Abstract: Maintenance and diagnosis of complex systems are common activities in the industrial world. Technological advances have led to a continuously increasing complexity of industrial systems. This complexity, which is due to an increasing number of components reduces in turn the reliability of plants. Therefore, fault diagnosis is becoming a growing field of interest. But fault diagnosis relies on sensors: efficient fault diagnosis procedures require a relevant sensor placement. This paper presents fundamental results for sensor placement based on diagnosability criteria. These results contribute to the design of sensor placement algorithms, which satisfies diagnosability specifications.

Keywords: fault diagnosis, diagnosability, sensor placement

1. INTRODUCTION

Sensor placement decisions depend on expected objectives. For instance, in control theory, the sensor placement is used to provide sufficient information for the control of systems. Criteria deal with observability and controllability of the variables. Madron and Veverka [1992] has proposed a sensor placement method which deals with linear system. This method makes use of the Gauss-Jordan elimination to find a minimum set of variables to be measured. This ensures the observability of variables while simultaneously minimizing the cost of sensors. In this theory, the observable variables include the measurable variables plus the unmeasured but deductible variables. Another method for sensor placement has been proposed in Maquin et al. [1997]. This method aims at guaranteeing the detectability and isolability of sensor failures. The proposed method is based on the concept of redundancy degree in a variable and the structural analysis of the system model. The sensor placement can be solved with a matricial analysis of a cycle matrix or using the technique of mixed linear programming. Commando et al. [2006] has proposed a method of sensor location. In this method, they defined a new set of separators (Irreducible Input Separators), which generates sets of system variables in which additional sensors must be implemented to solve the considered problem.

However, in fault diagnosis, the goal of sensor placement should be to satisfy detectability and diagnosability properties. Detectability is the possibility of detecting a fault on a component and diagnosability is the possibility of identifying a fault on a component without this creating ambiguity with any other fault.

Travé-Massuyès et al. [2001] has proposed a method based on consecutive additions of sensors, which takes into account diagnosability criteria. The principle of this method is to analyze the physical model of a system from a structural point of view. This structural approach is based on Analytical Redundancy Relations (ARR) Cassar and Staroswiecki [1997], which can be obtained from combinations of model constraints using bipartite graph Blanke et al. [2003] or elimination rules Ploix et al. [2005], and on the corresponding signature table Patton and Chen [1991]. In a signature table, rows and columns represent respectively, the set of analytical redundancy relations and the set of considered faults. However, this method requires an a priori design of all the ARR for a given set of sensors. This paper presents results for the design of sensor placement algorithms. Thanks to these results, the sensor placement satisfying diagnosability objectives becomes possible without designing ARR a priori. It is an important feature since it is no longer necessary to design all the possible ARR assuming all the variables are measured.

2. PROBLEM FORMULATION

In the following, the set of variables appearing in a constraint $k$ is denoted: $\text{var}(k)$ and the set of variables appearing in the set of constraints $K$: $\text{var}(K) = \bigcup_{k \in K} \text{var}(k)$. A system $\Sigma$ can be described by a tuple $(K_\Sigma, C_\Sigma)$. $\text{var}(K_\Sigma)$ is the set of variables that models phenomena influenced by $\Sigma$. The behavior is represented by constraints $K_\Sigma = \{\ldots, k_i, \ldots\}$ that establish relationships between variables of $\text{var}(K_\Sigma)$. It can be represented by a structural matrix $M_\Sigma$, which is an incidence matrix representing the application $M_\Sigma : \text{var}(K_\Sigma) \rightarrow K_\Sigma$. $C_\Sigma = \{\ldots, c_i, \ldots\}$ is a set of independent components constituting $\Sigma$. Each constraint in $K_\Sigma$ models one component and, conversely, a component can be modeled by at most one constraint.
∀k ∈ KΣ, \(comp(k) \in CΣ\)

where \(comp(k)\) refers to the component corresponding to the constraint \(k\). Let us introduce the concept of testable subsystem (TSS) and its relationship with the concept of ARR.

**Definition 1.** Let \(K\) be a set of constraints and \(v\) a variable in \(var(K)\) characterized by its domain \(dom(v)\). \(K\) is a solving constraint set for \(v\) if using \(K\), it is possible to find a value set \(S\) for \(v\) such that \(S \subseteq dom(v)\). A solving constraint set for \(v\) is minimal if there is no minimal solving constraint set for \(v\). A minimal solving constraint set \(K\) for \(v\) is denoted: \(K \vdash v\).

**Definition 2.** Let \(K\) be a set of constraints. \(K\) is testable if and only if there are two distinct subsets \(K_1 \subseteq K\), \(K_2 \subseteq K\) such that \(K_1 \nsubseteq K_2\) and \(K_2 \nsubseteq K_1\), and a variable \(v \in var(K)\) such that \(K_1 \vdash v\) and \(K_2 \vdash \neg v\). If this property is satisfied, it is indeed possible to check if the value set \(S_1\) deduced from \(K_1\) is consistent with the value set \(S_2\) deduced from \(K_2\): \(S_1 \cap S_2 \neq \emptyset\).

This definition also applies to models containing ordinary differential equations. Indeed, testable state space representations, including state space observers, always have equivalent parity space representations Staroswiecki et al. [1991].

Adding any constraint to a testable set leads also to a testable set of constraints. Only minimal testable sets are interesting.

**Definition 3.** A testable set of constraints is minimal if it is not possible to keep testability when removing a constraint.

A global testable constraint that can be deduced from a TSS is called ARR. Let \(RΣ = \{r_1, r_2, \ldots\}\) be the set of all the testable subsystems that can be deduced from \(KΣ\) according to Blanke et al. [2003], Ploix et al. [2005], Staroswiecki and Declerck [1989]. Because of the one-to-one relationships between constraints and components, notions of detectability and discriminability can be extended to constraints. Therefore, usual definitions Struss et al. [2002] related to continuous systems can be extended from faults to constraints. Let \(R\) be a set of TSS coming from \((KΣ, CΣ)\).

**Definition 4.** A constraint \(k \in KΣ\) is detectable (see Struss et al. [2002]) in \(R\) if \(\exists r_i \in R/k \in r_i\). By extension, the constraints \(K \subseteq KΣ\) are detectable in \(R\) if \(\forall k_i \in K\), \(k_i\) is detectable in \(R\).

**Definition 5.** Two constraints \((k_1, k_2) \in KΣ\) are discriminable (see Struss et al. [2002]) in \(R\) if: \(\exists r_i \in R/k_1 \in r_i\) and \(k_2 \notin r_i\) or if \(\exists r_j \in R/k_2 \in r_j\) and \(k_1 \notin r_j\). By extension, the constraints of a set \(K \subseteq KΣ\) are discriminable in \(R\) if: \(\forall (k_i, k_j) \in KΣ\), \(k_i\) and \(k_j\) are discriminable in \(R\) with \(k_i \neq k_j\).

Obviously, non detectability of both constraints \((k_1, k_2)\) implies nondiscriminability of \((k_1, k_2)\).

**Definition 6.** A constraint \(k \in KΣ\) is diagnosable (see Struss et al. [2002]) in \(R\) if: it is detectable and if \(\forall k_j \in (KΣ \setminus k), (k, k_j)\) are discriminable in \(R\). By extension, the constraints \(K \subseteq KΣ\) are diagnosable in \(R\) if: \(\forall k_i \in K\), \(k_i\) is diagnosable in \(R\).

In order to formulate the sensor placement problem, the notion of terminal constraint has to be introduced.

**Definition 7.** A terminal constraint \(k\) is a constraint that satisfies: \(\text{card}(\text{var}(k)) = 1\). A terminal constraint usually models a sensor or an actuator: \(\text{var}(k)\) is generally a measured or a controlled variable. It may also be a variable for which the value is assumed (such ambient temperature). It is thus a major concept in sensor placement.

In fault diagnosis, sensor placement has to satisfy specifications dealing with detectability and diagnosability. Because of the one-to-one relation between components and constraints, what is true for components is also true for constraints. Therefore, the components \(CΣ\) and the corresponding constraints \(KΣ\) may be decomposed into several sets:

- the set of components \(C_{\text{diag}}\) / constraints \(K_{\text{diag}}\) that has to be diagnosable
- the set of subsets of components \(C_{\text{nondi}}\) \(\cap C_{\text{diag}} = \emptyset\)
- constraints \(K_{\text{nondi}}\) / constraints \(K_{\text{diag}}\) that has to be nondiagnosable
- the set of components \(C_{\text{nondet}}\) / constraints \(K_{\text{nondet}}\) has to be non detectable

Specifications \(C_{\text{diag}}\), \(C_{\text{nondis}}\) and \(C_{\text{nondet}}\) of sensor placement problems are meaningful if the two following properties are satisfied:

1. Sets in specifications must not to overlap each other to make sense: constraint sets have to satisfy: \(\forall C_1 \in C_{\text{nondis}}, C_i \cap C_{\text{nondet}} = \emptyset\), \(\forall C_1 \in C_{\text{nondis}}, C_i \cap C_{\text{diag}} = \emptyset\) and \(\forall (C_i, C_j) \in C_{\text{nondis}}, C_i \cap C_j = \emptyset\) (no overlapping property).
2. The union of all the components appearing in \(C_{\text{diag}}\), \(C_{\text{nondi}}\) and \(C_{\text{nondet}}\) has to correspond to \(CΣ\): \(CΣ = C_{\text{diag}} \cup C_{\text{nondi}} \cup \bigcup C_i \in C_{\text{nondi}}\) (completeness property).

If these properties are satisfied the specifications are qualified as consistent in \(CΣ\). Replacing components by corresponding constraints leads to the same properties for specifications \(K_{\text{diag}}, K_{\text{nondis}}\) and \(K_{\text{nondet}}\) to be consistent in \(KΣ\).

Satisfying the specifications requires information delivered by sensors. Let \(Σ\) represent the system \(Σ\) with the additional sensors. \(Σ\) can be described by a tuple \((KΣ, CΣ)\) where \(CΣ\) represents the components of system \(Σ\) plus the additional sensors and \(KΣ\) represents the constraints of system \(Σ\) plus the additional terminal constraints which model the sensors. The sensor placement problem consists in determining the additional terminal constraints in \(KΣ\) that lead to the satisfaction of the specification \(K_{\text{diag}}, K_{\text{nondis}}\) and \(K_{\text{nondet}}\). Because of the relations between constraints and components, the results can be extended to components.

In the next sections, fundamental results are proposed for the design of sensor placement satisfying diagnosability.
and detectability specifications. Algorithms are not de-
tailed in this paper.

3. PRELIMINARY CONCEPTS

Before deducing diagnosability properties of constraint
sets, some concepts have to be introduced.

3.1 Value propagation as a theoretical tool

According to definition 3, a TSS is a minimal set of
constraints $K$ such that there exists a constraint $k \in K$
for which all the variables of $\text{var}(k)$ can be instantiated,
starting from terminal constraints. An ARR corresponding
to a TSS can be seen in different ways. The most common
approach is to consider an ARR as a global constraint.
Another way is to think of an ARR as a complete value
propagation from [1994] w.r.t. variables i.e. a propagation
that leads to information about the consistency of a
set of constraints, including terminal constraints that
contain known data. This approach has been adopted as
a theoretical tool to develop proofs. Relationship between
value propagation and ARR is detailed in this section.

Let $k_1$ and $k_2$ be two constraints. The propagation of a variable $v$ between $k_1$ and $k_2$ is possible only if $v \in \text{var}(k_1) \cap \text{var}(k_2)$. The variable $v$ is qualified as propagable
between $k_1$ and $k_2$. Consider a system, defined by $K_\Sigma = \{k_1, k_2, k_3, k_4, k_5\}$ with $\text{var}(k_1) = \{v_1, v_3\}$, $\text{var}(k_2) = \{v_1, v_2\}$, $\text{var}(k_3) = \{v_2, v_3\}$, $\text{var}(k_4) = \{v_2\}$ and $\text{var}(k_5) = \{v_3\}$. Terminal constraints $k_4$ and $k_5$ model sensors or actuators. Each terminal constraint contains known data. The set of all TSS that can be tested is represented by the
propagations drawn in figure 1.

![Propagation Diagrams](image)

Fig. 1. Set of propagations

A propagation starts by a terminal constraint, which
means that “a variable is equal to a known value”. In this
example, propagations start either with $k_4$ or $k_5$. Thanks
to these constraints, a value can be respectively assigned
to $v_2$ and $v_3$. Once values have been assigned to these
variables, new variables can then be instantiated. Propagation
continues until no more assignments are possible because terminal constraints or instantiated variables have
been reached. The set of constraints that appears in a
propagation, corresponds to a testable subsystem. These
constraints can be combined into a unique global con-
straint named ARR. Depending on the constraints chosen
for propagating values, different ARR may be obtained
(see figure 1). In the continuation of this paper, value
propagation is implicitly used and appears in the proofs
of the different lemmas and theorems.

3.2 Some characteristics of constraint sets

The concept of linked constraints is introduced because it is
important regarding sensor placement. Indeed, discrimi-

ability depends on this concept.

As mentioned in Blanke et al. [2003], the constraints of
a system $\Sigma$ may be modeled by a non directed bipartite
graph $(K_\Sigma, \text{var}(K_\Sigma), E_\Sigma)$ where $E_\Sigma$ is the set of edges.
Each edge $e = (k, v)$ models that $v \in \text{var}(k)$ . Let us
introduce new definitions useful for sensor placement.

Definition 8. A set of constraints $K \subset K_\Sigma$ is intercon-
ected by a set of variables $V \subset \text{var}(K_\Sigma)$ if there is a
tree $(K, V, E) \subset (K_\Sigma, \text{var}(K_\Sigma), E_\Sigma)$ with constraints
at extremities (see Bollobás [1998] for example), which
satisfies $\text{card}(V) = \text{card}(K) - 1$.

Definition 9. A set of constraints $K \subset K_\Sigma$ is linked in
$K_\Sigma$ by a set of variables $V \subseteq \text{var}(K_\Sigma)$ if $K$ is interconected
by $V$ and iff the other constraints of $K_\Sigma$ (i.e. $K_\Sigma \setminus K$)
do not contain any variable of $V$. The variables of $V$ are
called linking variables for $K$. They are denoted: $\text{var}_{\text{linking}}(K, K_\Sigma)$.

The shape of a structural matrix dealing with linked
constraints is drawn in figure 2.

![Structural Matrix](image)

Fig. 2. Structural matrix of a constraint set, which is linked
by path

The concept of linked constraints is strongly connected
with discriminability.

Lemma 10. A set of constraints $K \subset K_\Sigma$ linked by a set
of variables $V \subset V_\Sigma$ is necessarily non discriminable.

Proof. Indeed,

(1) because variables in $V$ only appear in the constraints
in $K$, the only way of propagating variables is to use
the constraints in $K$ and the variables in $V$,
(2) because there is a tree $(K, V, E) \subset (K_\Sigma, \text{var}(K_\Sigma), E_\Sigma)$
with constraints at extremities, instantiating all the
variables in $V$ involves at least the achievement of the
propagations defined by the tree.

Therefore, all the constraints are invariably found together
in TSS: $K$ is non discriminable.

In order to improve the clarity of these explanations, let
us introduce the notion of stump variables.

Definition 11. A set of variables $\text{var}(K)$ appearing in a
set of constraints $K$ but not in the other constraints of
\(K_\Sigma\) (i.e. \(K_\Sigma\backslash K\)) are named stump variables in \(K_\Sigma\). They are denoted: \(\text{var}_\text{stump}(K,K_\Sigma)\). For instance, the set of variables \(V\) that links a set of constraints \(K\) belong to the stump variables \(\text{var}_\text{stump}(K,K_\Sigma)\).

A set of constraints cannot be used to generate a TSS if they are linked and if there are additional variables that cannot be propagated. These constraints are qualified as isolated. Detectability depends on this concept.

**Definition 12.** A set of several constraints \(K \subset K_\Sigma\) is isolated in \(K_\Sigma\) by a set of variables \(V \subset \text{var}(K_\Sigma)\) if they are linked by \(V\) and if there is at least one variable in \(\text{var}(K)\backslash V\) that does not belong to other constraints of \(K_\Sigma\) (i.e. \(K_\Sigma\backslash K\)). If the set contains only one constraint, the link condition disappears but the other remains.

The shape of a structural matrix dealing with isolated constraints is drawn in figure 3.

![Structural matrix of a constraint set](image)

**Fig. 3.** Structural matrix of a constraint set, which is isolated by the set of variables \(V\)

The concept of isolated constraints is strongly linked with detectability.

**Lemma 13.** A set of constraints \(K \subset K_\Sigma\) isolated in \(K_\Sigma\) by \(V\) is necessarily non detectable.

**Proof.** The constraints \(K\) isolated in \(K_\Sigma\) by \(V\) will always come together in TSS because, by definition, they are linked by \(V\). Because of the fact that, in isolated constraints, there is at least one additional variable in \(\text{var}(K)\) which does not appear in other constraints (i.e. \(K_\Sigma\backslash K\)), it is not possible to instantiate this value and, therefore, this set of constraints cannot be involved into a TSS: \(K\) is non detectable.

### 4. CONSTRUCT SET AND DIAGNOSABILITY PROPERTIES

This section aims at setting up a direct link from sets of constraints to detectability and diagnosability properties. Firstly, it is obvious that adding additional constraints connected to all the variables \(\text{var}(k)\) appearing in a constraint \(k\), ensures the diagnosability of \(k\).

**Lemma 14.** Let \(k \in K_\Sigma\) be a constraint. If additional terminal constraints dealing with all the variables in \(\text{var}(k)\) are added, then the constraint \(k\) is diagnosable.

**Proof.** Because there are additional terminal constraints connected to each variable in \(V(k)\), a value can be assigned for each variable. Consequently, there is one TSS containing \(k\) plus additional terminal constraints connected to variables in \(\text{var}(k)\). Therefore, the constraint \(k \in K\) is necessarily diagnosable because there is one TSS that does not contain other constraints of \(K_\Sigma\) (i.e. \(K_\Sigma\backslash \{k\}\)).

Lemma 14 can be directly applied to all the constraints of a constraint set.

**Corollary 15.** If additional terminal constraints dealing with all the variables \(\text{var}(K)\) of a constraint set \(K \in K_\Sigma\), then each constraint \(k \in K\) is diagnosable.

In lemma 13, a relationship between isolated constraints and the detectability property has been presented. The next lemma generalizes the previous results.

**Lemma 16.** A sufficient condition for a subset of constraints \(K \subset K_\Sigma\) to be non detectable is that there is a tuple \((K_1,\ldots,K_m)\) of \(m\) sets of constraints making up a partition \(\mathcal{P}(K)\) of \(K\) such that each \(K_i\) is isolated in \(K_\Sigma\backslash \bigcup_{j<i} K_j\) (\(K_1\) is a limit case: it should be isolated in \(K_\Sigma\)).

**Proof.** The case of \(K_1\) has been discussed in lemma 13: because the constraints in \(K_1\) are isolated in \(K_\Sigma\), they are non detectable and therefore cannot be included in TSS. Then, the remaining candidate constraints for TSS belong to \(K_\Sigma\backslash K_1\). Because \(K_\Sigma\) is isolated in \(K_\Sigma\backslash K_1\), they are non detectable. The reasoning can be extended to any \(i\). Consequently, the constraints in \(K = \bigcup_i K_i\) are non detectable.

Figure 4 indicates the shape of a structural matrix of non detectable constraints.
16. If there are no additional terminal constraints containing \( v_1, v_2 \) and \( v_3 \), the subset \( K \) is non detectable.

**Lemma 17.** A sufficient condition for each set \( K_j \subseteq K \) belonging to a set of \( m \) constraint sets \( K = \{ K_1, \ldots, K_m \} \) such that \( \forall K_i \neq K_j, K_i \cap K_j = \emptyset \), to be non discriminable is that each \( K_j \) is linked by a set of variables \( V_i \).

**Proof.** This lemma is a direct application of lemma 10 to several sets of constraints.

Consider, for example, a system modeled by the following structural matrix:

\[
\begin{array}{c|ccccc}
 & v_1 & v_2 & v_3 & v_4 & v_5 \\
\hline
k_1 & 1 & 0 & 1 & 1 & 1 \\
k_2 & 1 & 1 & 1 & 1 & 0 \\
k_3 & 1 & 1 & 1 & 0 & 1 \\
k_4 & 0 & 1 & 1 & 0 & 0 \\
k_5 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

Assume that \( K = \{ k_1, k_2, k_3, k_4, k_5 \} \) is a constraint subset that should be non discriminable. Because the constraints \( k_1, k_2, k_3 \) and \( k_4 \) are linked by \( V = \{ v_1, v_2, v_3 \} \), lemma 17 is satisfied. Therefore, \( k_1, k_5, k_2, k_3 \) and \( k_4 \) are non discriminable provided that no additional terminal constraints contain a variable of \( V \).

The following theorem groups lemmas 14, 16 and 17.

**Theorem 18.** Let \( K_\Sigma \) be a set of constraints and \( K_{\text{nondet}}, K_{\text{diag}} \) be the specifications of a sensor placement problem consistent in \( K_\Sigma \). Sufficient conditions for the specifications to be fulfilled are:

1. there exists a tuple \( (K_1, \ldots, K_p) \) of \( p \) sets of constraints making up a partition \( \mathcal{P}(K_{\text{nondet}}) \) of \( K_{\text{nondet}} \) such that each \( K_i \) is isolated in \( K_\Sigma \setminus \bigcup_j K_j \) \((K_1 \) is a limit case: it should be isolated in \( K_\Sigma \)) see figure 4.
2. each set \( K_j \) belonging to \( K_{\text{nondet}} = \{ K_1, \ldots, K_m \} \) such that \( \forall K_i \neq K_j, K_i \cap K_j = \emptyset \), is linked by a set of variables \( V_i \) in considering only the constraints \( K_\Sigma \setminus K_{\text{nondet}} \).
3. Additional terminal constraints are added on the variables: \( V_{\text{candidate}} = \var(K_\Sigma) \setminus (\var(\text{stump}(K_{\text{nondet}}, K_\Sigma)) \cup \bigcup K_j \in K_{\text{nondet}} \var(\text{linking}(K_j, K_\Sigma \setminus K_{\text{nondet}}))) \) (see figure 5).

**Proof.** The proof relies on the resulting structure of the structural matrix, which directly stems from corollary 15 and lemmas 16 and 17. Note that point 2 could also be stated for the whole set of constraints \( K_\Sigma \). However, it is not useful to include non detectable constraints, which will not appear in resulting TSS: it would be less conservative.

Because of lemma 16 and 17, the variables of \( \var(K_{\text{diag}}) \) cannot contain variables appearing in the variables involved in (1) and (2) i.e. in \( \var(\text{stump}(K_{\text{nondet}}, K_\Sigma)) \) and in \( \bigcup K_j \in K_{\text{nondet}} \var(\text{linking}(K_j, K_\Sigma \setminus K_{\text{nondet}})) \). Then, \( \var(K_{\text{diag}}) \) satisfies: \( \var(K_{\text{diag}}) \subseteq V_{\text{candidate}} \). Because the variables of \( V_{\text{candidate}} \) can be instantiated with measured values, all the constraints of \( K_{\text{diag}} \) are diagnosable following corollary 15.

The point that has to be proved is that, in specifications, \( K_{\text{nondis}} \) defines non discriminable but detectable sets and not only non discriminable sets as in lemma 17: the detectability of sets in \( K_{\text{nondis}} \) has to be proved.

The variables \( \var(K_i) \) of a constraint set \( K_i \in \mathbb{K}_{\text{nondis}} \) can be decomposed into two sets: \( V^-_i \) and \( V^+_i \) where \( V^-_i = \var(\text{linking}(K_i, K_\Sigma \setminus K_{\text{nondet}})) \) contains the linking variables and \( V^+_i \) contains the remaining variables \( V^+_i = \var(K_i) \setminus V^-_i \). Because of lemma 16 and 17, the set \( V^+_i \) cannot contain variables in \( \var(\text{stump}(K_{\text{nondet}}, K_\Sigma)) \) and in \( \bigcup K_j \in K_{\text{nondet}} K_j \neq K_i \var(\text{linking}(K_j, K_\Sigma)) \). Therefore, \( V^+_i \) satisfies: \( V^+_i \subseteq V_{\text{candidate}} \).

Because of the third point of the theorem, all the variables of \( V_{\text{candidate}} \) are known: additional terminal constraints are indeed added, there is necessarily a TSS dealing with all the constraints in \( K_\Sigma \). It proves that the constraint set \( K_j \) is necessarily detectable. Because this result holds for any \( K_i \in \mathbb{K}_{\text{nondis}} \), it proves the theorem.

**Fig. 5.** Shape of a structural matrix Satisfying theorem 18 Satisfying theorem 18 guarantees that the specifications are satisfied. However, because the theorem provides only a sufficient condition for diagnosability, the number of additional terminal constraints is not necessarily minimal. It has to be checked afterwards.

The sensor placement problem has been studied without considering components. Let us now take components into account. Components of a system may be divided into three sets: the components on which faults need to be isolated, the components on which faults need to be detected but not necessarily localized and the components on which faults need to be non detectable. Because it has been assumed that each component is modeled by only one constraint, the results obtained for constraints can be extended to components using the application \( \Phi_\Sigma : K_\Sigma \rightarrow C_\Sigma \).

5. APPLICATION TO DAMADICS BENCHMARK

Several methods for fault isolation have been benchmarked on a pneumatic servo-motor actuated valve named DAMADICS (Development and Application of Methods for Actuator diagnosis in Industrial Control Systems). Spanache and Escobet [2004] has designed a sensor placement method for this problem that optimize the diagnosability level of the system. In this section, the method proposed in this paper, is applied on this benchmark. The system is defined by the following equations:
\[ k_1 : X = r_1(P_s, \Delta P) \]
\[ k_2 : F_V = r_2(X, \Delta P) \]
\[ k_3 : CVI = r_3(SP, PV) \]
\[ k_4 : P_s = r_4(X, CVI, P_z) \]
\[ k_5 : PV = r_5(X) \]

The corresponding structural matrix is given in Table 1.

<table>
<thead>
<tr>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
<th>( k_4 )</th>
<th>( k_5 )</th>
<th>( \Delta P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Let’s fix these specifications: \( K_{nondet} = \{k_1, k_2\} \), \( K_{nondis} = \{k_3, k_4\} \) and \( K_{diag} = \{k_5\} \).

The set of constraints \( K_{nondet} = \{k_1, k_2\} \) is linked by the path \( \{k_1, \Delta P, k_2\} \). Because of variable \( F_V \), \( K_{nondet} \) is isolated by the path \( \{k_1, \Delta P, k_2\} \).

The set of constraints \( K = \{k_3, k_4\} \in K_{nondis} \) is linked by the path \( \{k_3, CVI, k_4\} \). Then, according to theorem 18, no terminal constraints containing a variable from \( \{\Delta P, F_V, CVI\} \) have to be added i.e. these variables have not to be measured.

In order to satisfy the last item of theorem 18, all the variables of the system except \( \{\Delta P, F_V, CVI\} \) have to be measured.

The method proposed in Ploix et al. [2005] has been used to design all the ARR. It has led to the fault signature matrix drawn in Table 2.

<table>
<thead>
<tr>
<th>( \text{TSS} )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
<th>( k_4 )</th>
<th>( k_5 )</th>
<th>( k_6 )</th>
<th>( k_7 )</th>
<th>( k_8 )</th>
<th>( k_9 )</th>
<th>( k_{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{TSS}_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \text{TSS}_2 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \text{TSS}_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \text{TSS}_4 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

According to these results, the constraints that cannot be discriminated are: \( \{k_3, k_4\} \), the constraints that cannot be detected are: \( \{k_1, k_2\} \) and the diagnosable constraint is: \( \{k_5\} \). Applying the function \( \Phi : K_{ \Sigma } \rightarrow C_{ \Sigma } \), it is obvious that the components, which cannot be discriminated are: \( \{c_3, c_4\} \) and the components, which cannot be detected is: \( \{c_1, c_2\} \). The diagnosable component is: \( \{c_5\} \).

The results presented in this paper demonstrate that it is possible to design sensor placements which satisfy diagnosability criteria without designing ARR a priori.

6. CONCLUSION

New results for the design of sensor placement algorithms has been proposed. It manages, the specifications dealing with sets of constraints that have to be diagnosable, non discriminable or non detectable. These results apply to any system depicted by constraints, which may only be described by the variables appearing in them. Thanks to these results, sensor placements satisfying diagnosability specifications become possible without designing ARR a priori. It is a very important feature since it is no longer necessary to design all the possible ARR assuming that some variables are measured. An algorithm providing solutions to the sensor placement problem that contains a minimum number of sensors will be provided in the near future.

REFERENCES


S. Ploix, M. Désinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In 16th IFAC World Congress, Prague, Czech republic, 2005.


P. Struss, B. Rehfeld, R. Brignolo, F. Cascio, L. Console, P. Dague, P. Dubois, O. Dressler, and D. Millet. Model-based tools for the integration of design and diagnosis into a common process- a project report. In DX’02, Semmering, Austria, May 2–4 2002.