Preprints of the 7th IFAC Symposium on
Fault Detection, Supervision and Safety of Technical Processes
Barcelona, Spain, June 30 - July 3, 2009

Fuel Cell System Diagnosis based on a Causal Structural Model

Albert Rosich * **Fatiha Nejjari** * **Ramon Sarrate** *

* *Automatic Control Department, Universitat Politècnica de Catalunya, Rambla de Sant Nebridi, 10, 08222 Terrassa, Spain (e-mail: albert.rosich@upc.edu).*

Abstract: In this work, a diagnosis system is developed and applied to a fuel cell stack system. The paper shows the significance of structural models to solve diagnosis issues in large scale systems. The diagnosis system based on residual generation by means of the computation of causal MSO sets (Minimal Structural Overdetermined) is capable of detecting and isolating faults in the fuel cell system.

1. INTRODUCTION

Industrial processes can be affected by systems faults, having a serious impact on them when those are not promptly detected and identified. In order to diagnose these faulty behaviors, efficient and advanced diagnosis systems are of great importance for modern industries. Considerable research has been done in developing such diagnosis systems (Blanke et al., 2006). Most approaches to fault detection and isolation are based on consistency checking. They involve the comparison between the observed behavior of the process and a model. Traditionally, diagnosis systems have been developed for linear systems by first linearizing the model and then applying robust techniques to the residual generator design (Chen and Patton, 1999).

In this work, a diagnosis system is developed and applied to a fuel cell stack system (Pukrushpan, 2003). Fuel cell systems are receiving much attention in the last decade as good candidates for clean electricity generation. Here, a fuel cell system benchmark is used and some faults are defined to be diagnosed. The fuel cell system model is very complex, involving a wide range of non-linear equations (lookup tables, polynomial functions, saturations, non-linear dynamic equations, etc.), and the operating point may also change. So, a model linearization approach is not feasible. Nevertheless, the non-linear model thoroughly describes the non faulty behavior for different operating points. Therefore, in this work, non-linear equations are directly applied in the residual generation design. Internal unknown variables are evaluated through the convenient manipulation of system equations, which leads to a relationship between variables and equations that compute them. This concept is known, in Blanke et al. (2006) as a causal interpretation of the computability. Few works focus this causal assignment in the fault diagnosis field. For instance, in Ploix et al. (2008) causality is taken into account in the computation of the set of testable submodels whereas in Svärd and Nyberg (2008) causality is considered in derivative and integral computations.

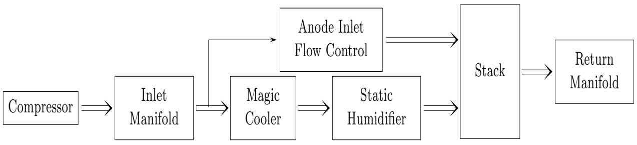

Fig. 1. Fuel Cell System scheme

The fuel cell system benchmark is briefly described in next section. The reader can find a more detailed description including the whole system equations, the full fault description and extended fault diagnosis simulations in Rosich et al. (2008).

2. FUEL CELL SYSTEM BENCHMARK

2.1 System description

The model of the PEM (Polymer Electrolyte Membrane) fuel cell system used in this work was proposed in Pukrushpan et al. (2004), and further information can be found in Pukrushpan (2003). The benchmark is widely accepted in the control community as a good representation of the behavior of a fuel cell system. The model, see Fig. 1, includes a very detailed description of the air compressor, the inlet and return cathode manifolds, the static air cooler, the static humidifier, the hydrogen flow and the PEM fuel cell stack. The fuel cell stack model is further decomposed in four main subsystems: stack voltage, cathode flow, anode flow and membrane hydration. In the model, it is assumed that the temperature is known and constant since its dynamic is much more slower than those of the rest of the model.

The available sensors in the system is another important issue to take into account for fault diagnosis. We assume the following sensors installed in the system: the stack current (i_{st}), compressor motor speed (ω_{cm}), compressor motor current (i_{cm}), compressor output pressure ($p_{cm,out}$), inlet manifold output temperature ($T_{im,out}$), cathode input pressure ($p_{ca,in}$), stack downstream pressure ($p_{st,ds}$), cathode output flow ($W_{ca,out}$) and cathode output pres-

* This work was supported in part by the Research Comission of the Generalitat of Catalunya (Grup SAC, ref. 2005SGR00537) and by CICYT (ref. DPI-2005-05415) of Spanish Ministry of Education

Table 1. Systems faults

	Fault description	**Occurrence time**
Fault 1	Electrical fault in the compressor motor	5 s
Fault 2	Mechanical fault in the compressor motor	7 s
Fault 3	Compressor fault. The relation described by the *compressor map* does not hold	10 s
Fault 4	Air leak in the *inlet manifold*	12 s
Fault 5	Humidifier fault. Output humidity does not follow desired humidity	8 s
Fault 6	Cathode *return manifold* fault	10 s
Fault 7	Exit cathode fault	14 s

sure ($p_{ca,out}$). Furthermore, there is a set of already known variables: the compressor voltage (v_{cm}) since it is a control variable, the desired temperature (T_{des}) and humidity (ϕ_{des}), both setpoints, the stack temperature (T_{st}) and all the ambient variables (pressure p_{amb}, temperature T_{amb} and humidity ϕ_{amb}).

2.2 Fault description

A set of faults has been defined for this benchmark. Each fault affects a primarily equation by changing a parameter or a variable, so that the relation between a fault and an equation is unique. Table 1 summarizes the faults considered in this work[1]. Other faults could be easily included in this set, that should be related to other model equations. Another assumption is that only single faults are allowed. This means that two or more faults can not occur in the system at the same time.

3. CAUSAL STRUCTURAL FRAMEWORK

3.1 Introduction

In a diagnosis system, consistency is checked by using a set of redundant sub-models. The equations of every redundant sub-model are combined to design what it is called a residual generator in the literature (Blanke et al., 2006). For linear systems, there exists a well developed theory (Chen and Patton, 1999). However, when non-linear equations are involved, this theory is not always applicable. One way to design a residual for these non-linear sub-models is to build them from the computation sequence (Blanke et al., 2006, Düştegör et al., 2006), which states how sub-model equations must be combined in a chained variable substitution in order to compute the residual. As an example, assume the following redundant sub-model (1), consisting of three equations (*i.e.*, e_1, e_2 and e_3), where y_1 and y_2 are known variables, and x_1 and x_2 are unknown variables.

$$\begin{array}{l} e_1 : x_1 = h_1(y_1) \\ e_2 : x_2 = h_2(x_1, y_2) \\ e_3 : h_3(x_1, x_2, y_1) = 0 \end{array} \tag{1}$$

Figure 4 shows a possible computation sequence that can be used to design a residual from the redundant sub-model (1), which assumes that equation e_1 solves variable x_1 and equation e_2 solves variable x_2.

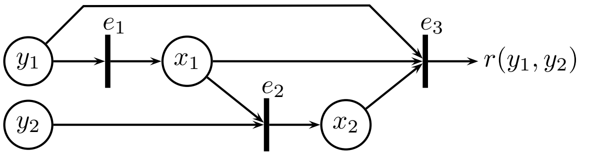

Fig. 2. Computation Sequence

From this computation sequence, it is straightforward to propagate the values to compute the residual as (2).

$$r(y_1, y_2) = h_3(h_1(y_1), h_2(h_1(y_1), y_2), y1) \tag{2}$$

where,

$$\begin{cases} r(y_1, y_2) \simeq 0 \text{ means that there is consistency} \\ r(y_1, y_2) \not\simeq 0 \text{ means that there is no consistency} \end{cases} \tag{3}$$

Using this procedure to obtain residuals in complex systems is an efficient technique. The nature of equations is not important and variables can be directly evaluated by propagating values through the computation sequence, avoiding the use of complex solver tools. However, solving a certain variable in a non-linear equation is not always possible, which ultimately poses restrictions on how to combine them when designing a residual generator. For instance, assume that equation e_2 in (1) has now the form $e_2 : h_2(x_1, x_2, y_2) = 0$, meaning that x_2 can not be expressed as an explicit variable in e_2. It is easy to see that now the value of x_2 has to be computed using other tools (numeric solvers, non-linear optimization, etc.) instead of value propagation from previously computed variables to the unknown variables. With such tools the solution is not always ensured and the computation cost can be large. This fact motivates the following approach based on a causal computability of variables when non-linear equations are involved.

An extended description of the following causal structural framework and its application to the sensor placement problem can be found in Rosich et al. (2009).

3.2 Causal Structural Model

Structural models (Blanke et al., 2006) are suitable to handle non-linear large-scale systems since efficient graph-based tools can be used to analyze them, which does not have numerical problems. However, only best case results are obtained. A structural model $G(M, X, A)$ is a coarse model description based on a bi-partite graph that can be defined from a set of model equations $M = \{\dots, e_i, \dots\}$, a set of variables $Z = \{\dots, z_j, \dots\}$ and the set of edges A, which states the following relation between equations and variables:

$$(e_i, z_j) \in A \text{ if variable } z_j \text{ appears in equation } e_i. \tag{4}$$

Extended information about structural modeling applied to fault diagnosis can be found in Krysander (2006) and Blanke et al. (2006).

[1] As already mentioned, a complete description of how these faults are modeled can be found in Rosich et al. (2008)

Given a structural model $G(M, X, A)$, edges in A will be qualified in order to take into account causal computability in the implementation of residuals based on the computation sequence. Causal computability accounts for how variables can be computed in an equation. In Section 3.1 it was remarked that, in non-linear equations, unknown variables can not always be computed as a function of the others. This leads to introduce Definition 1.

Definition 1. (Causal variable). Let $h(X) = 0$ be an equation of the model. Variable $x_i \in X$ is causal in h, if x_i can be computed using h, assuming that the remaining variables, $X \setminus x_i$, are known. We say that there is a causal relation between x_i and h.

From Definition 1 it holds that equation h can never be used in the computation sequence to compute non-causal variables.

Given a computation sequence, algebraic loops may also induce some restrictions on the computability of a set of variables. Consider the set of non-linear equations (5), where x_1 and x_2 are unknown variables and y_1 and y_2 are known variables. This set of equations form an algebraic loop.

$$\begin{array}{l} e_1 : x_1 - g_1(x_2, y_1) = 0 \\ e_2 : x_2 - g_2(x_1, y_2) = 0 \end{array} \tag{5}$$

It is clear that x_1 is a causal variable in e_1, whereas x_2 is a causal variable in e_2. Replacing variable x_2 in equation e_1 yields equation $e'_1 : x_1 - g_1(g_2(x_1, y_2), y_1) = 0$. Depending on the nature of equations e_1 and e_2, it turns out that x_1 could not necessarily be regarded as a causal variable in e'_1. In fact, non-linear algebraic loops could be tackled using complex tools such non-linear optimization that does no guarantee a solution. On the other hand, linear algebraic loops are easier to handle as long as linear coefficients are algebraically independent. Thus, in this work, algebraic loops will be accepted in a computation sequence, as long as it involves linear variables. This motivates Definition 2.

Definition 2. (Linear Variable). Let $h(X) = 0$ be an equation of the model. A set of variables $X_i \subseteq X$ is linear in h if h can be arranged as $\mathcal{L}(X_i)+g(X\setminus X_i) = 0$ and $|X_i| > 1$, where $\mathcal{L}$ is a linear function. We say that there is a linear relation between X_i and h.

Note that considering one single variable as a linear variable in an equation does not make sense. Linear relations are meant to be considered for identifying linear algebraic loops, and one single variable never forms a loop. Thus, linear relations are considered when two or more linear variables appear in the same equation.

Information on causal and linear relations can be well fitted in the structural model by an equivalent class partition on the set of edges $A = A_L \cup A_\times \cup A_\Delta$ where, according to the previous definitions:

- A_L is a subset of edges such that x_j is a linear variable in e_i.
- $A_\times$ is a subset of edges such that x_j is a causal (but not linear) variable in e_i
- A_Δ is the remaining subset of edges, where x_j is a non-causal variable in e_i

In the biadjacency matrix, edges in A_L are represented by an "L" symbol, edges in $A_\times$ are represented by a "$\times$" symbol and edges in A_Δ are represented by a "Δ" symbol.

Let us see some examples related to the fuel cell system model. Take, for instance, the non-linear equation $e_{p_{sat}}$ in (6) which is used, in each subsystem, to compute the saturation pressure from the gas temperature.

$$e_{p_{sat}} : log_{10}(p_{sat}) = a_4T^4 + a_3T^3 + a_2T^2 + a_1T + a_0 \tag{6}$$

This equation relates the saturation pressure p_{sat} in kPa and temperature T in °K (Pukrushpan, 2003). Coefficient values a_i should be estimated from measured data. According to equation $e_{p_{sat}}$, variable p_{sat} can be easily computed from T. However, numeric solvers should be used in order to compute variable T from p_{sat}, which are not advised, in this work. Thus, according to Definition 1, variable p_{sat} is a causal variable, whereas T is not. Therefore $(e_{p_{sat}}, p_{sat}) \in A_\times$ and $(e_{p_{sat}}, T) \in A_\Delta$.

Next, let us consider equation e_{67} in (7) which belongs to *fuel cell stack cathode* subsystem.

$$e_{67} : W_{O_2,reacted} = M_{O_2}\frac{nI_{st}}{4F} \tag{7}$$

This equation states that reacted oxygen flow rate $W_{O_2,reacted}$ is a function of stack current I_{st}, where M_{O_2} is the oxygen molar mass, n is the number of cells in the Stack and F is the Faraday constant (Pukrushpan, 2003). Thus according to Definition 2, both $W_{O_2,reacted}$ and I_{st} are linear variables. Therefore, $\{(e_{67}, W_{O_2,reacted}), (e_{67}, I_{st})\} \in A_L$.

3.3 Causal computability

First, assume that there are no linear relations, *i.e.* $A = \{A_\times \cup A_\Delta\}$. If there exists a complete matching $\mathcal{M}_G^X$ in X, such that $\mathcal{M}_G^X \subseteq A_\times$ and the well-constrained subgraph $G'(\partial^M \mathcal{M}_G^X, X)$ has no strong component with more than one equation, then the set of unknown variables, X, can be computed using the computation sequence. Note that $\partial^M \mathcal{M}_G^X$ denotes the subset of equations in M incident to edges in $\mathcal{M}_G^X$.

Algorithm 1 searches for the set of variables that can be computed as causal variables. This is iteratively done by finding equations that only contain one causal variable (the matching) and then pruning the graph, until no more equations-variable pairs can be found.

Algorithm 1 $\mathcal{X}$ = CausalVariable($G(M, X, A)$)

```
$X' = X$; $\mathcal{X} = \emptyset$;
while $\exists e \in M : |\text{var}_{X'}(e)| = 1 \wedge \{e, \text{var}_{X'}(e)\} \in A_\times$ do
    $X' = X' \setminus \text{var}_{X'}(e)$;
    $\mathcal{X} = \mathcal{X} \cup \text{var}_{X'}(e)$;
end while
```

Here, it is assumed that a set of variables can be solved if every variable can be matched with an equation and there are no algebraic loops.

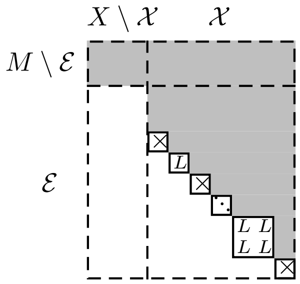

Fig. 3. Causal and linear computable decomposition

Now, assume that linear relations are considered (*i.e.* $A = \{A_L \cup A_\times \cup A_\Delta\}$), involving algebraically independent coefficients. Under this hypothesis, the Dulmage-Mendelsohn decomposition can be applied to determine whether a solution exists to a linear algebraic loop, and compute it. The Dulmage-Mendelsohn decomposition (Dulmage and Mendelsohn, 1958, Murota, 2000) defines a partition on the set of equations and the set of variables. This partition consists in the under-determined part, the just-determined part and the over-determined part, which contains the redundant equations.

The set of linear computable variables can be obtained by means of Algorithm 2. First, the subset of equations (E_L) that only depend on linear variables (X_L) is identified. Next, applying the Dulmage-Mendelsohn decomposition over $G(E_L, X_L, A_L)$, the set of linear computable variables, $\mathcal{X}_L$ is obtained as those variables that are in the just- and over-determined part. Note that this holds with the assumption that the linear coefficients are algebraically independent.

Algorithm 2 $\mathcal{X}_L = \text{LinearVariable}(G(M, X, A))$

```
$E_L = \{e \in M : \forall x \in \text{var}_X(e), \{e, x\} \in A_L\}$;
$\mathcal{X}_L = \text{var}_X(E_L^0) \cup \text{var}_X(E_L^+)$
```

The resulting structural decomposition is showed in Figure 3, where strong components with more than one equation does not involve causal relations but linear ones. White areas indicate no relation between equations and variables, whereas the shadowed ones mean that such a relation is possible. This decomposition is done by Algorithm 3, which iteratively alternates algorithms 1 and 2, and finally returns subgraph $G(\mathcal{E}, \mathcal{X}, A)$.

Algorithm 3 $\{\mathcal{X}, \mathcal{E}\} = \text{ComputableSystem}(G(M, X, A))$

```
$\mathcal{X} = \emptyset$;
**repeat**
    $\mathcal{X}' = \text{CausalVariable}(G(M, X \setminus \mathcal{X}, A))$;
    $\mathcal{X} = \mathcal{X} \cup \mathcal{X}'$;
    $\mathcal{X}_L = \text{LinearVariable}(G(M, X \setminus \mathcal{X}, A))$;
    $\mathcal{X} = \mathcal{X} \cup \mathcal{X}_L$;
**until** $\mathcal{X}' \cup \mathcal{X}_L = \emptyset$
$\mathcal{E} = \{e \in M : \text{var}_X(e) \subseteq (\mathcal{X})\}$
```

From the discussion above, it is clear that subgraph $G(\mathcal{E}, \mathcal{X}, A)$ contains the computable part of the model. Thus, all remaining equations, $M \setminus \mathcal{E}$, are not useful anymore, since they contain variables that can not be computed, $X \setminus \mathcal{X}$.

Remark that extracting the computation sequence given by subgraph $G(\mathcal{E}, \mathcal{X}, A)$, decomposed as in Figure 3, is straightforward since now the matching-diagonal establishes a true interpretation of which equation solves each variable.

3.4 Generating causal computable MSO

Finding redundant sub-systems for diagnosis is an important topic in the field of diagnosis based on structural models, there are several works devoted to this issue (Travé-Massuyès et al., 2006, Krysander et al., 2008, Ploix et al., 2008, Pulido and Gonzalez, 2004). A very efficient algorithm that computes the complete set of MSO (Minimal Structural Overdetermined) set was published in Krysander et al. (2008). An MSO set is a subset of model equations that is minimal redundant, i.e. no subset in an MSO set is redundant. Furthermore, the MSO set can be used to implement a residual generator. In this section, a modification of this algorithm is presented which only computes those MSO sets that can be used to generate a residual by means of the computation sequence. This kind of MSO set is called *causal computable MSO set.* Therefore, a causal computable MSO set is a MSO set that contains a computable structure, which means that it can be decomposed in the $(\mathcal{E}, \mathcal{X})$ structure depicted in Figure 3. For further details involved with the original algorithm see Krysander (2006).

Algorithm 4 $M_{MSO} = \text{findCCMSO}(G(M, X, A), R)$

```
1:  $M_{MSO} := \emptyset$;
2:  $\{\mathcal{X}, \mathcal{E}\} = \text{ComputableVariables}(G(M, X, A))$;
3:  **if** $(M \setminus \mathcal{E}) \cap R = \emptyset$ **then**
4:      $M = \mathcal{E}$;
5:      **if** $\bar{\varphi}M = 1$ **then**
6:          $M_{MSO} := M_{MSO} \cup M$;
7:      **else**
8:          **while** $R \not\supseteq M$ **do**
9:              Select an $e \in M \setminus R$;
10:             $E := M \setminus (M \setminus \{e\})^+$;
11:             **if** $E \cap R = \emptyset$ **then**
12:                 $R := R \cup E$;
13:                 $M' := M \setminus E$;
14:                 $M_{MSO} := M_{MSO} \cup \text{findCCMSO}(G(M', X, A), R)$;
15:             **else**
16:                 $R := R \cup E$;
17:             **end if**
18:         **end while**
19:     **end if**
20: **end if**
```

The new part in Algorithm 4 comprises steps 2:7, where Algorithm 3 is used to ensure that the computable part remains in the set of equations M. Since the causal computable MSO sets are a subset of all the MSO sets and the algorithm in Krysander et al. (2008) find all possible MSO sets, then Algorithm 4 finds all possible causal computable MSO sets. The structural model $G(M, X, A)$ defined in subsection 3.2 and $R = \emptyset$ are the inputs of this algorithm.

The fuel cell system presented in Section 2.1 is described as a causal structural model in Rosich et al. (2008) and applied to Algorithm 4. A set of 386 causal computable MSO sets is obtained. But not all this causal computable

Table 2. Relation between MSO sets and system faults

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
MSO_1							×
MSO_2						×	
MSO_3				×	×		
MSO_4			×	×			
MSO_5			×		×		
MSO_6		×		×			
MSO_7		×	×				
MSO_8	×						

MSO sets are needed to detect and isolate the system faults described in Table 1. So, a subset of 8 causal computable MSO has been selected. The equations that belong to each selected causal computable MSO set are described in (8). The selection is based on finding a reduced set that ensures fault detectability and isolability. Table 2 shows the causal computable MSO sets that are sensitive to each fault. An MSO set is sensible to a fault if the corresponding fault equation belongs to that MSO set. Note that all the faults are structurally detectable and structurally isolable from each other, according to Krysander and Frisk (2008).

$$\begin{aligned}
MSO_1 &= \{e_{75}, e_{124}, e_{125}, e_{126}\}\\
MSO_2 &= \{e_{31}, e_{32}, e_{33}, e_{36}, e_{81}, e_{103}, e_{104}, e_{105}, e_{109}, e_{116},\\
&\quad e_{124}, e_{125}\}\\
MSO_3 &= \{e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19}, e_{20}, e_{25}, e_{28}, e_{29},\\
&\quad e_{82}, e_{83}, e_{84}, e_{85}, e_{87}, e_{88}, e_{89}, e_{90}, e_{94}, e_{95}, e_{96}, e_{97},\\
&\quad e_{98}, e_{100}, e_{101}, e_{102}, e_{114}, e_{121}, e_{122}, e_{123}, e_{126}\}\\
MSO_4 &= \{e_7, e_9, e_{11}, e_{13}, e_{14}, e_{15}, e_{16}, e_{83}, e_{84}, e_{85}, e_{86}, e_{90},\\
&\quad e_{119}, e_{121}, e_{122}, e_{126}\}\\
MSO_5 &= \{e_7, e_9, e_{11}, e_{12}, e_{14}, e_{15}, e_{16}e_{18}, e_{19}, e_{20}, e_{25}, e_{28},\\
&\quad e_{29}, e_{82}, e_{83}, e_{84}, e_{85}, e_{86}, e_{87}, e_{88}, e_{89}, e_{94}, e_{95}, e_{96},\\
&\quad e_{97}, e_{98}, e_{100}, e_{101}, e_{102}, e_{114}, e_{119}, e_{121}, e_{122}, e_{123}\}\\
MSO_6 &= \{e_2, e_3, e_4, e_6, e_8, e_9, e_{13}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19}, e_{83},\\
&\quad e_{84}, e_{85}, e_{86}, e_{87}, e_{90}, e_{119}, e_{120}, e_{121}, e_{122}, e_{126}\}\\
MSO_7 &= \{e_2, e_3, e_4, e_6, e_7, e_8, e_9, e_{11}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19},\\
&\quad e_{83}, e_{84}, e_{85}, e_{86}, e_{87}, e_{119}, e_{120}, e_{121}, e_{122}\}\\
MSO_8 &= \{e_1, e_{117}, e_{119}, e_{120}\} \qquad (8)
\end{aligned}$$

Table 3. Computable structure of MSO_2

	x	$\dot{x}$	$W_{om,out}$	$p_{om,out}$	$W_{om,in}$	$T_{om,in}$	$p_{om,ds}$	$T_{ca,out}$	T_{st}	$p_{st,ds}$	$W_{ca,out}$
e_{31}	L			L							
e_{32}	L	L									
e_{33}		L	L		L						
e_{36}			×	Δ		Δ	Δ				
e_{116}				L						L	
e_{104}					L						L
e_{105}						L		L			
e_{103}							×				
e_{81}								L	L		
e_{109}									×		
e_{124}										×	
e_{125}											×

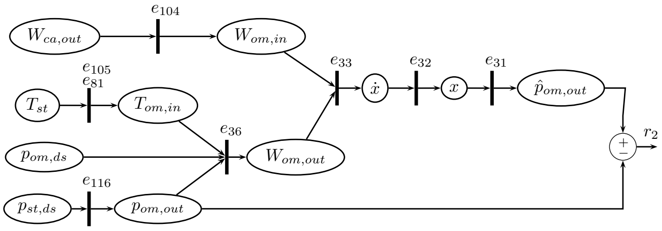

Fig. 4. Computation sequence from MSO_2

For instance, consider the causal computable MSO set MSO_2. The system equations that belong to this MSO set are listed in (9). Variables $p_{st,ds}$, T_{st} and $W_{ca,out}$ were already described in Section 2.1, and $p_{om,ds}$ is the downstream outlet manifold pressure, which equals to the ambient pressure. According to Section 2.1 they are all known (since they are measured or constant). Equations $\{e_{31}, e_{32}, e_{33}, e_{36}\}$ belong to the *outlet manifold* component. The causal structural model corresponding to (9) is depicted in Table 3. Equation e_{36} is a non-linear function that computes the output manifold flow. This function is not invertible, so it cannot be used to compute any of the input variables, which is consistent with the row assigned to e_{36} in Table 3. The same reasoning holds for each causal computable MSO set in (8). Remark that the structure in Table 3 can be decomposed as in Figure 3 (*i.e.*, it is a computable structure).

$$\begin{aligned}
&e_{31}: p_{om,out} = x\\
&e_{32}: \dot{x} = \frac{dx}{dt}\\
&e_{33}: \dot{x} = T_{om,in}\frac{R}{V_{om}}(W_{om,in} - W_{om,out})\\
&e_{36}: W_{om,out} = \text{NonlinearNozzle}(p_{om,out}, p_{om,ds}, T_{om,in})\\
&e_{81}: T_{ca,out} = T_{st}\\
&e_{103}: p_{om,ds} = 1atm\\
&e_{104}: W_{om,in} = W_{ca,out}\\
&e_{105}: T_{om,in} = T_{ca,out}\\
&e_{109}: T_{st} = 353K\\
&e_{116}: p_{om,out} = p_{st,ds}\\
&e_{124}: p_{st,ds} = p_{st,ds_{measured}}\\
&e_{125}: W_{ca,out} = W_{ca,out_{measured}}
\end{aligned} \qquad (9)$$

3.5 Causal residuals generator

Once the set of causal computable MSO sets is obtained, residuals can be easily implemented following the computation sequence. Note that the diagonal in the computable decomposition (see Figure 3) shows how every variable has to be computed. Therefore, implementing the computation sequence for a causal computable MSO set is straightforward. For instance, consider MSO_2, detailed in the previous section. The corresponding computation sequence is depicted in Figure 4 [2]. This computation sequence corresponds to the evaluation of residual r_2.

[2] Sensor equations $\{e_{124}, e_{125}\}$ and constant assignments $\{e_{103}, e_{109}\}$ have been omitted in order to make the figure simpler and readable.

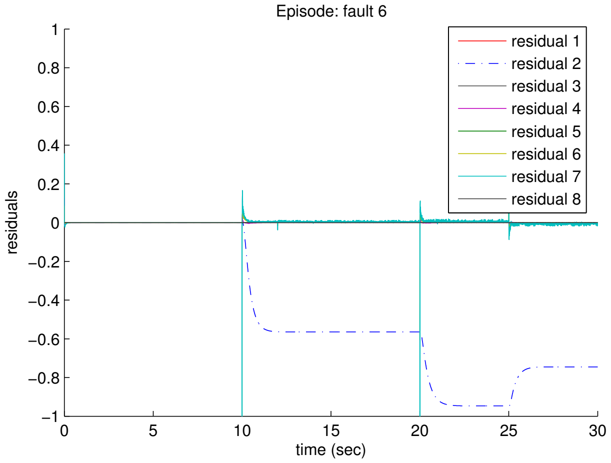

Fig. 5. Normalized residuals response for *fault 6*

Fault 6 has been simulated by an abrupt change of parameter V_{om} in equation e_{33}, from nominal value $5 \cdot 10^{-3} m^2$ to $4,5 \cdot 10^{-3} m^2$. As long as *fault 6* does not occur, $r_2 \simeq 0$ (*i.e.*, assuming model inaccuracies). However, when *fault 6* occurs, equation e_{33} will not hold and consequently $r_2 \not\simeq 0$, signaling the fault (*i.e.*, fault detection). Furthermore, according to (8), r_2 is the only residual which is sensible to *fault 6*. This is consistent with Table 2, and implies that a violation of residual r_2 indicates that *fault 6* has occurred (*i.e.*, fault isolation).

The same procedure has been followed for the other selected causal computable MSO sets in (8). Finally, a bank of eight residual generators, that are sensible to a different fault signature, according to Table 2, has been implemented.

4. FUEL CELL SYSTEM DIAGNOSIS

The faults presented in subsection 2.2 have been introduced in the benchmark simulator developed in Pukrushpan (2003). A simulation for the seven fault episodes (one for each system fault) has been run. The residuals signals have been generated and analyzed for fault detection and isolation using Table 2, and all faults have correctly been diagnosed.

For instance, Figure 5 shows the residuals response corresponding to *fault 6* episode (normalized in the interval $[-1, 1]$). Remark that all residuals except r_2 remain around zero: at time 10 s *fault 6* occurs and r_2 is the only residual activated. Therefore, fault detection is accomplished. As already discussed in subsection 3.5, the only fault that can affect r_2 is *fault 6*. Thus, the diagnosis system concludes that the fault that has occurred is indeed *fault 6*, which turns out to be true.

5. CONCLUSIONS

The results presented in this work show that faults can be detected and isolated by combining system equations and taking into account the causal computability of the variables involved in these equations. Working with structural models demonstrates that the nature of equations is not important to accomplish diagnosis specifications. However, some future work could be done to improve the results, *e.g.* residual threshold, dynamic state observation, etc.

Another important issue treated in this paper is the efficient generation of causal computable MSO sets. In general, its computational complexity increases exponentially with the number of equations. However, in this paper, an improvement has been done, since there is no need to find all the MSO sets and later reject those that are not causal computable. Instead, Algorithm 4 recursively generates all causal computable MSO sets.

REFERENCES

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control.* Springer, 2nd edition, 2006.

J. Chen and R. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems.* Kluwer Academic Publishers, Boston, 1999.

A. L. Dulmage and N. S. Mendelsohn. Covering of bipartite graph. *Canada J. Math*, 10:527–534, 1958.

D. Düştegör, E. Frisk, V. Cocquempot, Mattias Krysander, and Marcel Staroswiecki. Structural analysis of fault isolability in the DAMADICS benchmark. *Contr. Eng. Pract.*, 14:597–608, 2006.

M. Krysander. *Design and Analysis of Diagnosis Systems Using Structural Analysis.* PhD thesis, Linköping Univ., Linköping, Sweden, June 2006.

M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *IEEE Trans. Syst., Man, Cybern. A*, 38(6): 1398–1410, 2008.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained subsystems for model-based diagnosis. *IEEE Trans. Syst., Man, Cybern. A*, 38(1), 2008.

K. Murota. *Matrices and Matroids for Systems Analysis.* Springer, 2000.

S. Ploix, A. A. Yassine, and J. M. Flaus. An improved algorithm for the design of testable subsystems. Proc. of 17th IFAC World Congress, Seoul, Korea, 2008.

J. T. Pukrushpan. *Modeling and Control of Fuel Cell Systems and Fuel Processors.* PhD thesis, Univ. of Michigan, Ann Arbor, Michigan, 2003.

J. T. Pukrushpan, H. P., and A. G. Stefanopoulou. Analysis for automotive fuel cell systems. *Transactions of the ASME*, 126:14–25, 2004.

B. Pulido and C. A. Gonzalez. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. Syst., Man, Cybern. B*, 34(5):2192–2206, October 2004.

A. Rosich, R. Sarrate, and F. Nejjari. Fuel cell system benchmark. Technical report, Automatic Control Department, 2008. URL `http://websac.upc.es`.

A. Rosich, R. Sarrate, and F. Nejjari. Sensor placement for fault diagnosis based on causal computations. In *Proceedings of IFAC Safeprocess'09*, Barcelona, Spain, July 1–3, 2009.

C. Svärd and M. Nyberg. A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory. 19th International Workshop on Principles of Diagnosis (DX-08), Blue Mountains, Australia, 2008.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Trans. Syst., Man, Cybern. A*, 36(6):1146–1160, 2006.