

# FAULT DIAGNOSIS OF AN INDUSTRIAL GAS TURBINE USING NEURO-FUZZY METHODS

Vasile Palade<sup>1</sup>, Ron J. Patton<sup>1</sup>, Faisal J. Uppal<sup>1</sup>, Joseba Quevedo<sup>2</sup>, S. Daley<sup>3</sup>

1. *The University of Hull, Control and Intelligent Systems Engineering,  
Cottingham Road, HU6 7RX, Kingston upon Hull, United Kingdom,  
E-mai: {V.Palade,R.J.Patton,F.J.Uppal}hull.ac.uk*

2. *Automatic Control Department, Universidad Politécnic de Catalunya,  
Campus de Terrassa, Rambla Sant Nebridi, 10. 08222 Terrassa, Spain,  
E-mail: jquevedo@esaii.upc.es*

3. *ABB-Alstom Power, Cambridge Road, Whetstone, Leicester, LE8 6LH,  
Phone: +44 (0) 116 2015638, E-mail: {Steve.Daley}@energy.alstom.com*

Abstract: The paper focuses on the application of neuro-fuzzy techniques in fault detection and isolation. The objective of this paper is to detect and isolate faults to an industrial gas turbine, with emphasis on faults occurred in the actuator part of the gas turbine. A neuro-fuzzy based learning and adaptation of TSK fuzzy models is used for residual generation, while for residual evaluation a neuro-fuzzy classifier for Mamdani models is used. The paper is concerned on how to obtain an interpretable fault classifier as well as interpretable models for residual generation. *Copyright © 2002 IFAC*

Keywords: fault diagnosis, neural networks, fuzzy models, neuro-fuzzy, fault detection.

## 1. INTRODUCTION

In the last ten years, the field of diagnosis has attracted the attention of many researchers, both from the technical area as well as medical area. In the industrial field there is also an increasing need for safety, which conducted to the development of various techniques for an automatic diagnosis of faults. Generally, in an industrial control system a fault may occur in the process components, in the control loop (controller and actuators) and in the measurement sensors for the input and output variables. The conceptual diagram for a fault diagnosis system is depicted in Figure 1. The diagnosis consists of two sequential steps: residual generation and residual evaluation. In the first step a number of residual signals are generated in order to determine the state of the process. The objective of fault isolation is to determine if a fault has occurred and also the location of the fault, by analysing the residual vector.

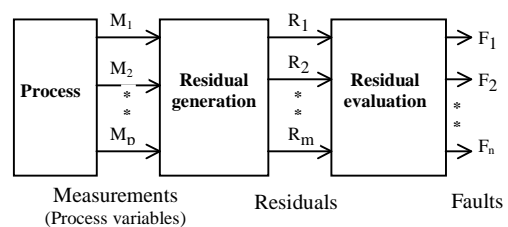


Fig. 1. The general structure of a diagnosis system

The problem of detecting and isolating faults to an industrial gas turbine was studied in other previous papers in the literature (Patton et al., 1999; Patton and Simani, 1999; Simani and Spina, 1998) using mainly observer based techniques. In this paper we investigate the problem of fault diagnosis of an industrial gas turbine using a neuro-fuzzy approach. For simulating purposes we used a SIMULINK model of such an industrial gas turbine, developed at ABB-Alstom Power, United Kingdom.

The structure of the paper is the following. Section 2 presents an overview on the use of neuro-fuzzy techniques in fault detection and isolation. Section 3 describes how residuals are generated using a TSK neuro-fuzzy based adaptation and learning technique. Section 4 is concerned on the development of a transparent fault classifier using neuro-fuzzy networks for Mamdani fuzzy models, in order to describe the task of fault classification. The paper ends with some conclusions and remarks.

## 2. NEURO-FUZZY IN FDI

Many authors have focussed on the use of neural networks in FDI applications (Marcu et al., 1999; Korbicz et al., 1999) for solving the specific tasks in FDI, such as fault isolation but mainly fault detection. Other authors (Koscielny et al., 1999) used fuzzy logic for fault diagnosis, especially for fault isolation, but some of them even for fault detection, using for example TSK fuzzy models. In the last few years there is also an increasing number of authors (Patton et al., 1999; Calado and Sa da Costa, 1999) who try to integrate neural networks and fuzzy logic in order to benefit of the advantages of both techniques for fault diagnosis applications.

Neural networks have been successfully applied to fault diagnosis problems due to their capabilities to cope with non-linearity, complexity, uncertainty, noisy or corrupted data. Neural networks are very good modelling tools for highly non-linear processes. Generally, it is easier to develop a non-linear neural network based model for a range of operating than to develop many linear models, each one for a particular operating point. Due to these modelling abilities, neural networks are ideal tools for generating residuals. Neural networks can also be seen as universal approximators. An usual 3 layered MLP neural network, with  $m$  inputs and  $n$  outputs, can approximate any non-linear mapping from  $R^m$  to  $R^n$  using an appropriate number of neurons in the hidden layer. Due to this approximation and classification ability, neural networks can also be successfully used for fault evaluation. The drawback of using neural networks for classification of faults is their lack of transparency in human understandable terms. Fuzzy techniques are more appropriate for fault isolation as it allows the integration in a natural way of human operator knowledge into the fault diagnosis process. The formulation of the decisions taken for fault isolation is done in a human understandable way such as linguistic rules.

The main drawback of neural networks is represented by their "black box" nature, while the disadvantage of fuzzy systems is represented by the difficult and time-consuming process of knowledge acquisition. On the other hand the advantage of neural network over fuzzy systems is learning and adaptation capabilities, while the advantage of fuzzy system is the human understandable form of knowledge representation. Neural networks use an implicit way of knowledge representation while

fuzzy and neuro-fuzzy systems represent knowledge in an explicit form, such as rules.

### 2.1 Methods of Neuro-Fuzzy Integration

The combination of neural networks and fuzzy systems can be done in two main ways:

- a) *Neural networks are the basic methodology and fuzzy logic is the second.* These hybrid systems are mainly neural networks, but the neural networks are equipped with abilities of processing fuzzy information. The systems are usually termed Fuzzy Neural Networks and they are networks where the inputs and/or the outputs and/or the weights are fuzzy sets, and they usually consist of a special type of neurons, called fuzzy neurons. Fuzzy neurons are neurons with inputs and/or outputs and weights represented by fuzzy sets, the operation performed by the neuron being a fuzzy operation.
- b) *Fuzzy logic is the basic methodology and neural networks the subsequent.* These systems can be viewed as fuzzy systems augmented with neural network facilities, such as learning, adaptation, and parallelism. These systems are usually called Neuro-Fuzzy Systems. Most authors in the field of neuro-fuzzy computation understand neuro-fuzzy systems as a special way to learn fuzzy systems from data using neural network type learning algorithms. Some authors (Shann and Fu, 1995) in the field term these neuro-fuzzy systems also fuzzy neural networks, but most of them like to term them as Neuro-Fuzzy Systems. Neuro-Fuzzy Systems (D. Nauck and R. Kruse, 2000) can be always interpreted as a set of fuzzy rules and can be represented as a feed-forward network architecture.

These two previous ways of neuro-fuzzy combination can be viewed as a type of fusion systems, as it is difficult to see a clear separation between the two methodologies. One methodology is fused into the other methodology, and it is assumed that one technique is the basic technique and the other is fused into it and augments the capabilities of information processing of the first methodology. Besides these fusion neuro-fuzzy systems, there is another way of hybridisation of neural networks and fuzzy systems, where each methodology maintains its own identity and the hybrid neuro-fuzzy system consists of modules structure which cooperate in solving the problem. These kind of neuro-fuzzy systems are called combination hybrid systems. The neural network based modules can work in parallel or serial configuration with fuzzy logic based modules and augments each other. In some approaches, a neural network (such as a self-organising map) can pre-process input data for a fuzzy system, performing for example data clustering or filtering noise. But, especially in FDI applications, many authors use a fuzzy system as a pre-processor for a neural network. In (Alexandru et al., 2000) the residuals signals are

fuzzified first and then fed into a recurrent neural network for evaluation, in order to perform fault isolation.

The most often used NF systems are fusion NF systems and the most common understanding for a Neuro-Fuzzy system is the following. A NF system is a neural network which is topologically equivalent to the structure of a fuzzy system. The network inputs/outputs and weights are real numbers, but the network nodes implement operations specific to fuzzy systems: fuzzification, fuzzy operators (conjunction, disjunction), defuzzification. In other words, a NF system can be viewed as a fuzzy system, with its operations implemented in a parallel manner by a neural network, and that's why it is easy to establish a one-to-one correspondence between the NN and the equivalent FS. Neuro-Fuzzy systems can be used to identify fuzzy models directly from input-output relationships, but they can be also used to optimize (refine/tune) an initial fuzzy model acquired from human expert, using additional data.

## 2.2 Neuro-Fuzzy Networks

In the area of neuro-fuzzy systems there are two principal types of neuro-fuzzy networks preferred by most of the authors in the field of neuro-fuzzy integration. In section 3 and 4, we will use kind of these structures, shortly presented below, for residual generation and for fault classification. The most common neuro-fuzzy network is used to develop or adjust a fuzzy model in Mamdani form given by relation (1), using input – output data. The network is a five layers network as shown in Figure 2. A Mamdani fuzzy model consists of a set of fuzzy if-then rules in the following form:

$$\text{If } x_1 \text{ is } X_{i_1} \text{ and } x_2 \text{ is } X_{i_2} \text{ and } \dots \text{ and } x_n \text{ is } X_{i_n} \\ \text{then } y \text{ is } Y_j \quad (1)$$

where:  $x_1, x_2, \dots, x_n$  are the system inputs,  $y$  is the output,  $X_{k i_k}$  with  $k=1, 2, \dots, n$  and  $i_k=1, 2, \dots, l_k$  are the linguistic values of the linguistic variable  $x_k$ , and  $Y_j, j=1, 2, \dots, l_y$  are the linguistic values of the output. Every linguistic variable  $x_k$  is described by  $l_k$  linguistic values  $X_{k1}, X_{k2}, \dots, X_{k l_k}$ .

Layer 1 is the input layer and each node corresponds to each input variable. Layer 2 is called membership function layer, the nodes from this layer mapping each input  $x_i$  to every membership function  $X_{ij}$  of the linguistic values of that input. It is possible to use, in the layer 2, a subnet of nodes to implement a desired membership function, instead of a single node. Each node in the layer 3 (called rule layer) performs the precondition matching – the IF part – of a fuzzy rule. The nodes from layer 4 combine the fuzzy rules with the same consequent, each node implementing a fuzzy OR operator, such as fuzzy max operator. Each node in the layer 5 corresponds to an output variable and acts as a defuzzifier. The integration and the activation functions of nodes for such a network are chosen (Shann and Fu 1995) so that to perform

specific operations in a fuzzy inference engine as described before.

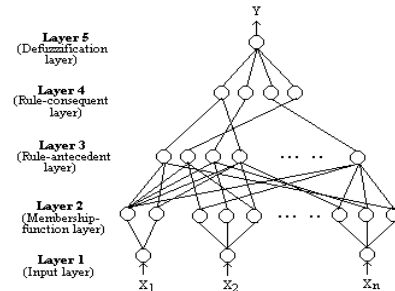


Fig. 2. The general structure of a neuro-fuzzy network for Mamdani models

Another major class of neuro-fuzzy networks are the neuro-fuzzy networks used to develop and adjust a Sugeno-type fuzzy model. The structure of such a neuro-fuzzy network is shown in figure 3. The first 3 layers are the same with those in a neuro-fuzzy network for Mamdani models. In the rule layer, it can be used the traditional fuzzy min operator, but many authors prefer to use a product operator as fuzzy intersection operator. Usually, all weights of this layer are set to 1. If some prior knowledge on process functioning is available, it can be established the number of nodes in layer 3 (the number of rules or fuzzy partition regions) and the corresponding links between layer 2 and 3.

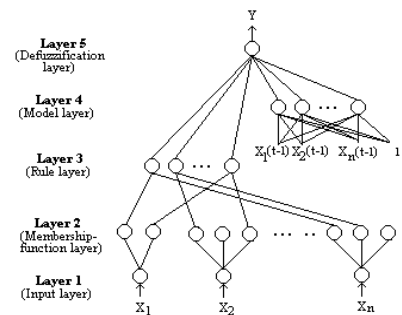


Fig. 3. Neuro-Fuzzy network for TSK fuzzy model implementation

In (Zhang and Morris 1996) the authors developed a neuro-fuzzy network for process modelling and fault diagnosis. The main shortcoming of this structure is that the user must partition the process operation into several fuzzy operating regions before training the fuzzy neural network. The partitioning is made empirically, looking to the process functioning, and it may be a very difficult task when the process has a complex nature. Different clustering techniques as well as genetic algorithms can be used to find the best fuzzy partition of the input space. Layer 4 is called the model layer, and each node implements a linear model corresponding to a rule node in the rule layer, respectively to a fuzzy operating region. The weights of a node are the parameters of the linear model and the inputs of the node are the past system inputs and outputs. Layer 5 consists of a single node, which performs the defuzzification. The most general Sugeno-type neuro-fuzzy network structure, is a network which implements a set of fuzzy rules with

ARMA models of higher order in the consequence part of the rules. The rules are in the following form:

**R<sub>k</sub>:** **If**  $x_1$  is  $X_{1i_1}$  **and**  $x_2$  is  $X_{2i_2}$  **and** ...  $x_n$  is  $X_{ni_n}$  **then**

$$y_k(t) = a_{0k} + \sum_{j=1}^{n_1} a_{jk}x(t-j) + \sum_{j=1}^{n_2} b_{jk}y(t-j) \quad (3)$$

where  $k=1,2, \dots, m$ ,  $m$  the number of rules, and  $x=(x_1, x_2, \dots, x_n)$  is the input vector, and  $a_{jk} = (a_{jk}^1, \dots, a_{jk}^n)$ .

When linear ARMA models of higher order are used, every node from layer 4 must be replaced by a subnet, which implements the ARMA model of the desired order. In figure 4, it is shown the subnet which corresponds to node  $k$  from layer 4, when  $n_1=n_2=2$ . The inputs of the subnet  $k$  from layer 4 are the previous inputs and outputs of the system.

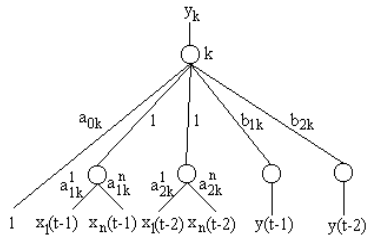


Fig. 4. The subnet corresponding to node  $k$  in layer 4

### 3. RESIDUAL GENERATION USING NEURO-FUZZY MODELS

The purpose of this paper is to detect and isolate mainly actuator faults, but also other types of faults such as components or sensor faults, occurred in an industrial gas turbine. In our turbine, air flows via an inlet duct to the compressor and the high pressure air from the compressor is heated in combustion chambers and expands through a single stage compressor turbine. A Butterfly valve provides a means of generating a back pressure on the compressor turbine (there is no power turbine present in the model). Cooling air is bled from the compressor outlet to cool the turbine stator and rotor. A Governor regulates the combustor fuel flow to maintain the compressor speed at a set-point value. For simulation purposes we used a Simulink prototype model of such an industrial gas turbine, presented in (Patton and Simani, 1999) and developed at ABB-Alstom Power, United Kingdom. The SIMULINK prototype simulates the real measurements taken from the gas turbine with a sampling rate of 0.08s. The model has two inputs and 28 output measurements which can be used for generating residuals. The Simulink model where validated in steady state conditions against the real measurements and all the model variables were found to be within 5% accuracy. All the neuro-fuzzy models we will develop later in the section for generating residuals purposes are driven by two inputs: valve angle ( $va$ ) and the fuel flow ( $ff$ ) which is also a control variable.

One common fault in the gas turbine is the fuel actuator friction wear fault. Other faults considered in our work were compressor contamination fault, thermocouple sensor fault and high-pressure turbine seal damage. Usually these faults in the industrial gas turbine develop slowly over of a long period of time. We will try to detect the actuator fault and to isolate it from other faults in the turbine. For simplicity, we present in the following only the results for the first two faults - fuel actuator friction wear fault and compressor contamination fault.

The residual signals are calculated as difference between estimated signal given by observer and the actual value of the signal. The residuals are generated using TSK neuro-fuzzy networks. We are concerned on how to find accurate neuro-fuzzy models for generating residuals, but we tried to have as much as possible a certain degree of transparency of the models. That's why we had to find a good structure of the model and therefore a good partition of the input space, using clustering for that. There is a compromise between the interpretability and the precision of the model.

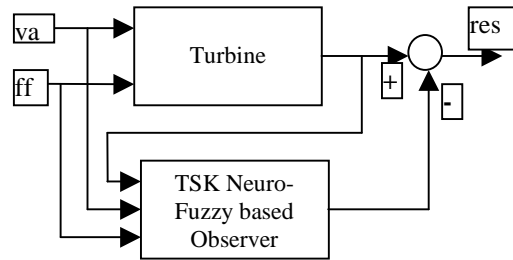
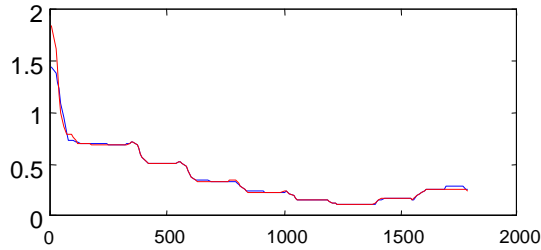


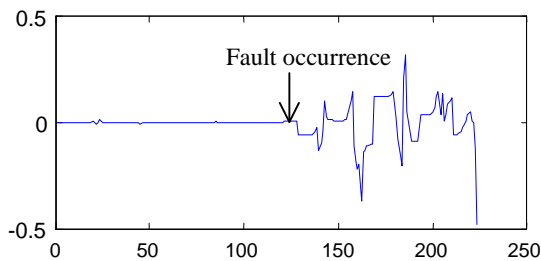
Fig. 5. Neuro-fuzzy based observer scheme for generating residuals.

First we developed a 3 inputs and one output NF-TSK network for the output measurement which is most affected by the actuator fault ( $ao$ ). We used, as input of the network, the present value for valve angle ( $va$ ) and fuel flow ( $ff$ ), and the previous value of the output affected by the fault. Three linguistic values were used for each input variable and grid partition of the input space. The performance of the model is shown in Figure 6a, the generated residual in Figure 6b, and the difference between the system output and the model output in Figure 6c. Unfortunately, due to the control loop action, this kind of fault can not be seen in steady state regime and can not be described as a ramp function in order to show what's happened in case of a gradual developing fault. But this actuator fault can be seen in dynamic regime, for different values of the valve angle. For isolating purposes we take the absolute value of this residual signal and the pass through a filter for obtaining a persistent residual signal. In order to see a gradually developing fault, we built a NF based observer for the output most affected by the compressor contamination fault. In a similar way we used also a network with 3 inputs and 3 membership functions per input and first order linear models in the consequent of the rules. The output ( $co$ ) most sensitive to a compressor ramp fault is depicted in Figure 7a and the residual generated in

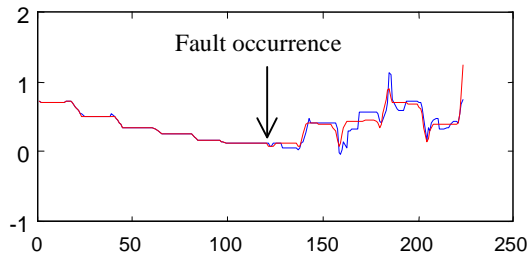
Figure 7b. The compressor fault affects also the output ao - most affected output by the actuator fault - but not in the same magnitude as the output co is affected. Then, the residual designed for output ao is sensitive to both faults.



a) Performance of the model



b) Generated residual



c) The system and the model output

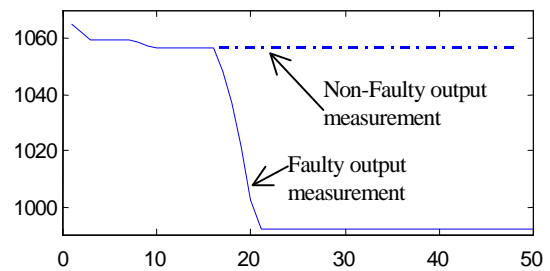
Fig. 6. Results for the fuel actuator friction wear fault

Further we developed several types of models for the residual sensitive to the actuator fault, in order to see a comparison between the accuracy and the transparency of the model. These results are summarised in Table 1. From this table, it can be seen that if you need a more transparent NF model for residual generation, you lose gradually the accuracy of the model. The first three NF models were generated using clustering methods and the following three were generated using a grid partition with 2, 3 and 4 membership functions for each input variable. The exceptional performance shown in the first case can be explained by the capabilities of Gustafson-Kessel clustering method, which can produce clusters with different shapes and orientation, and then more accurate model while keeping a reduced number of rules. On the other hand, it can be concluded that if the structure of the model is not properly chosen, then the transparency degree is reduced, but also the training time will be much increased and then more difficult to reach a desired performance of the model in a given time. Table 1 shows that there is not a very big difference

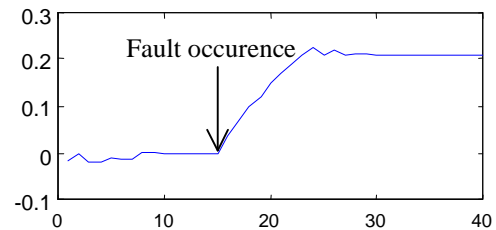
in performance between a model with 64 rules and a model with 2 rules, even if the number of the parameters were much bigger in the first case. That means the structure of the model in the first case where not appropriate and the training were not fully completed. In fact, after an appropriate and a complete training, the model with 64 rules should have overlapping membership functions and many input regions with about the same consequent, which require a post-processing of the model in order to simplify it. But this task, represented by a more difficult training and a simplification after that, is more complicate to perform than trying to predict first a right structure of the model at a desired degree of transparency and train the model after that.

Table 1

Transparency of the model	Performance of the model	Comments
2 rules	0.00180	Gustafson-Kessel
2 rules	0.007214	Substr. Clustering
3 rules	0.006087	Substr. Clustering
8 rules (2x2x2)	0.004194	Grid partition
27 rules (3x3x3)	0.001024	Grid partition
64 rules (4x4x4)	0.000887	Grid partition
Black box	0.000001	Neural network



a) Output affected by compressor fault



b) Generated residual for a ramp fault

Fig. 7. Results for compressor contamination fault.

#### 4. NEURO-FUZZY BASED RESIDUAL EVALUATION

In the residual generation part of a diagnosis system the user should be more concerned on the accuracy of neuro-fuzzy models, even desirable to have interpretable models also for residual generation, such as TSK models. For the evaluation part it is more important the transparency or the interpretability of the fault classifier, in human understandable terms, such as classification rules. The main problem in neuro-fuzzy fault classification is how to obtain an interpretable fuzzy classifier, which should have few meaningful fuzzy rules with

few meaningful linguistic rules for input/output variables. Neuro-fuzzy network for Mamdani models are appropriate tools to evaluate residuals and perform fault isolation, as the consequence of the rules contains linguistic values which are more readable than linear models in case of using TSK fuzzy models. As in previous section, the price paid for the interpretability of the fault classifier is the loss of the precision of the classification task.

The isolation table for the two faults used in the previous section is represented by Table 2. For training the neuro-fuzzy network in order to isolate these faults, 150 patterns for each fault were used. The NF-network decisions for the residual values were assigned in relation with the known faulty behaviour. In order to obtain a readable fault classifier we used NEFLASS neuro-fuzzy classifier (D. Nauck and R. Kruse, 2000). This neuro-fuzzy system has a slightly different structure than the neuro-fuzzy network for Mamdani models presented in section 2, but it allows to the user to obtain in an interactive manner and very easy an interpretable fuzzy fault classifier at the desired level and compromise accuracy/transparency. Considering conjunctive fuzzy rules for fault classification (both residual inputs in the antecedent), Table 3 summarises the results of the study.

Table 2 Fault isolation table

	Actuator fault	Compressor fault
R1	1	1
R2	0	1

Table 3 Transparency/accuracy of NF fault classifier

Transparency(no.of rules)	2	4	8	12
Accuracy (no. of patterns correctly classified-in %)	88.7	91.2	96.4	99.6

## 5. CONCLUSION

This paper studied the problem of diagnosis faults occurred in an industrial gas turbine. Neuro-fuzzy techniques have been applied in the paper both for residual generation and for fault classification. This study demonstrates that the combination of neural networks with fuzzy systems can produce better diagnostic results, especially when there is an interest on the transparency in human understandable terms of neuro-fuzzy models. Always there is a compromise between the interpretability and the precision of the model.

## ACKNOWLEDGMENT

This research has been conducted by funding from the European Framework 5 Research Training Network on Development and Application of Methods for Actuator Diagnosis for Industrial Control Systems – DAMADICS.

## REFERENCES

- Alexandru M., C. Combastel and S. Gentil (2000). Diagnostic decision using recurrent neural network. In: Proc. of the 4<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp 410-415, Budapest.
- Calado J.M.F. and J.M.G. Sa da Costa (1999). An expert system coupled with a hierarchical structure of fuzzy neural networks for fault diagnosis. *Journal of Applied Mathematics and Computer Science*, no 3, vol. 9, pp. 667-688.
- Leonhardt S. and M. Ayoubi (1997). Methods of fault diagnosis. *Control Engineering and Practice*, vol. 5, no. 5, pp. 683-692.
- Korbicz J., K. Patan and O. Obuchowicz (1999). Dynamic neural networks for process modelling in fault detection and isolation systems. *Journal of Applied Mathematics and Computer Science*, no 3, vol. 9, pp. 519-546.
- Koscielny J.M., M. Syfert and M. Bartys (1999). Fuzzy-logic fault diagnosis of industrial process actuators. *Journal of Applied Mathematics and Computer Science*, no 3, vol. 9, pp. 637-652.
- Marcu T., L. Mirea and P.M. Frank (1999). Development of dynamic neural networks with application to observer-based fault detection and isolation. *Journal of Applied Mathematics and Computer Science*, no 3, vol. 9, pp. 547-570.
- Nauck D. and R. Kruse (1998). Nefclass - a soft computing tool to build readable fuzzy classifiers. *BT Technol. Journal*, vol. 16, no 3, pp 89-103.
- Patton R.J., C.J. Lopez-Toribio and F.J. Uppal (1999). Artificial Intelligence Approaches to fault diagnosis for dynamic systems. *Journal of Applied Mathematics and Computer Science*, no 3, vol. 9, pp. 471-518.
- Patton, R. J., S. Simani, S. Daley and A. Pike (2000). Fault diagnosis of a simulated model of an industrial gas turbine prototype using identification techniques. In: Proceedings of the 4<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp 518-523, Budapest.
- Patton, R. J. and S. Simani (1999). Identification and fault diagnosis of a simulated model of an industrial gas turbine. Technical Research Report, The University of Hull-UK, Department of Engineering.
- Shann J.J. and H.C. Fu (1995). A fuzzy neural network for rule acquiring on fuzzy control systems, *Fuzzy Sets and Systems* vol. 71, pp. 345 – 357.
- Simani, S. and P. R. Spina (1998). Kalman filtering to enhance the gas turbine control sensor fault detection. In: Proceedings of the 6<sup>th</sup> IEEE Mediterranean Conference on Control and Automation, pp. 443-450, Alghero – Italy.
- Zhang J. and J. Morris (1996). Process modeling and fault diagnosis using fuzzy neural networks, *Fuzzy Sets and Systems* vol. 79, pp. 127-140.