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

Sensor Placement for Fault Diagnosis Based On Causal Computations

Albert Rosich [*] **Erik Frisk** [**] **Jan Åslund** [**] **Ramon Sarrate** [*]
Fatiha Nejjari [*]

[*] *Automatic Control Department, Universitat Politecnica de Catalunya, Rambla de Sant Nebridi, 10, 08222 Terrassa, Spain (e-mail: albert.rosich@upc.edu).*
[**] *Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden*

Abstract: This work develops a methodology to solve the sensor placement problem for fault detection and isolation. The proposed methodology is suitable to handle highly non-linear and large scale systems since it is based on structural models. Furthermore, causality is assigned in those variable-equation relations that the variable can be computed from the equation in order to guarantee the computability of the unknown variables in the residual generation design. Finally, the developed methodology is applied on an air compressor model.

1. INTRODUCTION

Fault diagnosis systems are an increasing and important topic in many industrial processes. The number of publications devoted to fault diagnosis has increased notably in the last years, as it can be seen in Blanke et al. (2006). In Model-based Fault Diagnosis, diagnosis is basically performed from the comparison between a process model and on-line process information. Since process information is usually obtained by means of the sensors installed in the process, it is important to develop methodologies to place the correct set of sensors in the process in order to guarantee some diagnosis specifications.

In this paper, diagnosis specifications are maximum detectability and isolability for single systems faults, though the method could be easily extended for single sensor faults. Large-scale diagnosis models may consist of many different types of descriptions, for example static/dynamic linear equations, lookup tables, logic rules, non-linear differential-algebraic equations, etc. One way to analyze such a general class of models in a general framework is to analyze the model structure. A structural model is a coarse model description, based on a bi-partite graph, that can be obtained early in the development process, without major engineering efforts. This kind of models are suitable to handle large scale systems since efficient graph-based tools can be used and does not have numerical problems. However, only best case results are obtained. More information about structural modeling applied to fault diagnosis can be found in Blanke et al. (2006).

Redundant sub-models are of central importance in diagnosis and there are several structural approaches to find redundant sub-models suggested in the literature. The name given to redundant sub-models depends on the approach, e.g. *analytical redundancy relations*, ARR (Travé-Massuyès et al., 2006), *minimal structurally overdetermined set*, MSO (Krysander et al., 2008), *testable subsystem*, TSS (Ploix et al., 2008) and *minimal evaluation chain*, MEC (Pulido and Gonzalez, 2004). All these approaches have in common the fact that each redundant sub-model has a complete matching in the variables set. Some works (Blanke et al., 2006, Düştegör et al., 2006, Yassine et al., 2008) use this matching to assign which equation computes each internal unknown variable. This concept is known, in Blanke et al. (2006) as a causal interpretation of the computability. The result is a directed bi-partite graph that shows how the internal values can be computed from the equations (value propagation) in every redundant sub-model.

Some other results devoted to sensor placement for diagnosis using graph tools can be found in Raghuraj et al. (1999), Krysander and Frisk (2008), Commault et al. (2008) and Rosich et al. (2007). All these works use a structural model-based approach and define different diagnosis specifications to solve the sensor placement problem.

In this work, sensor placement problem for diagnosis is solved in order to establish the set of sensors that are needed to achieve fault detectability and isolability when residuals are to be computed by means of a causal computation sequence.

2. PROBLEM MOTIVATION

In a diagnosis system, consistency is checked by using a set of redundant sub-models. All these sub-models have the property of a complete matching in the unknown variables plus an extra equation, the redundant equation, used for checking consistency. An example of a redundant sub-model is the set of equations (1) (y_1 and y_2 are known), where a possible complete matching is $\{(e_1, x_1), (e_2, x_2)\}$ and the redundant equation would be e_3.

[⋆] 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

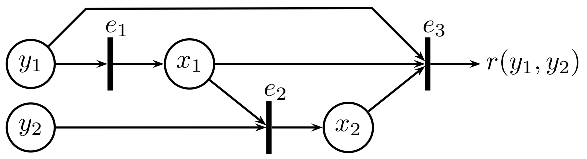

Fig. 1. Computation sequence.

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

One way to obtain a residual from these sub-models, when we handle non-linear equations, is to construct the computation sequence (Blanke et al., 2006) which describes how to compute the residual by convenient manipulation of the set of the sub-model equations. Following with the example, the matching is interpreted as e_1 solves x_1 and e_2 solves x_2. Figure 1 shows the computation sequence for the residual obtained from the redundant sub-model (1).

From this computation sequence, it is straightforward to propagate the values to compute the residual:

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

where,

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

Obtaining residuals in complex systems by forward propagation of computed variables as in the example is an efficient technique. Residuals that depend on submodels that include loops, both algebraic and differential, and non-invertible functions will be excluded. Then, no complex solving tools will be needed. This will lead to a restricted set of residuals.

To keep the simplicity of the approach and at the same time reduce the restrictiveness, the class of submodels is extended to also cover linear loops, since solving linear equations is not a complex task. A consequence of this extension is that existing structural methods for finding submodels and computational sequences have to be modified.

Note that the rejected submodels could still be used for diagnosis, but they would typically require complex solving tools.

The objective is to derive an algorithm that computes where to place sensors such that a required fault isolation performance is reached using only residuals based on the extended class of submodels.

Causal properties of the model have been considered in other works on residual generation and sensor placement, for example Travé-Massuyès et al. (2006), Blanke et al. (2006), Yassine et al. (2008). The main contribution, compared to these previous works, is that sensor placement analysis is done with respect to a class of residual generators that are particularly easy to implement and where causal properties of the model are taken into consideration. Also, this is done without computing the complete set of ARRs which is a complex task in most cases.

3. PRELIMINARY CONCEPTS

3.1 Causal structural model

Knowing when a redundant sub-model can be used to generate a residual (using the computation sequence) requires some properties in the sub-model equations. These properties are related to how variables can be computed in the equation. 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. Furthermore, a given set of variables will be computable as long as the following two conditions hold:

condition 1: there exists a complete matching with causal variables

condition 2: there are no algebraic loops

The non-existence of algebraic loops is motivated by the next small example. Consider the set of non-linear equations (4), where x_1 and x_2 are unknown variables and y_1 and y_2 are known variables.

$$\begin{array}{l} e_1 : h_1(x_1, x_2, y_1) = 0 \\ e_2 : h_2(x_1, x_2, y_2) = 0 \end{array} \tag{4}$$

Assume that x_1 is a causal variable in e_1 and x_2 is a causal variable in e_2. This means that e_1 and e_2 can be rearranged as $e_1' : x_1 = g_1(x_2, y_1)$ and $e_2' : x_2 = g_2(x_1, y_2)$. Then, replacing variable x_2 in the first equation, we obtain the equation $h_1(x_1, g_2(x_1, y_2), y_1) = 0$, where x_1 is not necessarily a causal variable. This kind of structure is known as an *algebraic loop.* There are several tools to compute unknown variables in an algebraic loop (*e.g.*, numeric solvers, non-linear optimization), but the solution is not always ensured and the computation cost can be large. In this work, a conservative approach consisting in rejecting all non-linear algebraic loops is adopted. Regarding linear algebraic loops, and assuming that the linear coefficients are algebraically independent, it is straightforward to know if a solution exists and compute it. Therefore, linear algebraic loops are accepted. Thus, it is important to know when a set of variables are linear. 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.

Returning to the set of equations (4), assume now that x_1 and x_2 are linear variables in both equations. This means that they can be arranged as $e_1 : \mathcal{L}_{1,1}(x_1) + \mathcal{L}_{1,2}(x_2) =$

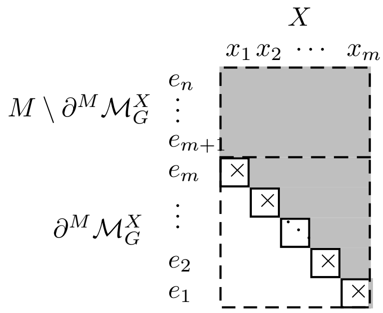
X
$x_1\, x_2\, \cdots\, x_m$
e_n
$\vdots$
e_{m+1}
e_m
$\vdots$
e_2
e_1
$M \setminus \partial^M \mathcal{M}_G^X$
$\partial^M \mathcal{M}_G^X$

Fig. 2. Causal computable structure

$f_1(y_1)$ and $e_2 : \mathcal{L}_{2,1}(x_1) + \mathcal{L}_{2,2}(x_2) = f_2(y_2)$. Then, as long as $[\mathcal{L}_{1,1}\mathcal{L}_{2,1}]'$ and $[\mathcal{L}_{1,2}\mathcal{L}_{2,2}]'$ are linearly independent, unknown variables can be easily computed.

We will now formalize the structural model as a bipartite graph $G(M, X, A)$, where $M = \{\dots, e_i, \dots\}$ is the set of model equations, $X = \{\dots, x_j, \dots\}$ the set of unknown variables and A the set of edges, such that $(e_i, x_j) \in A$ as long as $x_j \in X$ appears in equation $e_i \in M$. 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 "×" symbol and edges in A_Δ are represented by a "Δ" symbol.

3.2 Causal computability

Given a structural model, we need to know whether a set of unknown variables can be computed when causal and linear relations are considered. Let $G(M, X, A)$ be a structural model with $A = \{A_L \cup A_\times \cup A_\Delta\}$. First, for the sake of simplicity, 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 $\mathcal{M}_G^X \subseteq A_\times$ (*i.e.*, condition 1 holds) and the well-constrained subgraph $G'(\partial^M \mathcal{M}_G^X, X)$ has no strong component with more than one equation (*i.e.*, condition 2 holds), then the set of unknown variables, X, can be computed using the computation sequence. Note that $\partial^M \mathcal{M}_G^X$ denote the subset of equations in M incident to edges in $\mathcal{M}_G^X$. This means that the equations set and the variables set can be rearranged such that the biadjacency matrix has a triangular form with a diagonal of "×" symbols. Figure 2 shows this form where all unknown variables can be evaluated.

Here, it is assumed that a set of variables can not be solved in an algebraic loop (*i.e.*, condition 2 holds), so any strong component with more than one equation-variable pair is rejected. 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 diagonal matching) and then pruning the graph, until no more equation-variable pairs can be found.

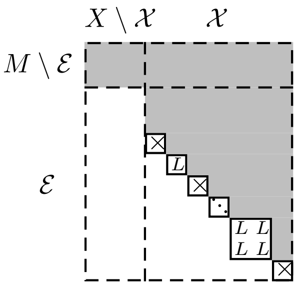

Fig. 3. Causal and linear computable decomposition

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

```
$X' = X$; $\mathcal{X} = \emptyset$;
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
```

Now, assume that linear relations are considered, *i.e.* $A = \{A_L \cup A_\times \cup A_\Delta\}$. In section 3.1, it was assumed that a subset of linear variables can be solved in an algebraic loop. Therefore, the Dulmage-Mendelsohn decomposition can be applied to determine the subset of linear computable variables.

The Dulmage-Mendelsohn decomposition (Dulmage and Mendelsohn, 1958) 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 diagonal matching presented in Figure 2 is now improved, by taking into account the linear computable variables set. The resulting structural decomposition is shown in Figure 3 where the triangular form remains, but now the strong components can take more than one variable, since linear relations are allowed. Note that the computable part is the subgraph $G'(\partial^{\mathcal{E}} \mathcal{M}_G^{\mathcal{X}}, \mathcal{X}, A) \subseteq G(M, X, A)$. This decomposition is done by Algorithm 3, which iteratively alternates algorithms 1 and 2, and finally returns subgraph $G(\mathcal{E}, \mathcal{X}, A)$.

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, $X \setminus \mathcal{X}$.

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 $\mathcal{X}' \cup \mathcal{X}_L = \emptyset$
$\mathcal{E} = \{e \in M : \text{var}_X(e) \subseteq (\mathcal{X})\}$
```

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.

4. STRUCTURAL DETECTABILITY AND ISOLABILITY WITH CAUSAL RELATIONS

According to the decomposition in Figure 3, there is, at least, one complete matching in $G(\mathcal{E}, \mathcal{X}, A)$: the diagonal. This means that $\mathcal{E}^- = \emptyset$. Since the matching in Figure 3 is complete in $\mathcal{X}$, it follows that the over-determined set of equations $\mathcal{E}^+$ contains part of this diagonal matching, so variables in $\mathcal{E}^+$ are computable. Now, fault diagnosis analysis can be performed over the over-determined part and the computation sequence can always be guaranteed to generate residuals. For extended information on the Dulmage-Mendelsohn decomposition applied to fault diagnosis see Krysander (2006).

In this work, faults are defined as a subset of equations, $F \subseteq M$, since a relation between an equation and a fault can be easily established (*i.e.*, a signal fault that affects an equation, or the assumption or support of an equation). In order to simplify the following theoretic development, only system faults will be considered (*i.e.*, sensor faults will be discarded).

4.1 Causal structural detectability

It is well known that the set of detectable faults can be defined from the over-constrained part (Blanke et al., 2006). Thus, given a set of faults $F \subseteq M$, they are structurally detectable, when using the computation sequence, if $F \subseteq \mathcal{E}^+$.

Algorithm 4 finds all detectable faults when the computable sequence is taken into account. Their inputs are a structural model $G(M, X, A)$ and a set of faults, $F \subseteq M$.

Algorithm 4 $F_d = \text{CausalDetectability}(G(M, X, A), F)$

```
$\{\mathcal{X}, \mathcal{E}\} = \text{ComputableSystem}(G(M, X, A))$;
$F_d = \mathcal{E}^+ \cap F$;
```

4.2 Causal structural isolability

Isolability analysis is based on detectability conditions. According to Krysander and Frisk (2008), a fault $f_i \in F$ is structurally isolable form $f_j \in F$ if f_j is detectable in sub-model $M \setminus \{f_i\}$. This means that for each detectable fault $f \in F_d$, there exist a set of isolable faults, $F_i(f)$, from f. Algorithm 5 computes the isolable fault set for each detectable fault.

Algorithm 5 $F_i = \text{CausalIsolability}(G(M, X, A), F_d)$

```
for $each f \in F_d$ do
    $M_f = M \setminus \{f\}$;
    $F_i(f) = \text{CausalDetectability}(G(M_f, X, A), F_d)$;
end for
```

5. SENSOR PLACEMENT BASED ON CAUSAL RELATIONS

Given a set of equations, the subset of unknown variables that can be computed will depend on the set of installed sensors (*i.e.*, known variables).

The main idea is to perform the fault detectability and isolability analysis with all sensors installed at once. Under this setting, the detectable faults set and the isolable faults set will rise an upper limit on the fault diagnosis specifications. Installing the same sensor more than once makes neither detectable a non-detectable fault nor isolable a non-isolable fault, except for sensor faults. Then, once maximum fault diagnosis specifications are known, the set with the minimum number of sensors that satisfies this specifications is sought.

5.1 Maximum detectability and isolability specifications

Maximum detectability specification is ensured when all candidate sensors are installed at least once. Therefore, it is straightforward to select those faults that can be detectable from those that can never be detectable, before the sensor placement analysis.

The set of candidate sensors can be defined as a subset of unknown variables, $S \subseteq X$. Each sensor has a corresponding sensor equation $x = f(y)$, with y being a measured variable. This equation has to be added to the model whenever the corresponding sensor is selected for installation. Note that adding this equation implies that x is now a causal variable.

Maximum detectability specification is computed by Algorithm 4 with all candidate sensors S installed once: $F_{d_{max}} = \text{CausalDetectability}(G(\{M \cup M_S\}, X, A), F)$.

There may be some useless sensors, that do not improve system fault detectability. Their corresponding sensor equations will not belong to the over-determined part of the computable subsystem. The set of sensor equations that are in the computable over-determined part is computed as $M_{S_d} = \mathcal{E}^+_{M \cup M_S} \cap M_S$. Thus, the new set of candidate sensors, $S_d \subseteq S$, is now defined as all sensors such that their corresponding sensor equation is in M_{S_d}. Therefore, there is no need to consider further sensors from $S \setminus S_d$ in the sensor placement analysis.

Now, maximum isolability specification is computed by Algorithm 5, with just the new set of sensors S_d installed, and for those system faults that are detectable F_d: $F_{i_{max}} = \text{CausalIsolability}(G(\{M \cup M_{S_d}\}, X, A), F_d)$.

5.2 Sensor placement algorithm

Once maximum detectability and isolability specifications are known, Algorithm 6 searches for the minimal set of sensors that satisfies them (*i.e.*, $F_{d_{max}}$ and $F_{i_{max}}$).

We have shown in section 5.1 that the algorithm will terminate since the set of candidate sensors, S_d, is one admissible set that fulfills the specifications.

Algorithm 6 S_{min} = CausalSensorPl($M, S_d, F_{d_{max}}, F_{i_{max}}$)

```
S_min = ∅; k = 0;
repeat
    $S_k$ = the minimal sensors set not previously tested from $S_d$;
    $F_{d,k} = \text{CausalDetectability}(G(M \cup M_{S_k}, X, A), F_{d_{max}})$;
    if $F_{d,k} = F_{d_{max}}$ then
        $F_{i,k} = \text{CausalIsolability}(G(M \cup M_{S_k}, X, A), F_{d_{max}})$;
        if $F_{i,k} = F_{i_{max}}$ then
            $S_{min} = S_k$;
        end if
    end if
    $k = k + 1$;
until $S_{min} \neq \emptyset$
```

6. CAUSAL COMPUTABLE MSO SETS

Finding redundant sub-models for diagnosis is an important topic in the field of diagnosis based on structural models. In Section 1, several works devoted to find the whole set of redundant sub-models given a structural model, were already referenced. In Krysander et al. (2008) an algorithm to compute all possible MSO sets is presented. An MSO set is a subset of model equations that is minimal redundant (*i.e.*, no subset in an MSO set is redundant). However, we are interested in those MSO sets which can be used to compute a residual based on a computation sequence (*i.e.*, causal computable MSO sets). Obtaining the set of all causal computable MSO sets resorts to choosing those MSO sets that are causal computable from the whole set of MSO sets.

One way to test whether a MSO set is causal computable or not is by means of Algorithm 3. Given a MSO set M and its corresponding graph $G(M, X, A)$, Algorithm 3 returns equation set $\mathcal{E}$ and computable variable set $\mathcal{X}$. Then, the MSO set is causal computable if $M = \mathcal{E}$. Thus, the MSO set has a causal matching as in Figure 2.

7. APPLICATION TO A COMPRESSOR FUEL CELL

Sensor placement analysis has been tested in a compressor model used in a fuel cell stack system presented in Pukrushpan et al. (2004). The model is given by the set of equations (5).

$$
\begin{aligned}
&e_1 : R \cdot i(t) + L\frac{di(t)}{dt} + k_v \cdot \omega(t) = v(t)\\
&e_2 : J\frac{d\omega(t)}{dt} = k_p \cdot i(t) - B \cdot \omega(t) - \tau_{cp}(t)\\
&e_3 : T_{cp}(t) = T_{amb} + \frac{T_{amb}}{\eta(t)}(p_r(t)^{\frac{\gamma-1}{\gamma}} - 1)\\
&e_4 : p_r(t) = \begin{cases} 0 \text{ if } p_{cp}(t)/p_{amb} \leq 0\\ p_{cp}(t)/p_{amb} \text{ if } 0 < p_{cp}(t)/p_{amb} < 1000\\ 1000 \text{ if } p_{cp}(t)/p_{amb} \geq 1000\end{cases}\\
&e_5 : \eta(t) = \text{LookUpTable}(W(t), p_r(t))\\
&e_6 : W(t) = \text{CompressorMap}(p_r(t), \omega(t))\\
&e_7 : \tau_{cp}(t) = \frac{C_p}{\omega(t)}\frac{T_{amb}}{\eta(t)}(p_r(t)^{\frac{\gamma-1}{\gamma}} - 1)W(t)
\end{aligned}
\tag{5}
$$

The compressor model has two parts. The first part (e_1 and e_2) is an electric motor model to compute the two dynamic states: compressor speed ω and motor current i. No restriction in the causal computation of differential states is assumed. Inputs for the electric motor are the motor voltage v and the compressor torque τ_{cp}. The second part (e_3, e_4, e_5, e_6 and e_7) comprises an static compressor map (e_6) which determines the air flow rate W, and thermodynamic equations (e_3 and e_7) used to calculate the exit air temperature T_{cp} and the required compressor torque τ_{cp}. A lookup table (e_5) is used to find the efficiency of the compressor η. The pressure ratio across the compressor p_r is calculated from the supply manifold pressure p_{cp} and bounded between 0-1000bar (e_4). The atmospheric temperature T_{amb} and pressure p_{amb} are assumed known and constant. Table 1 shows the biadjacency matrix of the structural model $G(M, X, A)$, extracted from (5) following the guidelines in section 3.1.

Table 1. Structural compressor model

	i	ω	τ_{cp}	T_{cp}	η	p_r	p_{cp}	W	v
e_1	L	L							L
e_2	L	L	L						
e_3				×	×	×			
e_4						×	Δ		
e_5					×	Δ		Δ	
e_6		Δ				Δ		×	
e_7		×	×		×	×		×	

System faults, f_1, f_2, f_3 and f_4 are related to the consistency of equations e_1, e_2, e_3 and e_6, respectively, *i.e.* $F = \{e_1, e_2, e_3, e_6\}$. Assume that there is no sensor installed, so the set of measurable variables is $S = \{i, \omega, \tau, T_{cp}, p_{cp}, W, v\}$. The goal is to place the minimal set of candidate sensors in order to obtain the maximum detectability and isolability specifications among systems faults.

First, maximum detectability and isolability analysis is performed. The set of sensors equations, M_S is added to the structural model by adding rows in Table 1 with a cross in the sensor variable column. For instance, motor speed sensor equation can be written analytically as e_ω : $\omega = h(\omega_{measured})$, where $e_\omega \in M_S$. Then, the structural representation of this equation is a single causal relation between measured variable ω and sensor equation e_ω, and this follows for each sensor equation in M_S. Applying Algorithms 4 and 5, it turns out that all faults are detectable (*i.e.*, $F_{d_{max}} = F$) and each fault is isolable from all others (*i.e.*, $\forall f \in F, F_{i_{max}}(f) = F \setminus \{f\}$). Furthermore, there are no useless sensor which can be omitted in the sensor placement analysis (*i.e.*, $S_d = S$).

Next, the sensor placement for diagnosis is solved. Algorithm 6 returns two possible solutions: $S_{min_1} = \{i, \tau_{cp}, T_{cp}, p_{cp}, v\}$ and $S_{min_2} = \{i, \omega_{cp}, T_{cp}, p_{cp}, v\}$. If causal relations were not taken into account, an optimal configuration with fewer sensors would be found ($\{i, T_{cp}, p_{cp}, v\}$), but some residuals would not be easily computed.

Assuming that the set of sensors S_{min_1} is installed in the system, 23 possible MSO sets are generated. Just 16 out of those 23 MSO sets are causal computable, so they can be used to compute the corresponding set of residuals using the computation sequence. Some of these causal computable MSO sets are given in (6).

Table 2 is deduced from the relation between faults and system equations, and taking into account sensor equa-

Table 2. Relation among MSO sets, system faults and sensors

	f_1	f_2	f_3	f_4	i	τ_{cp}	T_{cp}	p_{cp}	v
MSO_1		×		×	×	×		×	
MSO_2		×	×		×	×	×	×	
MSO_3	×			×	×	×		×	×
MSO_4	×		×		×	×	×	×	×
MSO_5	×	×		×		×		×	×

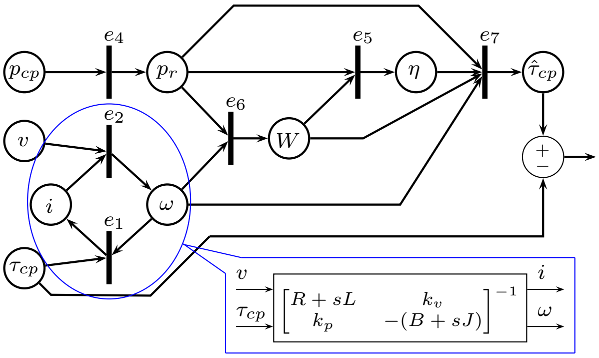

Fig. 4. Computation sequence from MSO_5

tions in each causal computable MSO set. It shows whether a fault affects the residual computed from a causal computable MSO set and the set of sensors that is needed to compute this residual.

$$\begin{aligned} MSO_1 &= \{e_2, e_4, e_5, e_6, e_7, e_{\tau_{cp}}, e_{p_{cp}}\} \\ MSO_2 &= \{e_2, e_3, e_4, e_5, e_7, e_i, e_{\tau_{cp}}, e_{T_{cp}}, e_{p_{cp}}\} \\ MSO_3 &= \{e_1, e_4, e_5, e_6, e_7, e_i, e_{\tau_{cp}}, e_{p_{cp}}, e_v\} \\ MSO_4 &= \{e_1, e_3, e_4, e_5, e_7, e_i, e_{\tau_{cp}}, e_{T_{cp}}, e_{p_{cp}}, e_v\} \\ MSO_5 &= \{e_1, e_2, e_4, e_5, e_6, e_7, e_{\tau_{cp}}, e_{p_{cp}}, e_v\} \end{aligned} \quad (6)$$

In Figure 4, the residual computation sequence corresponding to MSO_5 is presented. Similar computation sequences exist for each causal computable MSO set in (6). Note that the computation sequence in Figure 4 involves a linear algebraic loop among e_1, e_2, i and ω, as long as v and τ_{cp} are known. In this computation sequence, equations e_1 and e_2 should be replaced by the inverted matrix in Figure 4, where all parameters are linear independent and the equation is stable (*i.e.*, a solution can be computed).

8. CONCLUSIONS

This paper has addressed the sensor placement problem for model-based fault diagnosis systems. The novelty of this approach is that causal and linear relations between variables and equations are taken into account in order to guarantee the computability of the set of residuals later needed in the practical implementation of the diagnosis system. Moreover, in this approach no residual or redundancy relation is computed. An algorithm that returns the computable subsystem of a given set of equations has been introduced. Later, this algorithm has been applied to the sensor placement problem producing an algorithm that returns the minimal set of sensors that satisfy the maximum system faults diagnosis specifications. All algorithms have no hard computational complexity, except for Algorithm 6 in step 3.

In the near future, this methodology could be easily adapted to address three possible extensions. Firstly, stating the optimal sensor placement, by introducing an installation cost for each sensor as in Rosich et al. (2007). This would imply that step 3 in Algorithm 6 should be modified. Secondly, allowing sensor faults, which have an special interest since the sensor placement problem depends on whether a sensor is present or not. Then, maximum detectability and isolability specifications would depend on the tested sensors set. Lastly, allowing redundant sensors to measure the same variable. Note that the maximum detectability and isolability specifications, involving sensor faults, could then be ensured by installing all sensors twice.

REFERENCES

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

C. Commault, J. M. Dion, and S. Y. Agha. Structural analysis for the sensor location problem in fault detection and isolation. *Automatica*, 44(8):2074–2080, aug 2008.

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, 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.

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

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, A.G. Stefanopoulou, and H. Peng. *Control of Fuel Cell Power Systems Principles, Modeling, Analysis and Feedback Design*. Springer, 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.

R. Raghuraj, M. Bhushan, and R. Rengaswamy. Locating sensors in complex chemical plants based on fault diagnostic observability criteria. *AIChE J.*, 45(2):310–322, February 1999.

A. Rosich, R. Sarrate, V. Puig, and T. Escobet. Efficient optimal sensor placement for model-based FDI using and incremental algorithm. In *Proc. 46th IEEE Conference on Decision and Control*, pages 2590–2595, New Orleans, USA, December 12–14, 2007.

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.

A. A. Yassine, S. Ploix, and J. M. Flaus. A method for sensor placement taking into account diagnosability criteria. *Int. J. Appl. Math. Comput. Sci.*, 18(4):497–512, 2008.