



ELSEVIER

Contents lists available at ScienceDirect

# Engineering Applications of Artificial Intelligence

journal homepage: [www.elsevier.com/locate/engappai](http://www.elsevier.com/locate/engappai)

## PEM fuel cell fault diagnosis via a hybrid methodology based on fuzzy and pattern recognition techniques



Antoni Escobet<sup>a</sup>, Àngela Nebot<sup>b,\*</sup>, Francisco Mugica<sup>b</sup>

<sup>a</sup> Department of DIPSE, Edifici MN<sub>1</sub>, Technical University of Catalonia (UPC), Campus Manresa, Avinguda de les Bases de Manresa, 61-73, 08240 Manresa, Spain

<sup>b</sup> Department of LSI, Technical University of Catalonia (UPC), c/Jordi Girona 1-3, 08034 Barcelona, Spain

### ARTICLE INFO

#### Article history:

Received 12 November 2013

Received in revised form

12 June 2014

Accepted 7 July 2014

#### Keywords:

Fault diagnosis system

Fuel cell system

Fuzzy Inductive Reasoning

VisualBlock-FIR

### ABSTRACT

In this work, a fault diagnosis methodology termed VisualBlock-Fuzzy Inductive Reasoning, i.e. VisualBlock-FIR, based on fuzzy and pattern recognition approaches is presented and applied to PEM fuel cell power systems. The innovation of this methodology is based on the hybridization of an artificial intelligence methodology that combines fuzzy approaches with well known pattern recognition techniques. To illustrate the potentiality of VisualBlock-FIR, a non-linear fuel cell simulator that has been proposed in the literature is employed. This simulator includes a set of five fault scenarios with some of the most frequent faults in fuel cell systems. The fault detection and identification results obtained for these scenarios are presented in this paper. It is remarkable that the proposed methodology compares favorably to the model-based methodology based on computing residuals while detecting and identifying all the proposed faults much more rapidly. Moreover, the robustness of the hybrid fault diagnosis methodology is also studied, showing good behavior even with a level of noise of 20 dB.

© 2014 Elsevier Ltd. All rights reserved.

### 1. Introduction

Polymer electrolyte membrane (PEM) fuel cells are devices that allow the direct transformation of chemical energy (hydrogen) into electrical energy (Pukrushpan et al., 2004). This energy conversion is clean because the only by-products are water and heat. Moreover, this process is very efficient and has the same level of performance than the main fossil alternatives. Amongst others, these advantages are behind the recent increase in scientific production in the field in the last years. Potential applications are grouped into three categories: generation of electricity for stationary applications; residential and electronic applications and automotive applications. Applications in the automotive sector have been particularly attractive due to the fact that there is practically null emission of polluting agents (Rajashekara, 2000). This means that fuel cells are environmentally-friendly alternative to conventional fossil fuels that significantly reduce pollution and man-made greenhouse gases.

With all, PEM fuel cells are complex and interrelated systems and, as stated in recent research efforts (Feroldi, 2009), efficiency in control is crucial. Different control problems must be solved to obtain a correct operation of the system, i.e. control of the power flows in the system to fulfill the power load, conditioning of the generated power, handling of heat and water and suitable

hydrogen and air supply. Therefore, a set of auxiliary elements such as valves, compressors, sensors, regulators, etc., are needed to guarantee that the fuel cell works in an optimal way. For this reason, fuel cell systems are vulnerable to different set of faults that can imply its temporal or permanent damage. Fault diagnosis systems (FDS) become, therefore, fundamental in order to reduce as much as possible this vulnerability.

There has been intensive research activity in the fault diagnosis of fuel cell stack systems that includes quantitative as well as qualitative approaches. According to Aitouche et al. (2012), fault diagnosis methods can be classified on two types: model-based and knowledge-based approaches. In model-based approaches a priori knowledge about the model of the process is assumed. However, this information is not always available and the dynamic fuel cell model is characterized by multiple variables and a strong coupling with profound dynamics. Model-based approaches are primarily based on statistical techniques, first order logic, control theory, mathematical modeling, and computer simulation (Aitouche et al., 2011; Escobet et al., 2009; Rosich et al., 2014; Wu et al., 2008a, 2008b; Wang et al., 2011; Zhang and Huang, 2011). Knowledge-based approaches are based on the data available from the system to perform learning. Examples of knowledge-based approaches are signal processing, experimental methods and artificial intelligence. There is a large amount of research done in the area of knowledge-based FDS for fuel cell stack systems, specially using artificial intelligence methods, like expert systems, neural networks, and genetic programming (Chavez-Ramírez et al.,

\* Corresponding author. Tel.: +34 93 4137783; fax: +34 93 4137833.

E-mail addresses: [Toni@psem.upc.edu](mailto:Toni@psem.upc.edu) (A. Escobet), [angela@lsi.upc.edu](mailto:angela@lsi.upc.edu) (À. Nebot), [fmugica@lsi.upc.edu](mailto:fmugica@lsi.upc.edu) (F. Mugica).

## Nomenclature

PEM	polymer electrolyte membrane
FDS	fault diagnosis system
ANFIS	neural-networks fuzzy infer system
FIR	Fuzzy Inductive Reasoning
KNN	$k$ -nearest neighbor
5NN	5-nearest neighbor
EFP	equal frequency partition
MSE	mean square error
LPV	linear parameter varying
Q	acceptability measure

C	partial acceptability measure
$C_{rel}$	relative confidence
$I_a$	computing the maximum number of local cumulative errors possible
$\lambda_{o_2}$	oxygen excess ratio
$o_{2in}$	oxygen input
$o_{2react}$	oxygen reacted
$I_{cm}$	compressor current
$\omega_{cm}$	compressor speed
$V_{fc}$	voltages applied to the cell
$I_{fc}$	stack current
$V_{cm}$	voltages applied to the compressor

2010; Kamal et al., 2014; Liu and Wang, 2003; Nitsche et al., 2004; Yousfi Steiner et al., 2011; Zheng et al., 2013). However in recent years, the demand has arisen to develop FDS that are more robust to uncertainty. In this context, fuzzy logic and hybrid fuzzy approaches appear to offer a good alternative to other qualitative FDS methodologies (Hissel et al., 2004, 2007; Olteanu et al., 2012; Tao et al., 2005; Vural et al., 2009; Becker and Karri, 2010). Let us look a little closer to this research that is clearly related to our interests.

Hissel et al. (2004) presented a first step in the direction of fuel cell systems diagnosis, by proposing a Sugeno-type fuzzy model for two faults, i.e. an accumulation of nitrogen and/or water in the anode compartment and an important drying of the proton exchange membrane. The Sugeno model has been tuned using a genetic algorithm. The conclusions obtained state that the fuzzy diagnosis models help to improve the results. For instance, the authors point out that thanks to the fuzzy model only 150 s are needed to detect the accumulation of nitrogen and water in the anode component fault. Hissel et al. (2007) present a new approach to PEM diagnosis based on fuzzy  $k$ -means clustering. The fuzzy clustering algorithm produces three clusters on the two dimension feature space, each one corresponding to a specific fault, i.e. a specific behavior of the fuel cell stuck. Olteanu and co-workers also proposed a Sugeno-type fuzzy model. In this case the goal was to model the polyelectrolyte membrane fuel cell by means of a set of fuzzy rules built based on physical principles. Therefore, this research follows a model-based approach and it is not focused on fault diagnosis. Tao et al. use the adaptive neural-networks fuzzy infer system (ANFIS) to build the temperature model of PEM fuel cell which is used as the reference model of the control system, and adjusts the model parameters to control it online. Vural et al. also uses the ANFIS to model the PEM fuel cell under various operational conditions. The models obtained are able to predict fuel cell performance with a high accuracy in an easy, rapid and cost effective way. Becker and Karri also build predictive ANFIS models, in this case for hydrogen flow rate, electrolyzer system-efficiency and stack-efficiency. They found that these models are reliable predictive tools with an excellent accuracy. Notice that neither Tao et al. nor Vural et al. nor Becker and Karri research are focused on fault diagnosis.

Therefore, the number of works that can be found in the literature that propose fuzzy or hybrid fuzzy FDS for PEM fuel cells is quite low. This is the reason why we propose a FDS based on the Fuzzy Inductive Reasoning (FIR) methodology.

The main motivation of our research is to explore the added value of using a fuzzy knowledge-based approach as an alternative to other knowledge-based and model-based approaches, for cases in which the non-linear dynamics are insufficiently known. The goal of this research is to allow a tolerant fuel cell control by means of the addition of a fuzzy fault diagnosis system operating in real-time. With this idea in mind, the VisualBlock-FIR FDS based

on the Fuzzy Inductive Reasoning (FIR) methodology is presented in this work. The FIR methodology is based on a hybrid fuzzy pattern recognition approach and its conceptualization arises of the General System Problem Solving (Klir and Elias, 2002). This methodology of modeling and qualitative simulation is based on systems behavior rather than structural knowledge. It has the ability to describe systems that cannot easily be described by classical methods (e.g. differential equations), i.e., systems, whose physical processes are awkward to model.

The novelty of this work lies in using the FIR methodology to model the different fault scenarios and the plant without faults, and applying the detection and identification VisualBlock-FIR algorithm to detect and isolate the faults occurred as fast as possible.

Most of the inductive model identification techniques, such as neural networks and its hybridizations, assume a fixed (although often arbitrary) structure and map the knowledge contained in the training data set onto a set of parameter values. The training data are only used during the model identification phase, i.e., the modeling phase. Once the model has been identified, simulation runs are based solely upon the previously optimized parameter values. Such techniques suffer from the problem that they normally are unable to recognize, when the testing data lie outside the range, for which the model has been validated.

In contrast, FIR is a non-parametric technique. The training data are characterized and classified during the model identification phase, but they are not mapped onto parameter sets. Therefore, FIR refers back to the classified training data set also during the simulation or prediction phase. This property makes it impossible for FIR to extrapolate “generously” during simulation and, therefore, to predict values that are not physically possible. Finally, FIR has self-assessment capabilities that enable it to warn the user of the methodology if it makes “risky” predictions, i.e. predictions that are not well founded on the basis of the available training data.

To prove the usefulness and robustness of the proposed methodology, the PEM system simulator developed by Pukrushpan et al. (2004) and modified by Escobet et al. (2009), to include a set of typical fault scenarios, has been used. This allows comparison of the results obtained by the VisualBlock-FIR methodology to the results achieved by the model-based fault diagnosis methodology presented by Escobet et al. (2009). It is important to mention that we have chosen the fault scenarios described by Escobet et al. (2009), because the data was easily available. Moreover, in that work, the detection and identification of the faults is reported not only taking into account whether it is possible or not to detect and identify them; they also take into account the time needed to do it. We could not find many papers that present the time needed by the proposed methodologies to detect and identify the faults of PEM fuel cells.

The VisualBlock-FIR fault diagnosis methodology is introduced in Section 2. Section 3 describes the PEM fuel cell system. Section 4 presents the application of VisualBlock-FIR to the fuel cell system and the results obtained. Section 5 presents the study of robustness of

VisualBlock-FIR methodology. Finally, the conclusions of the research are outlined in Section 6.

## 2. Fault diagnosis methodology

It is necessary to clarify here that the Fuzzy Inductive Reasoning (FIR) methodology is responsible for identifying models that represent specific behaviors of a system. With a FIR model available the FIR prediction process forecasts the next output value of the system. On the other hand, the VisualBlock-FIR methodology is the FDS based on the FIR models previously mentioned.

A FDS aims to detect and identify the different faults that may occur in the system over time. VisualBlock-FIR performs these tasks in two modules, i.e. detection and identification, which the user can manage by means of a user-friendly framework. In the fault detection module, a FIR model is used to find out if a fault has occurred, setting off an alarm in that case. In the identification fault module, the type of fault is identified. In the following sections the VisualBlock-FIR is described and, when necessary, the FIR modeling and prediction processes are also explained.

### 2.1. VisualBlock-FIR fault detection

The fault detection process of VisualBlock-FIR is presented in Fig. 1. In this figure the gray boxes represent FIR processes, whereas the white boxes constitute the fault detection procedure. As in other diagnosis methods, a model allows checking the consistency between measured system output and its predicted behavior. If an inconsistency is detected, the existence of a fault is proved. In the proposed approach, the FIR methodology has been used for obtaining a qualitative behavioral model.

The data measured from the system is converted into qualitative triples (class, membership, and side) by means of the FIR *fuzzification* process. The class value represents a coarse discretization of the original real-valued variable. The fuzzy membership value denotes the degree of belonging of the data to a fuzzy set. Finally, the side value indicates whether the quantitative value is to the left or to the right of the peak value of the associated membership function. The side value, which is a specialty of the FIR technique since it is not commonly used in fuzzy logic, is responsible for preserving, in the qualitative triple, the complete knowledge that had been contained in the original quantitative value. The defuzzification is the inverse process of the fuzzification. Fig. 2 illustrates the process of fuzzification by means of an example. A temperature of 23 °C would hence be

fuzzified into the class *normal* with a side value of *right* and a fuzzy membership value of 0.755. The determination of the number of classes per variable and the distribution of the classes in the search space (shape of the membership functions) is defined by the application's experts or by the modeler that has available a set of clustering algorithms, such as fuzzy *c*-means, equal frequency partition, hierarchical clustering, *k*-means, etc. FIR also offers the possibility to use genetic algorithms to learn the granularity (number of classes per variable) and/or the shape of the membership functions. However, the computational time is quite large if this option is chosen.

The *fuzzy forecasting* process predicts the next qualitative output value from the qualitative input data using the best FIR model identified, which represents the current behavior of the system. The FIR model should be identified out of line, before the fault detection process takes place. A FIR model is composed by a structure, called mask, and a pattern rule base, named input/output matrix. A mask represents a possible causal and dynamic relation among the qualitative variables. It is represented by a matrix where each column corresponds to a measured variable. It has a certain number of rows, which represents the temporal domain that can influence the output. Each row is delayed relative to its successor by a time interval of  $\delta t$  representing the time lapse between two consecutive samplings. An example of a mask is presented in Fig. 3. This mask has four *m*-inputs,  $\{u_1(t-2\delta t), u_4(t-2\delta t), y_1(t-\delta t), u_2(t)\}$ , denoted by  $i_1, i_2, i_3, i_4$  respectively, and one *m*-output,  $O_1$ , associated to  $y_1(t)$ .

The best mask is found by means of an optimization process that determines the most deterministic state transition matrix. The optimality of the mask is evaluated with respect to the maximization of its forecasting power.

How can the one mask be found that, within the framework of all allowable masks, represents the most deterministic state transition matrix? This mask will optimize the predictiveness of the model. In the FIR methodology, the concept of a mask candidate matrix has been introduced. A mask candidate matrix is an ensemble of all possible masks, from which the best is chosen by a mechanism of exhaustive search. Another search strategy based on genetic algorithms has also been implemented.

The optimal mask function searches through all legal masks of complexity two, i.e., all masks with a single *m*-input, and finds the best one; it then proceeds by searching through all legal masks of complexity three, i.e., all masks with two *m*-inputs, and finds the best of those; and it continues in the same manner until the maximum allowed complexity (a parameter) has been reached. In all practical examples, the quality of the masks will first grow with increasing complexity, then reach a maximum, and then decay

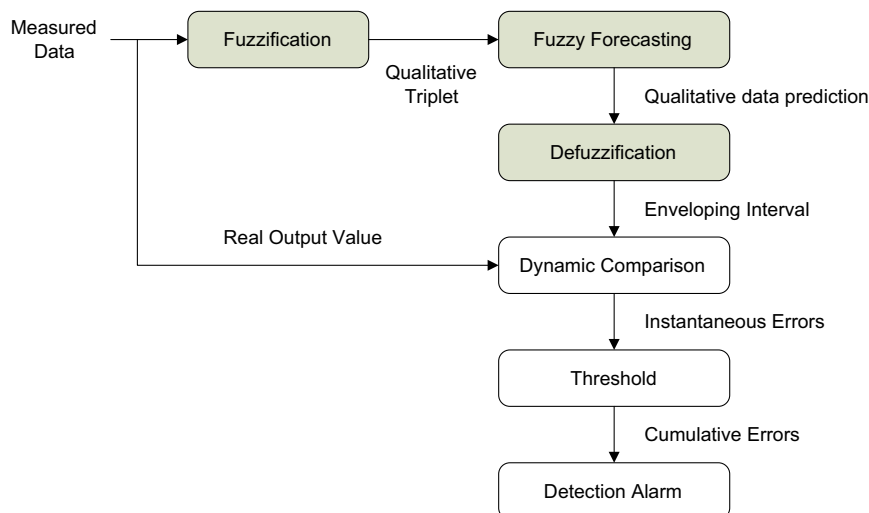


Fig. 1. VisualBlock-FIR fault detection process.

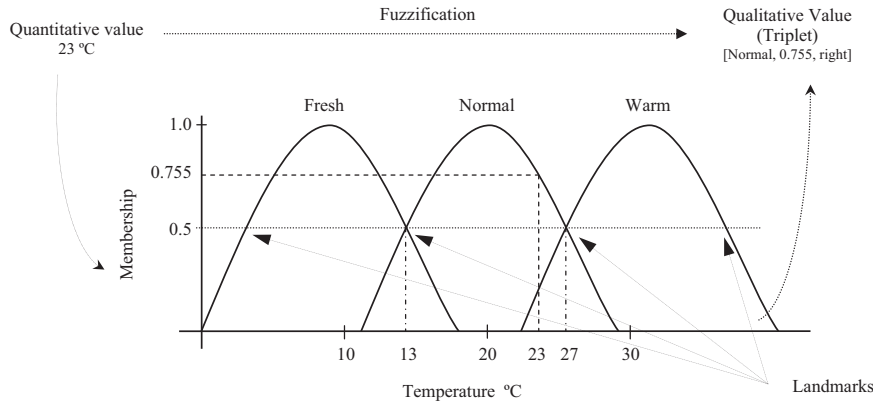


Fig. 2. FIR fuzzification process: example for ambient temperature variable.

$$\begin{array}{c}
 t/x \\
 t - 2\delta t \\
 t - \delta t \\
 t
 \end{array}
 \begin{array}{c}
 u_1 \\
 u_2 \\
 u_3 \\
 u_4
 \end{array}
 \begin{array}{c}
 u_1 \\
 u_2 \\
 u_3 \\
 u_4
 \end{array}
 \begin{array}{c}
 u_3 \\
 u_4 \\
 0 \\
 0
 \end{array}
 \begin{array}{c}
 u_4 \\
 0 \\
 0 \\
 0
 \end{array}
 \begin{array}{c}
 y_1 \\
 0 \\
 -i_3 \\
 O_1
 \end{array}$$

Fig. 3. Mask with four  $m$ -inputs ( $i_1, i_2, i_3, i_4$ ) and one  $m$ -output ( $O_1$ ).

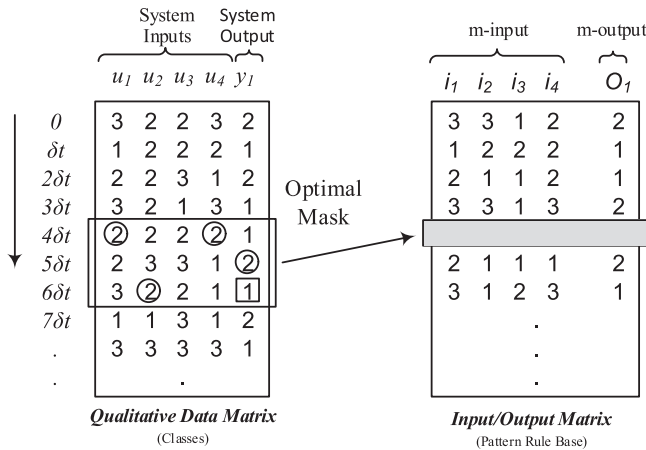


Fig. 4. Derivation of FIR pattern rule base.

rapidly. Each of the possible masks is compared to the others with respect to its potential merit. The optimality of the mask is evaluated with respect to the maximization of its forecasting power that is quantified by means of a quality measure, based mainly on Shannon entropy (Escobet et al., 2008).

Fig. 4 illustrates how the input/output matrix (pattern rule based) is obtained from the mask. The left hand side of Fig. 4 shows an excerpt of the qualitative data matrix that stores the class values. It shows the numerical rather than the symbolic class values presented in Fig. 2, i.e. instead of fresh, normal and warm linguistic values we use numerical values to represent them. The box symbolizes the mask that is shifted downwards along the qualitative data matrix. The round “holes” in the mask denote the positions of the  $m$ -inputs, whereas the square “hole” indicates the position of the  $m$ -output.

The class values are read out from the qualitative data matrix through the “holes” of the mask, and are placed next to each other in the input/output matrix that is shown on the right hand side of Fig. 4. Each row of the matrix represents one pseudo-static qualitative state or qualitative rule (also called pattern rule). For example, the shaded rule of Fig. 4 can be read as follows: “If all  $m$ -inputs ( $i_1, i_2, i_3, i_4$ ) have a value of 2 (corresponding to *medium*) then the  $m$ -output,  $O_1$ , assumes a value of 1 (corresponding to *low*)”.

In the forecasting phase, future qualitative output states can be predicted using the model by means of the inference engine (Escobet

et al., 2008). The FIR inference engine is based on a variant of the  $k$ -nearest neighbor (KNN) rule. The 5NN pattern matching algorithm is the core of the FIR inferencing process. The forecast of the output variable is obtained as a weighted average of the potential conclusions that result from firing the five rules, whose antecedents best match the actual state, as shown in Fig. 5. The qualitative simulation process predicts an entire qualitative triplet, from which a quantitative value can be obtained whenever needed.

The envelope drives the fault detection process (see Fig. 1) of VisualBlock-FIR. As shown in Fig. 6, the envelope is composed of an upper bound (maximum value) and a lower bound (minimum value), that delimit the area in which the real output signal is expected to lie. The envelope is computed by applying the defuzzification process to the five nearest neighbors obtained in the fuzzy forecasting process of FIR methodology. The upper bound is the neighbor predicting the highest value and the lower bound correspond to the neighbor predicting the lowest value.

A narrow enveloping interval implies that the five neighbors are very close to each other, meaning that the information available of the behavior of the system at that point in time is very rich and complete. By contrast, a wide enveloping interval means that there is not a lot of information available about the system at that point in time, and therefore, the nearest neighbors are far away from each other. It is important to keep in mind that the FIR methodology is driven by the system’s behavior rather than relying on structural knowledge, and therefore, the amount and richness of the data available from the system are crucial in order to assure the identification of an accurate and reliable model that represents it.

Fault detection process checks if the real signal lies inside the envelope. If the observed (real) value leaves the bounds specified by the envelope, an *instantaneous error* is recorded, meaning that the model used in the prediction no longer represents the system at that specific point in time.

In order to reduce the effect of instantaneous errors due to the presence of noise in the measurements or unmodeled dynamics, two filters are introduced, a *detection time window* and a *threshold error*. The instantaneous errors recorded inside a predetermined *detection time window* are accumulated. When the cumulative error surpasses the specified threshold, an alarm is triggered, and it is then necessary to identify the fault that has occurred (see Fig. 1).

Fig. 6 presents an example of FIR fault detection using the enveloping concept with a time window of 15 prediction points. The upper and lower dotted lines represent the upper and lower bounds of the envelope, respectively, whereas the continuous line is the real output signal. In the bottom part of the figure, the instantaneous errors are accumulated. As can be seen, the real value leaves the envelope for the first time at point number 6 where the observed value exceeds the upper bound of the envelope, causing an instantaneous error. The same occurs at points number 7 and 11.

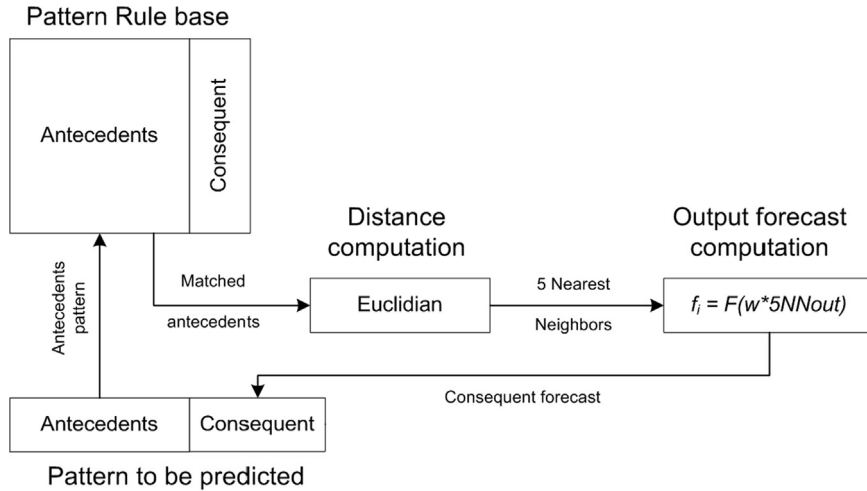


Fig. 5. FIR prediction process.

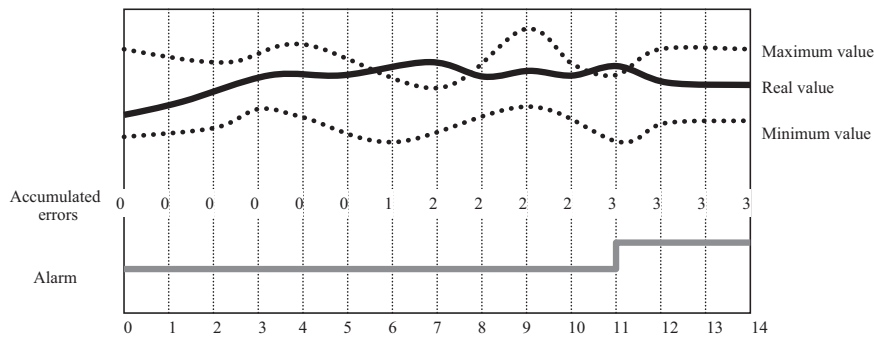


Fig. 6. Example of VisualBlock-FIR fault detection using the enveloping method.

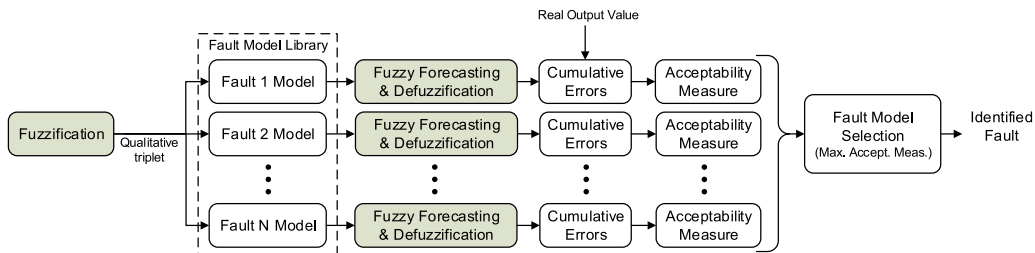


Fig. 7. VisualBlock-FIR fault identification process.

The threshold of cumulative errors specified in this example was 3, and therefore, the alarm is triggered at point number 11 when the third instantaneous error arrives.

### 2.2. VisualBlock-FIR fault identification

When an alarm is triggered, the fault identification process, described in Fig. 7, starts. The main part of this process is the fault model library, where each model has been identified, out of line, using the FIR methodology.

As in the detection process, the gray boxes represent FIR processes whereas the white boxes constitute the fault identification procedure.

At each instant of time, the measured data, once fuzzified, is used for each model stored in the fault model library to obtain the envelope interval. As in the detection process, the *instantaneous errors* recorded for each fault model are accumulated over time. A predetermined *identification time window* defines the number of prediction values that will be used in order to identify the fault that has occurred. Notice that a small-sized *identification time window* is

preferred because it implies fast model identification. In order to determine the size of the time window it is necessary to take into account the system's dynamics. Due to the fact that the FIR models for each of the different faults are already available, it is possible to experiment with them to establish a small useful time window.

The accumulated errors are used to compute the model acceptability measure. The acceptability measure is a relative index ranking the models in terms of their ability to predict the new behavior of the system. This measure allows us, in a reliable way, to identify the fault that has occurred. It also offers guidance when the identification process faces additional problems, e.g. when the produced fault is not a foreseen fault and therefore is not available in the fault model library, or when two different models can be identified that both are able to explain an observed fault.

The acceptability measure of the *i*th model,  $Q_i$ , is described by Eq. (1).

$$Q_i = C_i C_{rel_i} \tag{1}$$

where  $C_i$  is the partial acceptability measure of the *i*th model and  $C_{rel_i}$  is a relative confidence that takes the dispersion between the



$C_i$  values into account.  $C_i$  is obtained adding the cumulative errors of the  $i$ th model,  $I_{a_i}$ , and computing the maximum number of local cumulative errors possible (depending on the size of the identified time window),  $I_{a_{max}}$  as shown in Eq. (2).

$$C_i = 1.0 - I_{a_i} / I_{a_{max}} \quad (2)$$

$C_i$  is a confidence measure, i.e. a real-valued number in the range [0,1], where larger values indicate increased confidence. The relative confidence is obtained by Eq. (3).

$$C_{rel_i} = C_i / \sum_{k=1}^n C_k \quad (3)$$

If there is only one model with a high value of  $C_i$ ,  $C_{rel_i}$  will still be very high, but if there are other models with high  $C_i$  values as well,  $C_{rel_i}$  will be much smaller. The model with the largest acceptability measure is selected as the one that best represents the new behavior of the system, and therefore, the detected fault has been identified.

**Table 1**  
Parameters used in the PEM fuel cell.

Symbol	Variable	Value
$\rho_{m,dry}$	Membrane dry density	0.002 kg/cm <sup>3</sup>
$M_{m,dry}$	Membrane dry equivalent weight	1.1 kg/mol
$t_m$	Membrane thickness	0.01275 cm
$n$	Number of cells in stack	381
$A_{fc}$	Fuel cell active area	280 cm <sup>2</sup>
$d_c$	Compressor diameter	0.2286 m
$J_{cp}$	Compressor and motor inertia	$5 \times 10^{-5}$ kg m <sup>2</sup>
$V_{an}$	Anode volume	0.005 m <sup>3</sup>
$V_{ca}$	Cathode volume	0.01 m <sup>3</sup>
$V_{sm}$	Supply manifold volume	0.02 m <sup>3</sup>
$V_{rm}$	Return manifold volume	0.005 m <sup>3</sup>
$C_{D,rm}$	Return manifold throttle discharge coefficient	0.0124
$A_{T,rm}$	Return manifold throttle area	0.002 m <sup>2</sup>
$k_{sm,out}$	Supply manifold outlet orifice constant	$0.36293 \times 10^{-5}$ kg/(s Pa)
$k_{ca,out}$	Cathode outlet orifice constant	$0.21773 \times 10^{-5}$ kg/(s Pa)
$k_v$	Motor electric constant	0.0153 V/(rad/s)
$k_t$	Motor torque constant	0.0225 N-m/A
$R_{cm}$	Compressor motor circuit resistance	1.2 $\Omega$

**Table 2**  
Description of the fault scenarios used in this study.

ID	Fault description	Type	Magnitude
#1	There is suddenly a increase of friction in the mechanical component part of the compressor	Parametric abrupt	100% of increment (the constant is duplicated)
#2	The compressor motor suffers an overheating	Parametric abrupt	100% of increment (the constant is duplicated)
#3	The fluid resistance increases due to water blocking the channels or flooding in the diffusion layer	Parametric abrupt	50% of reduction of the water flow
#4	Air leak in the air supply manifold	Parametric abrupt	50% of reduction of the air flow
#5	Increase in the voltage value below which the compressor motor does not turn	Parametric abrupt	Increased from 30 V to 60 V

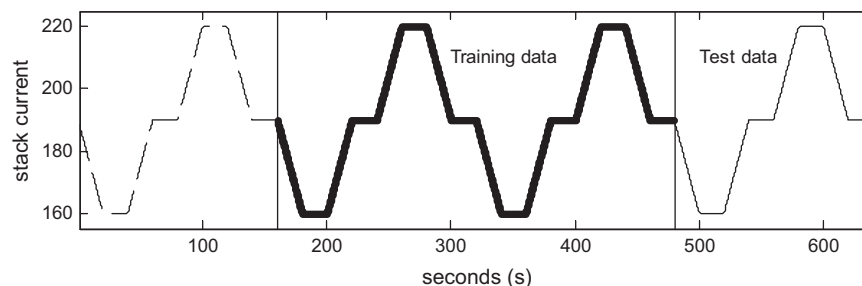
### 3. Case study: PEM fuel cell system

A PEM fuel cell system is an electrochemical cell where chemical energy is directly converted into electricity, by combining hydrogen fuel with oxygen from air, with waste heat and liquid water as by-products. In this paper, a simulator of the PEM fuel cell system is used as case study for fault diagnosis. The simulator system uses the non-linear model and controller strategies proposed by Pukrushpan et al. (2004) which represents a 75 kW fuel cell system with 381 cells. This PEM fuel simulator allows incorporating various faults (Table 2) within the Simulink environment.

A schematic diagram of the main subsystems of a PEM fuel cell system and system supervision variables is shown in Fig. 9. Table 1 describes the set of parameters of the PEM fuel cell used in this study.

The air supply system consists of an air compressor to feed atmospheric air to the fuel cell cathode, the supply manifold that connects the compressor output with the air cooler input, and the air cooler and the static humidifier that, respectively, refrigerates and humidifies the air before it goes into the stack. The majority of the parasitic power for an automotive fuel cell system is spent on the air compressor, thus, it is important to determine the proper air flow. The air flow excess is reflected by the term oxygen excess ratio, defined as the ratio of oxygen supplied to oxygen used in the cathode, i.e.  $\lambda_{o_2} = o_2 in / o_2 react$ .

The model guarantees the required stoichiometry by regulating the hydrogen, supplied from a pressurized or liquid hydrogen tank, by means of a controlled valve. The electrochemical principles of the fuel cell stack are also modeled in order to accurately evaluate the electricity production and the outputs of the stack. This implies specific model equation for the anode, the cathode, the membrane and the stack voltage. Finally, the cathode outlet manifold of the fuel cell is considered in the model as an external component. The simulated model only describes the normal operational mode and purges in the anode side are not considered, meaning that all the hydrogen in the anode side is consumed. It is also assumed that the temperature of the fuel cell stack is known and constant since its dynamic behavior is much slower than that of the rest of the model. The transient phenomena captured in the model include the flow and inertia dynamics of the compressor, the manifold filling dynamics (both anode and cathode), reactant partial pressures, and membrane



**Fig. 8.** Stack current periodic set-point signal.

humidity. For a deeper description of the PEM fuel cell simulator please refer to Lira et al. (2010).

The available sensors in the system are another important issue to take into account for fault diagnosis. In the study case, system provides measures of the compressor current  $I_{cm}$ , and its speed,  $\omega_{cm}$ , and the voltages applied to the compressor,  $V_{cm}$ , and the cell,  $V_{fc}$ . The stack current  $I_{fc}$ , is regarded as a measured disturbance to the system.

The simulator includes the five faults presented in Table 2. The fault #1 is simulated with an increment in the compressor constant and, similarly, the fault #2 is simulated with an increment in the compressor motor resistance. Both faults are translated in a change in the compressor torque. The fault #3 is simulated with an increment in the orifice constant of the cathode output which produces a change in the outlet air flow in the cathode. The fault #4 is simulated with an increment in the supply manifold outlet flow constant, which is translated into a change in the outlet air flow in the supply manifold. The fault #5 is simulated with an increment in the lower voltage that the controller supplies to the compressor motor, a boundary that also influences the compressor torque.

#### 4. Fault diagnosis of the PEM fuel cell system

Fault diagnosis requirements applied to PEM fuel cell system are to detect and isolate the five faults proposed as a study case. In order to do so, it is necessary to first identify the FIR qualitative model of the plant both in normal operation and in failure mode.

##### 4.1. FIR qualitative models identification

The first step for building the FDS is to obtain the FIR qualitative models, which constitute the model library. As Fig. 9

shows, the measured data are  $I_{fc}$ ,  $V_{cm}$ ,  $I_{cm}$ ,  $\omega_{cm}$ ,  $\lambda_{O_2}$  and  $V_{fc}$ , where the first two are input variables.

When load changes, the current delivered by the fuel cell stack varies. At that point, the fuel cell system controller adjusts the air pump in order to maintain appropriate air feeding. In this study, PEM fuel cell has been perturbed with a periodic load change. The load periodically draws the current signal shown in Fig. 8, with a period of 160 s and amplitude between 160 and 250 A. The second and third periods are used to train the models and the last one to test them.

The input and output variables are discretized into 3 classes. The number of classes per variable has been determined by the modelers after an analysis of the signals. The membership functions are defined following the equal frequency partition (EFP) algorithm, which distributes the membership functions in the search space in such a way that each class contains the same number of training data points. In this case, it is not necessary to use a genetic algorithm to optimize the discretization parameters, due to the fact that the prediction results obtained with the proposed discretization values are already really good.

Due to the fact that each FIR model can have only one output, it becomes necessary to identify 4 models for normal plant behavior and 4 for each fault studied. A total of 24 different models (4 outputs  $\times$  6 plant behaviors) should be identified before VisualBlock-FIR is ready to perform fault detection and identification.

For each of these 24 scenarios, the system is simulated during 3 complete signal periods of 160 s each, i.e. 480 s of simulation time, with a sampling rate of 0.1 s, generating 4800 data points. The first 320 s (3200 data points) are used by FIR to identify the model, and the final 160 s are used to verify the model. The masks obtained for the 24 scenarios are presented consecutively in Eq. (4).

	Normal behaviour			Fault #1			Fault #2			Fault #3			Fault #4			Fault #5		
$t/I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$	$I_{fc}$	$V_{cm}$	$I_{cm}$
$t-4\delta t$	-1	-2	0	0	0	0	-1	-2	0	-1	-2	0	-1	-2	0	-1	-2	-3
$t-3\delta t$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t-2\delta t$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t-\delta t$	-3	0	-4	0	0	-1	-3	0	-4	-3	0	-4	-3	0	-4	-4	-5	-6
$t$	0	0	1	-2	-3	1	0	0	1	0	0	1	0	0	1	0	0	1
$t/\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$	$I_{fc}$	$V_{cm}$	$\omega_{cm}$
$t-4\delta t$	-1	-2	0	0	0	0	0	0	0	-1	-2	0	-1	-2	0	0	0	0
$t-3\delta t$	0	0	0	-1	-2	0	-1	-2	0	0	0	0	0	0	0	0	0	-1
$t-2\delta t$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	-3	0
$t-\delta t$	-3	0	-4	0	0	0	0	0	0	-3	-4	0	-3	-4	0	0	0	0
$t$	0	0	1	-3	0	1	-3	-4	1	0	0	1	0	0	1	0	0	1
$t/\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$	$I_{fc}$	$V_{cm}$	$\lambda_{O_2}$
$t-4\delta t$	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	-1	-2	-3
$t-3\delta t$	-1	-2	0	0	0	0	-1	-2	0	0	0	0	0	0	0	0	0	0
$t-2\delta t$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t-\delta t$	0	0	-3	-1	-2	-3	0	0	0	0	0	-2	0	0	-2	-4	-5	-6
$t$	-4	0	1	0	0	1	-3	-4	1	-3	0	1	-3	-4	1	0	0	1
$t/V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$	$I_{fc}$	$V_{cm}$	$V_{fc}$
$t-4\delta t$	0	0	-1	0	0	0	0	0	0	-1	0	-2	-1	-2	-3	0	0	0
$t-3\delta t$	0	0	0	-1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0
$t-2\delta t$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$t-\delta t$	0	0	-2	0	0	0	0	0	-1	0	0	-3	-4	-5	-6	0	0	-1
$t$	-3	-4	1	-3	0	1	-2	-3	1	-4	0	1	0	0	1	-2	-3	1

(4)

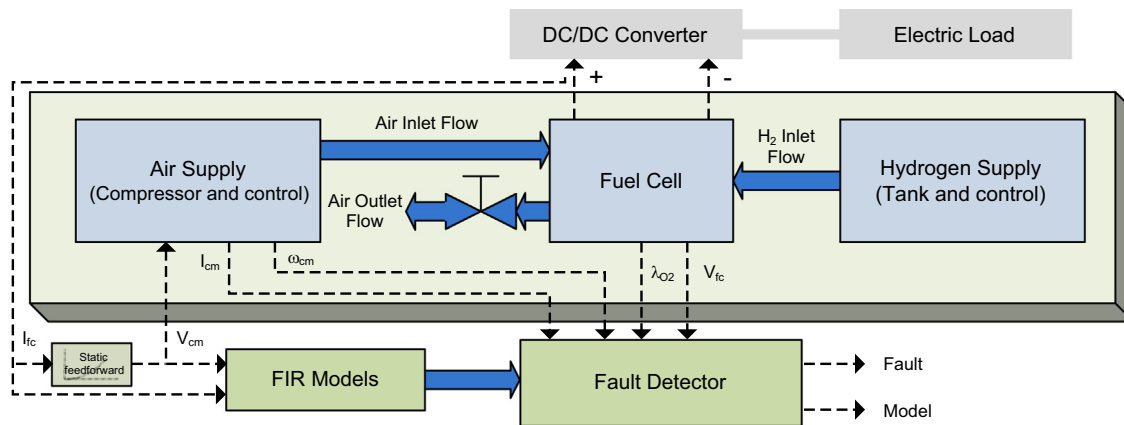


Fig. 9. Schematic diagram of FDS with PEM fuel cell system (adapted from Feroldi et al. (2007)).

The mean square error in percentage defined in Eq. (5) is used to compute the prediction error.

$$\text{MSE} = \left( E \left[ \frac{(y(t) - \bar{y}(t))^2}{\sigma^2} \right] \right) \times 100\%; \quad \sigma^2 = E \left[ y^2(t) \right] - \{E[y(t)]\}^2 \quad (5)$$

where  $E$  is the mean value,  $y(t)$  is the system output, QUOTE is the predicted value and  $\sigma^2$  denotes the variance.

For each of the 24 FIR models identified, mean square errors lower than  $1 \times 10^{-2}\%$  are obtained when predicting the test sets. These prediction errors are very low, meaning that the models obtained represent accurately each of the system's behaviors. The FIR models are stored in the fault model library and are used during the fault identification phase of VisualBlock-FIR.

It should be taken into account that if a possible fault is not included in the fault model library it can be detected but not identified. Therefore, the set of faults should be selected accurately. However, it is possible to include new faults in the model library when needed. On the other hand, the mask and the pattern rule base are stored in the model library for each fault and, therefore, if the number of faults becomes very large the memory burden for on-line applications could become a problem.

As has already been explained, FIR is a data driven methodology that infers the model from system measured trajectories. As all other inductive model identification techniques, FIR is not able to obtain "good" models with poor and/or scarce data. From statistical considerations each legal discrete state should be recorded at least five times (Law and Kelton, 1990). Thus, a relation exists between the total number of legal states and the number of data points required to base the modeling effort upon. The total number of legal states is computed multiplying the number of classes of all input variables. Obviously, fault detection and identification are more accurate when the fault models identified represent accurately its behavior. However, as mentioned earlier, FIR has self-assessment capabilities that enable to know if it makes predictions that are not well founded on the basis of the available training data.

#### 4.2. Simulation results

Turning to the detection and identification performance of VisualBlock-FIR, different experiments have been performed introducing the previously described 5 faults. The goal of these experiments was to study the usefulness of our approach detecting and identifying the faults. Fault #1 is introduced at second 150 when the plant is working properly. Figs. 10 and 11 show the dynamic comparison of the six FIR models when used for predicting oxygen excess ratio,  $\lambda_{o_2}$ , and compressor current,  $I_{cm}$ ,

output variables, respectively. The envelope is computed at the same time for the non fault model (most upper plot of Fig. 10), and the fault models (rest of the plots of Fig. 10). The maximum and minimum values of the enveloping are represented as dashed lines whereas the real data is represented as a continuous line. As explained before, the FIR models identified are very accurate (have low prediction errors) and, therefore, the upper and lower bounds of the enveloping are, in both figures, almost undistinguishable. In this study case, the detection time window lasts 1 s, the threshold of the cumulative error is 3 and the identification time window lasts 6 s.

Table 3 shows the detection and identification time (s) for each of the output variables and all of the faults analyzed. In the case of fault #1, VisualBlock-FIR needs only 0.4 s (4 samples) to detect a fault when oxygen excess ratio and compressor current output variables are analyzed. At second 150.4 an alarm is triggered and the identification phase starts, as seen in Figs. 10 and 11. For compressor speed,  $\omega_{cm}$ , and stack voltage,  $V_{fc}$ , output variables, VisualBlock-FIR needs 0.5 and 0.8 s, respectively, to detect that a fault has occurred.

In this case, a time window of 60 data points (corresponding to 6 s) is used to determine the fault that has occurred. Six seconds is enough for the acceptability measure to be sure that fault #1 has been produced. In this application acceptability measures of 1 (maximum value) are obtained. In Figs. 10 and 11 it can be seen that at second 156 the enveloping of fault #1 contains the real signal, meaning that fault #1 model represents the current behavior of the plant. Contrarily, envelopes of the rest of the models do not contain the real signal.

Almost identical results are obtained when the same analysis is carried out for fault #2, as summarized in Table 3. As can be seen in the second row of this table, the detection for each output variable is done very quickly (from 0.4 to 1.5 s). The identification performance is also very good, i.e. only 6 or 7 s is needed to know which fault has occurred.

If the same experiment is applied to fault #3 it is found that the behavior of compressor speed,  $\omega_{cm}$ , and stack voltage,  $V_{fc}$ , variables is extremely similar to the signals of the same variables when the plant is working properly (there is no fault). This is clearly seen in Fig. 12, for stack voltage signal. Notice that the real signal is inside the enveloping when the system is working properly (most upper plot of Fig. 12) as it also happens in fault #3 (middle plot of Fig. 12). For this reason, VisualBlock-FIR is not able to detect fault #3 when analyzing  $\omega_{cm}$  and  $V_{fc}$  variables, as shown in Table 3.

If the same experiment is applied to fault #3 it is found that the behavior of compressor speed,  $\omega_{cm}$ , and stack voltage,  $V_{fc}$ , variables is extremely similar to the signals of the same variables when the plant is working properly (there is no fault). This is clearly seen in



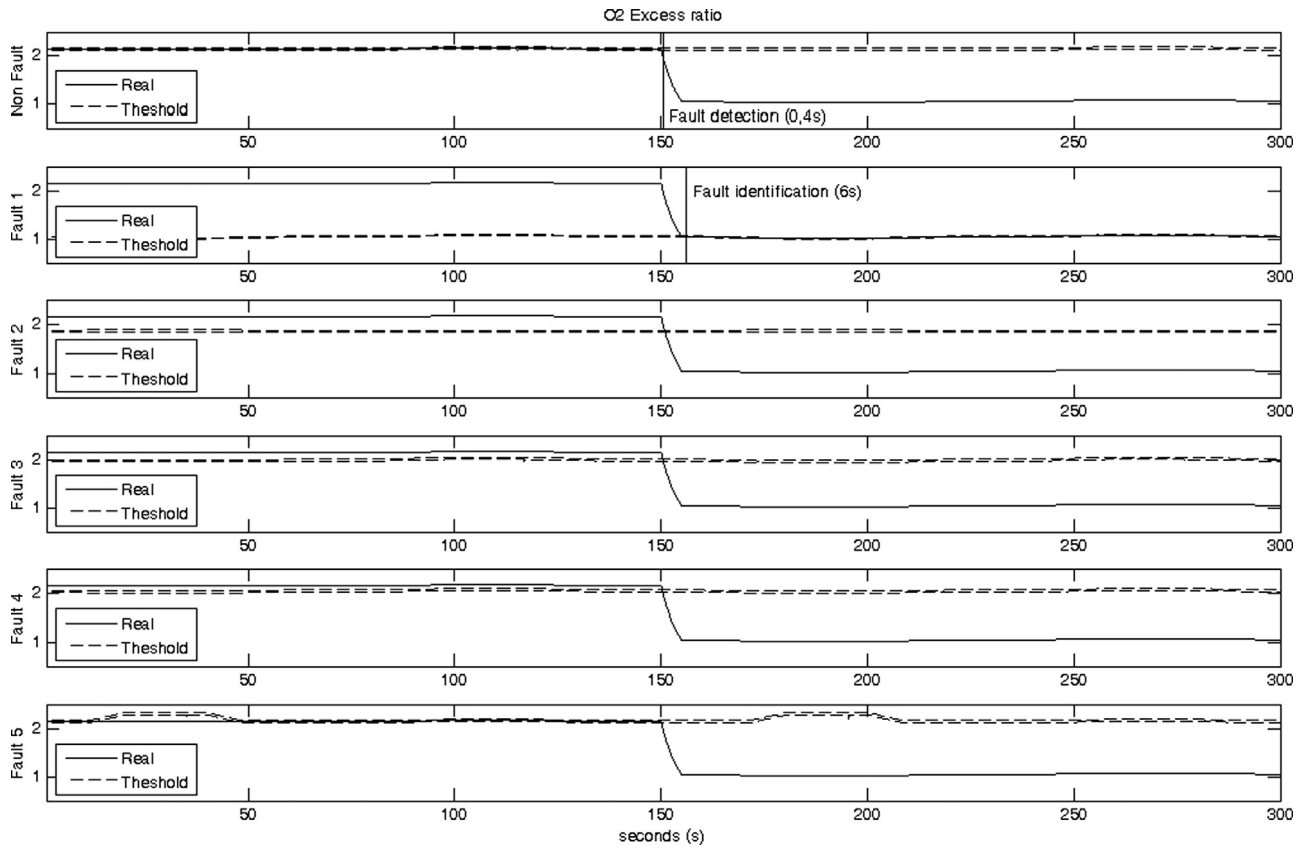


Fig. 10. VisualBlock-FIR detection and identification of fault #1 for oxygen excess ratio,  $\lambda_{O_2}$ , output variable.

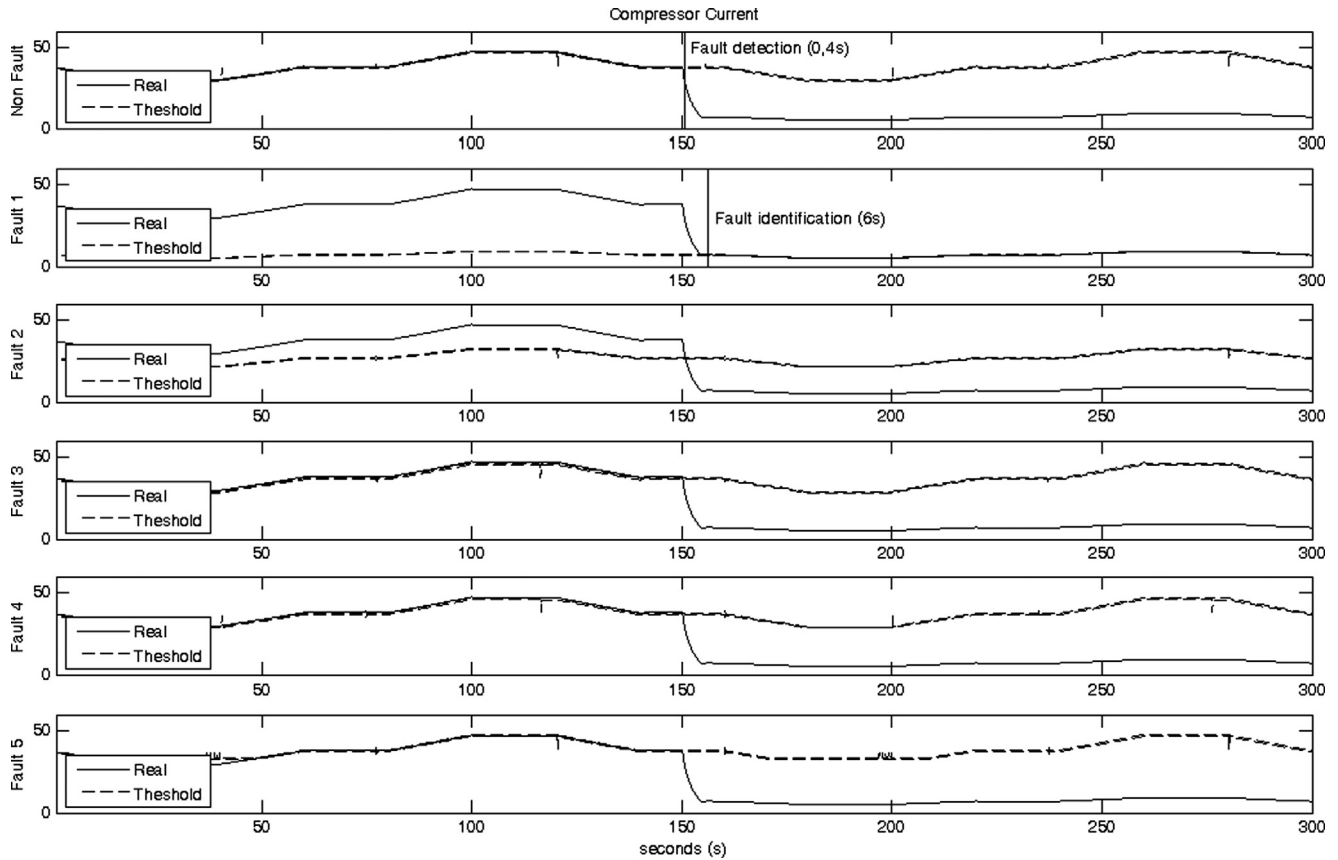


Fig. 11. VisualBlock-FIR detection and identification of fault #1 for compressor current,  $I_{cm}$ , output variable.

Fig. 12, for stack voltage signal. Notice that the real signal is inside the enveloping when the system is working properly (most upper plot of Fig. 12) as it also happens in fault #3 (middle plot of Fig. 12). For this reason, VisualBlock-FIR is not able to detect fault #3 when analyzing  $\omega_{cm}$  and  $V_{fc}$  variables, as shown in Table 3.

However, fault #3 is rapidly detected and identified when  $\lambda_{o_2}$  and  $I_{cm}$  variables are analyzed. This can be clearly seen in Fig. 13, which shows the dynamic comparison of the six FIR models when used for predicting oxygen excess,  $\lambda_{o_2}$ , output variable. In this case the fault is detected in 0.4 s and identified in 6 s.

VisualBlock-FIR also obtains good detection and identification results for fault #4. As before, it is able to detect the fault in 0.4 s and needs only 6 s to identify it. However, it is not able to detect the fault when the compressor speed,  $\omega_{cm}$ , variable is the only one taken into account. The same reason outlined in fault #3 is valid here.

Fault #5 should be approached in more detail due to the fact that the detection and identification of the four output variables become much more difficult than in the previous faults (see last row of Table 3). VisualBlock-FIR needs around 22–25 s to detect

the fault and 28–31 s to identify it. Notice that the seconds shown in Table 3 are absolute seconds, i.e. the identification time includes the detection time. This time increase needed to detect and identify faults is a consequence of the similarity between the behavior of the output signals of fault #5 and the behavior of the same signals when the system is working properly, mainly in the positive semi-period of the signals. During the negative semi-period, the signals behave differently from one another, allowing the detection of the fault.

Fig. 14 shows the detection and identification results obtained by VisualBlock-FIR for the oxygen excess ratio,  $\lambda_{o_2}$ , output signal in fault #5.

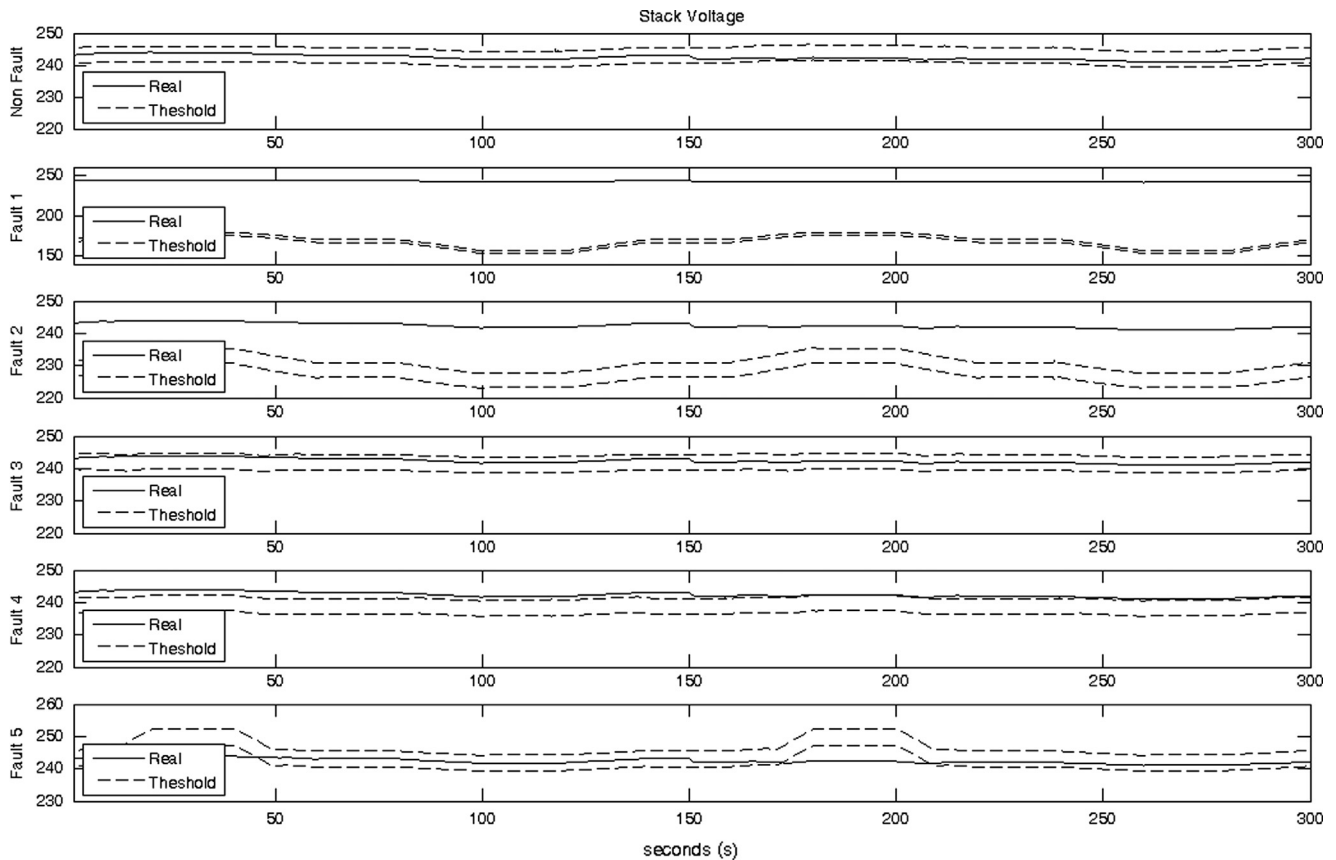
As in the previous experiments the fault has been introduced in the plant at second 150. VisualBlock-FIR detects the fault at second 172.7, and identifies it at second 178.

These results compare favorably to the detection and identification time obtained by the methodology based on computing residuals presented by Escobet et al. (2009), where more than 2 s were needed to detect faults #1 and #2 and more than 30 s were required to identify each of the first two faults. No specific results were reported in that research related to faults #3, #4 and #5. Therefore, no comparison can be made between their research and ours for these three faults.

On the other hand, the work of Lira et al. (2010) also analyzes five faults; two of them correspond to our #1 and #4 faults. Their methodology is based on computing residuals using a LPV observer. They conclude that with this methodology it is possible to detect and identify the faults studied. On the other hand, in the research of Kamal et al. (2014) and Kamal and Yu (2011) five faults are presented, one of them corresponding to our fault #4. In these studies the authors propose to use radial basis function networks. As in the previous case, the authors report that all the faults

**Table 3**  
VisualBlock-FIR performance for the five faults (s).

	Detection				Identification			
	$\lambda_{o_2}$	$\omega_{cm}$	$I_{cm}$	$V_{fc}$	$\lambda_{o_2}$	$\omega_{cm}$	$I_{cm}$	$V_{fc}$
<b>Fault #1</b>	0.4	0.5	0.4	0.8	6	6	6	6
<b>Fault #2</b>	0.6	0.9	0.4	1.5	6	6	6	7
<b>Fault #3</b>	0.4	–	0.4	–	6	–	6	–
<b>Fault #4</b>	0.3	–	0.4	0.4	6	–	6	6
<b>Fault #5</b>	22.7	23.5	22.3	25.3	28	29	28	31



**Fig. 12.** Enveloping and real signals of fault #3 for stack voltage,  $V_{fc}$ , output variable.

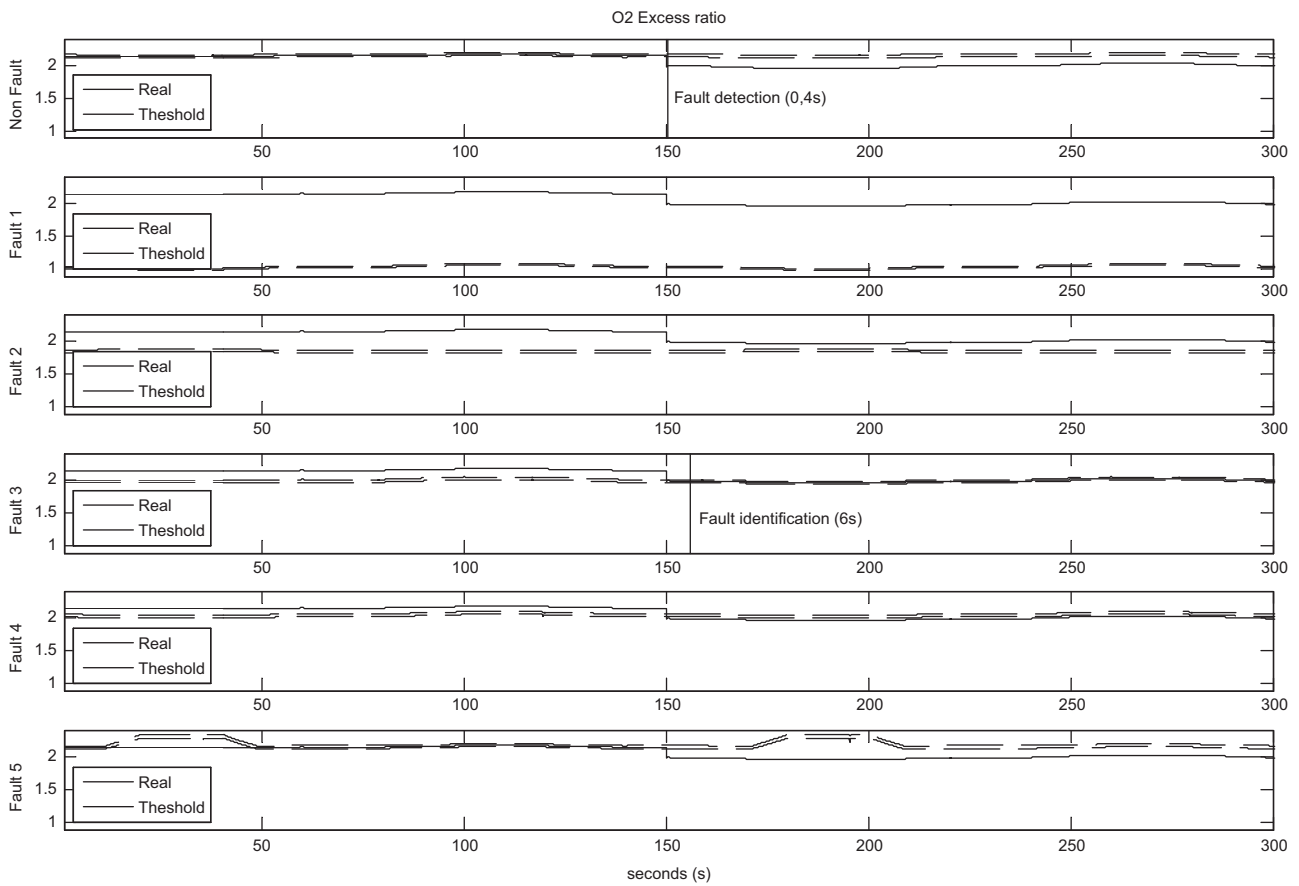


Fig. 13. VisualBlock-FIR detection and identification of fault #3 for oxygen excess ratio,  $\lambda_{O_2}$ , output variable.

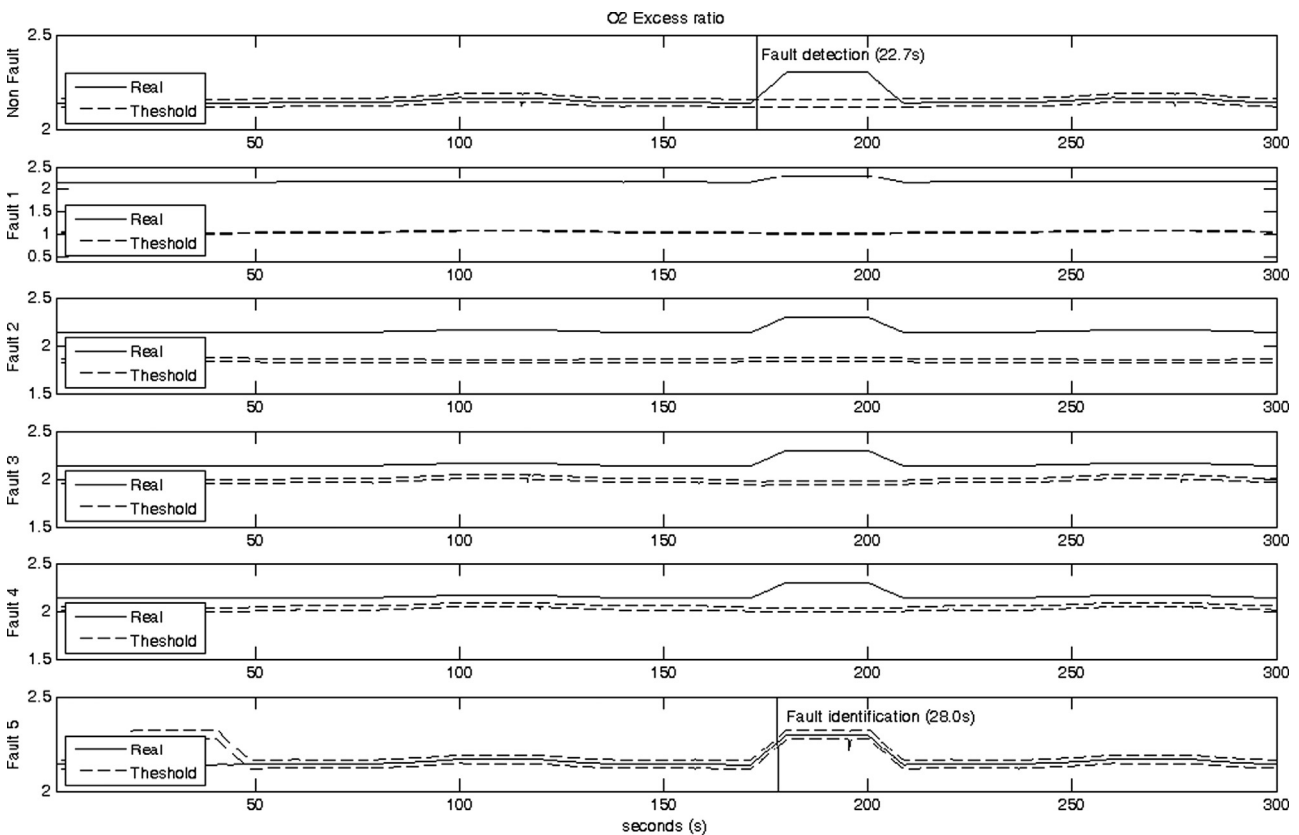


Fig. 14. VisualBlock-FIR detection and identification of fault #5 for oxygen excess ratio,  $\lambda_{O_2}$ , output variable.

**Table 4**

VisualBlock-FIR performance for the five faults with uniform white noise of 60, 40, 30 and 20 dB added to the output signals (s).

	Added noise	Increment enveloping (%)	Detection				Identification			
			$\lambda_{O_2}$	$\omega_{cm}$	$I_{cm}$	$V_{fc}$	$\lambda_{O_2}$	$\omega_{cm}$	$I_{cm}$	$V_{fc}$
<b>Fault #1</b>	0.1% (60 dB)	0	0.4	0.5	0.4	0.8	10	10	10	10
	1% (40 dB)	0	0.4	0.5	0.4	0.9	10	10	10	10
	3.5% (30 dB)	2.5	0.5	0.7	0.4	0.8	10	10	10	10
	10% (20 dB)	8	0.7	1.4	0.5	1.7	10	11	10	11
<b>Fault #2</b>	0.1% (60 dB)	0	0.6	0.9	0.4	1.8	10	10	10	11
	1% (40 dB)	0	0.5	1.3	0.4	2.0	10	11	10	11
	3.5% (30 dB)	2.5	0.7	1.6	0.5	2.3	10	11	10	12
	10% (20 dB)	8	2.3	2.7	0.7	7.6	12	12	10	17
<b>Fault #3</b>	0.1% (60 dB)	0	0.4	–	0.5	–	10	–	10	–
	1% (40 dB)	0	0.4	–	0.4	–	10	–	10	–
	3.5% (30 dB)	2.5	0.4	–	0.6	–	10	–	10	–
	10% (20 dB)	8	0.6	–	19.6	–	10	–	29	–
<b>Fault #4</b>	0.1% (60 dB)	0	0.3	–	0.4	0.4	10	–	10	10
	1% (40 dB)	0	0.3	–	0.4	0.4	10	–	10	10
	3.5% (30 dB)	2.5	0.3	–	0.5	0.7	10	–	10	10
	10% (20 dB)	8	0.6	–	45.0	–	10	–	54	–
<b>Fault #5</b>	0.1% (60 dB)	0	22.8	23.4	22.3	25.3	32	33	32	35
	1% (40 dB)	0	22.6	23.5	22.4	24.7	32	33	32	34
	3.5% (30 dB)	2.5	23.5	23.6	22.7	25.9	33	33	32	35
	10% (20 dB)	8	25.9	33.0	23.6	–	35	41	33	–

studied are detected and identified with the proposed methodology, but no time information is included.

It is interesting to remark that usually the papers found in the literature do not specify the time needed to detect and identify the faults studied. They only report if the fault has or has not been detected and identified. That is the reason why it is not possible to perform a deeper comparison here with other methodologies.

## 5. Robustness of VisualBlock-FIR

In order to study the robustness of the method developed, uniform white noise is added to the four output system variables. The robustness is tested for each of the five faults and for different noise levels. It is important to clarify that the noise is not included in the signals used to identify FIR models, i.e. FIR models are identified from data without noise, and are tested for fault detection and identification in a noise environment.

The reason not to use noisy data to obtain the models is twofold. On the one hand, we think that if the models are obtained without noise and afterwards they are used to detect and identify faults in a noisy environment, the robustness of the methodology is tested in higher depth. On the other hand, in real environments, it is always possible to perform a filtering of the data before using it in the model identification process. The decision to analyze the robustness studying what happens when noise is added to the sensors is a prerequisite to test VisualBlock-FIR with real data.

For these experiments the detection time window is set to 1 s, as it was in the previous section, whereas the identification time window has been increased from 6 to 10 s, and the cumulative errors needed to trigger an alarm remains to three. Table 4 shows the time needed by VisualBlock-FIR to detect and identify all faults and each of its four levels of noise studied, i.e. 60, 40, 30 and 20 dB. If we compare the detection results from Tables 3 and 4, it can be easily seen that when a noise of 60 dB and 40 dB is added to the signal, the detection time is almost the same than the one when the signal has no noise at all.

This makes sense since 60 dB and 40 dB correspond to only a 0.1% and 1% of noise, respectively. When the noise is increased, the time needed by VisualBlock-FIR to detect the fault is also

increased. However, the detection time is still acceptable when a 3.5% and a 10% of noise are added to the signal. In fact, it is very low compared to the results reported by Escobet et al. (2009).

Notice that when the percentage of noise increases more than 1% the enveloping signal is also increased in the percentage specified in the third column of Table 4. Remember that the FIR models obtained are very accurate thus, implying very narrow envelopes. Fig. 15 shows the enveloping signals obtained for fault #1 when a noise of 20 dB is added to the output signals. In this case, the percentage of increment of the enveloping is 8%, allowing fault detection and identification.

If we now compare the identification results from both tables, it can be seen that the identification time is increased from 6 to 10 s when the noise is added to the output signals. This is due to the fact that the identification time window has been increased from 6 to 10 s in order to get a good identification performance. Therefore, it is not possible to identify the faults in fewer seconds. However, the identification time obtained for all faults is very satisfactory and definitely lower than the time reported by other approaches for the same application (Escobet et al., 2009).

## 6. Conclusions

The VisualBlock-FIR fault diagnosis system is presented and applied to a fuel cell system. Five faults are studied in this work, i.e. an increase in the friction in the compressor motor, when the compressor motor suffers an overheating, the fluid resistance increases due to water blocking the channels or flooding in the diffusion layer, air leak in the air supply manifold and, finally, an increase in the voltage value below which the compressor motor does not turn.

For the first four faults, a detection time lower than a second and a half is accomplished and a correct identification is obtained in less than 7 s. Detection and identification of the last fault becomes more difficult due to the similarities in behavior between the compressor speed,  $\omega_{cm}$ , and stack voltage,  $V_{fc}$ , signals and that of the plant when working properly. For this fault, minimum detection times between 22 and 25 s and identification times between 28 and 31 s (i.e. absolute seconds, including detection

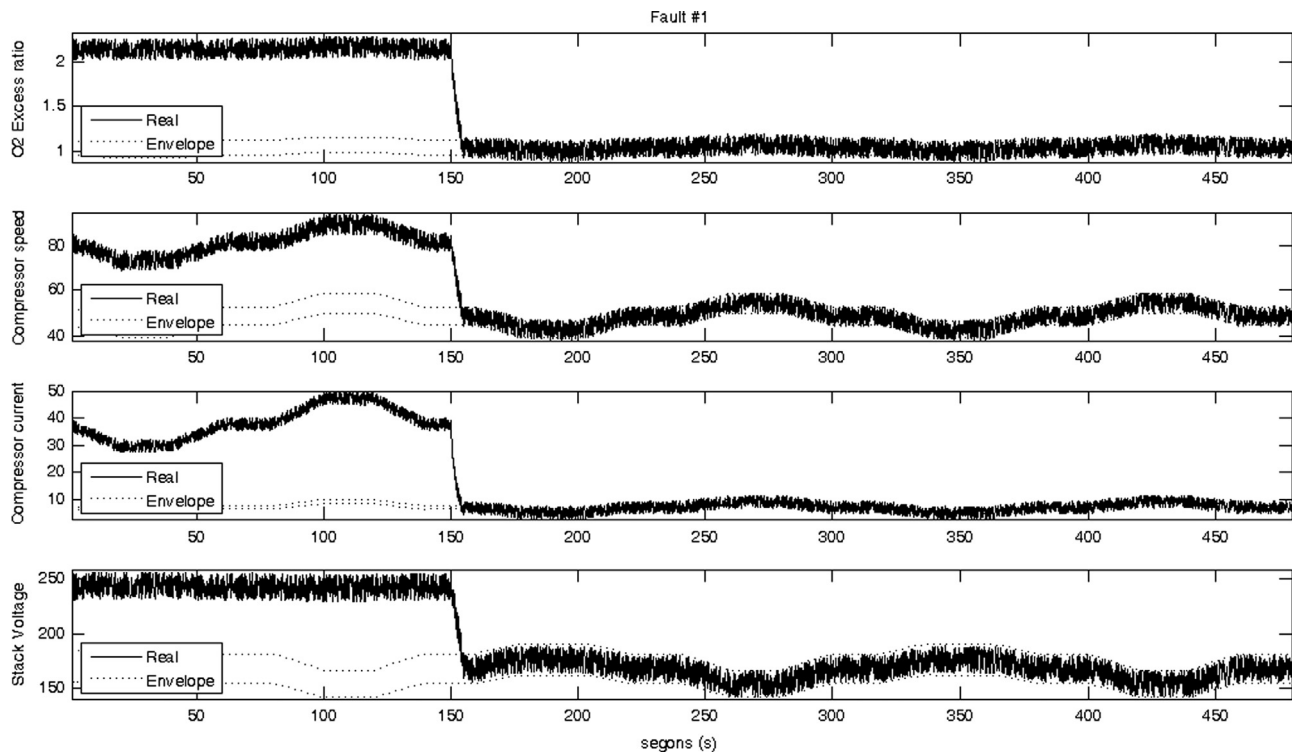


Fig. 15. Enveloping with an increment of 8% for fault #1, when a noise of 20 dB is added to the output signals.

time) have been found. These results compare favorably with those obtained by the computing residuals approach (Escobet et al., 2009). The robustness of VisualBlock-FIR has been also studied by adding different levels of noise to the output signals. The detection and identification time has been shown to increase slightly in the presence of noise. However, the performance is still very good for values of 60, 40 and 30 dB.

Here we have assumed that single fault occurs at a time. However, in the near future, we plan to extend this research to fault conditions where multifaults occur simultaneously. We strongly believe that the methodology developed in this work has the potential to be applied to real world fuel cell stacks and that it could be implemented for on-board applications. It should be taken into account that the identification of FIR models can be done out of line and that, once the model library is available, the FDS, i.e. VisualBlock-FIR, is able to perform the detection and identification in real time. Also notice that new models can be added to the fault model library anytime and no adjustment or reconfiguration of the FDS is required.

## References

- Aitouche, A., Yang, Q., Ould Bouamama, B., 2011. Fault detection and isolation of PEM fuel cell system based on nonlinear analytical redundancy. An application via parity space approach. *Eur. Phys. J. – Appl. Phys.* 54 (2), 23408 (11 pages).
- Aitouche, A., Olteanu, S.C., Ould Bouamama, B., 2012. A survey of diagnostic of fuel cell stack systems. In: Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safeprocess). Mexico City, Mexico, August 29–31, pp. 84–89.
- Becker, S., Karri, V., 2010. Predictive models for PEM-electrolyzer performance using adaptive neuro-fuzzy inference systems. *Int. J. Hydrog. Energy* 35 (18), 9963–9972.
- Chavez-Ramírez, A.U., Muñoz-Guerrero, R., Durón-Torres, S.M., Ferraro, M., Brunaccini, G., Sergi, F., 2010. High power fuel cell simulator based on artificial neural network. *Int. J. Hydrog. Energy* 35 (21), 12125–12133.
- Escobet, T., Feroldi, D., de Lira, S., Puig, V., Quevedo, J., Riera, J., Serra, M., 2009. Model-based fault diagnosis in PEM fuel cell systems. *J. Power Sources* 192, 216–223.
- Escobet, A., Nebot, A., Cellier, F.E., 2008. Visual-FIR: a tool for model identification and prediction of dynamical complex systems. *Simul. Pract. Theory* 16, 76–92.
- Feroldi, D., 2009. Control and Design of PEM Fuel Cell-based Systems (Ph.D.). Universitat Politècnica de Catalunya.
- Feroldi, D., Serra, M., Riera, J., 2007. Performance improvement of a PEMFC system controlling the cathode outlet air flow. *J. Power Sources* 169 (1), 205–212.
- Hissel, D., Péra, M.C., Kauffmann, J.M., 2004. Diagnosis of automotive fuel cell power generators. *J. Power Sources* 128 (2), 239–246.
- Hissel, D., Candusso, D., Harel, F.H., 2007. Fuzzy-clustering durability diagnosis of polymer electrolyte fuel cells dedicated to transportation applications. *IEEE Trans. Veh. Technol.* 4 (5), 1211–12420.
- Kamal, M.M., Yu, D.W., Yu, D.L., 2014. Fault detection and isolation for PEM fuel cell stack with independent RBF model. *Eng. Appl. Artif. Intell.* 28, 52–63.
- Kamal, M.M., Yu, D., 2011. Model-based fault detection for proton exchange membrane fuel cell systems. *Int. J. Eng. Sci. Technol.* 3 (9), 1–15.
- Klir, G.J., Elias, D., 2002. Architecture of Systems Problem Solving. Plenum Press, NY.
- Law, A., Kelton, D., 1990. Simulation Modeling and Analysis. McGraw-Hill, New York, NY.
- Lira, S., Puig, V., Quevedo, J., Husar, A., 2010. LPV model-based fault diagnosis using relative fault sensitivity signature approach in a PEM fuel cell. In: Proceedings of the 18th Mediterranean Conference on Control & Automation Congress. Palace Hotel, Marrakech, Morocco, June 23–25, pp. 1284–1289.
- Liu, W., Wang, X.G., 2003. Fault diagnosis of a PEM fuel cell system by sp model and on-line expert systems techniques. In: Proceedings of the IASTED International Conference on Modelling and Simulation. Palm Springs, CA, USA, February 24–26, pp. 188–191.
- Nitsche, C., Schroedl, S., Weiss, W., 2004. Onboard diagnostics concept for fuel cell vehicles using adaptive modeling. In: Proceedings of the IEEE Intelligent Vehicles Symposium. Parma, Italy, pp. 127–132.
- Olteanu, S.C., Aitouche, A., Ouedat, M., Jouni, A., 2012. PEM fuel cell modeling and simulation via the Takagi–Sugeno Fuzzy model. In: Proceedings of the International Conference on Renewable Energies for Developing Countries (REDEC). 28–29 November, pp. 1–7.
- Pukrushpan, J.T., Peng, H., Stefanopoulou, A.G., 2004. Control oriented modeling and analysis for automotive fuel cell systems. *Trans. ASME* 126, 14–25.
- Rajashekara, K., 2000. Propulsion system strategies for fuel cell vehicles. *Fuel Cell Technology for Vehicles*, pp. 179–187.
- Rosich, A., Sarrate, R., Nejari, F., 2014. On-line model-based fault detection and isolation for PEM fuel cell stack systems. *Appl. Math. Model.* 38 (11–12), 2744–2757.
- Tao, S., Si-jia, Y., Guang-yi, C., Xin-jian, Z., 2005. Modelling and control PEMFC using fuzzy neural networks. *J. Zhejiang Univ. Sci. A* 6 (10), 1084–1089.



- Vural, Y., Ingham, D.B., Pourkashanian, M., 2009. Performance prediction of a proton exchange membrane fuel cell using the ANFIS model. *Int. J. Hydrog. Energy* 34 (22), 9181–9187.
- Wang, H., Yuan, X.Z., Li, H., 2011. *PEM Fuel Cell Diagnostic Tools*. CRC Press, Boca Raton, FL, USA.
- Wu, J., Yuan, X.Z., Wang, H., Blanco, M., Martin, J.J., Zhang, J., 2008a. Diagnostic tools in PEM fuel cell research: Part I: electrochemical techniques. *Int. J. Hydrog. Energy* 33 (6), 1735–1746.
- Wu, J., Yuan, X.Z., Wang, H., Blanco, M., Martin, J.J., Zhang, J., 2008b. Diagnostic tools in PEM fuel cell research: Part II: physical/chemical methods. *Int. J. Hydrog. Energy* 33 (6), 1747–1757.
- Yousfi Steiner, N., Hissel, D., Moçotéguy, P., Candusso, D., 2011. Diagnosis of polymer electrolyte fuel cells failure modes (flooding and drying out) by neural networks modeling. *Int. J. Hydrog. Energy* 36 (4), 3067.
- Zhang, L., Huang, A.Q., 2011. Model-based fault detection of hybrid fuel cell and photovoltaic direct current power sources. *J. Power Sources* 196 (11), 5197–5204.
- Zheng, Z., Petrone, R., Péra, M.C., Hissel, D., Becherif, M., Pianese, C., Yousfi-Steiner, N., Sorrentino, M., 2013. A review on non-model based diagnosis methodologies for PEM fuel cell stacks and systems. *Int. J. Hydrog. Energy* 38 (21), 8914–8926.