

Sensor Placement for Fault Detection and Isolation based on Structural Models^{*}

A. Rosich^{*}

^{*} *Institut de Robòtica i Informàtica Industrial (CSIC-UPC),
Llorens i Artigas 4-6, 08028 Barcelona, Spain
(e-mail: albert.rosich@upc.edu).*

Abstract: This paper presents a new algorithm for solving the sensor placement problem in order to achieve the required fault detectability and isolability. The method is based on structural models, which means that non-linear differential systems can be efficiently handled. The developed algorithm does not need to compute any residual since it is based on structural model properties. Thus, the residual computation burden is avoided. The method can also handle faults of the extra placed sensors as well as the possibility of installing repeated sensors.

Keywords: Sensor placement, Structural model, Minimal Set Cover, Fault detection and isolation.

1. INTRODUCTION

The diagnosis task is often made difficult due to insufficient, incomplete or useless process observations. The problem on how to obtain the required process observations is known as the *sensor placement problem*. Several works have tackled this problem, where most of them presents a graph-based approach (Bagajewicz, 2000; Travé-Massuyès et al., 2006; Commault et al., 2008; Krysander and Frisk, 2008; Rosich et al., 2009). Graph-based model representations are suitable for the sensor placement problem since they allow to rid of the analytical expressions which are not always available at a first stage of the design. Also, graph-based tools are free of numerical problems and have in general better computational efficiency. However, only best-case results can be computed from graph-based methods. A graph model representation widely used in the area of model-based diagnosis is the *structural model* representation (Blanke et al., 2006), which will be used in this paper.

This paper specifically focuses on analysing which sensors should be installed in a process in order to achieve predefined fault detectability and isolability properties. The method presented here is rather close to Krysander and Frisk (2008), since the same diagnosis framework is used and exactly the same problem is solved. Furthermore, the sensor placement problem is also approached by first adding sensors for detectability and then, in a second step for isolability. Nevertheless, the strategy adopted here differs from Krysander and Frisk (2008) work in the sense that now the contribution of each sensor to the diagnosability and isolability of the system is sought.

A Matlab implementation of all algorithm presented here can be downloaded from <http://www.iri.upc.edu/people/arosich/Software.html>.

2. STRUCTURAL ANALYSIS REVIEW

An analytical model of a process typically consist of a set of equations describing the interaction among process variables. Then, a structural model is an abstraction of the model where the analytical expressions are neglected and only the structure of the model is preserved. More formally, given an analytical model, the corresponding structural model is represented by a bipartite graph with two vertex sets: the set M of model equations and the set X of unknown variables. An edge (e, x) , for $e \in M$ and $x \in X$, stands for variable x is involved in equation e . It should be noted that known process variables are not included in the structural model since they will not be used throughout the paper. Furthermore, $\text{var}(M)$ is introduced to denote the subset of unknown variables adjacent to the equations in M , that is $\text{var}(M)$ are the unknown variables involved in M .

Structural models have been widely studied in the fault diagnosis field. As a result of this, several model decompositions have been developed in order to exploit model properties. Next, *Dulmage-Mendelsohn decomposition* (Dulmage and Mendelsohn, 1958) is reviewed. A thorough description of this decomposition and its properties can be found in Murota (2000).

Given a structural model M , the following function, called *surplus function* (Lovász and Plummer, 1986), is defined as

$$p_o(E) = |\text{var}(E)| - |E| \quad (1)$$

for any $E \subseteq M$ and $|\cdot|$ denoting the cardinality of the set. It is worth noting that the surplus function p_o is a sub-modular function. Therefore, it holds that

$$p_o(E_1 \cup E_2) + p_o(E_1 \cap E_2) \leq p_o(E_1) + p_o(E_2) \quad (2)$$

for any two sets of equations E_1 and E_2 .

^{*} This work has been funded by the Spanish Ministry of Science and Technology through the CICYT project WATMAN (ref. DPI2009-13744), and by the European Commission through contract i-Sense (ref. FP7-ICT-2009- 6-270428).

From p_0 function, the family of all minimizers with minimal surplus can be defined as

$$\mathcal{L}_{min} = \{E \subset M \mid p_0(E) \leq p_0(E'), \forall E' \subseteq M\}. \quad (3)$$

This family of minimizers defines a lattice in \mathcal{L}_{min} . Hence, it can be stated that $E_i \cup E_j \in \mathcal{L}_{min}$ and $E_i \cap E_j \in \mathcal{L}_{min}$, for any $E_i, E_j \in \mathcal{L}_{min}$.

Let $E_0 \subset E_1 \subset \dots \subset E_{b-1} \subset E_b$ be any *maximal ascending chain* of \mathcal{L}_{min} . Then, the partitions of the Dulmage-Mendelsohn decomposition on the equation set is thereby computed from the sets in the maximal ascending chain according to

$$\begin{aligned} M_0 &= E_0, \\ M_k &= E_k \setminus E_{k-1}, \quad (k = 1, \dots, b), \\ M_\infty &= M \setminus E_b. \end{aligned}$$

In the diagnosis field, a coarse partition in three main parts is usually defined from the above partition as

- the *overdetermined* part $M^+ = M_0$.
- the *just-determined* part $M^0 = \cup_{k=1}^b M_k$.
- the *under-determined* part $M^- = M_\infty$.

From the diagnosis point of view, a key property in the over-determined part is that there are more equations than unknown variables, i.e., $|M^+| > |\text{var}(M^+)|$, and hence the existence of redundancy in M^+ . Indeed, the M^+ part of the model is the only useful part to perform diagnosis, see Blanke et al. (2006).

3. FAULT DIAGNOSIS FRAMEWORK FOR SENSOR PLACEMENT

The fault diagnosis framework on which this paper is based is presented in this section. First, some theoretical background and standard definitions will be reviewed. Then, sensor characterisation in the framework, as well as the sensor placement problem formalization, will be introduced.

3.1 Fault detectability and isolability analysis

Usually, to perform fault diagnosis based on structural models, it is assumed that different subsets of model equations describe the expected behaviour of process components. Then, when the component equations become inconsistent with the process observations, it may be suspected that the corresponding component is not working properly, i.e., the component is faulty. Here, it will be assumed that a fault can only cause inconsistency in one model equation. Thus, let F be the set of faults, then there exists a fault equation $e_f \in M$ for each fault $f \in F$.

The consistency of a fault equation, together with other model equations, can be checked if there are more equations than unknown variables. This fact motivates the following fault detectability definition (Krysander and Frisk, 2008).

Definition 1. (Fault detectability). A fault $f \in F$ is detectable in a model M as long as $e_f \in M^+$. \diamond

A fault $f_i \in F$ can be isolated from another fault $f_j \in F$ when the inconsistent set of equations involves the equation of the fault f_i but not the equation of the fault

f_j . Next, formal fault isolability definition (Krysander and Frisk, 2008) is introduced.

Definition 2. (Fault isolability) A fault $f_i \in F$ is isolable from fault f_j in a model M as long as $e_{f_i} \in (M \setminus \{e_{f_j}\})^+$. \diamond

Given a structural model M and a set of predefined faults F , diagnosability analysis can be performed from Definitions 1 and 2. The class of detectable faults is therefore computed as

$$D = \{f \in F \mid e_f \in M^+\}. \quad (4)$$

Fault isolability can be characterised by means of pairs of isolable faults. Here, it is assumed that all faults in the isolability analysis are detectable, which implies that the isolability relation is symmetric (Krysander and Frisk, 2008), i.e., if f_i is isolable from f_j then f_j is isolable from f_i . Let D be the ordered set of detectable faults, then fault isolability is characterised as

$$I = \{(f_i, f_j) \in D \times D \mid e_{f_i} \in (M \setminus \{e_{f_j}\})^+, \text{ for } i < j\}. \quad (5)$$

For convenience, it is introduced the diagnosability analysis as a procedure $(D, I) = \text{Diagnosability}(M, F)$ which, given a structural model M and a set of faults F , returns the set of detectable faults D and the set of all isolable faults pairs I , computed according to (4) and (5), respectively.

3.2 Sensor placement problem formulation

Sensors are regarded as system components that may be installed or not. Thus, installing sensors implies extending the model by adding those equations describing the sensor behaviour. Here, for the sake of simplicity, it is assumed that a sensor can be modelled by means of a single sensor equation of the form $y = x$, where $x \in X$ is the measured process variable and y represents the reading of the sensor, i.e., a known variables or observation. Therefore, the set of candidate sensors is characterised as a subset of unknown variables $S \subseteq X$.

It is worth noting that installing sensors increases the number of model equations whereas the number of unknown variables remains unaltered. This means that, in general, the more sensors are installed the better diagnosis performance can be expected. Indeed, maximum diagnosis performance can be determined by performing diagnosability analysis with all possible sensors installed in the system. Let M_S be the sensor equations of all candidate sensors in S , the maximum fault detectability and isolability attainable by placing sensors is determined as

$$(D_{max}, I_{max}) = \text{Diagnosability}(M \cup M_S, F).$$

This fact establishes an upper bound on the required fault detectability and isolability specifications in the sensor placement problem. However, diagnosis specifications other than the maximum ones may be desired. The diagnosis specifications, D and I , are feasible as long as $D \subseteq D_{max}$ and $I \subseteq I_{max}$.

Now, the sensor placement problem to be solved in this paper can be formulated as follows (Krysander and Frisk, 2008).

Problem 3. (Sensor placement problem). Given a structural model M of the process, a set of candidate sensors

S to be installed in the process and the required diagnosis specifications, D and I , defined from the set of process faults F , find all minimal sensors configurations $S' \subseteq S$ such that diagnosis specifications are fulfilled within the model $M \cup M_{S'}$. \diamond

Diagnosis specifications are fulfilled as long as $D \subseteq D'$ and $I \subseteq I'$, where D' and I' represent detectability and isolability derived from a sensor configuration $S' \subseteq S$, i.e., $(D', I') = \text{Diagnosability}(M \cup M_{S'}, F)$. Moreover, note that all minimal sensor configurations characterise all possible solutions since any combination of these minimal configurations is also a solution for the sensor placement problem.

4. SENSOR PLACEMENT FOR FAULT DIAGNOSIS

First, the theoretical concepts, on which the present approach is based, will be introduced. This will provide the basis to solve the sensor placement problem for fault detectability and isolability.

The sensor placement problem is then solved in two main steps. First step deals with those sensors that solve the problem only for fault detectability, while in the second step the sensors solving the fault isolability problem are computed. The *detectability* and the *isolability problem* are presented separately. Then, they are combined to finally solve the problem for both detectability and isolability.

4.1 Preliminary concepts

The basic idea of this approach is to study the detectability and isolability achieved by installing every candidate sensor individually. Then, the complete solution is derived from the result obtained when each individual sensor is considered. In order to do so, it is assumed that the model has no under-determined part, i.e., $M^- = \emptyset$. Lemma 4 shows that the overdetermined equations remain in the overdetermined part when new equations are taken into account.

Lemma 4. Let M_1 and M_2 be two arbitrary sets of equations such that $M_1 \subseteq M_2$, then it holds that

$$M_1^+ \subseteq M_2^+. \quad (6)$$

Proof. See Lemma 7 in Krysander et al. (2008). \square

Theorem 5 shows that the overdetermined part of the model with a sensor set S placed in it can be deduced from the overdetermined parts of the model obtained by placing single sensor $s \in S$ one at a time. Thus, from the diagnosability analysis of every single sensor, it can be straightforwardly obtained the diagnosability properties of the process for any combination of sensors.

Theorem 5. Let M be a structural model with no under-determined part and S a set of sensors. Then, it holds that

$$\bigcup_{s \in S} (M \cup \{e_s\})^+ = (M \cup M_S)^+, \quad (7)$$

where M_s is the sensor equation set of S and e_s is the equation of the sensor $s \in S$.

Proof. Let α be the minimal surplus of all subsets of M , then $p_0(M') \geq \alpha$ for any $M' \subseteq M$. Since $\text{var}(M_S) \subseteq$

$\text{var}(M)$, it holds that $p_0(M' \cup M_S) \geq \alpha - n$ for any $M' \subseteq M$ where n is the number of equations in M_S . This implies that any subset in $M \cup M_S$ with minimal surplus contains the equation set M_S . Hence, the set $(M \cup M_S)^+$ can be rewritten as $E \cup M_S$ where $E \subseteq M$. Then $(M \cup M_S)^+$ is the minimal set with minimum surplus only if E is the minimal set in M with $p_0(E) = \alpha$ and $\text{var}(M_S) \subseteq \text{var}(E)$.

Since M_S is an arbitrary set of sensor equations, the above reasoning also holds for single sensor equations. Therefore, it follows for any $e_s \in M_S$ that

$$(M \cup \{e_s\})^+ = K \cup \{e_s\}, \quad (8)$$

where $K \subseteq M$ is the minimal set in M with $p_0(K) = \alpha$ and $\text{var}(\{e_s\}) \subseteq \text{var}(K)$.

Now, let \mathcal{K} be the family of K sets obtained for every $e_s \in M_S$, i.e.,

$$\mathcal{K} = \{K \subseteq M \mid K \cup \{e_s\} = (M \cup \{e_s\})^+, e_s \in M_S\}, \quad (9)$$

and also let \mathcal{L}_α be the family of the surplus minimizers, i.e.,

$$\mathcal{L}_\alpha = \{U \subseteq M \mid p_0(U) = \alpha\}. \quad (10)$$

Then, it holds that the set E and all the set $K \in \mathcal{K}$ are contained in \mathcal{L}_α . Recall that the set of minimizers in \mathcal{L}_α define a lattice, which means that $\bigcup_{K \in \mathcal{K}} K \in \mathcal{L}_\alpha$. Note also that $\text{var}(M_S) \subseteq \text{var}(\bigcup_{K \in \mathcal{K}} K)$. Hence, since E is the minimal set in \mathcal{L}_α containing all the equations in M_S , it follows that $\bigcup_{K \in \mathcal{K}} K \supseteq E$. From this, it can be deduced that

$$\begin{aligned} \bigcup_{s \in S} (M \cup \{e_s\})^+ &= \left(\bigcup_{K \in \mathcal{K}} K \right) \cup \left(\bigcup_{s \in S} \{e_s\} \right) = \\ &= \left(\bigcup_{K \in \mathcal{K}} K \right) \cup M_S \supseteq E \cup M_S = (M \cup M_S)^+. \end{aligned} \quad (11)$$

Finally, from Lemma 4 it directly holds that $(M \cup \{e_s\})^+ \subseteq (M \cup M_S)^+$ since $e_s \in M_S$. Then, it can be straightforwardly stated that

$$\bigcup_{s \in S} (M \cup \{e_s\})^+ \subseteq (M \cup M_S)^+. \quad (12)$$

Thus, the proof is concluded from (11) and (12). \square

From Theorem 5 and taken detectability and isolability definitions into account, it can be intuitively seen how detectability and isolability properties of the process when a sensor configuration is installed can be deduced by gathering the detectability and isolability properties achieved by installing individual sensor one by one.

4.2 Sensor placement for detectability

In this subsection, the sensor placement sub-problem for fault detectability is first addressed. Recall that, $M^- = \emptyset$ by assumption and also that, according to Definition 1, any fault is detectable as long as its corresponding fault equation is in M^+ . Therefore, it is only needed to focus on the fault equations in the just-determined part of the model in order to fulfill detectability.

Now, let $F_D(s)$ be the class of detectable faults when sensor s is chosen for installation, i.e.,

$$F_D(s) = \{f \in D \mid e_f \in (M \cup \{e_s\})^+\}. \quad (13)$$

Based on Theorem 5, it can be stated that a sensor configuration $S' \subseteq S$ is a solution for the detectability problem as long as

$$\bigcup_{s \in S'} F_D(s) = D. \quad (14)$$

Hence, the set of $F_D(s)$ classes that covers D characterises a possible solution. Therefore, given the set $\mathcal{F}_D = \{F_D(s_1), F_D(s_2), \dots, F_D(s_n)\}$ (for $n = |S|$) of all detectability fault classes, the sensor placement problem for fault detectability is solved if all minimal subsets $C \subseteq \mathcal{F}_D$ that cover D are found.

Algorithm 1 is proposed for solving Problem 3 when only detectability specification is sought. First, the detectability fault classes $F_D(s)$ are generated. Then, all minimal covers are computed by means of the procedure `MinimalSetCovers`. In Section 5, this procedure is introduced. Finally, in last step, the minimal sensor sets $\mathcal{S}_D = \{S_1, S_2, \dots, S_m\}$ that solve the problem are obtained from the family of all minimal covers $\Gamma = \{\{C_1\}, \{C_2\}, \dots, \{C_m\}\}$.

Algorithm 1 $\mathcal{S}_D = \text{SPforDetectability}(M, S, D)$

```

 $\mathcal{F}_D = \emptyset$ 
for all  $s \in S$  do
   $F_D(s) = \{f \in D \mid e_f \in (M \cup \{e_s\})^+\}$ 
   $\mathcal{F}_D = \mathcal{F}_D \cup \{F_D(s)\}$ 
end for
 $\Gamma = \text{MinimalSetCovers}(\mathcal{F}_D, D)$ 
 $\mathcal{S}_D = \{S_i \subseteq S \mid \forall s \in S_i \Rightarrow F_D(s) \in C_i, C_i \in \Gamma\}$ 

```

4.3 Sensor placement for isolability

This subsection explains how to place sensors for fault isolability. Theorem 5 can be easily extended to the isolability case by replacing M with $M \setminus \{e\}$ in (7). Then, isolability performance achieved by installing a set of sensors can be determined by studying the isolability achieved with each individual sensor. Similarly to the detectability case, a class of isolable fault pairs $F_I(s)$ for each sensor $s \in S$ is defined,

$$F_I(s) = \{(f_i, f_j) \in I \mid e_{f_i} \in ((M \setminus \{e_{f_j}\}) \cup \{e_s\})^+\}. \quad (15)$$

Therefore, a sensor configuration $S' \subseteq S$ is a solution for the isolability problem if its corresponding isolability fault classes $F_I(s)$ cover all the pairs in I , i.e.,

$$\bigcup_{s \in S'} F_I(s) = I. \quad (16)$$

This means that the same approach for the detectability case in Subsection 4.2 can be developed here. Let $\mathcal{F}_I = \{F_I(s_1), F_I(s_2), \dots, F_I(s_n)\}$ be the set of all isolability fault classes, then all minimal subsets $C \subseteq \mathcal{F}_I$ that cover I characterise the solution of the sensor placement problem for fault isolability. Note, however, that now the elements to be covered are pair of faults instead of single faults. Algorithm 2 summarizes the sensor placement approach for the fault isolability case.

4.4 Sensor placement for fault detectability and isolability

As mentioned, the problem of placing sensors for fault detectability and isolability is directly solved in two steps.

Algorithm 2 $\mathcal{S}_I = \text{SPforIsolability}(M, S, I)$

```

 $\mathcal{F}_I = \emptyset$ 
for all  $s \in S$  do
   $F_I(s) = \{(f_i, f_j) \in I \mid e_{f_i} \in ((M \setminus \{e_{f_j}\}) \cup \{e_s\})^+\}$ 
   $\mathcal{F}_I = \mathcal{F}_I \cup \{F_I(s)\}$ 
end for
 $\Gamma = \text{MinimalSetCovers}(\mathcal{F}_I, I)$ 
 $\mathcal{S}_I = \{S_i \subseteq S \mid \forall s \in S_i \Rightarrow F_I(s) \in C_i, C_i \in \Gamma\}$ 

```

First, the detectability problem is solved. And then, based on the obtained results, the isolability problem is solved. Before solving the isolability problem, the sensor subset $S' \subseteq S$ computed in the detectability problem is added to the model by means of its corresponding sensor equations set, i.e., $M \cup M_{S'}$, whereas the sensors in S' are removed from the candidate sensor set, i.e., $S \setminus S'$. This procedure is performed for each sensor set $S' \in \mathcal{S}_D$.

Algorithm 3 is introduced to solve the sensor placement problem for fault detection and isolation. All minimal sensor subsets of S solving Problem 3 are computed.

Algorithm 3 $\mathcal{S} = \text{SPforFDI}(M, S, D, I)$

```

 $\mathcal{S} = \emptyset$ 
 $\mathcal{S}_D = \text{SPforDetectability}(M, S, D)$ 
for all  $S' \in \mathcal{S}_D$  do
  Construct  $M_{S'}$ 
   $M_i = M \cup M_{S'}$ 
   $S_i = S \setminus S'$ 
   $\mathcal{S}_I = \text{SPforIsolability}(M_i, S_i, I)$ 
   $\mathcal{S} = \mathcal{S} \cup \{S' \cup S_j \mid S_j \in \mathcal{S}_I\}$ 
end for
return  $\mathcal{S}$ 

```

Sensor faults As mentioned before, sensors are system components and thus can be faulty. However, fault in sensors are not taken into account in Algorithm 3. In order to handle these faults, here the sensor equation e_s will be used to represent the corresponding sensor fault.

The presence of a sensor fault in the analysis depends on whether the sensor is regarded installed, which hinders the characterization of the diagnosis specifications. Nevertheless, this characterization is relaxed by the following statements (Krysander and Frisk, 2008):

- (a) A fault of a sensor placed for solving the detectability problem is always detectable.
- (b) Faults of two sensors placed for solving the detectability problem are always isolable between them.
- (c) A fault of a sensor placed for solving the isolability problem is always isolable from any other fault.

Therefore, according to these statements, only isolability between fault of sensors placed for the detectability problem needs to be specified in order to handle sensor faults. Let $S_f = \{s_1, s_2, \dots, s_m\} \subseteq S$ be a subset of sensors, $F_{S_f} = \{f_{s_1}, f_{s_2}, \dots, f_{s_m}\}$ be the set of its corresponding sensor faults that need to be isolable from any other fault, and $S' \in \mathcal{S}_D$ be a solution of the detectability problem, then the isolability specification when sensor faults are regarded is defined as

$$I_{S_f} = \{(f_i, f_{s_j}) \in D \times F_{S_f} \mid f_i \in D, s_j \in (S' \cap S_f)\}. \quad (17)$$

These new isolability specifications should be computed for each sensor set $S' \in \mathcal{S}_D$ and inserted into the I set before the `SPforIsolability` call in Algorithm 3.

Repeated sensors There may exist the possibility of installing the same sensor more than once. From Theorem 5, it follows that repeated sensors are only useful to improve sensor fault isolability. Furthermore, let s' denote the repeated sensor of $s \in S$, then it holds that $(\{e_s, e_{s'}\})^+ = \{e_s, e_{s'}\}$, which implies that the fault of s becomes completely isolable from any other fault if the repeated sensor s' is installed in the process. This means that installing more than one repeated sensor per variable does not improve any diagnosis specification.

Based on this discussion and statement (c), it can be concluded that only the sensors placed to solve the detectability problem need to be repeated at most once. Let $S_r \subseteq S$ the subset of sensors that can be repeated and $S' \in \mathcal{S}_D$ a solution of the detectability problem, then the set of candidate sensors to solve the isolability problem is $(S \setminus S') \cup (S_r \cap S')$.

Algorithm 3 is modified in order to handle both sensor faults and repeated sensors. The resulting procedure is summarized in Algorithm 4 where now the set S_f of faulty sensors and the set S_r of repeated sensors are specified.

Algorithm 4 $S = \text{SPforFDI}(M, S, D, I, S_f, S_r)$

```

 $S = \emptyset$ 
 $S_D = \text{SPforDetectability}(M, S, D)$ 
for all  $S' \in \mathcal{S}_D$  do
  Construct  $M_{S'}$ 
   $I_{S_f} = \{(f_i, f_{s_j}) \mid f_i \in D, s_j \in (S' \cap S_f)\}$ 
   $M_i = M \cup M_{S'}$ 
   $S_i = (S \setminus S') \cup (S_r \cap S')$ 
   $I_i = I \cup I_{S_f}$ 
   $S_I = \text{SPforIsolability}(M_i, S_i, I_i)$ 
   $S = S \cup \{S' \uplus S_j \mid S_j \in S_I\}$ 
end for
return  $S$ 

```

Note that, now, the operator \uplus is used as the joint operation in the union of $S' \in \mathcal{S}_D$ and $S_j \in \mathcal{S}_i$ to indicate that multiple occurrences of the same element are allowed in the solution.

5. MINIMAL SET COVERS ALGORITHM

There exists a duality between the *hitting set problem* and the *set cover problem*. Hence, algorithms used to solve hitting set problems can be used to solve set cover problems and vice versa.

Let A be a set of elements and \mathcal{B} be a family of subsets of A . A subset $H \subseteq A$ is a *hitting set* if it has non-empty intersection with any set $B \in \mathcal{B}$, i.e., it is said that H hits each and every one set in \mathcal{B} . On the other hand, a subfamily of subsets $\mathcal{C} \subseteq \mathcal{B}$ is a *cover* if the union of its sets is A , then it is said that the sets in \mathcal{C} cover A . The duality of both problems can be seen by representing both sets in a bipartite graph $G(A, \mathcal{B}; E)$ where the set of edges is defined as: $(a, B) \in E$ if the element-node $a \in A$ is contained in the set-node $B \in \mathcal{B}$. Then, a hitting set is any subset of nodes in A such that their adjacent nodes

are all nodes in \mathcal{B} . Analogously, a cover is any subset of nodes in \mathcal{B} such that their adjacent nodes are all nodes in A .

In this paper, finding all minimal covers is of interest to solve the sensor placement problem. A cover $\mathcal{C} \subseteq \mathcal{B}$ is minimal if there is no subfamily $\mathcal{C}' \subset \mathcal{C}$ that is a cover. The algorithm presented in De Kleer and Williams (1987) to compute the minimal hitting sets is here used to find now all minimal covers. This is done by first determining the set of nodes in \mathcal{B} adjacent to each node in A and then solving the minimal hitting set problem.

Algorithm 5 $\Gamma = \text{MinimalSetCovers}(\mathcal{B}, A)$

```

 $\mathcal{E} = \emptyset$ 
for all  $a \in A$  do
   $\mathcal{E} = \mathcal{E} \cup \{B \in \mathcal{B} \mid a \in B\}$  {Find the adjacent nodes}
end for
Use iteratively the De Kleer and Williams (1987) algorithm for each set in  $\mathcal{E}$ 
Store the minimal hitting sets in  $\Gamma$ 
return  $\Gamma$ 

```

Algorithm 5 returns by means of Γ all minimal sets in \mathcal{B} that each of them covers A . It is important to point out that this algorithm is non-polynomial and hence computational issues may be expected for large number of sets in \mathcal{B} .

6. EXAMPLE

In this section, an illustrative example is shown in order to clarify the procedure steps of Algorithm 4. The same example introduced in Krysander and Frisk (2008) is used here. The analytical model consists of five equations, $M = \{e_1, e_2, e_3, e_4, e_5\}$, of the following form:

$$\begin{aligned}
 e_1: \quad \dot{x}_1 &= -x_1 + x_2 + x_5 \\
 e_2: \quad \dot{x}_2 &= -2x_2 + x_3 + x_4 \\
 e_3: \quad \dot{x}_3 &= -3x_3 + x_5 + f_1 \\
 e_4: \quad \dot{x}_4 &= -x_4 + x_5 + f_2 \\
 e_5: \quad \dot{x}_5 &= -5x_5 + u + f_3
 \end{aligned}$$

where x_i (for $i = 1, \dots, 5$) are the unknown process variables. Three faults $F = \{f_1, f_2, f_3\}$ are defined, which corresponds to inconsistencies in equations e_3, e_4 and e_5 , respectively (i.e., $e_{f_1} = e_3, e_{f_2} = e_4$ and $e_{f_3} = e_5$). Moreover, all unknown variables can be measured by introducing the set of candidate sensors $S = \{s_1, s_2, s_3, s_4, s_5\}$, where sensor s_i measures variable x_i .

The required diagnosis specifications are maximum fault detectability and isolability. Therefore, after performing the diagnosability analysis with all sensors installed, the following detectability and isolability sets are defined:

$$\begin{aligned}
 D &= D_{max} = \{f_1, f_2, f_3\}, \\
 I &= I_{max} = \{(f_1, f_2), (f_1, f_3), (f_2, f_3)\}.
 \end{aligned}$$

Furthermore, sensor faults and repeated sensors for all $s \in S$ will be taken into account, i.e., $S_f = \{s_1, s_2, s_3, s_4, s_5\}$ and $S_r = \{s_1, s_2, s_3, s_4, s_5\}$.

First, sensors for fault detectability are computed, i.e., $S_D = \text{SPforDetectability}(M, S, D)$. Algorithm 1 generates the following fault detectability classes for each sensor:

$$\begin{aligned}
F_D(s_1) &= \{f_1, f_2, f_3\} \\
F_D(s_2) &= \{f_1, f_2, f_3\} \\
F_D(s_3) &= \{f_1, f_3\} \\
F_D(s_4) &= \{f_2, f_3\} \\
F_D(s_5) &= \{f_3\}
\end{aligned}$$

Then, the minimal sets that cover D are

$$\Gamma = \{\{F_D(s_1)\}, \{F_D(s_2)\}, \{F_D(s_3), F_D(s_4)\}\},$$

which characterizes the solution

$$\mathcal{S}_D = \{\{s_1\}, \{s_2\}, \{s_3, s_4\}\}.$$

After solving the detectability problem, the isolability problem is solved for each set in \mathcal{S}_D . First, $S' = \{s_1\}$ is chosen with

$$\begin{aligned}
M_i &= M \cup \{e_{s_1}\}, & \text{for } e_{s_1} : x_1 = y_1, \\
S_i &= S, \\
I_i &= I \cup \{(f_1, f_{s_1}), (f_2, f_{s_1}), (f_3, f_{s_1})\}.
\end{aligned}$$

Algorithm 2 produces for each sensor in S_i , the following classes of isolable fault pairs:

$$\begin{aligned}
F_I(s_1) &= \{(f_1, f_{s_1}), (f_2, f_{s_1}), (f_3, f_{s_1})\} \\
F_I(s_2) &= \{(f_1, f_3), (f_2, f_3), (f_1, f_{s_1}), (f_2, f_{s_1}), (f_3, f_{s_1})\} \\
F_I(s_3) &= \{(f_1, f_2), (f_1, f_3), (f_2, f_3), (f_1, f_{s_1}), (f_3, f_{s_1})\} \\
F_I(s_4) &= \{(f_1, f_2), (f_1, f_3), (f_2, f_3), (f_2, f_{s_1}), (f_3, f_{s_1})\} \\
F_I(s_5) &= \{(f_1, f_3), (f_2, f_3), (f_3, f_{s_1})\}
\end{aligned}$$

The minimal covers of I_i are

$$\Gamma = \{\{F_I(s_1), F_I(s_4)\}, \{F_I(s_2), F_I(s_4)\}, \{F_I(s_1), F_I(s_3)\}, \{F_I(s_2), F_I(s_3)\}, \{F_I(s_3), F_I(s_4)\}\}.$$

These covers, together with the already installed sensors in S' , characterize the following partial solution

$$\begin{aligned}
\mathcal{S} &= \{\{s_1, s_1, s_4\}, \{s_1, s_2, s_4\}, \{s_1, s_1, s_3\}, \\
&\quad \{s_1, s_2, s_3\}, \{s_1, s_3, s_4\}\}.
\end{aligned}$$

The same procedure is repeated for $S' = \{s_2\}$ and $S' = \{s_3, s_4\}$. The new computed minimal sensor sets are added to the solution, which finally is

$$\begin{aligned}
\mathcal{S} &= \{\{s_1, s_1, s_4\}, \{s_1, s_2, s_4\}, \{s_1, s_1, s_3\}, \\
&\quad \{s_1, s_2, s_3\}, \{s_1, s_3, s_4\}, \{s_2, s_2, s_4\}, \\
&\quad \{s_2, s_2, s_3\}, \{s_2, s_3, s_4\}, \{s_3, s_3, s_4, s_4\}\}.
\end{aligned}$$

Any of these sensor configurations is suitable to achieve the specified fault detectability D and isolability I , as well as detectability and isolability of all sensor faults according to the chosen sensor configuration.

7. CONCLUSIONS

A novel approach for the sensor placement problem has been addressed in this paper. The computed solution characterises all possible sensor configurations that meet the required diagnosis specifications. Typically, maximum fault detectability and isolability are the most relevant specifications. However, since required specifications are explicitly expressed, Algorithm 4 can handle any different detectability and isolability specifications. Furthermore, faults in the extra sensors as well as repeated sensors are also taken into consideration.

A key step to characterise all possible solutions is to formalize the problem as a minimal set cover problem

where all the minimal covers need to be found. However, it should be noted that solving the minimal set cover problem may entail computational problems for large number of classes.

The method presented in this paper has some similarities with the method present in Krysander and Frisk (2008). Both methods solve first the detectability problem and then, based on the obtained results, the isolability problem is solved in order to finally compute all possible solutions. The main difference is the classes used to compute the solution. Here a class of faults induced by each sensor is used, whereas in Krysander and Frisk (2008) a class of sensors induced by each fault is required. This entails that the procedure to compute the solution for the detectability problem (and thereby also the isolability problem) is different in both works. However, the classes used in one paper can be easily derived from the classes used in the other paper which makes that, in terms of computational complexity, both works can be implemented with equivalent efficiency.

Finally, the main contribution of this paper is Theorem 5 which allows to verify that what can be done in terms of detectability and isolability with a set of sensors can be determined by the union of what can be done by each of them. This result can be useful in more sophisticated search algorithms devoted to the sensor placement problem for fault detection and isolation.

REFERENCES

- Bagajewicz, M. (2000). *Design and Upgrade of Process Plant Instrumentation*. Technomic Publishers, Lancaster, PA.
- Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*. Springer, 2nd edition.
- Commault, C., Dion, J.M., and Agha, S.Y. (2008). Structural analysis for the sensor location problem in fault detection and isolation. *Automatica*, 44(8), 2074 – 2080.
- De Kleer, J. and Williams, B.C. (1987). Diagnosing multiple faults. *Artificial Intelligence*, 32(1), 97 – 130.
- Dulmage, A.L. and Mendelsohn, N.S. (1958). Covering of bi-partite graph. *Canada J. Math*, 10, 527–534.
- Krysander, M., Åslund, J., and Nyberg, M. (2008). An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, 38(1).
- Krysander, M. and Frisk, E. (2008). Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, 38(6), 1398–1410.
- Lovász, L. and Plummer, M. (1986). *Matching Theory*. North-Holland.
- Murota, K. (2000). *Matrices and Matroids for Systems Analysis*. Springer-Verlag.
- Rosich, A., Sarrate, R., and Nejjari, F. (2009). Optimal sensor placement for fdi using binary integer linear programming. In *20th International Workshop on Principles of Diagnosis (DX-09)*. Stockholm, Sweden.
- Travé-Massuyès, L., Escobet, T., and Olive, X. (2006). Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, 36(6), 1146–1160.