



Ingeniería. Investigación y Tecnología

ISSN: 1405-7743

iit.revista@gmail.com

Universidad Nacional Autónoma de
México
México

Felipe-Altamirano, Salvador; Loboda, Igor; Cisneros-Azuara, Felipe de Jesús; Pérez-Ruiz,
Juan Luis

Gas Turbine Diagnostic Algorithm Testing Using the Software ProDiMES

Ingeniería. Investigación y Tecnología, vol. XVIII, núm. 1, enero-marzo, 2017, pp. 75-86

Universidad Nacional Autónoma de México

Distrito Federal, México

Available in: <http://www.redalyc.org/articulo.oa?id=40449649007>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative



Gas Turbine Diagnostic Algorithm Testing Using the Software ProDiMES

Prueba de algoritmo de diagnóstico de turbinas de gas usando el Software ProDiMES

Felipe-Altamirano Salvador

National Polytechnic Institute
School of Mechanical and Electrical Engineering
Department of Graduate Studies and Research, Unit Culhuacan
Email: svd.felipe@gmail.com

Loboda Igor

National Polytechnic Institute
School of Mechanical and Electrical Engineering
Department of Graduate Studies and Research, Unit Culhuacan
Email: iloboda@ipn.mx

Cisneros-Azuara Felipe de Jesús

National Polytechnic Institute
School of Mechanical and Electrical Engineering
Department of Graduate Studies and Research, Unit Culhuacan
Email: fjca87@gmail.com

Pérez-Ruiz Juan Luis

National Polytechnic Institute
School of Mechanical and Electrical Engineering
Department of Graduate Studies and Research, Unit Zacatenco
Email: perezruiz305@gmail.com

Abstract

Since gas turbines are very complex and potentially unreliable machines, the improvement of their monitoring systems becomes an essential part. Considering this necessity, the present paper performs a gas turbine diagnostic algorithm testing. The methodology proposed is formed by three stages. In the first stage, the commercial software ProDiMES (*Propulsion Diagnostic Method Evaluation Strategy*) is used to simulate an engine fleet and generate data with fault and no-fault conditions. In the second stage, a baseline model testing is implemented to improve the healthy engine performance approximation. Finally, a fault recognition stage based on a pattern recognition technique (Multi-Layer Perceptron) performs the diagnosis and calculates the probability of correct diagnostic decisions. The results obtained show that: a) the software ProDiMES is an easy and convenient tool to evaluate gas turbine diagnostic methods, b) the baseline model testing is a key step because it allows reducing the errors that can negatively influence the diagnostic process and c) the algorithm correctly performs the fault recognition task.

Keywords: gas turbine diagnostics, baseline model, fault recognition, ProDiMES, multi-layer perceptron.

Resumen

Debido a que las turbinas de gas son máquinas muy complejas y potencialmente no fiables, el mejoramiento de sus sistemas de monitoreo se vuelve una parte esencial. Considerando esta necesidad, el presente trabajo realiza una prueba de algoritmo de diagnóstico de turbinas de gas. La metodología propuesta se forma por tres etapas. En la primera, el Software comercial ProDiMES (Propulsion Diagnostic Method Evaluation Strategy) se emplea para simular una flota de motores y generar datos con condiciones de falla y sin falla. En la segunda etapa, se implementa una prueba de modelos de referencia para mejorar la aproximación de rendimiento de motor sano. Finalmente, una etapa de reconocimiento de fallas basada en una técnica de reconocimiento de patrones (Perceptrón Multicapa) que diagnostica y calcula la probabilidad de decisiones diagnósticas correctas. Los resultados obtenidos muestran que: a) el software ProDiMES es una herramienta fácil y conveniente para evaluar métodos de diagnóstico de turbinas de gas, b) la prueba de modelos de referencia es un paso clave porque permite reducir los errores que influyen negativamente al proceso de diagnóstico y c) el algoritmo realiza correctamente la tarea de reconocimiento de fallas.

Descriptores: diagnóstico de turbinas de gas, modelo de referencia, reconocimiento de fallas, ProDiMES, perceptrón multicapa.

INTRODUCTION

Gas turbines are very sophisticated and costly systems that have been used in the past decades for different industrial applications due to the capacity to produce great amount of energy and because of their high efficiencies (Boyce, 2006). Since faults and gradual deterioration affect extremely the reliability and maintenance costs, gas turbines need advanced condition based maintenance and condition monitoring systems to ensure a correct operation (Rao, 1996). The success of these systems depends on the enhancement of monitoring software, the degree to which engine critical elements are covered and the accuracy of diagnostic decisions. Nowadays, different gas turbine diagnostic algorithms are developed to identify as accurate as possible faults conditions of major components (e.g., compressor, combustion chamber and turbine). Gradual deterioration and abrupt faults (Meher *et al.*, 2001) and sensor faults (Kobayashi and Simon, 2008) can be detected as well. This is achieved by measuring principal gas path parameters such as pressure, temperature, rotation speed, fuel rate, etc.

In order to contribute to the diagnostic process improvement, the aim of this paper is to perform a gas turbine diagnostic algorithm testing. In previous works (Cisneros *et al.*, 2015; Felipe *et al.*, 2015), only preliminary results were obtained and some algorithms were tested separately. However, in gas turbine monitoring systems, an integrated approach is necessary to have more exact and reliable results. For this reason, the present investigation proposes a methodology comprising three unified stages. In the first stage, the gas turbine data is obtained using the software ProDiMES which works with no-fault conditions, faults and degradation mechanisms. This software is used because not only simulates a complete turbofan engine fleet based on a

high-fidelity thermodynamic model but also allows evaluating gas turbine diagnostic algorithms (Simon *et al.*, 2008 and 2013). To simulate real behavior for each engine, the software assigns unique deterioration profiles, noise levels and operation modes.

In the second stage, a baseline model testing is carried out to improve the healthy engine performance approximation. Besides, baseline models allow computing deviations, which are indicators of an engine's state (Loboda *et al.*, 2004). Three variations are proposed using simulated data with no-fault scenarios through ProDiMES and the model with the lowest total error is selected for the fault recognition stage. A baseline model can be developed based on a thermodynamic model or artificial neural networks (Loboda and Feldshteyn, 2010). The first option needs complex algorithms while the second one requires considerable execution time for training. Consequently, the present paper uses a polynomial function to ease the diagnostic algorithm. The least squares method is employed to calculate model coefficients determining the baseline model.

In the third stage, the baseline model selected is used as a basis to perform the fault recognition. With the intention of extracting the diagnosis information from raw data, it is necessary to use deviations calculated using actual measurements generated by simulating fault scenarios and healthy engine values. Three steps are proposed to compute deviations:

- 1) Initial deviation computation using a general model,
- 2) Creation of individual models and
- 3) Final deviation computation using individual models.

With these final deviations, normalized vectors (also called patterns), can be obtained to form a diagnostic space and perform the fault recognition. Gas turbine fault diagnostics, particularly the gas path fault identification, is

based on pattern recognition techniques such as Radial Basis Network (Loboda *et al.*, 2010), Probabilistic Neural Network (Tsalavoutas *et al.*, 2000), Bayesian Network (Romessis and Mathioudakis, 2004) and Multi-Layer Perceptron (MLP) (Roemer and Kacprzyński, 2000; Volponi *et al.*, 2000; Sampath and Singh, 2004). The latter technique has been applied widely in the past years and has shown that it is not inferior to other methods (Loboda *et al.*, 2010). Therefore, this work uses the MLP to calculate the probability of correct diagnosis. The methodology utilized is implemented in Matlab. Its neural network toolbox (Beale *et al.*, 2014) assists in an efficient development of the algorithm.

The paper is organized as follows. Section 2 gives the description of the methodology proposed. Section 3 describes the pattern recognition method used. Section 4 presents the results obtained for the baseline model testing and the gas turbine fault recognition.

METHODOLOGY PROPOSED FOR GAS TURBINE DIAGNOSTICS

The methodology proposed (Figure 1) includes the following main stages:

- 1) Gas turbine data simulation using the software ProDiMES (Propulsion Diagnostic Method Evaluation Strategy),
- 2) Baseline model testing to enhance the healthy engine performance approximation and
- 3) Gas turbine fault recognition and the calculation of diagnosis accuracy. The stages are described in the below subsections.

GAS TURBINE DATA SIMULATION USING THE SOFTWARE PRODiMES

Most of the real gas turbine faults are not so severe or rarely occur in practice to be sufficient for a complete fault description. Also, physical experimentation is not a feasible option because of the very high costs. Instead, mathematical models are applied (Saravanamuttoo and

MacIsaac, 1983). This work uses the software ProDiMES as a tool to simulate a fleet of engines and study the algorithms for gas turbine fault diagnostics. ProDiMES was developed by the NASA Glenn Research Center originally intended for benchmarking international gas path diagnostic methods (Simon *et al.*, 2008, 2013).

ProDiMES includes an Engine Fleet Simulation (EFS) that generates simulated measurement parameter histories for each engine of the fleet. To simulate real engine behavior, it works with a deterioration profile, noise level and operation mode (takeoff and cruise) unique for each engine. The EFS is implemented in Matlab and consists of a Graphical User Interface (GUI), a Case Generator and a C-MAPPS Steady State engine model (Simon, 2010). Each element is described below.

GRAPHICAL USER INTERFACE (GUI)

In the Graphical User Interface (GUI), the user can control the type and the number of faults occurring in the engine fleet. The maximal number of possible simulated faults is 18 plus a no-fault case. Table 1 shows these fault cases and their magnitudes, within which the faults are distributed uniformly. Module faults (ID 1-5) corresponding to Fan, LPC, HPC, HPT and LPT are simulated by adjusting at the same time efficiency and flow capacity parameters. Actuator faults (ID 6-7) corresponding to VSV and VBV result from a mis-scheduling between the commanded and current actuator position. There are also 11 different sensor faults (ID 8-18) whose magnitudes are in units of average measurement noise standard deviation σ . It is important to mention that in EFS, each individual engine only experiences a single fault type. Other aspects to consider are: the number of flights to generate the output data (the maximal number is 5000 flights per engine); the fault evolution type (abrupt or rapid); the flight of fault initiation (the 11th flight is the lowest value) and the sensor noise (on or off).

The EFS works with eleven sensed variables. The seven measured variables shown in Table 2 are available

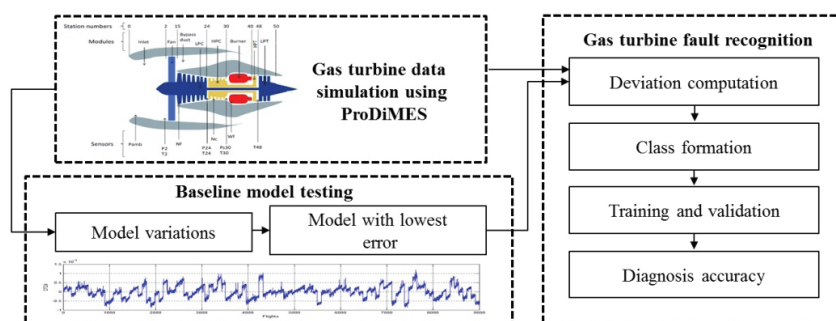


Figure 1. Methodology proposed for gas turbine diagnostics

for monitoring and are represented by a vector \vec{Y} . Table 3 shows four measured variables used as operating conditions represented by a vector \vec{U} . All the variables correspond to an engine standard measurement system.

CASE GENERATOR

The case generator produces parameter histories after the user has selected the number and type of faults in the GUI stage. One important characteristic is the random generation of unique faults, degradation profiles and operating history for each engine in the fleet. This also includes the following assignments: the date when the collection of the engine data starts, the city pairs for the takeoffs of the engine, the ambient pressure, the atmospheric temperature, the Mach number and power setting parameters for takeoff and cruise. The level and rate of gradual performance deterioration for each engine are also considered by the Case Generator. They emulate the degradation that an aircraft engine experiences during its lifetime due to different effects such as fouling, erosion, and corrosion of blades and vanes. The gradual deterioration is not considered a fault and its development is much slower than the produced by faults.

Table 1. Simulated fault cases (Simon, 2010)

| ID | Fault description | Fault magnitude |
|----|-------------------|-----------------------|
| 0 | No-fault | --- |
| 1 | Fan fault | 1 a 7% |
| 2 | LPC fault | 1 a 7% |
| 3 | HPC fault | 1 a 7% |
| 4 | HPT fault | 1 a 7% |
| 5 | LPT fault | 1 a 7% |
| 6 | VSV fault | 1 a 7% |
| 7 | VBV fault | 1 a 19% |
| 8 | Nf sensor fault | ± 1 a 10σ |
| 9 | Nc sensor fault | ± 1 a 10σ |
| 10 | P24 sensor fault | ± 1 a 10σ |
| 11 | Ps30 sensor fault | ± 1 a 10σ |
| 12 | T24 sensor fault | ± 1 a 10σ |
| 13 | T30 sensor fault | ± 1 a 10σ |
| 14 | T48 sensor fault | ± 1 a 10σ |
| 15 | Wf sensor fault | ± 1 a 10σ |
| 16 | P2 sensor fault | ± 1 a 10σ |
| 17 | T2 sensor fault | ± 1 a 10σ |
| 18 | Pamb sensor fault | ± 1 a 19σ |

*LPC=Low Pressure Compressor, HPC=High Pressure Compressor, HPT=High Pressure Turbine, LPT=Low Pressure Turbine, VSV= Variable Stator Vane, VBV=Variable Bleed Valve

Table 2. Monitored variables (Simon, 2010)

| ID | Variable | Symbol |
|----|---------------------------------|--------|
| 1 | Physical core speed | Nc |
| 2 | Total pressure at LPC outlet | P24 |
| 3 | Static pressure at HPC outlet | Ps30 |
| 4 | Total temperature at LPC outlet | T24 |
| 5 | Total temperature at HPC outlet | T30 |
| 6 | Total temperature at HPT outlet | T48 |
| 7 | Fuel flow | Wf |

Table 3. Operating conditions (Simon, 2010)

| ID | Variable | Symbol |
|----|--------------------------------|--------|
| 1 | Physical fan speed | Nf |
| 2 | Total pressure at fan inlet | P2 |
| 3 | Total temperature at fan inlet | T2 |
| 4 | Ambient pressure | Pamb |

C-MAPSS STEADY-STATE ENGINE MODEL

The *Commercial Modular Aero-Propulsion System Simulation* (C-MAPSS) Steady State is a high-bypass turbofan engine model created for diagnostics research (Simon, 2010). This model is run inside the EFS and receives the outputs from the Case Generator to produce the simulated measurement parameter histories for each engine, at takeoff and cruise of each flight. C-MAPSS works with two spool speeds (fan and core speed). Figure 2 shows the station numbers, the modules and the simulated sensor variables of the C-MAPSS Steady-State model.

BASLINE MODEL TESTING

In order to know the current gas turbine condition by means of measured gas path variables, it is necessary to describe correctly its healthy state. According to Loboda *et al.* (2004), a good approximation of healthy engine performance, also called baseline model, can be given by complete second order polynomials. Also, polynomials have shown to be better than other techniques (Loboda and Feldshteyn, 2010). Considering one monitored gas path variable as function of four operating condition arguments, the baseline model can be expressed as

$$Y_0(\vec{U}) = a_1 + a_2 u_1 + a_3 u_2 + a_4 u_3 + a_5 u_4 + a_6 u_1 u_2 + a_7 u_1 u_3 + a_8 u_1 u_4 + a_9 u_2 u_3 + a_{10} u_2 u_4 + a_{11} u_3 u_4 + a_{12} u_1^2 + a_{13} u_2^2 + a_{14} u_3^2 + a_{15} u_4^2 \quad (1)$$

where a_1, \dots, a_{15} are the model coefficients calculated using the least squares method for all monitored variables.

(1)

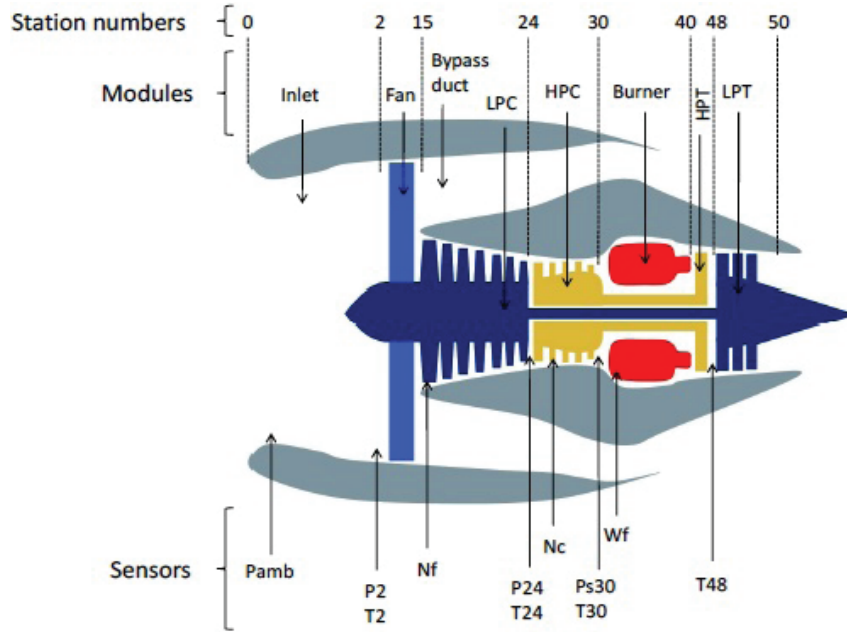


Figure 2. C-MAPSS Steady-State station numbers, modules and sensors (Simon, 2010)

To find an adequate baseline model, three model variations are proposed by simulating no-fault cases (healthy engine scenarios) through EFS for cruise operation point. These variations are specified in Table 4 containing their principal characteristics and are briefly described as follows. Variation 1 is created using 100 engines and 5000 flights per engine, however, only the first 90 flights are taken into consideration to form the model because the deterioration is not so great in that interval. Variation 2 works with 1 engine and 5000 flights. Variation 3 simulates 300 engines and 5000 flights per engine but, only 270 flights are considered for model creation. For all variations, the number of model coefficients is $k=15$ and sensor noise is not considered. The criterion to select the best variation is based on the lowest total model error. First, an error for one monitored variable δY is calculated as

$$\delta Y = \frac{Y^* - Y_0}{Y_0} \quad (2)$$

where Y^* and Y_0 are measured and baseline model values respectively. Then, the total error \bar{e} for each variation is obtained by using the Root Mean Square (RMS) of N ($m \times 1$)-vectors $\delta \bar{Y}$ produced by all engines and flights considered for model (see Table 4)

$$\bar{e} = \sqrt{\frac{1}{N} \sum_{i=1}^N \delta \bar{Y}_i^2} \quad (3)$$

After finding the model variation with the lowest \bar{e} , the model coefficients are passed to the gas turbine fault recognition stage within a matrix C .

Table 4. Characteristics of baseline model variations

| Baseline model | EFS fault type | Number of engines | Total flights per engine | Flights considered for model | Number of model coefficients | Sensor noise |
|----------------|----------------|-------------------|--------------------------|------------------------------|------------------------------|--------------|
| Variation 1 | No-fault | 100 | 5000 | 90 | 15 | Off |
| Variation 2 | No-fault | 1 | 5000 | 5000 | 15 | Off |
| Variation 3 | No-fault | 300 | 5000 | 270 | 15 | Off |

GAS TURBINE FAULT RECOGNITION

DEVIATION COMPUTATION

Due to the variation of gas turbine operation conditions, absolute gas path monitored parameters change as well. Since these changes are greater than the produced by faults, the latter remain hidden. Therefore, a diagnostic process requires an important step of deviation computation to reveal deterioration and faults effects (Loboda *et al.*, 2004). Three steps are proposed to calculate deviations needed for the gas turbine diagnostic algorithm:

- 1) Initial deviation calculation using a general model,
- 2) Creation of individual models,
- 3) Final deviation calculation using individual models.

1. *Initial deviations using a general model.* After selecting the variation model with the lowest error, the $(k \times m)$ -matrix C of model coefficients is used to form the general baseline model. It can be expressed as

$$\bar{Y}\phi = \bar{\omega}^T C \quad (4)$$

where $\bar{Y}\phi$ is a $(1 \times m)$ -vector and $\bar{\omega}$ is a $(1 \times k)$ -vector of components $1, u_1, u_2, \dots, u_3^2, u_4^2$ obtained from simulated data using fault cases. An initial deviation ΔY is obtained in the same way as equation (2) computing a relative difference between measured elements Y^* and general baseline values $Y\phi$ of a monitoring variable

$$\Delta Y = \frac{Y^* - Y\phi}{Y\phi} \quad (5)$$

Using the mean of these deviation vectors for the first $n=10$ simulated flights per engine (before the fault initiation) and considering one engine, we have

$$\delta \bar{Y}\phi = \frac{\sum_{j=1}^n \Delta \bar{Y}_j}{n} \quad (6)$$

where $\delta \bar{Y}\phi$ is a correction deviation vector and j is the index for flights.

2. *Creation of individual models.* As mentioned before, the Case Generator randomly assigns a unique operating history and deterioration profile to each engine in the fleet. However, the general model (4) does not contain these individualities. For this reason, individual models are needed before calculating the final deviations. Considering one flight and one engine, an indi-

vidual model vector for all monitored variables is given by

$$\bar{Y}\rho = \bar{Y}\phi(1 + \delta \bar{Y}\phi) \quad (7)$$

where $\delta \bar{Y}\phi$ and $\bar{Y}\phi$ are vectors of correction deviations and general model values respectively.

3. *Final deviation calculation using individual models.* Using individual baseline values $Y\rho$ for a monitored variable, we obtain

$$\delta Y\rho = \frac{Y^* - Y\rho}{Y\rho} \quad (8)$$

where $\delta Y\rho$ is a final deviation. These deviations are the base of fault class formation.

FAULT CLASS FORMATION

With the intention of having a homogeneous diagnostic process, deviations (8) are normalized as follows

$$Z = \frac{\delta Y\rho}{\sigma_Y} \quad (9)$$

where σ_Y is a mean deviation error. One vector \bar{Z} of m monitored variables represents a fault pattern to be recognized and built the diagnostic space where fault class formation is conducted. Since there is a considerable variation of engine faults in gas turbine diagnostics, they are separated into a limited number of classes. Based on the pattern recognition theory, a state D can be considered as only one of present classes

$$D_1, D_2, \dots, D_q \quad (10)$$

Thus,

$$\sum_{n=1}^q P(D_n) = 1 \quad \text{and} \quad P_{n \neq l}(D_n / D_l) = 0 \quad (11)$$

In ProDiMES, each fault class is constructed from patterns with the change of only one fault parameter (singular fault class).

TRAINING AND VALIDATION

In a pattern recognition process, the data can be separated into two parts: training and validation sets. Both sets are described shortly below and summarized in Table 5. The training set Z_T unites patterns of all classes and is employed to train the method under analysis. It

is formed by simulating all the 19 fault cases available in ProDiMES for better verification of the diagnostic algorithm (18 faults + 1 no-fault case), a determined number of engines per class and flights per engine for cruise operating point. The flight fault initiation selected is 11 with the option “fixed”, this means that the first 10 flights will not experience any fault. The fault evolution rate is selected as “rapid”. Since accuracy of fault classes’ description depends on the number of simulated patterns, 100 engines per class and 50 total flights per engine are considered. However, the number of flights per engine for rapid fault evolution is 34. For this reason, 3400 patterns per class are employed. Thus, the total size of the training set is 64600 patterns (19 fault cases \times 100 engines \times 34 flights).

The validation set Z_v is created to verify that the network can generalize the fault classes correctly. It is formed in the same manner as the training set Z_T ; however, its size is ten times smaller because it works with 340 patterns per class. Therefore, the total size of the validation set is 6460 patterns (19 fault cases \times 10 engines \times 34 flights). Every pattern in the validation set belongs to a known class.

Figures 3-5 exemplify the fault class formation using patterns of Z_T in the space of two normalized deviations: Figure 3 shows class 1 (no fault) and class 4 (HPC fault); Figure 4 shows faults 2-8 including component and actuator faults; Figure 5 shows sensor faults 10-14.

DIAGNOSIS ACCURACY

The fault recognition method selected classifies each pattern of the set Z_v producing the diagnosis d_i . Comparing d_i with a known class D_n for all validation set patterns, a confusion matrix is generated. Its diagonal is formed by correct pattern classification probabilities per class. A mean number \bar{P} is obtained from these probabilities representing the total diagnosis accuracy of the gas turbine fault recognition.

FAULT RECOGNITION METHOD

The fault recognition method chosen for this work is the Multi-Layer Perceptron (MLP). The MLP is an arti-

ficial neural network intended for classification problems. It uses a back-propagation algorithm that propagates a signal for a given input vector, producing an output and adapting unknown coefficients based on the error between a target and the network output. Figure 6 shows the general structure of the MLP. The input for each hidden layer neuron is given by the sum of an input vector \vec{Z} multiplied by weights in a matrix W_1 and a bias value (from a vector \vec{b}_1). The result is passed through a differentiable transfer function f_1 (tansig) producing the neuron output (contained in a vector \vec{a}_1) within an interval of $[-1, 1]$. For the output layer neurons, the computation is repeated in the same way using the vector \vec{a}_1 as input to the layer, a matrix W_2 , a bias vector \vec{b}_2 , and a transfer function f_2 (logsig), producing the network output contained in a vector \vec{a}_2 within an interval of $[0, 1]$. The MLP is trained on known pair’s vectors: the input vector and the target vector formed by 1 and 0 representing the membership of a class. The number of input layer neurons is seven, which represent the seven monitored variables. There are nineteen output layer neurons corresponding to the nineteen fault cases. Since the variation of the hidden neuron number produces different results, it is necessary to find the optimal setting by trial and error that yields the maximal diagnosis accuracy. Another parameter to set is the number of epochs, which is the number of iterations to update weights and biases. The network training also needs some additional parameters such as the type of activation function and the back-propagation method.

DIAGNOSTIC ALGORITHM RESULTS

ERRORS OF MODEL VARIATIONS

As shown in (Cisneros *et al.*, 2015), Variations 1 and 3 contain displacements in their plots of errors δY at certain intervals due to the influence of each engine. This situation was addressed in Subsection deviation computation to correct individualities of all engines. Figure 7 shows the errors of Variation 2 for 5000 flights and one monitored variable. Here, the engine degradation effect is observable through all flights. Each engine in the fleet experiences this inevitable situation.

Table 5. Characteristics of training and validation sets

| Set | Number of classes | Engines per class | Total flights per engine | Flight of fault initiation | Fault evolution rate | Flights for fault evolution | Sensor noise |
|------------|-------------------|-------------------|--------------------------|----------------------------|----------------------|-----------------------------|--------------|
| Training | 19 | 100 | 50 | 11 | Rapid | 34 | On |
| Validation | 19 | 10 | 50 | 11 | Rapid | 34 | On |

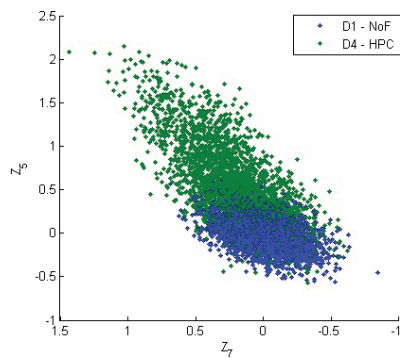


Figure 3. Class 1 (no fault) and class 4 (HPC fault)

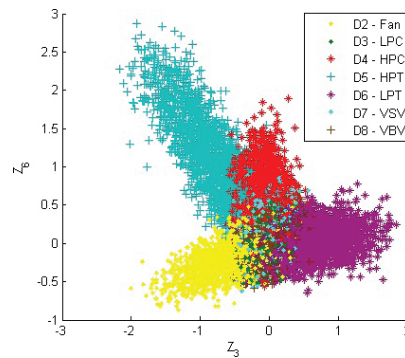


Figure 4. Component and actuator faults

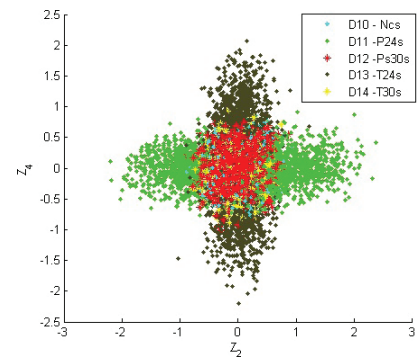


Figure 5. Sensor faults

The total error \bar{e} for each of the three variations was calculated using the RMS of the vectors $\delta\tilde{Y}$ for all engines and flights. Table 6 shows these results. Variation 2 has the greater error (a mean for all monitored variables of 0.0114). The reason of this could be because the degradation is greater using 5000 flights. Variation 1 and Variation 3 yield the lowest errors (a mean of 0.0021 and 0.0023 respectively). Since the errors remain very close, any variation of the two can be chosen. However, Variation 3 works with more data (300 engines x 270 flights) than Variation 1 (100 engines x 90 flights) resulting in

more execution time. For this reason Variation 1 is the model selected for the gas turbine diagnostic algorithm.

FAULT DIAGNOSIS ACCURACY

In order to have the highest diagnosis accuracy, two MLP parameters were tuned: the number of hidden layer neurons and the number of epochs. After performing different computations, 54 neurons and 2000 epochs produced the maximal validation probability. Figure 8 shows an example of this tuning. Figure 9 pre-

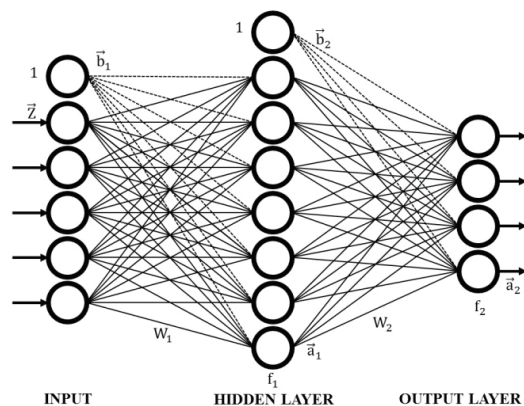


Figure 6. Structure of the MLP

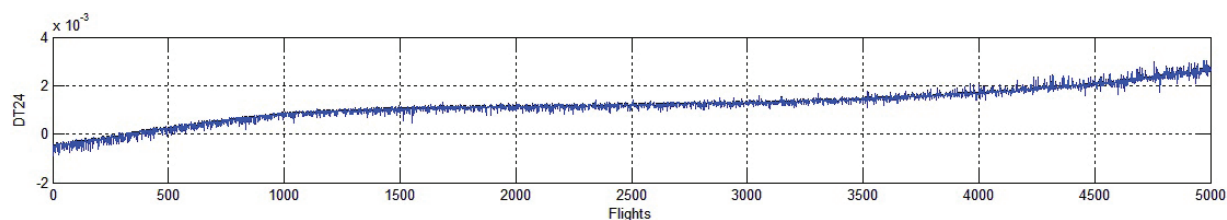

 Figure 7. Errors δY of Variation 2 (1 engine and 5000 flights)

Table 6. Total error \bar{e} for all model variations and monitored variables

| Baseline model | Y_1 | Y_2 | Y_3 | Y_4 | Y_5 | Y_6 | Y_7 | Mean |
|----------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Variation 1 | 0.0008 | 0.0010 | 0.0038 | 0.0003 | 0.0012 | 0.0024 | 0.0054 | 0.0021 |
| Variation 2 | 0.0012 | 0.0026 | 0.0276 | 0.0014 | 0.0054 | 0.0296 | 0.0123 | 0.0114 |
| Variation 3 | 0.0008 | 0.0010 | 0.0040 | 0.0004 | 0.0013 | 0.0031 | 0.0052 | 0.0023 |

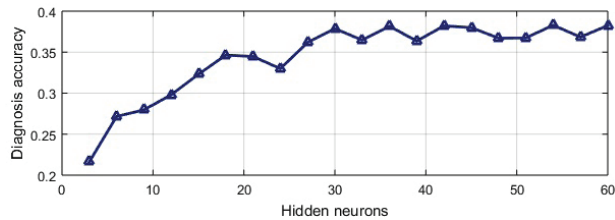


Figure 8. Optimal number of hidden neurons

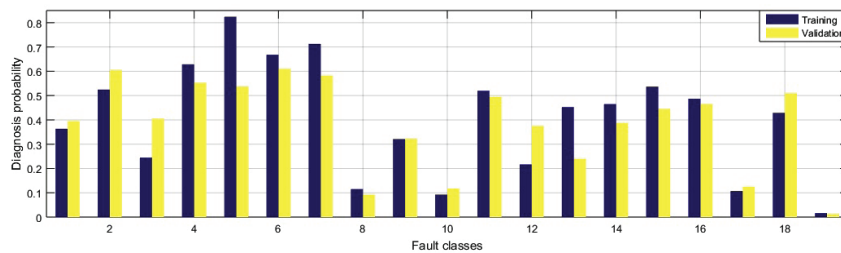


Figure 9. Diagnosis probability of each fault class

Table 7. Diagnosis probability of each fault class (training and validation)

| Fault class | Training probability | Validation probability |
|-------------|----------------------|------------------------|
| Class 1 | 0.3632 | 0.3950 |
| Class 2 | 0.5247 | 0.6050 |
| Class 3 | 0.2447 | 0.4050 |
| Class 4 | 0.6288 | 0.5525 |
| Class 5 | 0.8244 | 0.5375 |
| Class 6 | 0.6682 | 0.6100 |
| Class 7 | 0.7126 | 0.5825 |
| Class 8 | 0.1150 | 0.0925 |
| Class 9 | 0.3209 | 0.3225 |
| Class 10 | 0.0924 | 0.1175 |
| Class 11 | 0.5197 | 0.4950 |
| Class 12 | 0.2168 | 0.3750 |
| Class 13 | 0.4526 | 0.2400 |
| Class 14 | 0.4650 | 0.3875 |
| Class 15 | 0.5365 | 0.4450 |
| Class 16 | 0.4871 | 0.4650 |
| Class 17 | 0.1071 | 0.1250 |
| Class 18 | 0.4288 | 0.5100 |
| Class 19 | 0.0165 | 0.0125 |
| \bar{P} | 0.4066 | 0.3829 |

sents the diagnosis probability of each fault class for training and validation sets. Table 7 shows these values as well. For validation, the higher probabilities are obtained from classes 2, 6 and 7 (60.50 %, 61.00% and

58.25% of recognition respectively) while the lowest probabilities are produced by classes 8, 10 and 19 (9.25 %, 11.75 % and 1.25 % respectively). The differences between the probability values of both sets are ex-

plained by the limited pattern number of the validation set. The increase of this number can produce more accurate and closer results. The total diagnosis accuracy for training is 40.66% and 38.29% for validation. \bar{P}

CONCLUSIONS

The data generation using the software ProDiMES allowed the simulation of healthy and faulty conditions of an engine fleet in an appropriate environment facilitating the diagnostic process. This enabled the test of our diagnostic approach on the data of a new engine. The stage of baseline model testing permitted finding the best healthy engine performance approximation. This stage is an important part of the diagnosis process because it directly affects the diagnosis accuracy. After performing the calculations, the conclusion is that the baseline model selected has low level of errors and the deviations computed with this model adequately reflect engine health conditions. In the gas turbine fault recognition stage, the Multi-Layer Perceptron was used to classify the fault patterns.

The results showed that this network correctly performs this task; however, the diagnosis accuracy for both sets (training and validation) seems to be relatively low. Some objective reasons of this could be: the great number of fault classes and the low fault severity randomly assigned to them. The other possible explanation is that we did not take into consideration all peculiarities of the engine simulation in ProDiMES because we are not authors of this simulator.

Future works can consider working with different recognition methods such as Radial Basis Network, Probabilistic Neural Network or Support Vector Machines in order to ensure that the fault recognition process was carried out correctly. The major issue is to increase the diagnosis accuracy.

ACKNOWLEDGMENTS

The work has been carried out with the support of the National Polytechnic Institute of Mexico (IPN) and the National Council of Science and Technology (CONACYT).

NOMENCLATURE

| | | | |
|------------------------|--|-------------------------------------|--|
| C-MAPSS | Commercial Modular Aero-Propulsion System Simulation | \tilde{Y} | Vector of monitored gas path variables |
| EFS | Engine Fleet Simulation | Y_p, \tilde{Y}_p | Scalar and vector of individual baseline values |
| GUI | Graphical User Interface | Y_ϕ, \tilde{Y}_ϕ | Scalar and vector of general baseline values |
| HPC | High Pressure Compressor | Z, \tilde{Z} | Scalar and vector of normalized deviations / neuron input vector |
| HPT | High Pressure Turbine | Z_T, Z_V | Training and validation sets |
| LPC | Low Pressure Compressor | γ | Flow capacity |
| LPT | Low Pressure Turbine | $\Delta Y, \Delta \tilde{Y}$ | Scalar and vector of initial deviation values |
| MLP | Multi-Layer Perceptron | $\delta Y, \delta \tilde{Y}$ | Scalar and vector of model errors |
| VBV | Variable Bleed Valve | $\delta Y_p, \delta \tilde{Y}_p$ | Scalar and vector of final deviations |
| VSV | Variable Stator Vane | $\delta \tilde{Y}_\phi$ | Deviation correction vector |
| \vec{a}_1, \vec{a}_2 | Layer output vectors | η | Efficiency |
| \vec{b}_1, \vec{b}_2 | Vectors of bias values | σ | Standard deviation |
| C | Matrix of model coefficients | σ_y | Mean deviation error |
| d | Diagnosis | SUBSCRIPTS AND SUPERSSCRIPTS | |
| D | Fault class | 0 | Baseline value |
| \bar{e} | Total error of model variation | * | Measured value |
| f_1, f_2 | Network transfer functions | i | Index for errors |
| m | Number of monitored variables | j | Index for flights |
| \bar{P} | Mean probability of correct diagnosis | k | Index for model coefficients |
| \vec{w} | Vector of operating conditions | l, n | Fault class indices |
| W_1, W_2 | Vector of components U | q | Number of fault classes |
| | Matrices of weight coefficients | | |

REFERENCES

- Beale M.H., Hagan M.T., Demuth H.B. *Neural network toolbox user's guide R2014a*, Natick, Massachusetts, MathWorks, Inc., 2014.
- Boyce M.P. *Gas turbine engineering handbook*, 3rd ed., Oxford, Elsevier, 2006, pp. 3-56.
- Cisneros-Azuara F.J., Loboda I., Felipe-Altamirano S. Gas turbine baseline model creation through the software ProDiMES (in spanish), Proceedings of the XV National Congress of Electromechanical and Systems Engineering (CNIES 2015), Mexico, Oct. 13-17, 2015.
- Felipe-Altamirano S., Loboda I., Cisneros-Azuara F.J. Gas turbine fault recognition based on the universal software ProDiMES (in spanish), Proceedings of the XV National Congress of Electromechanical and Systems Engineering (CNIES 2015), Mexico, Oct. 13-17, 2015.
- Kobayashi T. and Simon D.L. Aircraft engine on-line diagnostics through dual-channel sensor measurements: Development of an enhanced system, Proceedings of ASME Turbo Expo 2008, ASME Paper GT2008-50346, Berlin, Germany, 2008.
- Loboda I. and Feldshteyn Y. Polynomials and neural networks for gas turbine monitoring: A comparative study, Proceedings of ASME Turbo Expo 2010, ASME Paper GT2010-23749, Glasgow, Scotland, UK, 2010.
- Loboda I., Miró-Zarate L.A., Leal-Bolaños A.E. Radial basis functions for gas turbine fault recognition, Aerospace Technic and Technology. *Journal National Aerospace University, Kharkov, Ukraine*, volume 10 (issue 77), 2010: 182-186.
- Loboda I., Yepifanov S., Feldshteyn Y. Deviation problem in gas turbine health monitoring, proceedings of IASTED International Conference on Power and Energy Systems, Clearwater Beach, Florida, USA, 2004.
- Meher-Homji C.B., Chaker M.A., Motiwala H.M. Gas turbine performance deterioration, Proceedings of the Thirtieth Turbomachinery Symposium, Houston, Texas, Texas A&M University, 2001, pp. 139-175.
- Rao B.K.N. *Handbook of Condition Monitoring*, Oxford, Elsevier, 1996, pp. 285-321.
- Roemer M.J. and Kacprzynski G.J. Advanced diagnostics and prognostics technologies for gas turbine engine risk assessment, Proceedings of ASME Turbo Expo 2000, ASME paper GT2000-0030, Munich, Germany, 2000.
- Romessis C. and Mathioudakis K. Bayesian network approach for gas path fault diagnosis, Proceedings of ASME Turbo Expo 2004, ASME Paper GT2004-53801, Vienna, Austria, 2004.
- Sampath S. and Singh R. An integrated fault diagnostics model using genetic algorithm and neural networks. *ASME Journal of Engineering for Gas Turbines and Power*, volume 128 (issue 1), 2004: 49-56.
- Saravanamuttoo H.I.H. and MacIsaac B.D. Thermodynamic models for pipeline gas turbine diagnostics. *ASME Journal of Engineering for Power*, volume 105 (issue 4), 1983, pp. 875-884.
- Simon D.L., Bird J., Davison C., Volponi A., Iverson R.E. Benchmarking gas path diagnostic methods: a public approach, Proceedings of ASME Turbo Expo 2008, ASME Paper GT2008-51360, Berlin, Germany, 2008.
- Simon D.L., Borguet S., Léonard O., Zhang X.F. Aircraft engine gas path diagnostic method: public benchmarking results, Proceedings of ASME Turbo Expo 2013, ASME Paper GT2013-95077, San Antonio, Texas, USA, 2013.
- Simon D.L. Propulsion diagnostic method evaluation strategy (ProDiMES) user's guide, NASA/TM-2010-215840, 2010.
- Tsalavoutas A., Mathioudakis K., Stamatidis A., Smith M. Identifying faults in the variable geometry system of a gas turbine compressor, Proceedings of ASME Turbo Expo 2000, ASME paper GT2000-0033, Munich, Germany, 2000.
- Volponi A.J., DePold H., Ganguli R., Daguang C. The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparative study, Proceedings of ASME Turbo Expo 2000, ASME paper GT2000-0547, Munich, Germany, 2000.

Citation for this article:**Chicago style citation**

Felipe-Altamirano, Salvador, Felipe de Jesús Cisneros-Azuara, Igor Loboda, Juan Luis Pérez-Ruiz. Gas Turbine Diagnostic Algorithm Testing Using the Software ProDiMES. *Ingeniería Investigación y Tecnología*, XVIII, 01 (2017): 75-86.

ISO 690 citation style

Felipe-Altamirano S., Cisneros-Azuara F.J., Loboda- I., Pérez-Ruiz J.L. Gas Turbine Diagnostic Algorithm Testing Using the Software ProDiMES. *Ingeniería Investigación y Tecnología*, volume XVIII (issue 1), January-March 2017: 75-86.

ABOUT THE AUTHORS

Salvador Felipe-Altamirano. Received his B.S. degree in mechanical engineering from the National Autonomous University of Mexico in 2013. He is currently enrolled in the M.S. program in energy systems engineering at the National Polytechnic Institute. His main research interest is in the field of gas turbine diagnostics.

Felipe de Jesús Cisneros-Azuara. Received his B.S. degree in mechanical engineering from the Technological Institute of Ciudad Madero in 2010. He is currently enrolled in the M.S. program in energy systems engineering at the National Polytechnic Institute. His main research interest is in the field of gas turbine diagnostics.

Igor Loboda. Received the M.S. and Ph.D. degrees in aircraft engine engineering from the Kharkov Aviation Institute (Ukraine) in 1979 and 1994, respectively. He was an investigator, lecturer and assistant professor at the Kharkov Aviation Institute from 1992 to 2001. Since 2001, he has been research professor at the National Polytechnic Institute of Mexico. He is member of the SNI (National Researchers System of Mexico). His research interests are in the areas of simulation and condition monitoring of gas turbines and common theory of pattern recognition. Particular issues of interest are gas turbine thermodynamic models, model identification and fault identification techniques.

Juan Luis Pérez-Ruiz. Received his B.S. degree in mechanical engineering from the Technological Institute of Tuxtla Gutierrez in 2012 and his M.S. degree in energy systems engineering from the National Polytechnic Institute in 2015. He is currently enrolled in the Ph.D. program in mechanical engineering at the National Polytechnic Institute. His research interests are in the areas of gas turbine diagnostics, gas turbine performance simulation and pattern recognition.