable for visualization of isolated parameter planes. However, for preserving multivariate anomalies and comparisons of MQEs of test datasets classified by a SOM produced by a normal dataset with the threshold for the purpose of fault detection, following normalization is applied to the CHT data instead, (12). Considering that the average EGT value is approximately 3.8 times greater than the average CHT value this puts normalized CHT values in the same range with EGT for each cylinder i:

$$T_{N,CHT,i} = 3.8 T_{CHT,i} \qquad i = 1, \ldots, 6 \tag{12}$$

F. Data Visualization

Following are three examples of visualization of engine logs using SOMs. The training process is accomplished with normalized vectors, however legend corresponds to real temperatures. SOM corresponding to training data from a normally operating engine is shown in Fig. 12. In Fig. 13 there is another example of a normally operating engine, similar to one in Fig. 12, and in Fig. 14 is an example of faulty engine operation (see EGT4 plane).

- SOM Training data (normal engine operation, Flt#192)

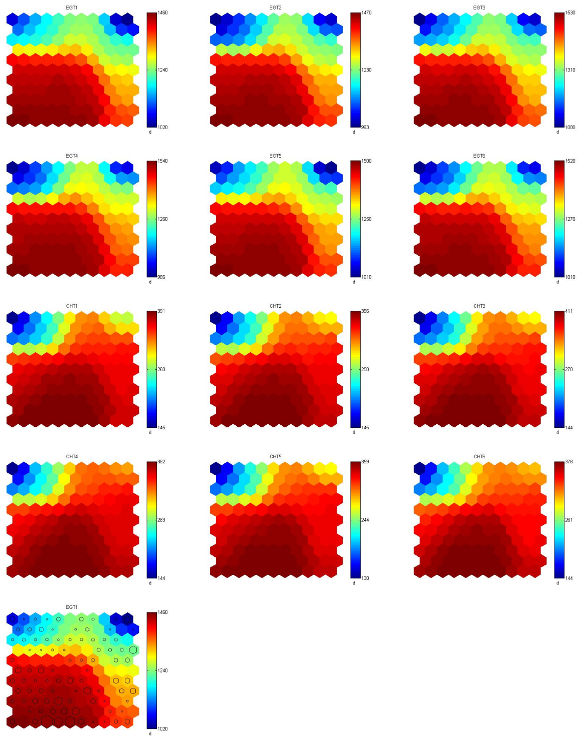

Figure 12. Visualization of training data in SOM (normal engine operation), planes EGT1-6 and CHT1-6 plus hit map

- SOM Test data (normal engine operation, Flt#193)

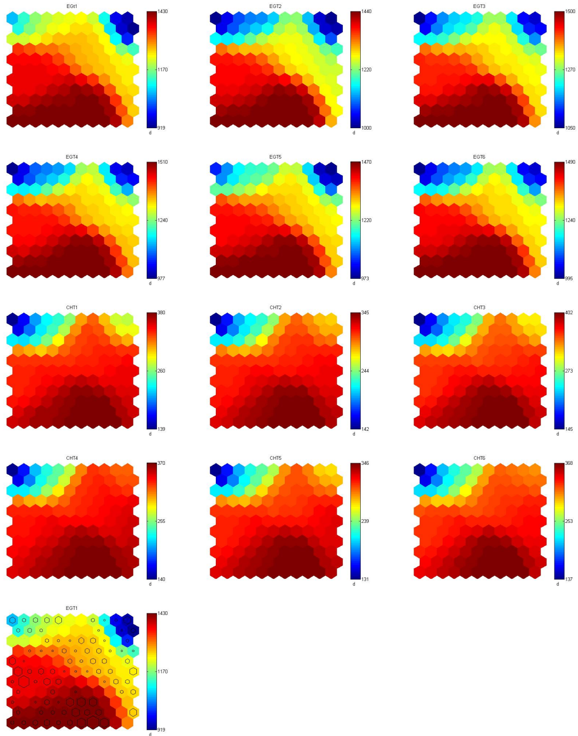

Figure 13. Visualization of test data in SOM (normal engine operation), planes EGT1-6 and CHT1-6 plus hit map.

- SOM Test data (example of problematic engine operation, artificial data: Failure 3)

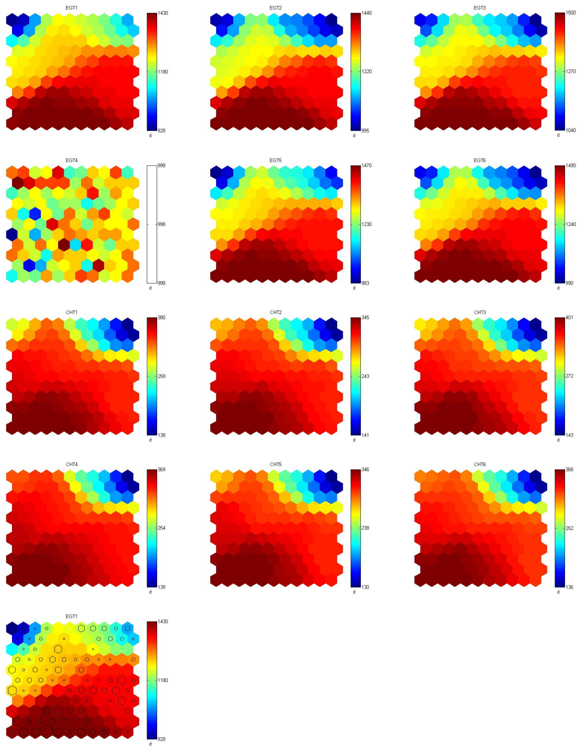

Figure 14. Visualization of test data in SOM (abnormal engine operation, Failure 3), planes EGT1-6 and CHT1-6 plus hit map

G. *Fault Detection of Failure Datasets*

Artificial data and MQEs for abnormal engine operation (eight different failures) were also analyzed. Examples of eight different engine problems were included in eight test datasets. Example of visualization of abnormal engine operation for one failure (Failure 3) is shown in Fig. 14. MQEs (e_{MQE}) were calculated for different data sets and results are shown in Table III, including the original, non-normalized data and data normalized according to (12). These values could be used for determination of the threshold L by multiplying with a constant α for the safety margin (e.g. α=0.8), (13). More failure examples are needed for more precise value of L.

TABLE III. MQE FOR SOM AND VARIOUS DATASETS

Dataset	e_{MQE} [1]	e_{MQE} [2]
Training data (Flt#192)	19.6937	32.756
Normal data (Flt#193)	33.412	58.8373
Failure 1	389.6016	605.1097
Failure 2	721.1847	747.0054
Failure 3	360.6092	374.6447
Failure 4	500.2829	694.2098
Failure 5	342.6307	500.4933
Failure 6	333.4395	419.8564
Failure 7	466.7379	522.2226
Failure 8	229.1554	825.1329
Min (for Failures 1-8)	229.1554	374.6447

[1]non-normalized data, [2]normalized data according to (12)

$$L = \alpha \min_{k}(e_{MQE,k}) \quad k=1, 2, \dots , 8 \tag{13}$$

In previous experiment one SOM is used for all engine operating regimes. Better accuracy could be achieved if separate SOM is prepared for each particular operating regime. Methods for regime selections are given in [3].

VII. CONCLUSION

Engine monitors for aircraft piston engines record large amount of data that may be difficult to visualize and analyze. SOMs with its unsupervised learning algorithm are able to produce the two-dimensional mapping of multivariate data. This provides dimensionality reduction while topologically preserving similarities in the input space. Such two-dimensional representation is suitable for visualization. Resulting SOM can be further analyzed and used for the fault detection. Use of SOM for monitoring of aircraft piston engine operation is proposed. By comparing MQE of engine logs from engines under the test with the threshold it is possible to detect suspicious engine operation. On-line detection is also possible by comparing input vector to best matching unit (BMU) and comparing quantization error with the predetermined threshold.

REFERENCES

[1] D. Miljković, “Engine Monitors for General Aviation Piston Engines Condition Monitoring”, CrSNDT Info Journal, Vol. 3, No. 2 (10), pp. 19-23, 2013

[2] Pilot’s Guide - Engine Data Management, EDM-730, EDM-830, EDM-740, J. P. Instruments, Inc., 2010

[3] D. Miljković, “Regime Dependent Aircraft Piston Engine Monitoring”, Proc. MIPRO 2014, Opatija, Croatia, 26-30 May 2014, pp. 1311-1316

[4] J. P. Instruments, Technical Support - Video Tutorials, https://www.jpinstruments.com/technical-support/video-tutorials/

[5] EzTrends (Download and Plotting Software), JPI, California, USA, Available: http://www.jpitech.com/softwaredownload.php

[6] D. Miljković, “Statistical Properties of Aircraft Piston Engine Monitor CM Data”, Proc. MATEST 2013, Zagreb, 9 October 2013

[7] T. Kohonen, Self-Organizing Maps, 2nd ed., Springer 1997

[8] T. Kohonen, MATLAB Implementations and Applications of the Self-Organizing Map, Unigrafia, Helsinki, Finland, 2014

[9] R. Ji-Hong, C. Jiang-Cheng and W. Nan, “Visual Analysis of SOM Network in Fault Diagnosis”, Proc. 2011 International Conference on Physics Science and Technology (ICPST 2011), Hong Kong, 12 – 13 December 2011, pp. 333-338

[10] E. Come, M. Cottrell, M. Verleysen and J. Lacaille, “Aircraft engine health monitoring using Self-Organizing Maps”, Proc. 10th Industrial Conference, ICDM 2010, Berlin, Germany, 12-14 July 2010, pp. 405-417

[11] A. Zachrison, “Fluid Power Applications Using Self-Organising Maps in Condition Monitoring”, PhD Thesis, Linköping University, Linköping, Sweden, 2008

[12] J. Tian, M. H. Azarian. and M. Pecht, “Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm”, Proc. European Conference of the Prognostics and Health Management Society 2014, Nantes, France, 8-10 July 2014, pp. 110-118

[13] R. Mayer, D. Baum, R. Neumayer, A. Rauber: "Analytic Comparison of Self-Organising Maps", Proc. 7th International Workshop on Self-Organizing Maps (WSOM 2009), St. Augustine, Florida, USA, 8-10 June 2009, pp. 182-190

[14] A. A. Akinduko and E. M. Mirkes, “Initialization of Self-Organizing Maps: Principal Components Versus Random Initialization. A Case Study”, CoRR abs/1210.5873, 2012

[15] M. Cottrell, E. de Bodt and M. Verleysen, “A Statistical Tool to Assess the Reliability of Self-Organizing Maps”. In: Advances in Self-Organising Maps, Springer, London, 2001

[16] S. Kaski and K. Lagus, “Comparing self-organizing maps”, Proc. ICANN’96, Bochum, Germany, July 16 - 19, 1996, pp. 809-814

[17] J. Vesanto, J. Himberg, E. Alhoniemi and J. Parhakangas, SOM Toolbox for MATLAB 5, Helsinky University of Technology, April 2000

[18] P. Demartinesand and F. Blayo, “Kohonen Self-Organizing Maps: Is the normalization necessary?”, Journal of Complex Systems, Vol. 6, Iss. 2, pp. 105-123, 1992