Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Efficient Optimal Sensor Placement for Model-based FDI using an Incremental Algorithm

Albert Rosich, Ramon Sarrate, Vicenç Puig and Teresa Escobet

Abstract— **The problem of optimal sensor placement for FDI consists in determining the set of sensors that minimizes a pre-defined cost function satisfying at the same time a pre-established set of FDI specifications for a given set of faults. Existing approaches are mainly based on formulating an optimization problem once the sets of all possible ARRs has been generated, considering all possible candidate sensors installed. However, the associated computational complexity is exponential with the number of possible sensors. The main goal of this paper is to propose an incremental algorithm for FDI sensor placement that tries to avoid the computational burden. To show the effectiveness of this approach, an application based on a fuel-cell system is proposed.**

I. Introduction

Process faults, if undetected, have a serious impact on process economy, product quality, safety, productivity and pollution level. In order to detect, diagnose and correct these abnormal process behaviors, efficient and advanced automated diagnostic systems are of great importance to modern industries. Considerable research has gone into the development of such diagnostic systems [1]. Most approaches for fault detection and isolation (FDI) in some sense involve the comparison of the observed behavior of the process to a reference model. The process behavior is inferred using sensors measuring the important variables in the process. Hence, the efficiency of the diagnostic approach depends critically on the location of sensors used for monitoring process variables. The emphasis of most of the work on model-based fault diagnosis has been more on procedures to perform diagnosis given a set of sensors and less on the actual location of sensors for efficient identification of faults. The problem of sensor placement for FDI consists in determining the optimal set of instruments such that a predefined set of faults are detected and isolated. In many cases, this set is defined in order to design some remedial actions such that the control loop is able to continue operating (fault-tolerant control). The usual objective to minimize in the sensor placement problem is the sensor cost. There are several articles devoted to the study of the design of sensor networks using goals corresponding to normal monitoring operation. Aside from cost, different other objective functions such as precision, reliability, or simply observability were used. Different techniques were also used, such as graph theory, mathematical programming, genetic algorithms and multiobjective optimization, among others. The problem has also been extended to incorporate upgrade considerations and maintenance costs. In [2], Bagajewicz reviews all these methods, but noticed that the problem of sensor placement in the model-based FDI community is still an open problem. However, some contributions have already been done in this direction [3], [4], [5], [6], [7], [8], [9], among others. In [6], the sensor placement problem is solved by the analysis of a set of possible Analitical Redundacy Relations (ARR) using algorithms of cycle generation in graphs. More recent approaches consist in finding the set of all possible ARRs under the assumption that all possible sensors are installed [3]. For sensor placement it is required to use an ARR generation algorithm that is complete. Otherwise, the sensor placement could exclude from consideration some sensor configurations just because some ARRs have not been generated. The excluded configurations could provide better FDI results that the ones that have been generated. Or, even in some dramatic cases, the sensor placement could not find solution because of this lack of completeness, when in fact if all ARRS were generated the solution would have been found. Just recently, several exhaustive methods have been developed that claim that are able to generate the complete set of ARRs [10], [11], [12].

In [13], optimal sensor placement for model-based FDI require to find the set of all possible analytical redundancy relations (ARRs), considering that all possible candidate sensors are installed. Then, a set of sensors that minimizes the total cost of the network is selected such that the resulting ARRs satisfy that a pre-established set of faults can be detected and isolated. However, the associated computational complexity is exponential with the number of possible sensors. The main goal of this paper is to propose an efficient algorithm that tries to avoid the practical limitations of the approach presented in [13]. This algorithm solves incrementally the optimal sensor location problem, by adding at each iteration one sensor and just generating the set of ARRs that are required. The effectiveness of this approach is illustrated through an application based on a fuel-cell system.

The structure of the paper is as follows. In Section II, the sensor placement problem for FDI is presented. Section III presents an incremental approach to optimal sensor placement problem. Section IV presents the proposed algorithm. Finally, in Section V, the effectiveness of the proposed approach is demonstrated with an application on a fuel-cell system.

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

All authors are with the Automatic Control Department, Universitat Politècnica de Catalunya, Rambla de Sant Nebridi, 10, 08222 Terrassa, Spain albert.rosich@upc.edu

1-4244-1498-9/07/$25.00 ©2007 IEEE.

II. Sensor Placement Problem

A. Problem set-up

Model-based FDI is based on comparing the real system behavior measured using a set of sensors with the estimated behavior obtained using a set of analytical redundancy relations (ARRs). The difference between the real and estimated behavior is known as residual. The ARRs comes from the set of model equations M describing system behaviors and the set of installed sensors J. They present a fault signature matrix (FSM) with respect to a pre-specified set of faults F. In particular, this matrix stores the influence of the considered faults on the residual set: an ij-element of the matrix contains the pattern 1 if fault f_j is expected to affect ARR$_i$, otherwise it is equal to 0. In this matrix, fault signatures establish fault detection and isolation properties of the considered set of faults F. Fault isolation using FDI algorithms just consists on finding the observed fault signature (vector of residuals) that is the closest to the fault signature column stored in the fault signature matrix.

In [4], the set of ARRs are generated using algebraic methods and transformations of the primary set of equations that comes directly from elementary model equations. On the other hand, in [1], they are generated using structural analysis and perfect matching algorithms on directed graphs. In order to present the sensor placement problem for FDI, a set of preliminary definitions are introduced.

Definition 1 (Configuration Sensor ARR set): Given a system modeled by a set of equations M and a set of installed sensors J, then $ARR(J)$, is the set of all analytical redundancy relations that exclusively use the whole set of sensors J.

Note that $ARR(J)$ does not include ARRs which use a proper subset of J.

Let 2^J be the set of all subsets of J and $\mathcal{J} \subseteq 2^J$ be a set of sensor sets. Then, $ARR(\mathcal{J})$ will denote the set of all analytical redundancy relations that are a function of any element of $\mathcal{J}$. This set can be incrementally built in the following way:

$$ARR(\mathcal{J}) \triangleq \bigcup_{J \in \mathcal{J}} ARR(J) \tag{1}$$

The sensor placement problem for FDI can be formulated in the following way:

Definition 2 (Sensor placement for Model-Based FDI): Given a system described by a set of model equations M, a set of candidate sensors I, and a set of FDI specifications S, the problem of sensor placement for FDI consists on finding a subset of sensors $J \subseteq I$ such that the set of FDI specifications S is fulfilled.

For each particular sensor configuration J, a function (algorithm) P can be defined such that it verifies whether the corresponding fault signature matrix FSM satisfies the set of FDI specifications according to:

$$P(M, J, S) = \begin{cases} 1 \text{ if FDI specifications } S \text{ are fulfilled} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The specifications S can be any set of required properties of FSM to design a diagnosis system. Fault detectability and isolability are examples of FDI specifications that can be assessed by inspection of FSM, according with the following definitions:

- *Fault Detectability*: A set of faults are detectable if their effects on the system can be observed on the available set of ARRs.
- *Fault Isolability*: A set of faults are (fully) isolable if their effects on the system can be discriminated one of each other considering the available set of ARRs.

These specifications can be easily verified by examining the fault signature matrix FSM. A fault f_j is detectable if there is, at least, a 1 present in the j^th-column of FSM. Two faults f_j and f_k are isolable if the j^th-column and the k^th-column of FSM are different.

Function P computes the matrix FSM, that results from the set of all possible ARRs for the considered sensor configuration J, $ARR(2^J)$. So, an ARR generation algorithm to generate such set of ARRs with the completeness property, is required.

Let I be the set of candidate sensors. Function P introduced in (2) induces a two-class partition in the set 2^I.Let $[2^I]^+$ be the set of all the subsets of sensors that are feasible solutions of the sensor placement problem for FDI. If each $J \in [2^I]^+$ has associated a cost, we can get a $J^* \in [2^I]^+$ such that it corresponds to the optimal solution. An easy way to associate a cost to a given configuration of sensors J is accomplished by associating a cost to each single sensor. This cost can be related to price, reliability, etc.

Definition 3 (Sensor Cost): Let $i \in I$ be any sensor. Then, this sensor has associated a cost, $C(i) > 0$.

Definition 4 (Configuration Sensor Cost): Let $J \subseteq I$ be a subset of sensors. Then, the cost of J is defined by:

$$C(J) \triangleq \sum_{j \in J} C(j) \tag{3}$$

These definitions lead to introduce the following definition:

Definition 5 (Optimal sensor placement problem): Given a system described by a set of model equations M, a set of candidate sensors I, their cost function $C(i)$ and a set of specifications for FDI S, the optimal sensor placement problem can be formulated as:

$$\begin{aligned} &\min_{J \subseteq I} \quad C(J) \\ &subject\ to: \\ &\quad P(M, J, S) = 1 \end{aligned} \tag{4}$$

where P has been introduced in (2).

B. Drawbacks of existing algorithms

One possible way to solve the optimal sensor placement problem (5) consists in generating the set $ARR(2^I)$ for all possible candidate sensors I [13]. This approach will be called in the following ***absolute approach***, in order to differentiate it from the approach proposed in the present

paper, that will be called ***incremental***. From this set of ARRs, the associated fault signature matrix can be obtained. Then, a binary discrete optimization problem can be formulated introducing a binary vector q, that denotes which candidate sensors are installed or not (*i.e*, means that i-sensor in the candidate sensor set I is installed, otherwise it is not installed). Moreover, the set of FDI specifications that the resulting fault signature matrix, obtained from the selected candidate sensors, should fulfil can be easily expressed as linear constraints relating the elements of vector q. Then, using mixed integer optimization algorithms based on branch and bound methods can be solved. Alternatively, this problem can be formulated to be solved using genetic algorithms [3]. However, this approach suffers from the curse of dimensionality since the number of ARRs grows exponentially with the number of possible sensors. This fact has led to imagine a more efficient way of solving the optimal sensor placement problem that avoids the generation of the set of all possible ARRs considering all possible sensor candidates.

III. Incremental Approach

A. The basic idea

The goal of the incremental approach to sensor placement is the same as the absolute algorithm presented in the previous section. That is, to solve the problem presented in (4). However, the basic difference of the incremental approach proposed here is on how the implementation of function P is done. In particular, the incremental approach does not require generating beforehand the complete set of all possible ARRs ($ARR(2^I)$) in order to solve the optimization problem (4). But instead, the optimization problem (4) is solved in such a way that at each iteration, only the subset of necessary ARRs, $ARR(2^J)$, to test the subset of sensors $J \subseteq I$ is generated online, re-using previously generated ARRs.

Let us consider that at a given iteration of the incremental algorithm a subset of sensors $J \subseteq I$are considered as possible candidates to be installed. Then, only the set of ARRs given by $ARR(2^J)$ are needed to test if J is a solution. The solution that the absolute approach would have produced would be the same but at a higher computational cost. This is because it finds the subset of necessary ARRs by removing all unnecessary ARRs, $ARR(2^I \backslash 2^J)$, from the complete set of ARRs,

$$ARR(2^J) = ARR(2^I) \backslash ARR(2^I \backslash 2^J) \tag{5}$$

whereas the incremental algorithm finds the subset of necessary ARRs directly without having to generate the complete set of ARRs.

B. Incremental computation of ARR

Moreover, generating the set of for each subset $ARR(2^{J_k})$, at each iteration k of the incremental algorithm, is not the most efficient way to work, since several ARR are generated several times. This leads to incrementally generate the set of required ARR as well as re-using the ones previously computed.

Let $J_k, J_l \subseteq I$ be two subsets of sensors such that $J_l \subseteq J_k$, then, it can be easily seen that:

$$ARR(2^{J_l}) \subseteq ARR(2^{J_k}) \tag{6}$$

and, moreover, the set $ARR(2^{J_k})$ can be expressed as:

$$ARR(2^{J_k}) = ARR(2^{J_l}) \cup ARR(2^{J_k} \backslash 2^{J_l}) \tag{7}$$

Thus, if subset $ARR(2^{J_l})$is ensured to have been previously computed, the only subset needed to be generated will be $ARR(2^{J_k} \backslash 2^{J_l})$. Following this guideline, no ARR is computed more than once. So, this methodology is more efficient than computing $ARR(2^{J_k})$ for each subset $J_k \subseteq I$.

C. Optimal subset of sensors

From Definitions 3 and 4, given two subsets of sensors $J_k, J_l \subseteq I$, the following holds:

$$J_l \subset J_k \Rightarrow C(J_l) < C(J_k) \tag{8}$$

The incremental algorithm, in the k^{th}-iteration, chooses a subset of sensors J_k:

$$J_k = \arg \min_{J \in \mathcal{J}_k} C(J) \tag{9}$$

where $\mathcal{J}_k = 2^I \backslash \overset{k-1}{\underset{l=1}{\cup}} \{J_l\}$.

According to (9), the subset of sensors J_k is the subset with minimal cost of all subsets not chosen in previous iterations.

Notice that if $P(M/J_k) = 1$ (see (2)), then J_k is an optimal solution to the optimal sensor placement problem stated in (4) and the algorithm finishes.

D. Computation of new ARRs

Computing $ARR(2^{J_k})$ implies to know which subsets of $ARR(2^{J_k})$ have been generated in previous iterations (see III-B). Moreover, the choice of a subset of sensors, in the incremental algorithm, is based on the cost computed as indicated above. Therefore, according to (8) all the subsets $J \in 2^{J_k} \backslash \{J_k\}$ have a subset sensor cost $C(J)$ such that $C(J) < C(J_k)$. So, according to (9) all $J \in 2^{J_k} \backslash \{J_k\}$ have been tested before. Two conclusions can be drawn:

- No subset $J \in 2^{J_k} \backslash \{J_k\}$ is a solution.
- $ARR(2^{J_k} \backslash \{J_k\})$ has been generated previously.

Next, according to (7):

$$ARR(2^{J_k}) = ARR(2^{J_k} \backslash \{J_k\}) \cup ARR(J_k) \tag{10}$$

Now, just $ARR(J_k)$ needs to be generated in the k^{th}-iteration to compute $ARR(2^{J_k})$.

IV. Incremental Algorithm Implementation

A. Model representation

A real system without faults can be expressed as a set of equations:

$$f(X) = 0 \tag{11}$$

where X is the set of unknown internal variables. In this paper, system faults are only considered as unknown variables

that affect the normal system behavior [10]. Thus, a system model with faults is represented as:

$$M_{system} : \begin{cases} f(X,F) = 0 \\ F = 0 \end{cases} \quad (12)$$

Using this representation, as soon as a system fault (f_i) occurs, equation $e_{f_i} : f_i = 0$ becomes inconsistent. However, the system equations $f(X,F) = 0$ remain consistent.

The subset of sensors J has to be modeled by a set of equations. This leads to next definition:

Definition 6 (Sensor Equation): Let $j \in J$ be a sensor that measures the value of internal variable x_j , then a sensor equation for sensor j can be defined as:

$$e_j : y_j = x_j \quad (13)$$

Variable y_j represents a known system variable. Equation e_j becomes inconsistent whenever sensor j is faulty. In those cases where a sensor measures a function of more than one unknown variable, just include an extra equation in the model [9] and the corresponding sensor equation $x_{new} = h(x)$.

Let J be a set of sensors, then the corresponding set of sensor equations is:

$$M_J = \bigcup_{j \in J} \{e_j\} \quad (14)$$

To generate the set $ARR(J_k)$, the system model has to be reconfigured at each iteration, adding the set of sensor equations associated to the sensors in J_k. Thus, a reconfigured model $M^*_{J_k}$ for a subset of sensors J_k can be expressed from (12) and (14) as:

$$M^*_{J_k} = M_{system} \cup M_{J_k} \quad (15)$$

B. Optimal Incremental Algorithm

Algorithm 1 solves incrementally the optimal sensor placement problem described in Definition 5, This is achieved using the ideas presented in previous sections.

Algorithm 1 $J^* = SensorPlacement(M, I, C(i), S)$

1: **repeat**
2: Search the optimal subset of sensors $J_k \subseteq I$, according to (9).
3: Build the corresponding set of sensor equations M_{J_k},according to (14).
4: Reconfigure the model $M^*_{J_k}$, according to (15).
5: Generate the subset $ARR(J_k)$
6: Store $ARR(J_k)$.
7: Obtain $ARR(2^{J_k})$ by retrieving the set of previously computed ARRs:
$ARR(2^{J_k}) = \bigcup_{j \subseteq J_k} ARR(j)$
8: Generate the fault signature matrix FSM_{J_k} from $ARR(2^{J_k})$.
9: **if** FSM_{J_k} satisfies the FDI specifications **then**
10: $P(M, J_k, S) \Leftarrow 1$
11: **else**
12: $P(M, J_k, S) \Leftarrow 0$
13: **end if**
14: **until** $P(M, J_k, S) = 1$ or $J_k = I$
15: **if** $P(M, J_k, S) = 0$ **then**
16: *There is no solution*
17: **else**
18: $J^* \Leftarrow J_k$
19: **end if**

TABLE I
STRUCTURAL MODEL M_{system}

	x_1	x_2	x_3	x_4	f_1	f_2
e_1	1	0	1	0	1	0
e_2	0	1	1	1	0	1
e_3	1	1	0	1	1	0
e_4	0	0	0	0	1	0
e_5	0	0	0	0	0	1

TABLE II
STRUCTURAL SENSOR EQUATION, M_{j_1}

	x_1	x_2	x_3	x_4	f_1	f_2
e_{j_1}	1	0	0	0	0	0

Step 2 of Algorithm 1 deals with the minimal configuration sensor cost search in (9) whereas steps 4-13 implement and evaluate $P(M, J, S)$ presented in (2), that assesses whether the current sensor configuration satisfies the set of FDI specifications. In step 5, an adaptation of the algorithm proposed in [11] has been used.

C. Academic example

Now, an academic example is presented to show how Algorithm 1 works. This is useful to view the structural reconfiguration of the model and the generation of the fault signature matrix FSM. Nevertheless, the optimal search of a subset of sensors (Step 2) is not presented in this section.

Assume the following linear static system:

$$M_{system} : \begin{cases} e_1 : x_1 + x_3 + f_1 = 0 \\ e_2 : x_2 + x_3 + x_4 + f_2 = 0 \\ e_3 : x_1 + x_2 + x_4 + f_1 = 0 \\ e_4 : f_1 = 0 \\ e_5 : f_2 = 0 \end{cases} \quad (16)$$

This system model has four internal variables that can be measured $X = \{x_1, x_2, x_3, x_4\}$ and two system faults $F = \{f_1, f_2\}$, included in the system model as in (12). Table I shows the structural model that corresponds to (16).

In this case, the set of possible sensors is considered to be the whole set of internal variables X, $I = \{j_1, j_2, j_3, j_4\}$, where it is assumed that sensor j_i is used to measure internal variable x_i.

The sensor placement problem in Definition 2 also requires the FDI specifications S to be formulated. In this example, the required FDI specifications are that system faults F and installed sensors faults are detectable and isolable, according to the definitions introduced in section II-A.

Assume that the first optimal subset of sensors is $J_1 = \{j_1\}$, where j_1 denotes a sensor that measures internal variable x_1. The reconfigured model $M^*_{J_1}$ (Step 4) is obtained adding row e_{i_1} in Table II to Table I.

Once the model is reconfigured, $ARR(J_1)$ can be generated (Step 5). In this specific case, $ARR(J_1) = \emptyset$ which clearly does not satisfies the FDI specifications. That would conclude first iteration of Algorithm 1. After 8 iterations,

TABLE III
ARR GENERATED AFTER 8 ITERATIONS

k	J_k	e_1	e_2	e_3	e_4	e_5	e_{j_1}	e_{j_2}	e_{j_3}	e_{j_4}
1	$\{j_1\}$	-	-	-	-	-	-	-	-	-
2	$\{j_2\}$	-	-	-	-	-	-	-	-	-
3	$\{j_3\}$	-	-	-	-	-	-	-	-	-
4	$\{j_4\}$	-	-	-	-	-	-	-	-	-
5	$\{j_1, j_2\}$	1	1	1	1	1	1	1	0	0
6	$\{j_1, j_4\}$	1	1	1	1	1	1	0	0	1
7	$\{j_2, j_4\}$	1	1	1	1	1	0	1	0	1
8	$\{j_1, j_2, j_4\}$	0	0	1	1	0	1	1	0	1
		1	1	0	1	1	1	1	0	1
		1	1	1	0	1	1	1	0	1

TABLE IV
FAULT SIGNATURE MATRIX FOR J_8

	f_1	f_2	f_{j_1}	f_{j_2}	f_{j_4}
ARR_1	1	1	1	1	0
ARR_2	1	1	1	0	1
ARR_3	1	1	0	1	1
ARR_4	1	0	1	1	1
ARR_5	1	1	1	1	1
ARR_6	0	1	1	1	1

Table III is obtained. In this table an ij-element is 1 if equation in column j is needed to compute ARR in row i, and 0 otherwise.

Proceeding with the algorithm execution, at iteration $k = 8$, the set $ARR(2^{J_8})$ is computed as (Step 7):

$$ARR(2^{J_8}) = ARR(J_1) \cup ARR(J_2) \cup ARR(J_4) \cup ARR(J_5) \cup ARR(J_6) \cup ARR(J_7) \cup ARR(J_8) \quad (17)$$

The fault signature matrix (Step 8) can be easily generated by taking into account the inconsistent equations for each fault. In this case, $f_1 \rightarrow e_4$, $f_2 \rightarrow e_5$ and $f_{j_i} \rightarrow e_{j_i}$. Table IV shows the resulting FSM_{J_8}.

Finally, the possible solution J_8 is verified to fulfill FDI specifications (Step 9). Looking at Table IV, all considered faults are detectable and isolable, so J_8 is a solution, and Algorithm 1 finishes.

V. APPLICATION EXAMPLE: A FUEL CELL SYSTEM

A. Fuel-cell based system description

A PEM Fuel Cell System model is used to test the proposed algorithm. A model for a PEM Fuel Cell was proposed in [14]. This model is widely accepted nowadays in the control community as a good representation of the behavior of an actual fuel cell for control purposes.

The structural relations between model equations and system variables are presented in Appendix section. Variables are classified into the following categories:

- Control variables: measured variables required for control purposes.
- Unmeasurable variables: since sensors are not available, sensor installation is not possible or too expensive, etc.
- Measurable variables: define possible sensor locations.
- Faults.

TABLE V
SENSOR COST ASSOCIATED TO EACH VARIABLE

x_i	ω_{cp}	p_{sm}	$W_{sm,out}$	p_{ca}	$W_{ca,out}$	p_{an}	$W_{an,in}$	$W_{rm,out}$
$C(j_i)$	50	10	25	70	35	60	90	40

TABLE VI
EFFICIENCY OF ALGORITHM 1

	generated ARRs	relative efficiency [a]
Absolute approach	9029	**0,41%**
Algorithm 1	335	**11,05%**

[a] relative efficiency $= 100 \times card(ARR(2^{J^*}))$/generated ARRs

Each measurable variable has an associated sensor which together constitute the set of candidate sensors I. Solving the optimal sensor placement problem requires a cost to be associated to each candidate sensor (see Table V). Changing these cost values, different solutions would be produced by Algorithm 1.

The FDI specifications which have to be fulfilled in this application are that system faults and installed sensors faults must be detectable and isolable, as defined in section II-A.

B. Optimal Sensor Placement Solution for the Fuel Cell System

Algorithm 1 has been applied to the Fuel Cell System and an optimal solution has been found that satisfies the fault detectability and isolability specifications for FDI. The optimal solution consists in adding two sensors, to the set of already installed sensors for control purposes (see Appendix), that measure the following two internal variables: cathode output flow ($W_{ca,out}$) and anode pressure (p_{an}).

The cost of this solution is 95. Using these two additional sensors and the four control sensors, a set of 37 ARRs can be generated (i.e. $card(ARR(2^{J^*})) = 37$), that satisfies the FDI specifications.

To assess the effectiveness of the incremental approach proposed in this paper over the absolute approach, the number of ARRs generated by both approaches will be compared. On the one hand, the absolute approach would require to generate all ARRs considering a fully sensored system, that amounts to a total of 9029. On the other hand, the incremental algorithm finds the solution in 27 iterations and the total number of ARRs generated during these iterations is 335. Table VI compare the relative efficiency in both approaches, which proves that ARRs generation using incremental algorithm is more efficient than using the absolute approach.

VI. CONCLUSIONS

This paper presents an efficient algorithm for solving the problem of optimal sensor placement for FDI. It allows determining the set of sensors that minimizes a pre-defined cost function satisfying at the same time a pre-established

set of FDI specifications for a given set of faults. Existing algorithms are mainly based on formulating an optimization problem once the sets of all possible ARRs have been generated, considering all possible candidate sensors installed, following the so-called absolute approach. However, the associated computational complexity is exponential with the number of possible sensors. The main goal of this paper is to propose an incremental algorithm that tries to avoid the computational burden. To show the effectiveness of the approach proposed an application based on a fuel-cell system has been proposed.

Nevertheless, there are still some open issues which could be considered as a further research. Firstly, the causality constraints involved in the structural modeling of dynamic equations are not taken into account. Secondly, faults that change the structure of the model are not considered, only additive faults on measurable variables are handled. Lastly, sensor redundancy is not concerned, but could be easily included as in [9].

APPENDIX

In this appendix, the structural relations between model equations and system variables are showed. Model equations are derived from [14] and they are classified according to the system component that they describe:

- *Air Supply Compressor:*

  $e_1 : f(\omega_{cp}, \tau_{cm}, \tau_{cp}) = 0$
  $e_2 : f(\tau_{cm}, V_{cm}, \omega_{cp}) = 0$
  $e_3 : f(\tau_{cp}, \omega_{cp}, p_{sm}, W_{cp}, f_{p_{sm}}) = 0$
  $e_4 : f(W_{cp}, p_{sm}, \omega_{cp}, f_{p_{sm}}) = 0$
- *Air Supply Manifold:*

  $e_5 : f(W_{sm,out}, p_{sm}, p_{ca}, f_{p_{sm}}, f_{W_{sm,out}}) = 0$
  $e_6 : f(W_{cp}, W_{sm,out}, f_{W_{sm,out}}) = 0$
- *Fuel Cell Stack:*

  $e_7 : f(W_{ca,out}, W_{v,inj}, I_{st}, W_{sm,out}, f_{W_{sm,out}}, f_{I_{st}}, f_n) = 0$
  $e_8 : f(W_{an,in}, I_{st}, f_{I_{st}}, f_n) = 0$
- *Anode Manifold:*

  $e_{10} : f(W_{an,in}, p_{an}) = 0$
- *Return Manifold:*

  $e_{11} : f(W_{ca,out}, W_{rm,out}, f_{W_{rm,out}}) = 0$
  $e_{12} : f(p_{ca}, W_{ca,out}, W_{rm,out}, f_{W_{rm,out}}) = 0$
- *Fault equations:*

  $e_{13} : f_{p_{sm}} = 0$
  $e_{14} : f_{W_{sm,out}} = 0$
  $e_{15} : f_n = 0$
  $e_{16} : f_{I_{st}} = 0$
  $e_{17} : f_{W_{rm,out}} = 0$

Next, main system variables are described:

- *Control Variables:*

  V_{cm}: Compressor voltage
  W_{cp}: Air flow through the compressor
  I_{st}: Stack current
  V_{st}: Stack voltage
- *Unmeasurable variables:*

  τ_{cm}: Compressor motor torque
  τ_{cp}: Load torque
  $W_{v,inj}$: Humidifier injector flow
- *Measurable variables:*

  ω_{cp}: RPM compressor
  p_{sm}: Supply manifold pressure
  $W_{sm,out}$: Supply manifold exit flow
  p_{ca}: Cathode pressure
  $W_{ca,out}$: Cathode output flow
  p_{an}: Anode pressure
  $W_{an,in}$: Anode input flow
  $W_{rm,out}$: Return manifold exit flow
- *Fault variables*

  $f_{p_{sm}}$: Compressor fault
  $f_{W_{sm,out}}$: Supply manifold fault
  f_n: Cell fault
  $f_{I_{st}}$: Fuel Cell Stack fault
  $f_{W_{rm,out}}$: Return manifold fault

REFERENCES

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer, 2003.

[2] M. Bagajewicz, *Design and Upgrade of Process Plant Instrumentation*. Lancaster, PA: Technomic Publishers, 2000.

[3] S. Spanache, T. Escobet, and L. Travé-Massuyès, "Sensor placement optimisation using genetic algorithms," in *Proc. 15^{th} International Workshop on Principles of Diagnosis (DX'04)*, Carcassonne, France, June 23–25, 2004.

[4] F. Nejjari, R. Pérez, T. Escobet, and L. Travé-Massuyès, "Fault diagnosability utilizing quasi-static and structural modelling," *Math. Comput. Mod.*, vol. 45, pp. 606–616, 2006.

[5] M. Staroswiecki, G. Hoblos, and A. Aitouche, "Sensor network design for fault tolerant estimation," *Int. J. Adapt. Control Signal Process.*, vol. 18, no. 1, pp. 55–72, 2004.

[6] D. Maquin, M. Luong, and J. Ragot, "Fault detection and isolation and sensor network design," *Europ. J. Autom.*, vol. 31, no. 13, pp. 396–406, 1997.

[7] R. Raghuraj, M. Bhushan, and R. Rengaswamy, "Locating sensors in complex chemical plants based on fault diagnostic observability criteria," *AIChE J.*, vol. 45, no. 2, pp. 310–322, Feb. 1999.

[8] C. Commault and J.-M. Dion, "Sensor location for diagnosis in linear systems: A structural analysis," *IEEE Trans. Automat. Contr.*, vol. 52, no. 2, pp. 155–169, 2007.

[9] E. Frisk and M. Krysander, "Sensor placement for maximum fault isolability," in *Proc. 18^{th} International Workshop on Principles of Diagnosis (DX'07)*, Nashville,TN,USA, May 29–31, 2007.

[10] D. Düştegör, E. Frisk, V. Cocquempot, M. Krysander, and M. Staroswiecki, "Structural analysis of fault isolability in the DAMADICS benchmark," *Contr. Eng. Pract.*, vol. 14, pp. 597–608, 2006.

[11] M. Krysander, "Design and analysis of diagnosis systems using structural analysis," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, June 2006.

[12] L. Travé-Massuyès, T. Escobet, and X. Olive, "Diagnosability analysis based on component supported analytical redundancy relations," *IEEE Trans. Syst., Man, Cybern. A*, vol. 36, no. 6, pp. 1146–1160, 2006.

[13] R. Sarrate, V. Puig, T. Escobet, and A. Rosich, "Optimal sensor placement for model-based fault detection and isolation," in *Proc. 46^{th} IEEE Conference on Decision and Control*, New Orleans, USA, Dec. 12–14, 2007.

[14] J. Pukrushpan, H. Peng, and A. Stefanopoulou, "Control-oriented modeling and analysis for automotive fuel cell systems," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 126, pp. 14–25, 2004.