Influence Matrix Approach for an Optimal Sensor Placement

Abid Sabrina[1], Haffaf Hafid[2]
[1]Department of Computer science
Laboratory of Research of Industrial Computing and
Network, University Oran1 Ahmed BenBella
Bp 1524 El M'Naouer Oran, Algeria
abid.sabrina@univ-mascara.dz, haffaf.hafid@univ-oran.dz

ABSTRACT: *Fault Detection and Isolation (FDI) procedure increase assurance on quality, reliability and safety in industrial systems. A suitable installed sensors in an industrial process is a necessary condition for fault diagnosis. Sensor placement for diagnosis purposes consists to study which process variables have to be measured to satisfy diagnosis specifications. Analytical redundancy relations (ARRs) are used frequently in the area of diagnosis as well as optimizing, analyzing, and validating of sensors of the system. This paper presents the optimal sensor placement approach based on structural analysis methods using tripartite graph approach. The proposed approach allows to study which sensors are required to be installed in a process in order to improve certain fault diagnosis specifications; and it includes 2 phases: (i) development of systematic and efficient approaches for the generation of ARRs set allowing to generate "influence matrix", (ii) then the multi-criteria optimization is applied by selecting robust sensor placements (Pareto Optimal Solutions) leading to a sensor placement algorithm. The proposed method has been validated on a robot dynamic model taken as a benchmark where the benefits of the method are clearly shown.*

Keywords: Optimal Sensor Placement, Fault Detection and Isolation, Structural Analysis, Influence Matrix, Tripartite Graph, Pareto Optimal Solution

Received: 4 August 2018, Revised 10 October 2018, Accepted 1 November 2018

1. Introduction

Maintenance and diagnosis of complex systems are common activities in the industrial world. The failures within these systems can cause disruption to the operational functionality. Effective FDI approaches enhance the operational functionality by reducing the maintenance cost. Indeed, the performance of a diagnostic system highly depends on the number and location of sensors.

The problem of sensor placement for FDI consists in determining the optimal set of instruments such that the places of sensors allow that predefined set of faults could be detected and isolated. The usual objective to be minimized in the sensor placement problem is the global sensor cost with regard to industrial constraints. The sensor placement problem can then be viewed as a combinatorial problem classified N-P hard which consists in finding a sensor combination that fulfills diagnosis specifications. Solving the sensor placement for diagnosis can be treated from many different points of view. Indeed, such a problem depends on the kind of system description, the required diagnosis specifications, as well as the technique used to implement the diagnosis system.

This paper deals with the optimal sensors placement based on a given diagnosis performance specification.

The problem of sensor placement has been widely investigated in the FDI literature and several approaches proposed by researchers have been selected to be discussed in section 2. Most methods mentioned are limited to the optimization criteria related to the sensors placement as well as how the system model can be presented.

Unlike previous work, we present a tripartite graph representation model and introduce a new maximum diagnosability metric in the optimization model. For this purpose, we develop new approach for extracting all useful information in the system initially configured, and then we construct a table which we call influence matrix. This paper presents results for a multi-criteria optimization of the design sensor placement satisfying diagnosability objectives with minimizing both the global cost and the sensors number to be settled.

The present work can be considered as a new structural representation able to extract residual cycles from tripartite graphs and yielding more efficiently to redundancy relationships rather than extracting a matching from bipartite graph methods [1][2].

In order to demonstrate the efficiency of this optimal sensor placement formulation, an industrial system is chosen as a benchmark. It consists of a robot dynamic model and some faults are defined to be diagnosed. In this paper, fault diagnosis systems are based on consistency checking by means of structural models. The required diagnosis specifications to be fulfilled are fault detection and isolation for a predefined set of faults.

The paper is organized as follows: The related works are discussed in section 2. Section 3 introduces preliminary concepts and gives a background of the theoretical tools used in the development of the proposed method. The proposed methodology and a formal problem formulation is presented in Section 4. Section 5 describes the algorithm used to solve the aforementioned problem. In Section 6, the dynamic model of the robot system benchmark is presented, section 7 presents the system implementation and the results of placement. Finally, some conclusions and remarks are given in Section 8.

2. Related Work

The ability to detect and to isolate faults which may affect the system depends essentially on the instrumentation's architecture. This is why, before designing an industrial supervision system, determination of monitoring ability based on technical specifications is important. Fig.1 gives an overview about the state of the art dealing with this problem. Three categories of methods can be distinguished : the data based method, the model based method and the hybrid method.

A sensor placement method basically depends on the objectives it aims to achieve. For example, in the theory of command, the sensor placement is used to provide sufficient information to the control system so we have to check the classical controllability and observability properties. Several methods have been proposed for Fault Tolerant Control (FTC)[3][4][5]

A sensor placement method for structural health monitoring (SHM) is proposed in [6][7], an other method proposed in [8] with the objective of damage detection.

However, in fault diagnosis, the goal of sensor placement is to satisfy detectability (observability) and diagnosability (monitorability or surveillability) properties. For the sensor placement problem dealing with FDI objective, we distinguish three types of methods: data based methods, model based methods and hybrid methods.

2.1 Data based Methods

In Data-based methods, the knowledge of the system is based on large amounts of recorded and collected data of the process under nominal and faulty conditions. Based on this information, different methods can be used to extract features from the process history so that it can be used as knowledge to the diagnosis system. According to the available works, most are based on neuronal approaches (RN) [9], genetic algorithms (GA) [10], Binary Particular Swarm Optimization(BPSO) Algorithm [11] and Fuzzy Feature Selection Approach [12].

The most advantage of data based approaches is not requiring the knowledge of an analytical or structural model of the system. They only require information collected in databases through the history of the system functioning. The main drawback of the use of these methods consists in the fact that they need recognition pattern step, the physical knowledge is also omitted and

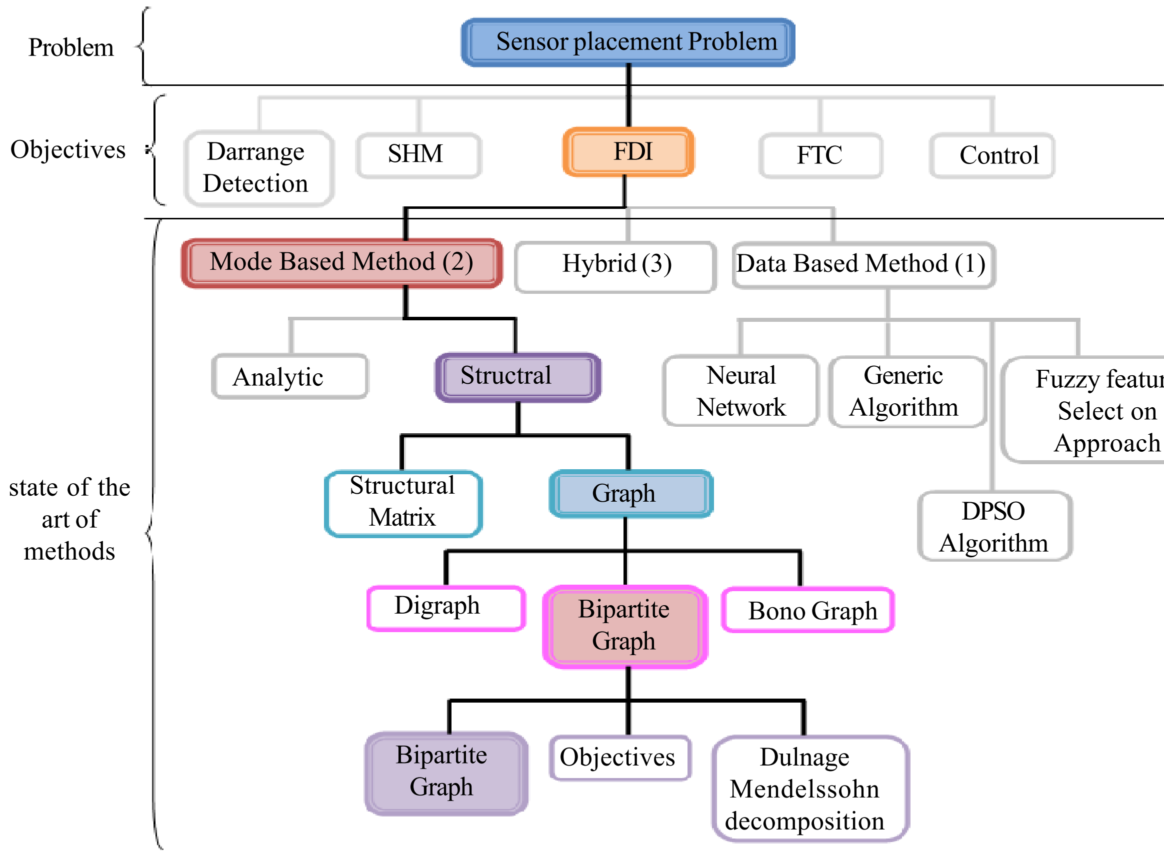

Figure 1. State of the art methods of sensor placement problem

then the sensor placement methods are mainly reduced to heuristics.

2.2 Model based Methods

On the other hand, Model-based methods use the analytical redundancy relations (ARRs) for which is applied the sensor placement algorithm; Indeed, it is possible to monitor the system through to the redundancy of information in the system. Furthermore, in addition to the modelling step problem, ARRs generation is not trivial and needs complicated unknown variables theory elimination [1].

In model-based FDI, faults are modeled as deviations of parameter values or unknown signals, and diagnostic models are often brought back to a residual form. The main approaches to construct residuals are based on using ARRs.

To be able to perform model-based supervision, some redundancy is needed, and this redundancy is typically provided by physical sensors judiciously placed in the industrial process.

Analytical redundancy has to find relations between known variables of the system. These relations are satisfied when the system is in its normal mode and not satisfied in the presence of failure. Model-based FDI algorithms depend highly on the accuracy of the system model. The model that represents the system can be in analytical form [13][14][15][16] or structural form. The drawback of model-based diagnosis is the need for the knowledge about the behavioral model.

Furthermore, analytical approaches require accurate models and numerical values of the parameters, which are not always available in real systems. This is why structural approaches based on the system model can be an alternative to study fault diagnosability and fault recoverability possibilities.

In a stream of research, we have reported two types of structural approaches: one based on graph representation and the other, on structural Matrix [17][18][19].

In graphical models, the nodes represent the system variables and the system equations, while the arcs connect each variable node to its associated equation node. Among these methods we can cite, for example, bipartite graphs, bond graphs [20][21] [22] [23]and digraph [24] can be mentioned.

Existing graphical approaches allow us to generate structural properties of the system without knowing numerical values of the parameters. For example the method based on a bi-partite graph can be obtained early in the development process, without major engineering efforts. This kind of models is suitable to handle large scale systems since efficient graph-based tools can be used and does not have numerical problems. We have divided this later into three approaches; one based on matching [25][26], an other on Dulmage Mendelssohn decomposition [27][28][29] and an novel contribution is the use of tripartite graph [2][30].

2.3 Hybrid Methods

Hybrid methods can be viewed as a bridge across the Artificial Intelligence (AI) and FDI approaches to model-based sensor placement.

[31] have proposed and discussed a novel method for computing Minimal Additional Sensor Sets (MASS) by exploiting recent techniques based on the symbolic compilation of qualitative system models within a structural approach suitable for numerical equations models.

3. Preliminary Concepts

In order to present the sensor placement problem for FDI, a set of preliminary definitions are introduced.

Combinatorial optimization problems are characterized for having discrete decision variables, but an objective function and constraints formulation could take many forms (i.e. linear or non linear) [32]. Such problems have been studied for several years in mathematics and computer science where they have attracted a lot of attention, mainly because of their wide applicability. However, because of their complexity (i.e. combinatorial optimization problems tends to be NP-hard or NP-complete), the use of approximation algorithms (mainly metaheuristics) to solve them has become relatively popular in the last few years [33]. On the other hand, many real-world problems have two or more objectives (often conflicting) to be optimized at the same time .For example, we aim to minimize the time to complete the task, but simultaneously, we wish the task to be as cheap as possible, which is an objective that normally opposes to the previous one. These problems are known to be multi-objectives and their solution involves finding not a single solution, but several that correspond the the best possible trade-offs among all the optimized objectives. Numerous mathematical programming techniques exist to deal with multi-objective optimization problems [34, 35], however, the use of metaheuristics in this field has become increasingly investigated domain [36, 37].

3.1 Multi-objective Combinatorial Optimization Problem

A Multi-Objective Combinatorial Optimization Problem (MOCOP) is defined as:

$$\textit{Optimize } F(x) = (f_1(x)\ f_2(x), ..., f_n(x) \text{ with } x \in D \qquad (1)$$

Where n is the number of objectives ($n \geq 2$), $x = (x_1, x_2, ..., x_k)$ is the vector of decision variables, D is the set of feasible solutions and $F(x)$ is the objective funtion where each objective function $f_i(x)$ has to be optimized (i.e. minimized or maximized). Unlike single-objective optimization, the solution of a MOCOP is not unique, but is composed instead of a set of solutions representing the best possible solution space. Such solutions are contained in the so-called Pareto optimal set(PO)[38]. When plotting the objective function values corresponding to the solutions stored in the PO, we obtain the Pareto front of the problem.

3.2 Dominance and Pareto Optimality

In multi-objective optimization, there is a different notion of optimality than in singleobjective optimization, since in this case, we are interested in finding good compromises (or trade-offs) among the objectives that we wish to optimize. The notion of optimality most commonly adopted is that originally proposed by Francis YsidroEdgeworth in 1881 [39] and later (in 1896) generalized by Vilfredo Pareto. Although some authors call this notion the Edgeworth-Pareto optimality, the most commonly accepted term is Pareto optimality.

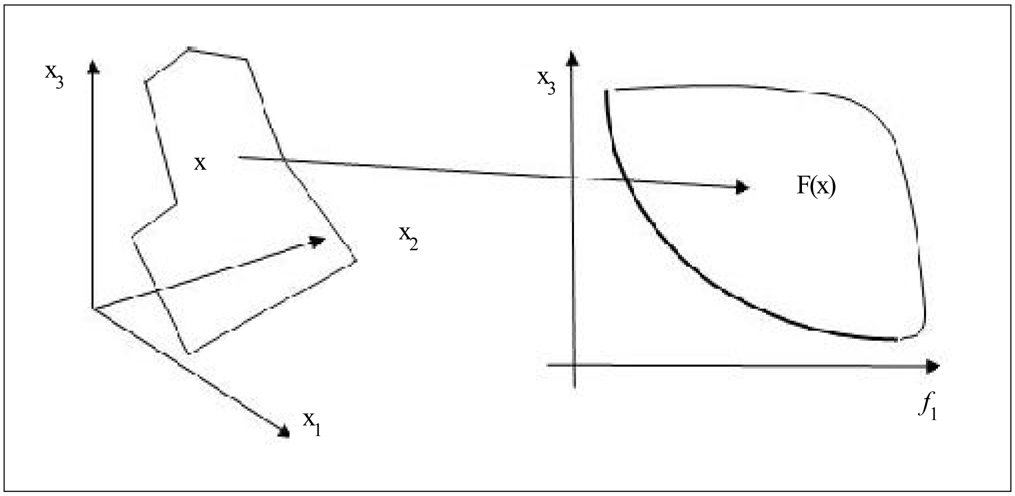

Figure 2. Mapping from decision space at left to objective space at right

A feasible solution $x^*\in D$ is called **Pareto optimal** (also called efficient or nominated) if and only if there is no solution $x\in D$ such that x dominates x^* . A solution $y = (y_1, y_2,..., y_k)$ **dominates** a solution $z = (z_1, z_2,..., z_k)$, in a minimization context, iff $\forall i\in [1..n]$, $f_i(y) < f_i(z)$ and $\forall i\in[1..n]$ such that $f_i(y) \leq f_i(z)$ and $\exists i\in[1..n]$. In this context, any solution of the Pareto optimal set may be considered as optimal, since no improvement may be found for an objective without degrading another a objective value. In the case of bi-objective minimization problem, the Pareto front of the efficient solutions obtained may be easily plotted (see the thick line in objective space from Figure 2).

3.3 Graphical Structural Analysis Approach

A general framework for diagnostic analysis feasibility based on structural analysis approach is proposed in [40]. The main principle of this method is to identify the measurements subsystems in the plan that contain redundant information independently from the detailed knowledge of parameters.

The types of variables in a diagnostic context are: (i) the known variables corresponding to measurements and controller input; (ii) the unknown variables, typically internal states and unknown inputs that should not influence the residual, and the faults to be detected.

Formally, the structural model of the system is defined as follows:

$R = \{R_1, R_2 ,..., R_m\}$ the set of structural equations.

$K= \{k_1, k_2 ,..., k_c\}$ the set of the known variables.

$X= \{x_1, x_2 ,..., x_n\}$ the set of the unknown variables.

$Z = X \cup K$ is the set of all the variables. $|Z| = n + c$.

A constraint R impose a relation between variables and parameters, belonging to $Z : R_j(Z_1, Z_2, ...,Z_{|z|}) = 0; j = 1, ... , m$.

The strength of structural approach is the fact that only relationships between variables and relations are considered regardless the nature of these relations (linear or not, static or not, ...)

3.4 Tripartite Graph

Furthermore, the structure of the model will be used to derive the diagnosis properties. The structural model can therefore be represented by a tripartite graph.

Tripartite Graph $G = (K, R, X, A)$ is constituted of three nodes parts where each pair of parts is a bipartite graph as shown in Figure 3. The set of edges A is then partitioned into A_c and A_x linking K to R in one hand and R to X in the other hand respectively. So we have two bipartite sub-graphs: $G_{B_1} = ((K, R, A_c)$ and $G_{B_2} = ((R, X, A_x)$ [2].

1. A ***residual*** is a relation where all variables are known.

2. A ***residual cycle*** is a closed path free loop (cycle) starting from K and ending in K in the tripartite graph; see Fig. 4 (all the variables involved by residual cycle will be known by deduction). Among all cycles in the tripartite graph, only this kind of cycle will be investigated [2].

3. ***Detectability*** is the possibility of detecting an occurrence of a fault and ***diagnosability*** is the possibility of identifying (without ambiguity) a fault on a component.

4. ***Fault signature matrix*** is a table where every line corresponds to a residue and every column to a failure. This table enables us to check the structural detectability and localizability property [41][42].

A failure is detectable if its signature comprises at least one "1", two failures are localizable, or identifiable if their signatures are different. The structural analysis gives only the structural proprieties.

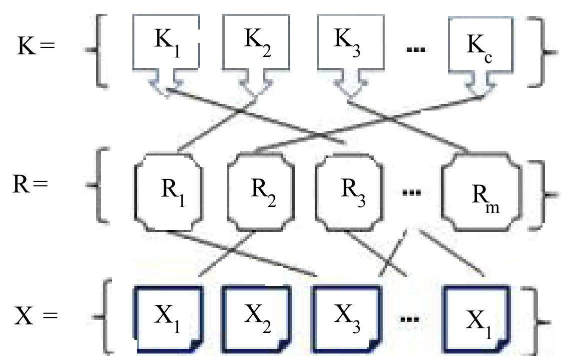

Figure 3. Tripartite graph associated to a system

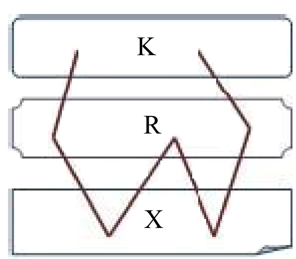

Figure 4. A residual cycle in a tripartite graph

4. Proposed Methodology

Fault diagnosis is of great importance for industrial systems. It represents an important factor for quality of service (related with fast maintenance response to fault situations). In these systems, it is obvious that only a limited number of sensors can be installed due to budget constraints. Since improper selections may seriously hamper diagnosis performance, the development of a sensor placement strategy has become an important research issue in recent years. Ideally, a sensor placement should be configured to facilitate fault detection and maximize diagnosis performance strategy under given sensor cost limitations. A multi-criteria sensor placement is defined as a minimum sensor configuration that achieves the minimum cost while observing pre-specified performance criteria.

4.1 Problem Formulation

In this paper, we deal with the sensor placement problem represented in our work as an optimization problem where the best sensor configuration fulfilling some given diagnosis specifications is sought [30]. The fault diagnosis based on extraction of residual cycles from tripartite graph [43] representation optimizes the diagnosis process by incorporating sensors information. So, we adopt this method to construct our optimized sensor placement algorithm. First of all, some basic concepts have to be defined.

Formal definition of the Problem

For the all installation there exist *n* sensors to measure *n* variables

• Constraints:

$$\hat{C} = min\sum_{i=1}^{n}(C_{S_i} \times Nb_{S_i}) \tag{2}$$

$$\hat{N}b_s = min\sum_{i=1}^{n} Nb_{S_i} \tag{3}$$

$$\hat{O} = min\sum_{i=1}^{n} F_i = max\sum_{i=1}^{n} D_{V_i} \tag{4}$$

• Objective function:

$$f(Nb_{S_1}, Nb_{S_2}, ..., Nb_{S_n}) = \hat{C} + \hat{N}b_s + \hat{O} \tag{5}$$

With:

n : Total of sensors which can be present in the system

$\hat{C}$: Total cost of the placement

C_{S_i} : Cost of the the *i* Sensor

Nb_{S_i} : Number of the sensors which measure the i^{th} variable in the placement (note that the we can measure a variable with several sensors)

$\hat{N}b_s$: Total number of sensors in the placement

$\hat{O}$: Total redundancy degree of the placement

D_{V_i} : Redundancy degree specification for the i^{th} variable (note that the i^{th} sensor correspond to the i^{th} variable)

F_i : fault in the i^{th} sensor (is a binary variable; 1 if a fault is present, 0 if not)

Let *S* be the feasible solutions and *f* the objective function. The objective is to find

$x^* = (Nb_{S_1}, Nb_{S_2}, ..., Nb_{S_n}) \in S$ such that $f(x^*) = \underbrace{min f(x)}_{x \in S}$.

4.2 Optimization of Sensors Placement for FDI

The problem of optimal sensor placement for FDI consists in determining the set of sensors that minimizes a pre-defined cost function, and satisfying at the same time, FDI specifications set. The Sensor placement objectives include:

• Cost reduction.

• Minimum number of sensors

• Max diagnosability and isolability property.

The latter property is difficult to express, and has not been solved formally by the research.

4.2.1 Multi-criteria optimal sensor placement procedure

Here, we describe the various stages of system design diagnostic program that we have proposed. We have divided the process of the placement procedure for diagnosis in 3 parts:

1. The calculation of the pareto solutions and objective precision i.e. the redundancy degree that we must ensure and sensors placement based on them,

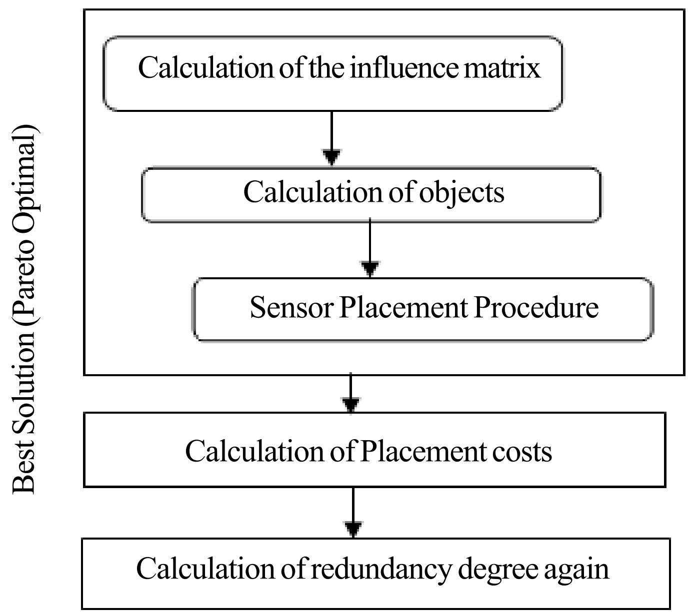

Figure 5. Proposed sheme of the optimal sensor placement

In the next, we discuss more in depth these parts.

4.2.2 Best Solution (Pareto Optimal)

Based on basic concepts as the structural analysis of systems and well known observability and monitorability structural criterion, a computer method of sensors placement has been proposed in [43]; this uses graph theory tools to bear all possible combinations. However, even if one may think that exploring all possibilities is a complex method, we solved efficiently the problem in a polynomial time. This method relies on the generation of residual cycles and paths through a representation of the system with a tripartite graph. The algorithm of generation of residual cycles is reliable and based on the development of an n tree and then extraction of all paths leading from the father node to the leaves. Thanks to various generated information (degree of redundancy and objectives); we can see what effectively better place is. [(a)]

1. Calculation of the Influence Matrix

The calculation of the Influence Matrix boils down to paths generating procedure during the first stage, its size is Nb_unknown_var × Nb_unknown_var which represent is what has provided the knowledge of each variable on other variables as degree of redundancy, the diagonal values are ones "1" (each variable adduce to itself a single redundancy degree; see Table 1). The elimination of all paths whose variables are not physically measurable leads to vanishe values in the Influence-Matrix, what brings the addition of these sensors on these variables (the meaning of influence).

Example. The knowledge of a variable x_1 allows us to estimate the variable x_3 and x_4 in her calculate path. So it generates one redundancy degree to these variables and to itself, x_4 is not measurable. So from this matrix we can clearly simulate the placement of new sensors (if we add a sensor to measure x_3 we can estimate the value of the variable x_4).

	x_1	x_2	x_3	x_4
x_1	**1**	0	1	1
x_2	0	**1**	0	0
x_3	0	0	**1**	0
x_4	0	0	0	0

Table 1. Influence Matrix example

2. Calculation of Objectives

The calculation of the objectives is to extract what remains to make the surveillance system, i.e. the redundancy degree. The objectives are the difference between what is requested in specifications and what is present in the system (redundancy generated during the first generation stage). This is the result of the Hamming distance between the vector of redundancy degrees and vector of specifications. The objective vanishes when it is no difference.

3. Sensors Placement

We are limited to deploy a small number of sensors, and thus have to carefully choose where to place them, so we place a sensor to measure a potential variable (which involve after its calculate paths more variables figuring in specifications), that's why this investment will lead to more redundancy for one place that minimizes the installation of new sensors [44].

We have created a solution quality threshold m (reliability threshold) that exceed it deteriorates the quality of solution. p in equation 6 is the number of sensors to be installed to verify the specifications without taking into account any optimality (in the worst case)

$$p = \sum_{i=1}^{n} Objectif_i \tag{6}$$

And b in equation 7 is the number of sensors in the best placement

$$b = \sum_{i=1}^{n} BestPlacement_i \tag{7}$$

To maintain the effectiveness of the solution it defects that number of sensors in placements not exceed the boundary b; so the threshold m depends on b .

This phase of the procedure is to find the most relevant sensors on the process towards its diagnosis. Based on analysis of adding sensors by the Influence-Matrix and the objectives, we made a correspondence by choosing to add a sensor on a variable at each time increasing the maximum redundancy degree to other variables contained in the objectives. The placement is done by calculating a distance (or metric) which is a measure of similarity between the two vectors. The two variables assigned to the vectors which ensures a minimum distance (or a maximum degree of similarity) with the vector objective are chosen. So we choose to place in the i^{th} placement the first best variable, the other placements take the second best variable. We take the maximum amount of information to verify the specifications redundancy degrees with installing minimum of sensors. These placements represent the Pareto optimal solutions presented in Figure 6 (by translating the maximum diagnosability by minimum faults).

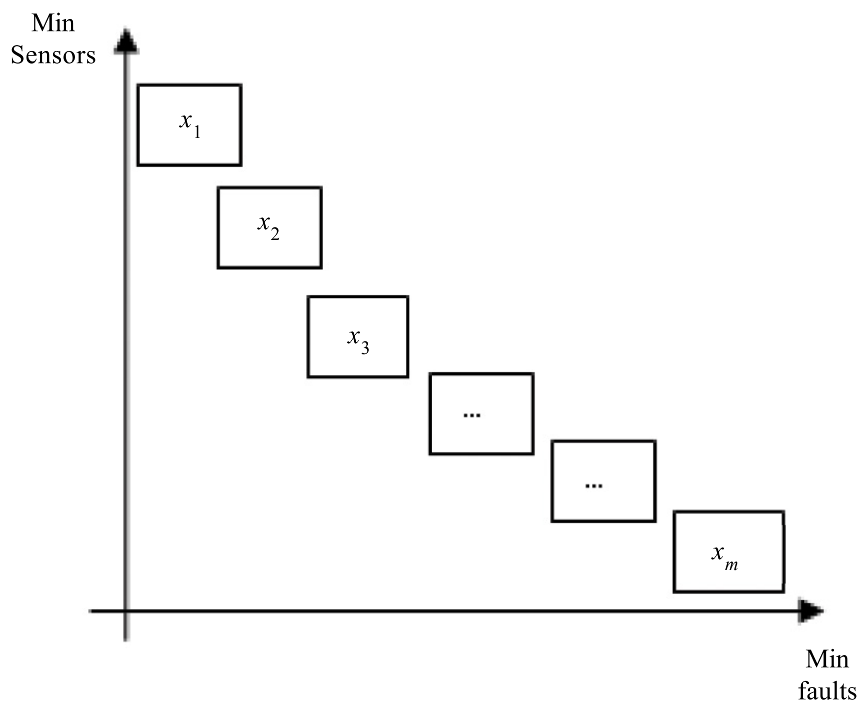

Figure 6. Graphical representation of the *m* Pareto solutions

4.2.3 Placement Cost

We calculate the cost of each placement.

$$Tcost_j = \sum_{i=1}^{n}(CS_i * NB_{S_{ij}}) \tag{8}$$

With:

$Tcost_j$: Total cost of j^{th} placement

CS_i: Cost of i^{th} Sensor

$NB_{S_{ij}}$: Number of i^{th} Sensor in j^{th} placement

4.2.4 Redundancy Degree Gain

We have calculated what additional gains in diagnosis can be made with which placement. This is accomplished by calculate the exceeding degrees in Influence matrix. Not only we have checked the specifications with a minimum number of sensors, but also a better quality of diagnosabilty property through redundancy degree calculated with influence matrix, so this matrix helps us to control adding sensors process.

$$TGain_j = \sum_{k=1}^{n}[\sum_{i=1}^{n}(Infuence\ Matrix\ line\ i * Placement_{ij}) - Objectif\,] \tag{9}$$

With:

$TGain_j$: Total gain in j^{th} placement

InfMat : Influence Matrix

5. Algorithm

Algorithm 1 ALGORITHM OF SENSORS PLACEMENT

Require: degree: array

{represents redundancy degrees in the system.}

specifications : array

{represents redundancy degrees of the specifications.}

Costs: array {represents cost of every sensor}

m: const {represents threshold of pareto solutions.}

1. **for** i:=1 to nb_inknown_var **do**
2. Placement_Vector[i]:=0;
3. **end for**
4. Calculate Influence_Matrix() [30];
5. Cancel the lines of Influence_Matrix where the variables are not physically measurable.
6. Calculate objectives = Hamming distance(specifications, degree)
7. **if** (objectives <= 0) **then**
8. End of Placement {Specifications verified with the initial placement.}
9. **else**
10. **while** (objectives >= 0) **do**
11. Select();

{Procedure which is to select the relevant variable. See algo2}

12. Place();

{Sensor Placement Procedure. See algo3}

13. Update objectives {ex- objectives what brings the placement as redundancy degrees.}
14. **end while**
15. **end if**
16. Calculate Total cost of each placement(); See algo4.
17. Calculate redundancy degree gain of each placement(); See algo5

Algorithm 2 Procedure Select()

1. **for** i:=1 to nb_unknown_var **do**
2. Counter[i]:=0;
3. **end for**
4. **for** i:=1 to nb_inknown_var **do**

```
5.        if (objectifs > 0) then
6.        for m:=1 to nb_inknown_var do
7.         if (Influence_Matrix[m][i]> 0) then
8.    Counter[m]:=Counter[m]+1;
{Calculate Metric between objectives and influence Matrix to ordering the best sensors}
9.        end if
10.       end for
11.       end if
12. end for
```

Algorithm 3 Procedure Place()

```
1. TRI: sort array of Counter;
{m represents the number of the best placements}
2. for i:=1 to m do
3. for j:=i+1 to m do
4.  Place_Vector[Tri[1]][i]=Place_Vector [Tri[1]][i]+1;
{place sensor in the first best place of Tri in the first placement}
5.  Place-Vector[Tri[2]] [j]=Place-Vector[Tri[2]] [j]+1;
{place sensor in the two second best place of Tri in the other placements }
6. end for
7. end for
```

Algorithm 4 Calculate Total cost of each placement()

```
1. for i:=1 to nb_inknown_var do
2. Totalcost[i]:=0;
3. end for
4. for i:=1 to m do
5.     for j:=1 to nb_inknown_var do
6.          Total_cost[i]:= Place-Vector[j] *cost[j]+Totalcost[i];
7.     end for
8. end for
```

Algorithm 5 Calculate redundancy degree gain of each placement()

```
for i:=1 to nb_inknown_var do
        for j:=1 to nb_inknown_var do
                  if (Influence-Matrix[i][j] > 0) then
                            Count[i]:= Count[i] +1;
                 end if
        end for
end for
for i:=1 to m do
        for j:=1 to nb-inknown-var do
         TGain[i]:= Place-Vector [i] [j]*Count[j]+objectives;
end for
end for
```

6. Application to Dynamic Model of the Robot

We applied our placement algorithm for diagnosis for several benchmark systems [43, 30]. In this section, the method proposed in this paper, is applied on the dynamic model of robot as benchmark. The model is expressed as:

$$\Gamma = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f_v\dot{q} + f_s sign(\dot{q}) \quad (10)$$

Where $\ddot{q}$ is the $n\times1$ accelerations vector, $\ddot{q}$ is the $n\times1$ velocities vector, $\dot{q}$ is the $n\times1$ coordinates vector and Γ is the $n\times1$ external torques vector. $M(q)$ represents the $n\times n$ positive definite inertia matrix, $C(q, \dot{q})$ is the $n\times n$ Coriolis and centrifugal forces matrix and $G(q)$ is the $n\times1$ gravitational torques vector. f_v is the $n\times n$ viscous frictions diagonal matrix, and f_s is the Coulomb frictions diagonal matrix, but these matrix will be ignored throughout our work. The Figure 7 shows the robot that we will study:

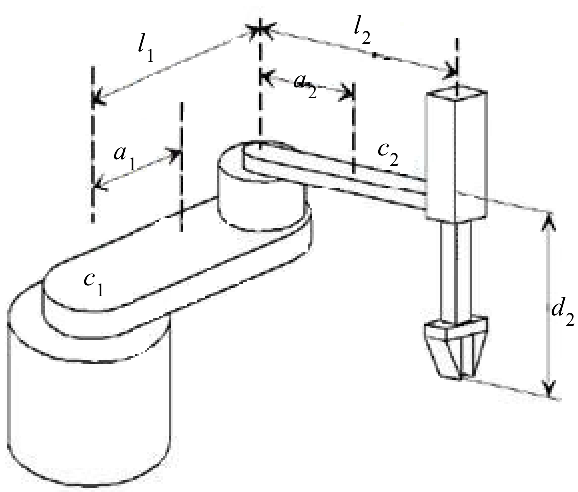

Figure 7. The Robot under study

The components of the robot are described by the following relations:

$$C1:\ddot{q}_1 = a\Gamma_1 + b\Gamma_2 + c\Gamma_3 + \dot{q}_1(a8{,}2428\sin x(3)x(4) - b4{,}1214\sin x(3)x(2) + cm_3 g) \quad (11)$$

$$C2:\ddot{q}_2 = d\Gamma_1 + e\Gamma_2 + f\Gamma_3 + \dot{q}_2(b4{,}1214\sin x(3)x(4) + fm_3 g) \quad (12)$$

$$C3:\ddot{q}_3 = g\Gamma_1 + h\Gamma_2 + i(\Gamma_3 + m_3 g) \quad (13)$$

$$m_1 : y_1(t) = q_1(t) \quad (14)$$

$$m_2 : y_2(t) = q_2(t) \quad (15)$$

$$m_3 : y_3(t) = q_3(t) \quad (16)$$

$$d_1 : \dot{q}_1(t) = \frac{dq_1(t)}{dt} \quad (17)$$

$$d_2 : \ddot{q}_1(t) = \frac{d\dot{q}_1(t)}{dt} \quad (18)$$

$$d_3 : \dot{q}_2(t) = \frac{dq_2(t)}{dt} \quad (19)$$

$$d_4 : \ddot{q}_2(t) = \frac{d\dot{q}_2(t)}{dt} \quad (20)$$

$$d_5 : \dot{q}_3(t) = \frac{dq_3(t)}{dt} \quad (21)$$

$$d_6 : \ddot{q}_3(t) = \frac{d\dot{q}_3(t)}{dt} \quad (22)$$

The analytical model of the robot is a set of:

• Three constraints: $C1$, $C2$ and $C3$

• Three measures: m_1, m_2 and m_3

• Six derivations : d_1 , d_2 , d_3 , d_4 , d_5 and d_6

This model connects nine unknown variables; it implies that we can generate three relations of analytical redundancy. It includes the following components:

• The inputs Γ_1, Γ_2 and Γ_3 of the three axis of robot.

-The position, the speed and acceleration (q_1, $\dot{q}_1$ and $\ddot{q}_1$)for axis 1, and so on for axis 2 and axis 3 .

• y : output,

• a,b, c, d, e,f , g, h, i coefficients,

• g : gravity.

7. System implementation

Given the structural model of the dynamic model of the robot (Fig. 8 represent the tripartite graph of the robot), and consider the following specification considered as input of the program implemented in Java:

a) The global cost of the sensors

b) The degree of redundancy of each variable to be monitored (see Table 6). The program allows to introduce any structured model which is presented by variables set and relations set.

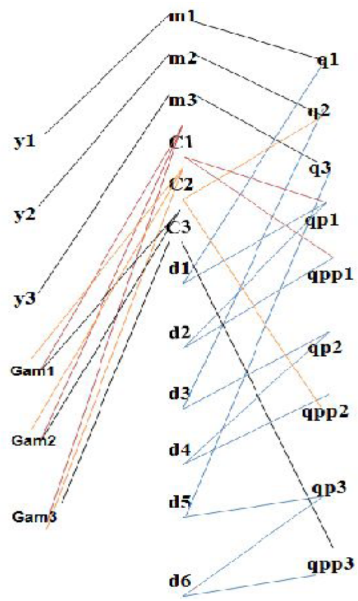

Figure 8. Tripartite graph of the example system

	q_1	$\dot{q}_1$	$\ddot{q}_1$	q_2	$\dot{q}_2$	$\ddot{q}_2$	q_3	$\dot{q}_3$	$\ddot{q}_3$
q_1	**1**	1	1	0	0	0	0	0	0
$\dot{q}_1$	1	**1**	1	0	0	0	0	0	0
$\ddot{q}_1$	1	1	**1**	0	0	0	0	0	0
q_2	0	0	0	**1**	1	1	0	0	0
$\dot{q}_2$	0	0	0	1	**1**	1	0	0	0
$\ddot{q}_2$	0	0	0	1	1	**1**	0	0	0
q_3	0	0	0	0	0	0	**1**	1	1
$\dot{q}_3$	0	0	0	0	0	0	1	**1**	1
$\ddot{q}_3$	0	0	0	0	0	0	0	0	**1**

Table 2. The influence matrix generated from the initially installed robot [30]

Gam1	C_1	$\dot{q}_1$	d_1	q_1		
Gam1	C_1	$\ddot{q}_1$	d_2	$\dot{q}_1$	d_1	q_1
Gam2	C_1	$\dot{q}_1$	d_1	q_1		
Gam2	C_1	$\ddot{q}_1$	d_2	$\dot{q}_1$	d_1	q_1
Gam3	C_1	$\dot{q}_1$	d_1	$\dot{q}_1$		
Gam3	C_1	$\ddot{q}_1$	d_2	$\dot{q}_1$	d_1	q_1
y1	m_1	q_1				
Gam1	C_1	$\dot{q}_1$				
Gam1	C_1	$\ddot{q}_1$	d_2	$\dot{q}_1$		
⋮						

Table 3. Calculation paths

Variable	redundancy degree
q_1	6
$\dot{q}_1$	6
$\ddot{q}_1$	6
q_2	6
$\dot{q}_2$	6
$\ddot{q}_2$	6
q_3	3
$\dot{q}_3$	3
$\ddot{q}_3$	1

Table 4. Redundancy degrees

The Figure 3 represents all possible paths to get the variables of the system and a tablecontaining their redundancy degrees before placement.

The cycles are shown in Figure 6, which lead to the residuals:

$R1(\Gamma_1 y_1) = 0$

$R2(\Gamma_1 y_1) = 0$

$R3(\Gamma_1 y_2) = 0$

$R4(\Gamma_1, y_2) = 0$

$R5\ (\Gamma_1, y_3) = 0$

$R6\ (\Gamma_2, y_1) = 0$

$R7(\Gamma_2, y_1) = 0$

$R8\ (\Gamma_2, y_2) = 0$

$R9\ (\Gamma_2, y_2) = 0$

$R10(\Gamma_2, y_3) = 0$

$R11\ (\Gamma_3, y_1) = 0$

$R12\ (\Gamma_3, y_1) = 0$

$R13\ (\Gamma_3, y_2) = 0$

$R14\ (\Gamma_3, y_2) = 0$

	Γ1	Γ2	Γ3	y1	y2	y3
R1	1	0	0	1	0	0
R2	1	0	0	1	0	0
R3	1	0	0	0	1	0
R4	1	0	0	0	1	0
R5	1	0	0	0	0	1
R6	0	1	0	1	0	0
R7	0	1	0	1	0	0
R8	0	1	0	0	1	0
R9	0	1	0	0	1	0
R10	0	1	0	0	0	1
R11	0	0	1	1	0	0
R12	0	0	1	1	0	0
R13	0	0	1	0	1	0
R14	0	0	1	0	1	0

Table 5. Signature table

If the variables to be monitored are the input output set, the signature table obtained from the system is shown in Table 5.

So, this system is structurally monitorable, all the faults that may effect the input or output variables are detectable and isolable.

Gam1	C1	qp1	d1	q1	m1	y1				
Gam1	C1	qpp1	d2	qp1	d1	q1	m1	y1		
Gam1	C2	qp2	d3	q2	m2	y2				
Gam1	C2	qpp2	d4	qp2	d3	q2	m2	y2		
Gam1	C3	qpp3	d6	qp3	d5	q3	m3	y3		
Gam2	C1	qp1	d1	q1	m1	y1				
Gam2	C1	qpp1	d2	qp1	d1	q1	m1	y1		
Gam2	C2	qp2	d3	q2	m2	y2				
Gam2	C2	qpp2	d4	qp2	d3	q2	m2	y2		
Gam2	C3	qpp3	d6	qp3	d5	q3	m3	y3		
Gam3	C1	qp1	d1	q1	m1	y1				
Gam3	C1	qpp1	d2	qp1	d1	q1	m1	y1		
Gam3	C2	qp2	d3	q2	m2	y2				
Gam3	C2	qpp2	d4	qp2	d3	q2	m2	y2		

Figure 9. Residual cycles

Variable	redundancy degree
q_1	8
$\dot{q}_1$	8
$\ddot{q}_1$	7
q_2	7
$\dot{q}_2$	6
$\ddot{q}_2$	7
q_3	4
$\dot{q}_3$	3
$\ddot{q}_3$	4

Variable	Cost
q_1	5
$\dot{q}_1$	6
$\ddot{q}_1$	4
q_2	6
$\dot{q}_2$	9
$\ddot{q}_2$	10
q_3	11
$\dot{q}_3$	9
$\ddot{q}_3$	5

Table 6. Specifications degrees and costs of sensors

Solving the optimal sensor placement problem requires a cost to be associated to each candidate sensor. Consider sensor costs in Table 6; costs are dimensionless and have been assigned according to the ease of installation and the price of their corresponding sensors. The six optimal solution as presented in Figure 10 which represents the best solutions and their corresponding global cost with excess of redundancy degree for each placement.

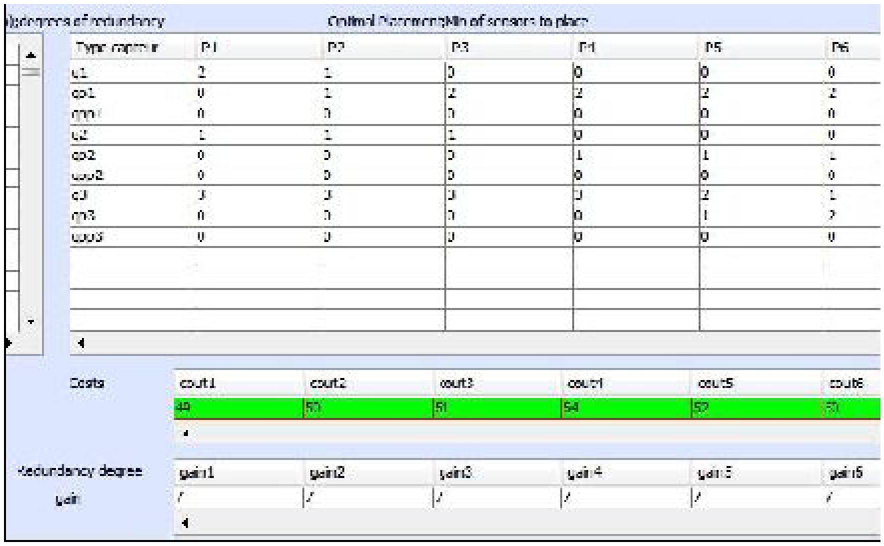

Figure 10. Results of placement

7.1 Analysis of the Results

With these tests, we have solved the sensor problem optimally:

• We have generated all information present in the system initially installed; and this is thank to the generating cycles algorithm and calculate paths [30], this will help us to do not lose any information.

• Calculate new objectives specifications with calculating difference between specifications and redundancy degrees generated from system.

• We have created a threshold *m* to preserve solution fiability.

• We have chosen the tow best sensors to be installed in the placements from the influence Matrix witch gives maximum to the objectives for installing just one sensor.

• Finally, and thanks to influence matrix, the solution gives directly the desired redundancy degrees of the variables with a gain.

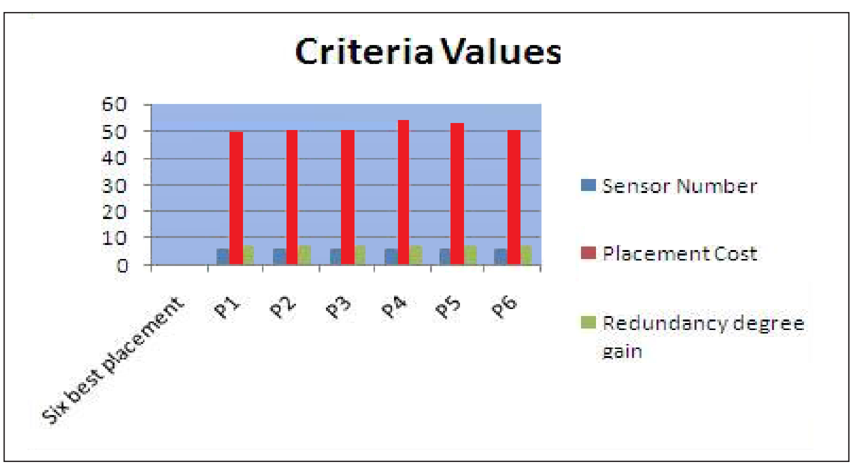

Figure 11. Graphical representation of the *m* Pareto solutions

8. Conclusion

A key issue for practical diagnosis in industry is the trade-off between installing the minimal sensors and getting a high degree of fault isolation and diagnosis. Under economic constraints, industrial systems are typically configured with the minimum set of sensors needed for control and protection. In this work, and in multi-criterion optimization context, a new tripartite graph based methodology to solve the sensor placement problem for FDI has been addressed and applied to the Dynamic Model of the Robot. The sensor placement problem has been discussed and a formal model has been presented, leading to optimal algorithm, which was implemented in Java program. This method could also be applied to other kind of systems if given their structural model. A key contribution of this work is the definition of the redundancy degrees index of initial system and generating influence matrix witch covers all information of the system. This two tools allows us to set up a different best sensor placement pareto solutions based on a fault diagnosis performance maximization criteria. In brief, the results are very encouraging concerning the utility of influence matrix to avoid exhaustive research and found the solutions in minimum time. Future research work will include other criteria optimization and comparison study with semi formal and meta heuristic approaches on the same benchmark.

References

[1] Maquin, D., al. (1994). Sureté de fonctionnement et redondance analytique, 1er congrés Pluridisciplinaire Qualité et sureté de fonctionnement, Compiègne France, 1994.

[2] Ould Bouamama, B., Haffaf, H. (2008). enumerating cycles algorithm in tripartite graph for FDI, *Journal of the Franklin Institute*, 349, Issue 1, p 112_125, March 2008.

[3] Sinha, U. V. S., Rajaram, R. (2016). Operator theoretic framework for optimal placement of sensors and actuators for control of nonequilibrium dynamics, *Journal of Mathematical Analysis and Applications*, 440, p 750_772, August 15, 2016.

[4] Ambrosio, F. R. P., Ripamonti, F. (2013). A sensor and actuator placement technique for active vibration control using h2 norm, 978-1-4673-1388-9/13/IEEE, 2013.

[5] Suryanarayanan, M. T. S. (2007). Appropriate sensor placement for fault-tolerant lanekeeping control of automated vehicles, *IEEE/ASME Transactions on mechatronics*, Vol. 12, pp. 465_471, August.

[6] Mosleh, A., Pourali, M. (2013). A functional sensor placement optimization method for power systems health monitoring, *IEEE Transactions on industry applications,* 49, 4, p 1711_1719, July/August, 2013.

[7] Tod, M. D., Flynn, E. B. (2010). A bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing, In Elseiver. *Mechanical Systems and Signal Processing*, 24, p 891_903, 2010.

[8] Saitta, S., Kripakaran, P., et al. (2007). Optimal sensor placement for damage detection: Role of global search", 18th *International Workshop on Database and Expert Systems Applications*, IEEE, pp.302_306, 2007.

[9] Burrows, A. P., Worden, K. (2001). Optimal sensor placement for fault detection, Engineering Structures 23, Elsevier Science Ltd, pp. 885_901, 2001.

[10] Chi, G., et al. (2012). Sensor placement for fault diagnosis using genetic algorithm, *IEEE*, p 4673-473, 2, 2012.

[11] Minjie, D., et al. (2011). ARRs based sensor placement optimization for fault diagnosis, Procedia Engineering, Published by Elsevier Ltd, International Workshop on Automobile, *Power and Energy Engineering*, p 42_47, 2011.

[12] Hedjazi, L., *et al*. (2010). Sensor placement and fault detection using an efficient fuzzy feature selection approach, 49th *IEEE Conference on Decision and Contro*l, Atlanta, GA, USA, p 6827_6832, December 15-17, 2010.

[13] Rahim, M. A., Akram, M., Khalid, H. M. (2013). Sensor location optimization for fault diagnosis with a comparison to linear programming approaches, Int J Adv Manuf Technol 65, Springer-Verlag London, p 1055_1065, 2013.

[14] Yang, Q., Aitouche, A., Ould Bouamama, B.(2010). Model based fault detection and isolation of pem fuel cell, Conference on Control and Fault Tolerant Systems Nice, *France IEEE*, p 825-830, October 6-8 2010.

[15] Frisk, E., Krysander, M., Aslund, J. (2009). Sensor placement for fault isolation in linear differential-algebraic systems, Automatica 45, Elsevier Ltd, p 364_371, 2009.

[16] Sarrate, R., Puig, V., Escobet, T., Rosich, A. (2007). "Optimal sensor placement for modelbased fault detection and isolation, In Proceedings of the 46th IEEE Conference on Decision and Control New Orleans, LA, USA, IEEE, pp. 2584_2589, Dec. 12-14.

[17] Chen, Q., Ahmed, Q., et al. (2015). Model-based fault diagnosis of an automated manual transmission shifting actuator, Elsevier, 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes Safeprocess 2015 Paris, vol. 48, pp. 1479_1484, 2015.

[18] Yassine, A., Ploix, S., Flaus, J. (2008). A method for sensor placement taking into account diagnosability criteria, *Int. J. Appl. Math. Comput. Sci*., 18 (4), pp. 497_512, 2008.

[19] Ahmed, Q., Arasu, M., Zhang, J., Riz zoni, G. (2016). Sensors installation guide to monitor automatic transmission performance, IFAC(International Federation of Automatic Control) Elsevier, 64, p 736_741, 2016.

[20] Benallel, M., Haffaf, H., Meghebbar, A. (2016). "Sensor placement for monitoring systems modelled by bond graph, Electrotechnica, *Electronica, Automatica*, 64, p105_116, 2016.

[21] Djeziri, M. A., et al. (2009). Optimal sensor placement for fault diagnosis, *In*: Proceedings of the IEEE international Conference on Mechatronics, Malaga, Spain, April 2009.

[22] Khemliche, M., Ould Bouamama, B., Haffaf, H. (2006). Sensor placement for component diagnosability using bond-graph, Sensors and Actuators, elseiver science direct, p 547_556, 2006.

[23] Latreiche, S., Mostefai, M., Khemliche, M. (2014). Optimal sensors placement for monitoring a steam condenser of the distillation column using bond graph approach, *Leonardo Journal of Sciences* ISSN 1583-0233, p 97_110, (July-December 2014).

[24] Salem, S., Benazzouz, D. (2013). Optimal sensor placement for fault detection and isolation by the structural adjacency matrix, *International Journal of the Physical Sciences*, 8(6), p 225_230, 16, February 2013.

[25] Commault, C., Dion, J-M. (2003).Sensor location for diagnosis in linear structured systems with disturbances, *In*: Proceedings of the 42nd IEEE, Conference on Derision and Control Mae, Hawaii USA, December 2003.

[26] Commault, C., Dion, J. (2007). Sensor location for diagnosis in linear systems: A structural analysis, *IEEE TransactionsRon Automatic Control* 52, (2), p 155_169, February 2007.

[27] Krysander, M., Frisk, E. (2008). Sensor placement for fault diagnosis, *IEEE Transactions on systems*, Man and cybernetics_PART A: Systems and humans, vol. 38, NO. 6, pp. 1398_1410, November 2008.

[28] Rosich, A., Yassine, A., Ploix, S. (2010). Efficient optimal sensor placement for structural model based diagnosis, 21st International Workshop on Principles of Diagnosis, 2010.

[29] Yassine, A., Rosich, A., Ploix, S. (2010). An optimal sensor placement algorithm taking into account diagnosability specifications, 2010.

[30] Abid, S., Haffaf, H. (2012). Optimal sensor placement for diagnosis: Industrial process of glue preparation station, In: CITIM,Algeria, 2012.

[31] Torta, G., Torasso, P. (2011). A structural approach to sensor placement based on symbolic compilation of the model, 2011.

[32] Papadimitriou, C., Steiglitz, K. (1998). Combinatorial optimization. algorithms and complexity, Dover Publications, Inc., New York, 1998.

[33] Ehrgott, M., Gandibleux, X. Hybrid metaheuristics for multiobjective combinatorial optimization.

[34] Ehrgott, M. (2005). Multicriteria optimization, 2ndedn. Springer, Berlin, 2005.

[35] Miettinen, K.M. (1999). Nonlinear multiobjective optimization, Kluwer Academic Publishers, Boston, 1999.

[36] Coello Coello, C.A., Lamont, G. B., Van Veldhuizen, D.A. (2007). Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York.

[37] Deb. K. (2001). Multi-objective optimization using evolutionary algorithms, John Wiley and Sons.Chichester.

[38] Carlos, A., et al. (2010). Advances in multi-objective nature inspired computing, Springer Verlag Berlin Heidelberg, pp. 1_21,

2010.A.

[39] Edgeworth, E. (1981). Mathematical psychics. p. keagan, London, 1981.

[40] Yassine, A., et al. (2010). An optimal sensor placement algorithm taking into account diagnosability specifications, Cluj-Napoca, Romania IEEE, doi:978-1-4244-6724-2, 2010.

[41] Gondran, M., Minoux, M., (1995). Graphs and algorithms, Edition Eyrolles, 1995.

[42] Dustegor, D. (2010). Automated graph-based methodology for fault detection and location in power systems, *IEEE Transactions on power delivery*, 25 (2), p 638_646, april 2010.

[43] Abid, S., Haffaf, H. (2013). A tripartite graph approach for optimizing sensor placement problem, 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013, Sardinia, Italy, IEEE 2013 ISBN 978-1-4673-2479-3, pp. 108 _ 113, July 1-5, 2013.

[44] Torta, G., Torasso, P.A. (2011). Structural approach to sensor placement based on symbolic compilation of the model, Springer Varlag London, p 251_264 doi:10.1007/978-0-85729-130-1_18, 2011.

Author Biographies

Sabrina Abid was born in Mascara, Algeria in 1984. She received a M.S. degree in 2010, from Mascara University, Algeria. Currently she has a position as an assistant professor at the Department of Computer science at Mascara University, and she is doctoral student in 6th year. Current research interests include structural analysis for diagnosis, sensor placement optimization, and Multicriteria optimization.

Correspondence address: Department of Computer science, Laboratory of Research of Industrial Computing and Network, University Oran1 Ahmed BenBella, Bp 1524 El M'Naouer Oran,Algeria, e-mail: abidsabrina2006@yahoo.fr

Hafid Haffaf is a full Professor and senior lecturer at the University of Oran (Algeria) after Obtaining Doctor degree in computer Science in 2000. He also heads the " Research in Industrial computer science and Networks" Laboratory at Computer science department. His researchers concern different domain as Automatic control, information system Design, Sensor networks, ..but, the more important works are about optimization algorithms, especially those based on graph theory and matroid theory as well as their application in Bond graph and monitoring.

He has many collaborations projects with European laboratory: Polytech Lille in the domain of Engineering and Pau (France) in the domain of Wireless sensor Networks (CMEP Tassili project).

Correspondance adress: Department of Computer science, Laboratory of Research of Industrial Computing and Network, University Oran1 Ahmed BenBella, Bp 1524 El M'Naouer Oran,Algeria, e-mail: haffaf.hafid@univ-oran.dz