Sensor Placement Optimisation Using Genetic Algorithms

Stefan Spanache[1] and **Teresa Escobet** [2] and **Louise Travé-Massuyès**[3]

Abstract. The efficiency of a diagnosis system depends on the relevance of the information it can retrieve from the diagnosed plant; or, in other words, the efficiency of a sensor system can be measured by the diagnosability degree it provides. However, the same diagnosability level may be obtained for different sensor configurations. Moreover, in some cases, less than the highest level of diagnosability might be sufficient. Thus, appears the necessity to take into account the economic issue. Following the work of [11], this paper proposes a method to design a cost optimal sensor system for a certain diagnosability degree using an evolutionary approach.

1 INTRODUCTION

The efficiency of a diagnosis system depends on the information it can retrieve from the diagnosed plant. Obviously, if the information is insufficient, the diagnosis system is not able to perform its task. However, an increased number of sensors alone does not guarantee that the diagnosis system will have a better performance. The relevance of the information brought by an additional sensor must also be taken into account. In some cases, knowing the values of a given variable brings no information from the diagnosis point of view. Thus, the efficiency of a sensor system can be measured by the diagnosability degree it provides. However, the same diagnosability level may be obtained for different sensor configurations. Moreover, in some cases, the highest level of diagnosability might not be necessary.

Therefore, economical issues may come into place. When designing a sensor system, one must search for those combinations of sensors that can provide a specified diagnosability level at lowest possible cost. Following the work of [11], this paper proposes a method to design a cost optimal sensor system for a certain diagnosability degree using an evolutionary approach. Different combinations of sensors are codified in the chromosomes of a first population, and then a genetic algorithm searches for the most advantageous ones in terms of diagnosability degree over cost ratio. A practical example illustrates the method.

The paper is structured as follows: section 2 summarises an already existing method to perform a diagnosability analysis of a given system, introduces the objectives and formulates the problem, section 3 presents the genetic algorithm approach to the problem, section 4 gives an application example and section 5 outlines a few conclusions and further directions of work.

[1, 2] Dept. of ESAII, Universitat Politècnica de Catalunya, Barcelona, Spain, e-mail: (stefan.spanache, teresa.escobet)@upc.es

[3] LAAS-CNRS, Toulouse, France, louise@laas.fr

2 SENSOR SETS AND DIAGNOSABILITY

2.1 Analytical redundancy

The behaviour of a physical system – or its components – can be described by a set of constraints applied to known and unknown variables. The constraints are in fact limitations imposed to the evolution of these variables by physical laws or other restrictions. In a structural analysis approach, the model of the system can be represented as a bipartite graph $G = (E \cup V, A)$, where E is the set of constraints relating the variables in V and A is the set of arches such that $a(i,j) \in A$ iff variable v_i is involved in relation e_j [10].

Considering a system $\Sigma(E, V)$, where E is the set of constraints - also called *primary relations (PRs)* - and V is the set of variables, V can be partitioned as $V=O \cup U$, where O is the set of observed (measured) variables and U the set of unknown variables. Then Σ is said to be:

- *under-determined* if CARD(E) < CARD(U);
- *just-determined* if CARD(E) = CARD(U) and
- *over-determined* if CARD(E) > CARD(U).

If the system is over-determined, a perfect matching $\boldsymbol{M}(E_m \subset E\ ,\ U)$ can be found between the variables in U and the relations in E. A matching $\boldsymbol{M}(E_m\ ,\ U)$ is said to be *perfect* when the bipartite graph $G_m = (E_m \cup U, A)$ does not have two arches connected by a common relation or a common variable and the system (E_m, U) is analytically resolvable, e.g. some of the relations in E_m might not be invertible. A method to obtain such a matching is proposed in [3].

The remaining relations, those not involved in the perfect matching are called Redundant Relations (RRs)[3]. Every RR produces an Analytical Redundant Relation (ARR) when the unknown variables involved in the RR are replaced by their formal expression determined analytically by resolving the perfect matching up to observed variables. An ARR hence arises from a causal interpretation of the underlying model primary relations. It only contains observed variables and can be evaluated from the observations.

In a component-oriented model, the primary relations are matched to the system physical components whose behaviour they describe. The set of components whose corresponding primary relations underlie a given ARR is called the support of the ARR and noted *Supp*(ARR) [6]. In the general case, *Supp*(ARR) also includes the set of sensors used to measure the variables involved in the ARR and can be splited in a component support *Comp_Supp*(ARR) and a sensor support *Sens_Supp*(ARR).

The ARRs obtained from a given perfect matching are called *primary* ARRs. Additional *combined* ARRs can be obtained by substituting the expression derived from one ARR for a variable in

another ARR. The combined ARRs inherit the component supports of their ascendant ARRs, while the sensor support is constructed from the sensors used to measure the variables involved in the newly obtained ARR [6].

However, there is one issue that concerns the conditions under which a variable substitution can be actually performed. A given relation (primary or not) can be interpreted in a causal way, i.e. for determining one or the other of the involved variables. The possible causal interpretations may be submitted to validity conditions depending on the corresponding mathematic analytical form. For example, the relation $x=y\times z$ has two additional possible interpretations : $y=x/z$ under validity condition $z\neq 0$, and $z=x/y$ under validity condition $y\neq 0$. Every relation hence gives rise to a set of causal-relations with their associated validity conditions. Each final ARR will also inherit the evantual validity conditions of its ascendant relations.

2.2 Component-oriented Fault Signature (FS) matrix

In the FDI terminology, the fault signature (FS) matrix crosses ARRs in rows and (sets of) faults in columns [5]. Let us assume that F_j denotes a fault on component C_j, then in this matrix, the interpretation of some entry s_{ij} being 0 is that component C_j does not belong to $Supp(ARR_i)$, i.e. the occurrence of the fault F_j does not affect ARR_i, meaning that ARR_i is satisfied in the presence of that fault. s_{ij} = 1 means that C_j belongs to $Supp(ARR_i)$, i.e. if detected, fault F_j will affect ARR_i. The columns of such a matrix are the fault signatures. Two faults with identical signatures are said to be non-discriminable.

Example 1:

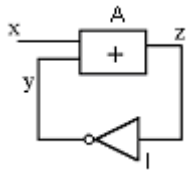

Figure 1. A simple example

Let's consider the system in figure 1 containing an adder (A) and an inverter (I): Σ, with $V = \{x, y, z\}$. $S(i)$ will denote the sensor measuring the values of variable i. The set E of primary relations is:

PR_1: $z = x + y$, describing the behaviour of component A
PR_2: $y = -z$, describing the behaviour of component I

In the case that $O = \{x\}$ and $U = \{y, z\}$ Σ is just-determined. No ARRs can be found since no redundancy exists. Let's consider now that $O = \{x, z\}$ and $U = \{y\}$. Σ becomes over-determined and an ARR appears: ARR_3: $x = 2z$ with $Comp\_Supp = \{A, I\}$ and $Sens\_Supp = \{S(x), S(z)\}$. ARR_3 inherits the component supports of its ascendant relations. The FS matrix in this case is:

Table 1. The FS matrix with y unmeasured.

ARR	Comp_Supp		Sens_Supp		
	A	I	S(x)	S(y)	S(z)
ARR3	×	×	×		×

With this system instrumentation (adder, inverter and the two sensors $S(x)$ and $S(z)$) at an eventual detection of a fault it is impossible to discriminate between the four component faults. Now, by adding a new sensor for y all the variables are measured - $O = \{x, y, z\}$ and $U = \varnothing$ - so the primary relations will become actually ARRs themselves. The FS matrix in this case is:

Table 2. The FS matrix with all variables measured.

ARR	Comp_Supp		Sens_Supp		
	A	I	S(x)	S(y)	S(z)
ARR1	×		×	×	×
ARR2		×		×	×

In this system instrumentation the sets that can be discriminated are $\{A, S(x)\}$, I and $\{S(y), S(z)\}$.

It is easy to see that the discriminability level of the system grows with the number of ARRs. Since an ARR contains exclusively measured variables, increasing the number of ARRs is achieved by increasing the number of sensors. Obviously, the maximum discriminability level is obtained when all the variables in the set V are measured.

2.3 Optimal sensor sets for diagnosability

A definition of the concept of diagnosability can be found in [4]: "A system is diagnosable with a given set of sensors S if and only if (i) for any relevant combination of sensor readings there is only one minimal diagnosis candidate and (ii) all faults of the system belong to a candidate diagnosis for some sensor readings".

The first condition ensures that there are no two identical columns in the system's FS matrix and the second that the FS matrix doesn't contains void columns, case in which the system is fully diagnosable. However, in practice, this situation is seldom encountered. Most physical systems are only partially diagnosable as several faults share the same signature. Given a system Σ and a set of faults F, a subset $F' \subset F$ of non-discriminable faults is called a *D-class* [11]. The number of *D-classes* of a fully diagnosable system is equal to the number of faults, CARD(F). The faults are assumed to be detectable.

The method is based on assuming that all the unknown (measurable) variables have a hypothetical sensor and producing the corresponding hypothetical ARRs (H-ARRs). The corresponding FS matrix is called Hypothetical Fault Signature (HFS) Matrix [11]. This matrix has several fields, each corresponding to H-ARR attributes – component and sensor supports, causal interpretations, validity condition, etc. – that are traced along the combinations. It hence summarises all the required information to perform a complete diagnosability assessment. The HFS matrix can easily be obtained from the H-ARRs matrix by removing all fields but the component support and the sensor support fields, complemented by the corresponding validity condition field. Let us note with $\boldsymbol{S}^*$ the set of sensors associated to all measurable variables in a given system Σ. Then $\boldsymbol{S}^*$ can be written as $\boldsymbol{S}^* = \boldsymbol{S_a} \cup \boldsymbol{S_h}$, where $\mathbf{S_a}$ is the set of already available sensors and $\boldsymbol{S_h}$ is the set of hypothetical sensors. If the method is implemented in the phase of design $\boldsymbol{S_a} = \varnothing$.

Definition 1. (Alternative Fault Signature Matrices). Given a system Σ with a set of sensors $S \subseteq \boldsymbol{S}^*$, its FS matrix is obtained from the HFS matrix by removing all the columns corresponding to hypothetical sensors not included in S and all the H-ARRs whose sensor supports do not intersect S. The Alternative Fault Signature (AFS) Matrices are given by all the FS matrices corresponding to all the possible sensor sets $S \subseteq P(\boldsymbol{S}^*)$, where $P(\boldsymbol{S}^*)$ are the parts of $\boldsymbol{S}^*$.

Obviously, the most suitable combination of hypothetical sensors with respect to fault diagnosis is the one whose AFS matrix provides the same number of *D-classes* as the HFS matrix and, in the same time, is of minimal cost. However, when evaluating the

number of *D-classes*, one must not allow to a hypothetical sensor to alter this number. A combination of hypothetical sensors with different fault signatures may generate an AFS matrix with a fairly large number of *D-classes* without improving the discriminability between the system components. So, the number of *D-classes* given by an AFS matrix must be evaluated without considering the columns corresponding to hypothetical sensors. On the other hand, each newly added sensor will introduce a new possible fault in the system: itself. This can be penalised by taking into account the reliability specifications of each sensor when evaluating the cost of a hypothetical sensor set.

Considering the above, we propose the following formula for evaluating the efficiency – with respect to fault diagnosis - of a set $S = \{S_1, ..., S_n\} \subset \boldsymbol{S_h}$ of hypothetical sensors:

$$\varepsilon = \frac{D}{\sum_{i=1}^{n} C_i} \qquad (1)$$

where D is the number of *D-classes* evaluated as already explained and C_i, $i = 1, ..., n$, are the costs of the sensors in the set S. The costs are evaluated as $C_i = P_i + I_i - R_i$, where P_i is the price of the sensor, I_i measures the ease of installation and/or replacement of the sensor and R_i is its reliability specified by the provider.

So, finding the optimal sensor configuration w.r.t. fault diagnosis means to set D at D_{max} and to look for the most cost effective hypothetical sensor sets (i.e. to maximise ε), where D_{max} is the number of *D-classes* given by the HFS matrix.

3 GENETIC ALGORITHM

In order to find the most suitable combination of additional sensors that provides a system with the maximum level of diagnosability one must compare the discriminability level (number of *D-classes*) and the cost effectivenes of 2^n AFS matrices, where n is the number of hypothetical sensors. In the case of complex systems, with a large number of measurable variables this can be time consuming. On the other hand, finding the most efficient sensor set w.r.t. fault diagnosis can actually be formulated as an optimisation problem with ε as objective function. So path-oriented global optimization methods present themselves as a faster alternative to an exhaustive search. However, the same (optimal) value of ε may result from different combinations of hypothetical sensors. The genetic algorithms have the capacity to evolve to a final population that can store different individuals with the same fitness value, namely the optimal solution to the problem.

Such and algorithm starts from an initial set of candidate individuals called the initial population and, using genetic operators – *crossover, mutation, selection* - which try to mimic natural selection laws, simulates the biological evolution producing new populations with better individuals at each iterative step. After a number of iterations, which depends on the complexity of the problem, the algorithm finds the optimal solution to the problem as the best fit individual [8].

Every individual is uniquely identified by a code called chromosome which is mapped to a certain value of the objective function (the function to optimise) representing the fitness of the individual. With the objective to improve the fitness of the individuals at each step, the algorithm uses the genetic operators to generate new individuals and to select the fittest ones. The *crossover* operator combines the information of two different chromosomes to generate a new individual, whilst the *mutation* operator generates new individuals by randomly altering the information in one chromosome. The *selection* operator chooses the individuals which survive to the next generation [7][1].

The following picture illustrates the main steps of a simple genetic algorithm:

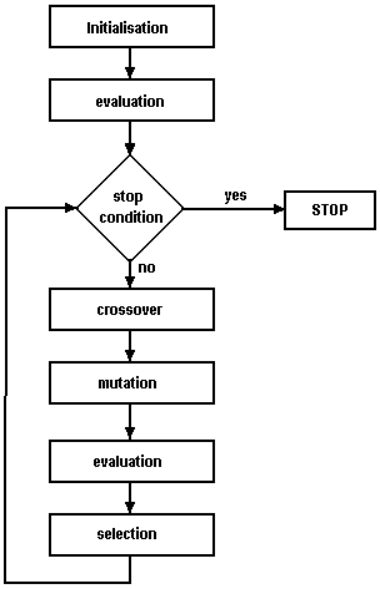

Figure 2. Basic genetic algorithm

In our case, the individuals of the initial population are binary codified hypothetical sensor sets: the optimal sensor combination candidates. The mapping between the candidates and the objective function ε is done through the AFS matrix generated by each candidate. For example, let's consider that a given system Σ has the following set of unmeasured variables: $U = \{x, y, z\}$. Then the chromosome [1 0 1] will be the code for the set $\boldsymbol{S_h} = [1\ 0\ 1] \times [S(x)\ S(y)\ S(z)]^T = \{S(x), S(z)\}$ of hypothetical sensors. The fitness of this chromosome is evaluated as $\varepsilon = D/(C_x + C_z)$, where D is the number of *D-classes* given by the corresponding AFS matrix constructed as indicated in definition 1.

Example 2:

HFS matrix	11010	*AFS*	**11010**	1101**0**
	10110	*matrices*	1010**0**	1011**0**
	01110		**01110**	0111**0**
	10111		1010**1**	**10111**
	10101		1110**1**	**10101**
Chromosomes	111		101	110

The first two columns correspond to the component support and the last three to the sensor support. The digits marked in bold characters are eliminated from the HFS matrix, constructing the corresponding AFS matrices.

The crossover and mutation operators are implemented in the simple way of [8]. To illustrate the behaviour of the crossover operator let's note the male chromosome as $m = [m_1, ... ,m_i, ... ,m_s]$, the female chromosome as $f = [f_1, ... ,f_i, ... ,f_s]$, where $s = \text{CARD}(\boldsymbol{S_h})$. *Crossover* randomly selects two individuals of the current population, then randomly generates a crossover point (*cp*) in the interval $[1, s]$. Then it mixes the genetic information between the two, generating two offsprings as follows: $child_1 = [m_1,.... , m_{cp}, f_{cp+1}, ... , f_s]$ and $child_2 = [f_1,.... , f_{cp}, m_{cp+1}, ... , m_s]$. This mechanism is called *one point crossover*. The *mutation* operator also randomly generates a mutating point (*mp*) in the interval $[1, s]$ and switches the binary value of the *mp* bit of the original chromosome.

The selection operator uses the *q-tournament* mechanism. There are several selection mechanisms, a complete description of which

can be found in [7]. Tournament selection randomly chooses q individuals from the current population, evaluates them, and passes the fittest one to the next generation. Here $q = 2$ was used, the so-called *binary tournament*.

The main steps of the algorithm are given below:

Step 1 (optional): in case of very large HFS matrices, the algorithm starts with a refining step consisting in evaluating one by one all hypothetical sensors and eliminating the ones that do not increase the diagnosability degree given by the already available sensors. Let us note with s the number of remaining sensors.

Step 2: the initial population is generated by constructing the set of $C_s^{\mathrm{int}(s/2)}$ from the remaining sensors. This results in the maximum number of possible combinations, which will guarantee a more robust search. Example: if s=7 than the initial population is formed of all possible combinations of binary vectors with 7 elements, 3 of which are 1 and the rest of 4 are 0. However, if the number of hypothetical sensors is very large, for computational reasons, the population should be truncated to a number between 50 to 200 individuals [1].

Step 3: running the algorithm.

- **Step 3.1.** *evaluation*: the fitness values of all individuals in the population are calculated.
- **Step 3.2.** *elitism*: the best fit individuals are just passed to the next generation.
- **Step 3.3.** *crossover*: from the rest of the population two individuals are randomly selected and their genetic information is mixed as described above, generating new individuals. The operation is repeated until there are no more individuals left.
- **Step 3.4.** *mutation*: one of each two offsprings has a gene switched.
- **Step 3.5.** *selection (binary tournament)*: two individuals are randomly selected and the fittest one is passed to the next generation. The operation is repeated as many times is necessary to complete the new generation.
- **Step 3.6.** Replace the old generation with the new one and go to Step 3.1.

Stop condition: when the difference between the most fit and less fit value of the current population is smaller or equal to a value set according to the application.

4 APPLICATION TO DAMADICS BENCHMARK

The benchmark actuator selected is a final control element or simply named actuator, which interacts with the controlled process. These actuators are used in the evaporation station of a sugar factory in Poland [2][12]. The set-point of the position actuator is the output of the process controller (flow or level controller) and the actuator modifies the position of the valve allowing a direct effect on the primary variable in order to follow the flow or pressure set point. In this example, it is used to control the flow on the valve outlet (F).

The actuator consists in three main components (figure 3):

- control valve or hydraulic (H)
- pneumatic servo-motor or mechanics (M)
- positioner, which in turn can be divided in three sub-components:
  - position controller (PC)
  - electro/pneumatic transducer (E/P) with pressure supplier (PSP)
  - displacement transducer – or positioner feedback – (DT)

Control valve is the mean used to prevent, allow and/or limit the flow of fluids through control systems. Changing the state of the control valve is accomplished by a servomotor.

A pneumatic servomotor can be defined as a compressible (air) fluid powered device in which the fluid acts upon the flexible diaphragm, to provide linear motion of the servomotor stem.

Positioner is a device applied to eliminate the control-valve-stem miss-positions produced by the external or internal sources such as friction, pressure unbalance, hydrodynamic forces etc. It consists in an inner loop with a P controller of a cascade control structure, including the output signal of the outer loop of the flow or level controller and the inner loop of the position controller.

- Additional external components:
  - V1, V2 - cut-off valves
  - V3 - by-pass valve
  - PT - pressure transmitters
  - FT - volume flow rate transmitter
  - TT - temperature transmitter

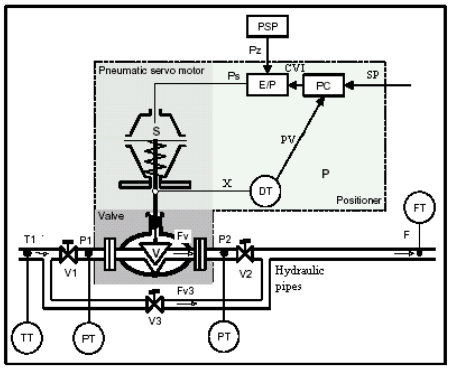

Figure3. Actuator scheme

The set V of system variables is:

- X - servomotor's rod displacement
- PV - process variable
- F_v - volumetric flow on the valve outlet
- P_s - the pressure in the servomotor's chamber
- P_z - the supply pressure (600 MPa)
- SP - the set point
- CVI - the control current (controller's output)
- ΔP - pressure difference across the valve ($P_1 - P_2$)

The following table summaries the equation for each component:

Table 3. The component-oriented primary relations of the process

Component	Equation	Invertible	Validity cond.
Pneumatic servomotor	$X = r_1(P_s, \Delta P)$	Not	none
Control valve	$Fv = r_2(X, \Delta P)$	in respect of X	$\Delta P \neq 0$
Position controller	$CVI = r_3(SP, PV)$	in respect of PV	none
E/P transducer + pressure supplier	$P_s = r_4(X, CVI, P_z)$	Not	none
Positioner feedback	$PV = r_5(X)$	in respect of X	none

Following the ideas presented in the section 3 the goal is to design an instrumentation system that will optimise the diagnosability level of the system. The components that can be faulty are: {M, P, H, DT} and the hypothetical sensors S_h = {S(x), S(Ps), S(Fv), S(CVI), S(PV), S(dP), S(Pz)}. Ideally, the discriminability level of the system should be 4 as the number of possible faulty components.

Considering all the variables in V measured, the number of H-ARRs is 44. This simple example was chosen in order to compare the results obtained with the genetic algorithm with those obtained with an exhaustive search. With an initial population of 35 individuals calculated with the formula given in Step 2 in the previous section the genetic algorithm converges to the same solution as the one obtained with the exhaustive search in 5, 6 iterations. This means it has to evaluate 175 or 210 combinations of sensors, a value bigger than 2^7 which is the number of sensor combinations evaluated in an exaustive search. However, if the initial population is truncated to 15 the same result is obtained in the same number of generations, sensible faster than with an exhaustive search. It seems that in cases with larger HFS matrixes the discrepancy between the time needed by the genetic algorithm to converge and the time needed to scan all the matrix in an exhaustive search will increase.

Table 4. The optimal sensor set.

	Component-Support				Sensor-Support				
	M	P	H	DT	S(*ps*)	S(*fv*)	S(*pv*)	S(*dp*)	S(*pz*)
ARR1	x		x		x	x		x	
ARR2	x			x	x		x	x	
ARR3			x	x		x	x	x	
ARR4	x	x			x		x	x	x
ARR5		x	x		x	x	x	x	x
ARR6		x		x	x		x		x
ARR7	x	x		x	x			x	x
ARR8	x	x	x			x	x	x	x
ARR9	x	x	x	x		x		x	x
ARR10		x	x	x	x	x		x	x
ARR11	x	x		x			x	x	x
ARR12	x	x	x	x	x	x		x	x

5 CONCLUSIONS AND FUTURE WORK

The analytical redundancies existing in an industrial system provide means to discriminate between its functional components in the eventuality of a fault. The number of such redundancies depends of the availability of measured variables. Thus, increasing the number of sensors will increase de level of discriminability between system components. Here was proposed a method to search for an optimal (cost wise) set of sensors that can be implemented in the stage of instrumentation system design.

The genetic algorithm used encountered the optimal additional sensor set in a fairly small number of iterations showing itself as a good way to avoid an exhaustive search. For the future remains to test the approach with more complicated examples with larger HFS matrices. Another issue still under study is the possibility to incorporate in the procedure a method to take into account the eventual conflicts between the validity conditions under which the resulting ARRs can be evaluated.

ACKNOWLEDGEMENTS

The authors acknowledge the founding support of EC RTN DAMADICS (contract RTN-1999-00392), the support received from the Research Commission of the Generalitat of Catalunya (ref. 2001SGR00236) and from Spanish CICYT (ref. DPI2002-02147).

REFERENCES

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, (1996).

[2] M. Bartys, *Specification of actuators intented to use for benchmark definition*, http:// diag.mchtr.pw.edu.pl/damadics/, (2002).

[3] J.P. Cassar and M. Staroswiecki, 'A structural approach for the design of failure detection and identification systems2', *Proceedings of IFAC, IFIP, IMACS Conference on Control of Industrial Systems,* Belfort, France, 329-334, (1997).

[4] L. Console, C. Picardi and M. Ribaudo, 'Diagnosis and Diagnosability Analysis using process algebra', *Proc. DX'2000*., Morelia, Mexico, (2000).

[5] M-O. Cordier, P. Dague, F. Lévy, M. Dumas, J. Montmain , M. Staroswiecki and L. Travé-Massuyès, 'AI and automatic control approaches of model-based diagnosis: links and underlying', *Proeedings of the IFAC Workshop SAFEPROCESS*, Budapest, Hungary, (2000).

[6] M-O. Cordier, P. Dague, F. Lévy, M. Dumas, J. Montmain , M. Staroswiecki and L. Travé-Massuyès, 'Conflicts versus analytical redundancy relations – A comparative analysis of the model based diagnosis approach from the AI and Automatic Control perspective', *Submitted to IEEE Trans. On SMC*, (2002).

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, (1989).

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, (1975).

[9] M. Luong, D. Maquin, C. T. Huynh and J. Ragot, 'Observability, redundancy, reliability and integrated design of measurement system', *Proceedings of 2nd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, SICICA'94, Budapest, Hungary, (1994).

[10] M. Staroswiecki and P. Declerck, 'Analytical redundancy in non-linear interconnected systems by means of structural analysis', *IFAC Advanced Information Processing in Automatic Control* (AIPAC'89), Nancy, France, 51-55, (1989).

[11] L. Travé-Massuyès, T. Escobet and S. Spanache, 'Diagnosability analysis based on component supported analytical redundancy relations', *Proc. of the IFAC Workshop SAFEPROCESS 2003*. Washington D.C., U.S.A, (2003).

[12] P. Wasiewich, 'Specification of sugar production processes connected to the actuators intended to use for benchmark definition.', *http:// diag.mchtr.pw.edu.pl/damadics/*, (2001).