2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Fault diagnosis by refining the parameter uncertainty space of nonlinear dynamic systems

Esteban R. Gelso, Sandra M. Castillo, and Joaquim Armengol

University of Girona, E17071 Girona, Spain.

{esteban.gelso, sandra.castillo, joaquim.armengol}@udg.edu

Abstract— **This paper deals with fault detection and isolation problems for nonlinear dynamic systems. Both problems are stated as constraint satisfaction problems (CSP) and solved using consistency techniques. The main contribution is the isolation method based on consistency techniques and uncertainty space refining of interval parameters. The major advantage of this method is that the isolation speed is fast even taking into account uncertainty in parameters, measurements, and model errors. Interval calculations bring independence from the assumption of monotony considered by several approaches for fault isolation which are based on observers. An application to a well known alcoholic fermentation process model is presented.**

I. INTRODUCTION

Early and accurate fault detection and diagnosis for industrial processes can minimize downtime, increase the safety of plant operations, and reduce costs.

Different techniques have been developed in recent years that are intended to detect and diagnose faults. These techniques can be classified in different ways [1], [2]. For example, a distinction can be made between model-based techniques and techniques based on other kinds of knowledge, such as heuristic approaches, statistical approaches, learning systems, artificial neural networks, etc.

Among others, all the fault detection and isolation techniques have to face the challenge of dealing with uncertainty. This can be achieved in several ways, e.g. by statistical data processing, averaging, or using intervals.

This paper introduces a fault diagnosis approach based on a model that takes into account the uncertainties in the measured signals and in the model by means of intervals. These uncertainties are caused by, for example, non-modeled effects, electrical disturbances, model simplifications, and so on.

Several engineering problems such as system and state estimation, fault detection, robustness analysis, robust control design, risk assessment, and worst case behavior analysis, can be solved when interval uncertainties are considered.

As matters stand, some interval methods have been proposed in the context of fault detection and diagnosis, e.g. [3], [4] and [5]. A fault detection approach based on constraint propagation is proposed by Stancu et al. in [6]. In [7], the fault detection problem is solved using a tool known as IntervalPeeler, based on constraint projection algorithms (2B-consistency) to reduce interval domains of variables without bisections.

Consistency methods are used to perform results of this article. They are a combination of interval methods and constraint satisfaction techniques. Constraint satisfaction techniques implement local reasoning on constraints to remove inconsistent values from variable domains. In practice, the set of inconsistent values is computed by means of interval reasoning.

To introduce the results of this papers it is necessary to mention a method based on parameter partitioning and the monotony of an observer prediction error for fault isolation which is proposed in [8]. The method applies for fault isolation in non-linear dynamic systems and assumes that the fault is detected once it occurs, so the isolation procedure is triggered at this time. Its authors emphasize the approach speed, being quicker than other methods based on adaptive observers.

Regarding the approach proposed in [8], the main contributions of this paper are: (i) the isolation problem is based on parameters uncertainty refining instead of partitioning, (ii) the isolation problem is stated as a Constraint Satisfaction Problem (CSP) and solved by means of consistency techniques. A sliding time window is used to reduce the computational effort. And (iii) interval calculations allow the proposed approach to be independent of the assumption (about the type of nonlinear systems) that the system dynamics is a monotonous function with respect to the considered parameters.

The aim of this paper is to show the usefulness of the consistency methods to solve not only the fault detection problem, but also the isolation problem when a fault appears as a parameter deviation for non-linear dynamic systems. The method provides the estimation of the faulty parameter range, which is very useful information for the controller reconfiguration in the Fault Tolerant System (FTC).

In section II the fault detection and isolation problems are shown to be constraint satisfaction problems and the resolution of them is achieved by the solver RealPaver [9]. An alternative, which is to use an efficient combination of Hull- and Box- consistency, is explored.

The proposed approach effectiveness is illustrated by means of a well known alcoholic fermentation process presented in [10], [11], [12], [8], [13] and [14], for instance.

E. R. Gelso, S. M. Castillo, and J. Armengol are with the Institut d'Informàtica i Aplicacions (IIiA), University of Girona

In section III, the used model is described, two simulation scenarios are considered, and fault detection and isolation results are presented. Some conclusions and future work are stated in section IV.

II. Fault detection as a constraint satisfaction problem

Many engineering problems can be formulated in a logical form by means of some kind of first order predicate formulas: formulas with the logical quantifiers (universal and existential), a set of real continuous functions (equalities and inequalities), and variables ranging over real interval domains.

As defined in [15], a numerical constraint satisfaction problem is a triple $\mathcal{CSP} = (\mathcal{V}, \mathcal{D}, \mathcal{C}(x))$ defined by

1) a set of numeric variables $\mathcal{V} = \{x_1, \ldots, x_n\}$,
2) a set of domains $\mathcal{D} = \{D_1, \ldots, D_n\}$ where D_i, a set of numeric values, is the domain associated with the variable x_i,
3) a set of constraints $\mathcal{C}(x) = \{C_1(x), \ldots, C_m(x)\}$ where a constraint $C_i(x)$ is determined by a numeric relation (equation, inequality, inclusion, etc.) linking a set of variables under consideration.

The fault detection problem can be represented by a CSP similar to the one presented in [16], which deals with the problem of nonlinear state estimation. For example, consider a discrete-time nonlinear dynamic system described by:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{g}(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}, \boldsymbol{w}(k)) \\ \boldsymbol{y}(k) = \boldsymbol{h}(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}) + \boldsymbol{v}(k) \end{cases}, \quad (1)$$

where:

◇ $\boldsymbol{u}(k) \in \Re^{n_u}$, $\boldsymbol{y}(k) \in \Re^{n_y}$, and $\boldsymbol{x}(k) \in \Re^{n_x}$ are the input, output, and state vector, respectively.

◇ $\boldsymbol{w}(k) \in \Re^{n_w}$ and $\boldsymbol{v}(k) \in \Re^{n_y}$ are the perturbation and measurement noise vectors, which are unknown but bounded. The perturbation vector takes into account, for instance, unmodeled dynamics of the actual plant, unknown inputs, or an error due to the discretization procedure.

◇ $\boldsymbol{\theta} \in \Re^{n_p}$ is a vector of uncertain parameters.

The dynamic system (1) can be represented as a CSP:

$$\begin{aligned}
\mathcal{V} &= \{\boldsymbol{\theta}, \tilde{\boldsymbol{y}}(1), \ldots, \tilde{\boldsymbol{y}}(k), \hat{\boldsymbol{x}}(1), \ldots, \hat{\boldsymbol{x}}(k+1), \tilde{\boldsymbol{u}}(1), \ldots, \tilde{\boldsymbol{u}}(k) \\
&\quad \boldsymbol{w}(1), \ldots, \boldsymbol{w}(k), \boldsymbol{v}(1), \ldots, \boldsymbol{v}(k)\} \\
\mathcal{D} &= \{\boldsymbol{\Theta}, \tilde{\boldsymbol{Y}}(1), \ldots, \tilde{\boldsymbol{Y}}(k), \hat{\boldsymbol{X}}(1), \ldots, \hat{\boldsymbol{X}}(k+1), \tilde{\boldsymbol{U}}(1), \ldots, \tilde{\boldsymbol{U}}(k) \\
&\quad \boldsymbol{W}(1), \ldots, \boldsymbol{W}(k), \boldsymbol{V}(1), \ldots, \boldsymbol{V}(k)\} \\
\mathcal{C} &= \{\hat{\boldsymbol{x}}(2) = \boldsymbol{g}(\hat{\boldsymbol{x}}(1), \tilde{\boldsymbol{u}}(1), \boldsymbol{\theta}, \boldsymbol{w}(1)) \\
&\quad \tilde{\boldsymbol{y}}(1) = \boldsymbol{h}(\hat{\boldsymbol{x}}(1), \tilde{\boldsymbol{u}}(1), \boldsymbol{\theta}) + \boldsymbol{v}(1) \\
&\quad \vdots \\
&\quad \hat{\boldsymbol{x}}(k+1) = \boldsymbol{g}(\hat{\boldsymbol{x}}(k), \tilde{\boldsymbol{u}}(k), \boldsymbol{\theta}, \boldsymbol{w}(k)) \\
&\quad \tilde{\boldsymbol{y}}(k) = \boldsymbol{h}(\hat{\boldsymbol{x}}(k), \tilde{\boldsymbol{u}}(k), \boldsymbol{\theta}) + \boldsymbol{v}(k)\}.
\end{aligned}$$

A problem finding the CSP solution is the continuous increment with time in the computational effort. An alternative for overcoming this problem is the use of a *sliding time window*. The time interval from the initial time point to the current one is called *time window* w [17].

Consistency techniques can be used to contract the domains of the variables involved removing inconsistent values [18], [19], [20]. In particular for the fault detection application, they are used to guarantee that the observed behavior and the model are inconsistent when there is no solution. The algorithms that are based on consistency techniques are actually "branch and prune" algorithms, i.e., algorithms that can be defined as an iteration of two steps [18]:

1) Pruning the search space by reducing the intervals associated with the variables until a given consistency property is satisfied.
2) Generating subproblems by splitting the domains of a variable

Most interval constraint solvers are based on either hull-consistency (also called 2B-consistency) or box-consistency, or a variation of them [19]. Box-consistency tackles the problem of hull-consistency for variables with many occurrences in a constraint. The aforementioned techniques are said to be local: each reduction is applied over one domain with respect to one constraint. Better pruning of the variable domains may be achieved if, complementary to a local property, some global properties are also enforced on the overall constraint set.

In this paper, the solution of the fault detection and isolation CSP is achieved by using the solver RealPaver [9]. The BC4 algorithm, an efficient combination of hull and box consistency, is used in Section III.

In this paper, only the case where the fault is caused by a change of a singular parameter is considered. For each parameter, its initial domain is set to its possible range in practice and the initial domains of the other parameters are equal to the nominal intervals. For example, if we have three parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$, and the corresponding nominal intervals, $\boldsymbol{\Theta^0} = (\Theta_1^0, \Theta_2^0, \Theta_3^0)$, and possible range in practice, $\boldsymbol{\Theta^p} = (\Theta_1^p, \Theta_2^p, \Theta_3^p)$, then three constraint satisfaction problems are taken into account. For the first, the set of initial domains of the parameters is: $(\Theta_1^p, \Theta_2^0, \Theta_3^0)$, for the second, $(\Theta_1^0, \Theta_2^p, \Theta_3^0)$, and finally, for the third, $(\Theta_1^0, \Theta_2^0, \Theta_3^p)$.

As a novelty, in this paper the fault isolation problem is also stated as a CSP and solved using the same reasoning for solving the fault detection problem described above. Thus, the CSP for the fault isolation is similar to the one for fault detection. The fault isolation task starts once the fault has been detected. The sliding time window goes up from its smallest value until it gets its maximum possible value. When no CSP solution is found, we can judge that the fault is not caused by a change of the parameter θ_i, in which the initial domain is the possible range Θ_i^p. Satisfactory simulation results are presented in Section III-A.

III. Application example: The alcoholic fermentation process

A well-known dynamical example of an alcoholic fermentation process [8] will be used to explain the proposed method for fault detection and isolation.

The fermentation consists in growing a population of microorganisms by feeding them appropriate nutrients or substrates, provided the environmental conditions are propitious [14].

The model obtained from the mass balance considerations is composed of the following differential equations:

$$\begin{cases} \frac{\mathrm{d}C(t)}{\mathrm{d}t} = \mu(t)C(t) - D(t)C(t) \\ \frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{1}{Y_{c/s}}\mu(t)C(t) + D(t)S_a - D(t)S(t) \\ \frac{\mathrm{d}P(t)}{\mathrm{d}t} = \frac{Y_{p/s}}{Y_{c/s}}\mu(t)C(t) + D(t)P(t) \end{cases} \quad (2)$$

where $C(t)$, $S(t)$ and $P(t)$ represent respectively the biomass, substrate, and product concentrations in the bioreactor. The dilution rate $D(t)$ is used as the control variable. S_a represents the substrate concentration in the feeding. $Y_{c/s}$ and $Y_{p/s}$ are the yield coefficients and it is assumed that they are known and constant. The measurable state is the substrate concentration $S(t)$. $\mu(t)$ represents the growth rate of the biomass, and it is a nonlinear function of the variable $S(t)$ described by

$$\mu(t) = \mu_m \frac{S(t)}{K_s + S(t)} \quad (3)$$

where μ_m is the maximum growth rate and K_s is the saturation constant.

Faults are modeled as a single parameter change in the process parameters μ_m and K_s.

The interval method presented in this paper uses discrete-time models. In this case a discretization is obtained by using a first order approximation:

$$\boldsymbol{x}(t + T_s) \simeq \boldsymbol{x}(t) + T_s\, \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \quad (4)$$

where the sample time, T_s, is equal to 3 minutes.

Thus, from (2), the following discrete-time model can be obtained:

$$\begin{aligned} \hat{C}(k+1) &= \hat{C}(k) + T_s(\mu(k)\hat{C}(k) - \tilde{D}(k)\hat{C}(k)) + w_1(k) \\ \hat{S}(k+1) &= \hat{S}(k) - T_s(\tfrac{\mu(k)}{Y_{c/s}}\hat{C}(k) - \tilde{D}(k)(S_a - \hat{S}(k))) + w_2(k) \\ \hat{P}(k+1) &= \hat{P}(k) + T_s(\tfrac{Y_{p/s}}{Y_{c/s}}\mu(k)\hat{C}(k) + \tilde{D}(k)\hat{P}(k)) + w_3(k) \\ \tilde{S}(k) &= \hat{S}(k) + v(k) \end{aligned} \quad (5)$$

where $w_i(k)$ is the perturbation vector at time k, and it takes into account, for example, an error due to the discretization procedure. $v(k)$ is the measurement noise of the interval measurement $\tilde{S}(k)$.

A. Simulation results

The nominal values of model parameters used as well as the yield coefficients are obtained from real applications and are given by [8]:

$$\begin{aligned} \mu_m &= 0.38h^{-1} \\ K_s &= 5g/l \\ Y_{c/s} &= 0.07 \\ Y_{p/s} &= 0.44 \\ S_a &= 100g/l \end{aligned}$$

The possible value ranges, i.e. experimental considerations, of the parameters in practice are given by $K_s \in [0.5, 5.1]$ and $\mu_m \in [0.2, 0.53]$.

In this paper two faulty scenarios are considered: (i) the faulty parameter is μ_m and its value is 0.3, and (ii) the faulty parameter is K_s and its value is 3.1.

In the simulation, $D(t)$ is selected as a rectangular wave varying between 0.1 and 0.27 with a period of 30 hours.

Fault detection results are obtained by using the BC4 consistency technique and a window length equal to 100 samples (5h). When no solution is found to the CSP, a fault is detected. Otherwise, when the observed behavior and the model are not proven to be inconsistent, this means there is not a fault or it could not be detected. In this way, the proposed approach prioritizes avoiding false alarms over missed alarms.

1) First scenario:

This scenario analyzes a fault appearing as a deviation of the parameter μ_m. Regarding the nominal range of $\mu_m \in [0.36, 0.41]$, obtained results for the faulty parameter $\mu_m = 0.3$ are shown in Fig. 1. "FD" indicates there is a fault and "NF—FND" means there is not a fault or one could not be detected.

As shown in this figure, there is no false alarm in the absence of a fault. The fault begins at $70h$ and is detected from $70.05h$.

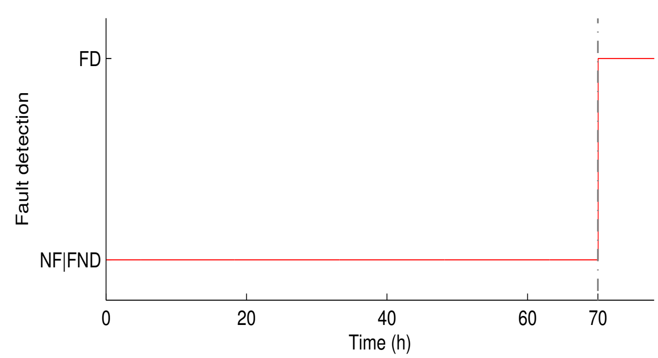

Fig. 1. First scenario fault detection. Fault in parameter μ_m beginning at time $t = 70h$. The fault is detected from $70.05h$.

Once the fault is detected, the fault isolation algorithm starts. Isolation results are presented in Fig. 2 and Fig. 3.

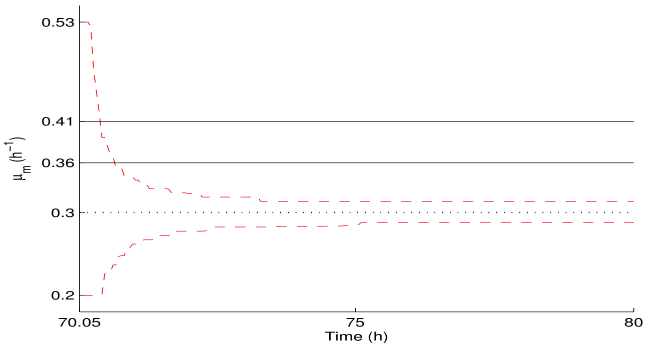

Fig. 2. First scenario fault isolation (Faulty parameter $\mu_m = 0.3$). Consistent values for μ_m when the parameter K_s is equal to its nominal interval $K_s = [4.9, 5.1]$.

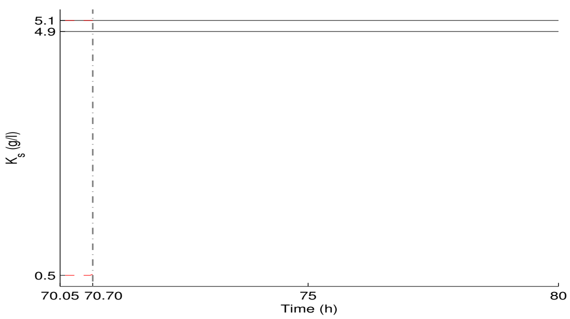

Fig. 3. First scenario fault isolation (Faulty parameter $\mu_m = 0.3$). Consistent values for K_s when the parameter μ_m is equal to its nominal interval $\mu_m \in [0.36, 0.41]$.

Fig. 2 shows the consistent values for μ_m when the parameter K_s is equal to its nominal interval, $K_s = [4.9, 5.1]$. After almost 10 hours of detecting the fault, the consistent interval of μ_m is equal to $[0.282, 0.314]$, which includes the faulty parameter value $\mu_m = 0.3$.

Similarly, Fig. 3 shows the consistent values for K_s when the parameter μ_m is equal to its nominal interval, $\mu_m = [0.36, 0.41]$. Since there is no consistent region of K_s in its feasible range of variation, the fault is not in the parameter K_s. Therefore the fault associated with a deviation in this parameter can be discarded at time $70.70h$.

2) *Second scenario:*

The second scenario considers a deviation of parameter K_s from its nominal region $K_s \in [0.5, 5.1]$. In Fig. 4, obtained results for the faulty parameter $K_s = 3.1$ are shown. The fault begins at $70h$ and is detected from $70.35h$.

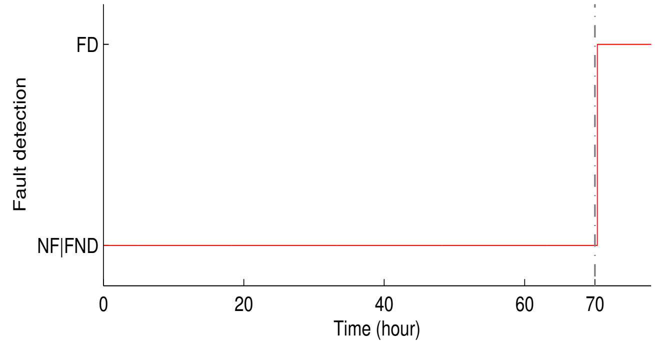

Fig. 4. Second scenario fault detection. Fault in parameter K_s beginning at time $t = 70h$. The fault is detected from $70.35h$.

Fig. 5 shows the consistent values for μ_m when the parameter K_s is equal to its nominal interval, $K_s = [4.9, 5.1]$. Since there is no consistent region of μ_m in its feasible range of variation, the fault is not in the parameter μ_m. Therefore the fault associated with a deviation in this parameter can be discarded at time $75.20h$.

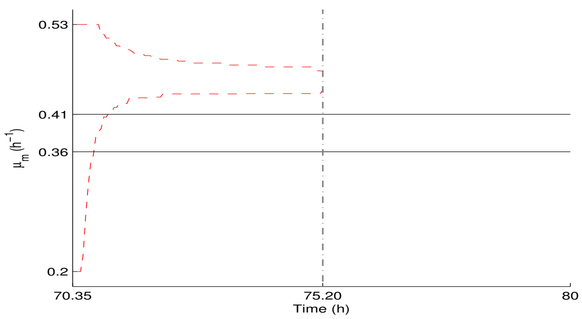

Fig. 5. Second scenario fault isolation (Faulty parameter $K_s = 3.1$). Consistent values for μ_m when the parameter K_s is equal to its nominal interval $K_s = [4.9, 5.1]$.

Fig. 6 shows the consistent values for K_s when the parameter μ_m is equal to its nominal interval, $\mu_m = [0.36, 0.41]$. After almost 10 hours of detecting the fault, the consistent interval of K_s is equal to $[2.89, 3.25]$, which includes the faulty parameter value $K_s = 3.1$.

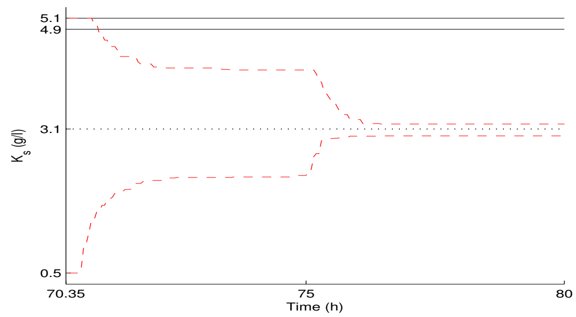

Fig. 6. Second scenario fault isolation (Faulty parameter $K_s = 3.1$). Consistent values for K_s when the parameter μ_m is equal to its nominal interval $\mu_m \in [0.36, 0.41]$.

IV. Conclusions and Future Works

When interval uncertainties are considered, consistency methods can be used to solve fault detection problems. In this paper, through the obtained results, consistency techniques are shown to be particularly efficient to solve the isolation problem when a fault can be represented as parameter deviations. The speed of fault isolation is fast (similar to the one obtained in [8]) even dealing with uncertain measurements, parameters, and model errors. Interval calculations allow the proposed approach to be independent of monotony assumptions. In the future, the case of multiple faults (or a fault caused by the changes of multiple parameters), must be studied in depth.

V. Acknowledgments

This work has been partially funded by the European Social Fund of the European Union, the Spanish Government through the coordinates research projects No. DPI2006-15476-C02-02 and No. DPI2005-08668-C03-02, by the grant No. 2005SGR00296 and the Departament d'Innovació, Universitats, i Empresa of the Government of Catalonia.

References

[1] K. Balakrishnan and V. Honavar, "Intelligent diagnosis systems," *Journal of Intelligent Systems*, vol. 8, no. 3, pp. 239–290, 1998.

[2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, pp. 293–311, 2003.

[3] J. Armengol, J. Vehí, L. Travé-Massuyès, and M. Á. Sainz, "Interval model-based fault detection using multiple sliding time windows," in *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2000. Budapest, Hungary*, 2000, pp. 168–173.

[4] V. Puig, J. Quevedo, and T. Escobet, "Model-based fault diagnosis of dynamic processes: comparing FDI and DX," in *IV Jornadas de ARCA: Sistemas Cualitativos y Diagnosis*, 2002, p. 91.

[5] S. Ploix and C. Follot, "Fault diagnosis reasoning for set-membership approaches and application," in *IEEE International Symposium on Intelligent Control*, 2001.

[6] A. Stancu, V. Puig, and J. Quevedo, "Gas turbine model-based robust fault detection using a forward - backward test." in *COCOS*, 2003, pp. 154–170.

[7] V. Puig, C. Ocampo-Martínez, S. Tornil, and A. Ingimundarson, "Robust fault detection using set-membership estimation and constraints satisfaction," in *17th International Workshop on Principles of Diagnosis DX*, 2006, pp. 227–234.

[8] Z. Li and B. Dahhou, "Fault isolation for nonlinear dynamic systems based on parameter intervals," *International Journal of Systems Science*, vol. 38, pp. 531 – 547, January 2007.

[9] L. Granvilliers and F. Benhamou, "Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques." *ACM Trans. Math. Softw.*, vol. 32, no. 1, pp. 138–156, 2006.

[10] Z. Li, "Contribution à l'élaboration d'algorithmes d'isolation et d'identification de défauts dans les systèmes non linéaires," PhD Thesis, Institut National des Sciences Appliquées, Toulouse (France), 2006.

[11] Z. Li and B. Dahhou, "An observers based fault isolation approach for nonlinear dynamic systems," Marrakech (Maroc), Mars 2006.

[12] Z. Li, B. Dahhou, and G. Roux, "A fault isolation method for nonlinear dynamic systems based on monotonous observers," Beijing (Chine), October 2006.

[13] S. Rajaraman, "Robust model-based fault diagnosis of chemical process systems," PhD Thesis, Texas A&M University, Texas (EEUU), 2005.

[14] N. Kabbaj, M. Polit, B. Dahhou, and G. Roux, "Adaptive observers based fault detection and isolation for an alcoholic fermentation process," Antibes - Juan les Pins (France), October 2001.

[15] S. Shary, "A new technique in systems analysis under interval uncertainty and ambiguity," *Reliable Computing*, vol. 8, pp. 321–418, 2002.

[16] L. Jaulin, "Consistency techniques for the localization of a satellite," in *COCOS*, 2002, pp. 157–170.

[17] J. Armengol, J. Vehí, L. Travé-Massuyès, and M. Á. Sainz, "Application of multiple sliding time windows to fault detection based on interval models," in *12th International Workshop on Principles of Diagnosis DX 2001. San Sicario, Italy*, S. McIlraith and D. Theseider Dupré, Eds., 2001, pp. 9–16.

[18] H. Collavizza, F. Delobel, and M. Rueher, "Comparing partial consistencies," *Reliable Computing*, vol. 5, pp. 213–228, 1999.

[19] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget, "Revising hull and box consistency," in *Proceedings of the International Conference on Logic Programming*, Las Cruces, NM, 1999, pp. 230–244.

[20] J. Cruz and P. Barahona, "Maintaining global hull consistency with local search for continuous CSPs." in *COCOS*, 2002, pp. 178–193.