

Constraint Satisfaction Techniques under uncertain conditions for Fault Diagnosis in Nonlinear Dynamic Systems

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

Abstract—The speed of fault isolation is crucial for the design and reconfiguration of fault tolerant control (FTC). In this paper the fault isolation problem is stated as a constraint satisfaction problem (CSP) and solved using constraint propagation techniques. The proposed method is based on constraint satisfaction techniques and uncertainty space refining of interval parameters. In comparison with other approaches based on adaptive observers, the major advantage of the presented method is that the isolation speed is fast even taking into account uncertainty in parameters, measurements and model errors and without the monotonicity assumption. In order to illustrate the proposed approach, a case study of a nonlinear dynamic system is presented.

I. INTRODUCTION

Fault diagnosis of industrial processes becomes more important in light of increased automation in industry. The early detection and isolation of faults can help avoid major system breakdowns. The development of model-based fault diagnosis began at various places in the early 1970s. Methods for fault detection in dynamic systems have been receiving more and more attention over the last two decades, because of economical and safety related matters. Fault detection and isolation methods are based on detecting discrepancies between the system outputs and model outputs.

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 using intervals. These uncertainties are caused by, for example, non-modeled effects, electrical disturbances, model simplifications, and so on.

Some interval methods have been proposed in the context of fault detection and diagnosis, e.g. [1], [2] and [3]. [4] include constraint propagation to solve fault detection problems. In [5], the problem is solved using a tool known as IntervalPeeler, based on constraint projection algorithms (2B-consistency) to reduce interval domains of variables without bisections.

When interval uncertainties are considered, consistency methods which combine interval methods and constraint satisfaction techniques can be used to solve different problems such as system and state estimation, fault detection,

robustness analysis, robust control design problems, risk assessment, and worst case behavior analysis.

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. In section II the fault detection and isolation problems are shown like constraint satisfaction problems and the resolution of them is performed by the solver RealPaver [6]. The alternative to use an efficient combination of Hull and Box consistency, is explored.

In [7] a method of fault isolation for non-linear dynamic systems is presented, which assumes that the fault is detected once it occurs and the isolation procedure is triggered at this time. That method is based on the monotonous characteristic of an observer prediction error and parameter partitioning, the authors remark its speed, being quicker than other methods based on adaptive observers.

The main contribution of this paper is that, the isolation problem is based on parameters uncertainty refining instead of partitioning, stated and solved as a Constraint Satisfaction Problem (CSP). A sliding time window is used to reduce the computational effort. Interval calculations lead the proposed approach to be independent of the assumption that the system dynamics is a monotonous function with respect to the considered parameters.

Thus, 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.

In order to illustrate the proposed approach effectiveness, it is applied to a well known alcoholic fermentation process presented, for instance in [7], [8], [9] and [10]. In section III, the used model is described, two simulation scenarios are considered and fault detection and isolation results are presented. Finally, section IV provides some conclusions and outlines the future work.

II. CONSTRAINT SATISFACTION TECHNIQUES

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.

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 Department d'Innovació, Universitats, i Empresa of the Government of Catalonia.

Authors are with the Institut d'Informàtica i Aplicacions, Universitat de Girona, Campus de Montilivi, E-17071 Girona, Spain, {sandra.castillo,esteban.gelso,joaquim.armengol}@udg.edu

As defined in [11], a numerical constraint satisfaction problem is a triple $CSP = (\mathcal{V}, \mathcal{D}, \mathcal{C}(x))$ defined by

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

A solution to a numeric constraint satisfaction problem $(\mathcal{V}, \mathcal{D}, \mathcal{C}(x))$ is an instantiation of the variables of \mathcal{V} for which both inclusion in the associated domains and all the constraints of $\mathcal{C}(x)$ are satisfied. All the solutions of the constraint satisfaction problem thus constitute the set

$$\Sigma = \{x \in \mathcal{D} \mid \mathcal{C}(x) \text{ is satisfied}\}.$$

Consistency techniques can be used to contract the domains of the variables involved removing inconsistent values [12], [13], [14]. 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 [12]:

- 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 satisfaction solvers are based on either hull-consistency (also called 2B-consistency) or box-consistency, or a variation of them [13]. 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 performed by using the solver RealPaver [6]. The BC4 algorithm, an efficient combination of hull and box consistency, is used in Section III.

A. Fault detection

The fault detection problem can be represented by a CSP similar to the one presented in [15], which deals with the problem of nonlinear state estimation. For example, consider a discrete-time nonlinear dynamic system described by:

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

where:

- $\mathbf{u}(k) \in \mathbb{R}^{n_u}$, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ and $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ are the input, output and state vector, respectively.

- $\mathbf{w}(k) \in \mathbb{R}^{n_w}$ and $\mathbf{v}(k) \in \mathbb{R}^{n_y}$ 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.
- $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ is a vector of uncertain parameters.

The dynamic system (1) can be represented as a CSP_{fd} :

$$\begin{aligned} \mathcal{V} &= \{\boldsymbol{\theta}, \tilde{\mathbf{y}}(1), \dots, \tilde{\mathbf{y}}(k), \hat{\mathbf{x}}(1), \dots, \hat{\mathbf{x}}(k+1), \tilde{\mathbf{u}}(1), \dots, \tilde{\mathbf{u}}(k) \\ &\quad \mathbf{w}(1), \dots, \mathbf{w}(k), \mathbf{v}(1), \dots, \mathbf{v}(k)\} \\ \mathcal{D} &= \{\boldsymbol{\Theta}^0, \tilde{\mathbf{Y}}(1), \dots, \tilde{\mathbf{Y}}(k), \hat{\mathbf{X}}(1), \dots, \hat{\mathbf{X}}(k+1), \tilde{\mathbf{U}}(1), \dots, \tilde{\mathbf{U}}(k) \\ &\quad \mathbf{W}(1), \dots, \mathbf{W}(k), \mathbf{V}(1), \dots, \mathbf{V}(k)\} \\ \mathcal{C} &= \{\hat{\mathbf{x}}(2) = \mathbf{g}(\hat{\mathbf{x}}(1), \tilde{\mathbf{u}}(1), \boldsymbol{\theta}, \mathbf{w}(1)) \\ &\quad \tilde{\mathbf{y}}(1) = \mathbf{h}(\hat{\mathbf{x}}(1), \tilde{\mathbf{u}}(1), \boldsymbol{\theta}) + \mathbf{v}(1) \\ &\quad \vdots \\ &\quad \hat{\mathbf{x}}(k+1) = \mathbf{g}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}, \mathbf{w}(k)) \\ &\quad \tilde{\mathbf{y}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \boldsymbol{\theta}) + \mathbf{v}(k)\}. \end{aligned}$$

where:

- $\boldsymbol{\Theta}^0$ is defined by the nominal range of parameter variation, and
- $\tilde{\mathbf{Y}}(k)$ and $\tilde{\mathbf{U}}(k)$ are given by interval measurements, which are inaccurate due to the analog to digital conversion errors, the calibration of the sensors, etc.

The fault-free operating mode is characterized by the nominal vector $\boldsymbol{\Theta}^0$.

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* ω [16], Fig. 1. By using sliding time windows, the parameters of the system are allowed to vary in time at a speed depending on the size of the window (they are assumed to be invariant within each window). The smaller is the window, the higher is the allowed variation speed.

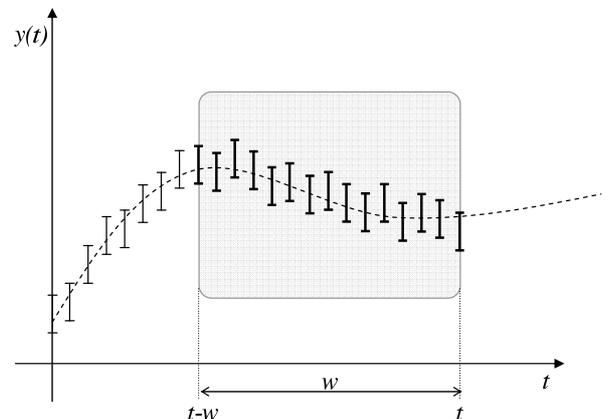


Fig. 1. Fault detection by means of the consistency between the model and all the measurements within a window of length ω .

When no solution is found for CSP_{fd} , a fault is detected. Otherwise, when the observed behavior and the model are not proven to be inconsistent, means that there is not a fault or, if there is a fault, it can not be detected.

B. Fault isolation

In this paper the fault isolation problem is also stated as a CSP, similar to the one for fault detection. Considering the CSP_{fd} , a general approach for fault isolation can be stated replacing the initial domain of θ , Θ^0 by Θ^p . Where Θ^p is the feasible range of parameter variation, given by, e.g. practical considerations. Then, the isolation problem is solved, by estimate the consistency parameter range under fault conditions.

A particular case is when at a time, a single independent parameter of the system may be faulty. As a novelty, under this single-fault hypothesis, and considering that:

- k_{fd} is the fault detection time
- ω_{max} is the maximum sliding time window for fault isolation, and
- \mathcal{F} is the set of fault candidates (possible faulty parameters),

the generic algorithm of the proposed approach can be represented as:

Algorithm 1 Algorithm for Fault Isolation

```

1: begin
2:    $\omega = \min(k - k_{fd}, \omega_{max})$ 
3:   for  $i = 1$  to  $n_p$  do
4:      $\Theta = \begin{cases} \Theta_i = \Theta_i^p \\ \Theta_j = \Theta_j^0 \quad \forall j \neq i \in \{1, \dots, n_p\} \end{cases}$ 
5:      $\mathcal{V} = \{ \theta, \tilde{\mathbf{y}}(k-\omega), \dots, \tilde{\mathbf{y}}(k), \hat{\mathbf{x}}(k-\omega), \dots, \hat{\mathbf{x}}(k+1),$ 
6:        $\tilde{\mathbf{u}}(k-\omega), \dots, \tilde{\mathbf{u}}(k), \mathbf{w}(k-\omega), \dots, \mathbf{w}(k),$ 
7:        $\mathbf{v}(k-\omega), \dots, \mathbf{v}(k) \}$ 
8:      $\mathcal{D} = \{ \Theta, \tilde{\mathbf{Y}}(k-\omega), \dots, \tilde{\mathbf{Y}}(k), \hat{\mathbf{X}}(k-\omega), \dots, \hat{\mathbf{X}}(k+1),$ 
9:        $\tilde{\mathbf{U}}(k-\omega), \dots, \tilde{\mathbf{U}}(k), \mathbf{W}(k-\omega), \dots, \mathbf{W}(k),$ 
10:       $\mathbf{V}(k-\omega), \dots, \mathbf{V}(k) \}$ 
11:      $\mathcal{C} = \{ \hat{\mathbf{x}}(k-\omega+1) = \mathbf{g}(\hat{\mathbf{x}}(k-\omega), \tilde{\mathbf{u}}(k-\omega), \theta, \mathbf{w}(k-\omega))$ 
12:        $\tilde{\mathbf{y}}(k-\omega) = \mathbf{h}(\hat{\mathbf{x}}(k-\omega), \tilde{\mathbf{u}}(k-\omega), \theta) + \mathbf{v}(k-\omega)$ 
13:        $\vdots$ 
14:        $\hat{\mathbf{x}}(k+1) = \mathbf{g}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \theta, \mathbf{w}(k))$ 
15:        $\tilde{\mathbf{y}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k), \tilde{\mathbf{u}}(k), \theta) + \mathbf{v}(k) \}$ 
16:      $CSP_{fi} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ 
17:      $\Sigma = \text{solution}(CSP_{fi})$ 
18:     if  $\Sigma = \emptyset$  then
19:       Erase  $\theta_i$  from  $\mathcal{F}$ 
20:     end if
21:   end for
22: end

```

In this algorithm, the fault isolation task starts once the fault has been detected. 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 $\theta = (\theta_1, \theta_2, \theta_3)$, and the corresponding nominal intervals, $\Theta^0 = (\Theta_1^0, \Theta_2^0, \Theta_3^0)$, and feasible range of variation in practice, $\Theta^p = (\Theta_1^p, \Theta_2^p, \Theta_3^p)$, then three constraint satisfaction problems are solved. 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, $(\Theta_1^0, \Theta_2^0, \Theta_3^p)$.

The sliding time window goes up from its smallest value until it gets its maximum possible value. Considering as initial domain the feasible range of variation Θ_i^p , when no CSP solution is found, we can decide that the fault is not caused by a value change of the parameter θ_i , because no value of $\theta_i \in \Theta_i^p$ can explain measurement data.

Satisfactory simulation results are presented in Section III. Only the case where the fault is caused by a change of a singular parameter is considered.

III. CASE STUDY

1) Process model:

A well-known dynamical example of an alcoholic fermentation process [7] 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 [10]. The model obtained from the mass balance considerations is composed of the following differential equations:

$$\begin{aligned} \frac{dC(t)}{dt} &= \mu(t)C(t) - D(t)C(t) \\ \frac{dS(t)}{dt} &= -\frac{1}{Y_{c/s}}\mu(t)C(t) + D(t)S_a - D(t)S(t) \end{aligned} \quad (2)$$

where,

- $C(t)$ and $S(t)$ represent respectively the biomass, and substrate concentrations in the bioreactor.
- The dilution rate $D(t)$ is used as the control variable.
- S_a represents the substrate concentration in the feeding.
- $Y_{c/s}$ is the yield coefficient and it is assumed that it is 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

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

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

The interval method presented in this paper uses discrete-time models, in this case a discretization is obtained by using a first order approximation, where T_s is the sample time:

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

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

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

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 $\tilde{S}(k)$.

In the simulation, $D(t)$ is selected as a rectangular wave varying between 0.1 and 0.27 with a period of 30 hours. The sample time, T_s , is equal to 3 minutes. The feasible ranges of parameter variation, i.e. experimental considerations in practice are given by $\mu_m \in [0.2, 0.53]$ and $K_s \in [0.5, 5.1]$.

The nominal values of model parameters used as well as the yield coefficients are obtained from real applications and are given by [7] (see Table I):

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

TABLE I
PARAMETERS AND YIELD COEFFICIENTS: NOMINAL VALUES

2) Fault detection results:

Considered faults are modeled as a single parameter change in the process parameters μ_m and K_s . Two faulty scenarios are considered.

The fault detection results (see Table II) are obtained by using BC4 consistency technique and a sliding window length equal to 100 samples (5h).

Scenario	Faulty parameter	Nominal range	Faulty value	Fault occurrence time (h)	Detection time (h)
(i)	μ_m	[0.36, 0.41]	0.3	70	70.05
(ii)	K_s	[4.90, 5.10]	3.1	70	70.35

TABLE II
FAULT DETECTION RESULTS

In Fig. 2 and Fig. 3, obtained results for both fault scenarios are shown. “FD” indicates there is a fault and a “NF—FND” means there is not a fault or one could not be detected. As shown in figures, there is no false alarm in the absence of a fault.

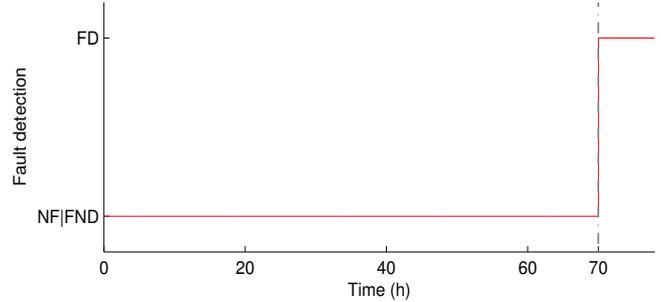


Fig. 2. Scenario (i). Fault in parameter μ_m detection.

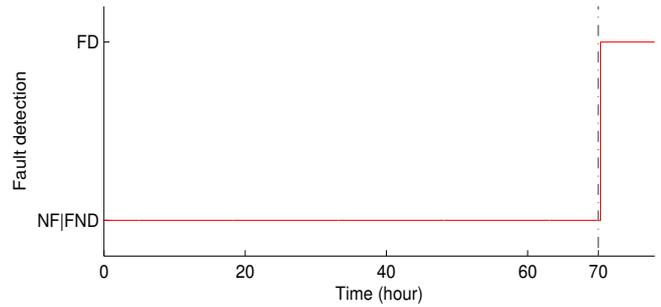


Fig. 3. Scenario (ii). Fault in parameter K_s detection.

3) Fault isolation results:

Once the fault is detected, the fault isolation algorithm starts. Since the parameters of this case study are μ_m , and K_s , two CSP must be solved. For the first, the set of initial domains of the parameters is: (μ_m^p, K_s^0) , and for the last one (μ_m^0, K_s^p) . When no consistent region is found in the feasible range of parameter variation, a fault associated with a value change of the refined parameter can be discarded. Obtained results for fault isolation in both of the scenarios, are summarized in Table III.

Scenario	Faulty parameter	Feasible range	Faulty value	Estimate range	Isolation time (h)
(i)	μ_m	[0.20, 0.53]	0.3	[0.282, 0.314]	70.70
(ii)	K_s	[0.50, 5.10]	3.1	[2.890, 3.250]	75.20

TABLE III
FAULT ISOLATION RESULTS

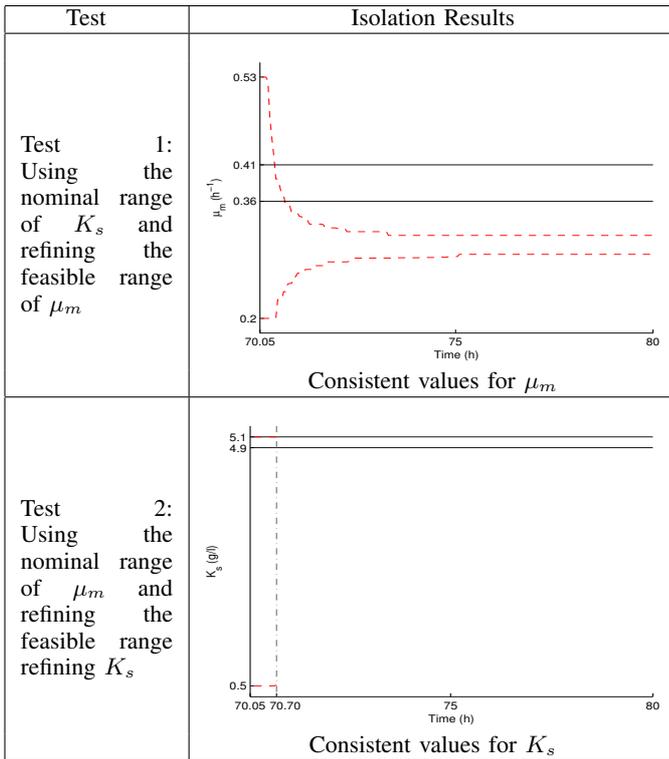


TABLE IV

SCENARIO(I). FAULT ISOLATION RESULTS

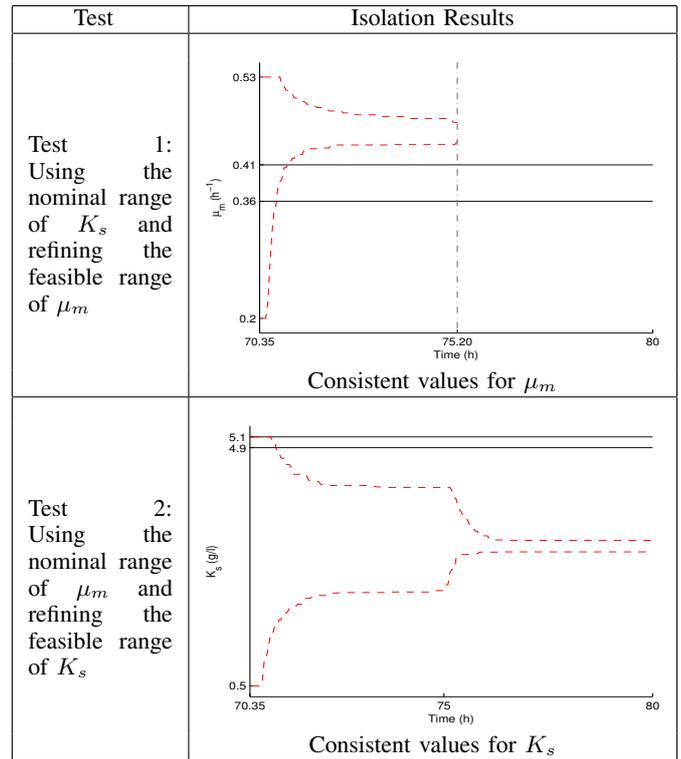


TABLE V

SCENARIO(II). FAULT ISOLATION RESULTS

Notice that in both scenarios the estimate range of the faulty parameter, after almost 10 hours of detecting the fault, includes the considered faulty value.

In Table IV and Table V, the results of the isolation test are presented in a more detailed way.

- Faulty parameter μ_m

Since there is no consistent region of K_s in its feasible range of variation, the fault is not in the parameter K_s and it can be discarded at time 70.70h (See Table IV).

- Faulty parameter K_s

Since there is no consistent region of μ_m in its feasible range of variation, the fault is not in the parameter μ_m and it can be discarded at time 75.20h (See Table V).

4) Isolation time:

Even taking into account uncertainty in measurements, parameters, and model errors, the isolation time in both scenarios obtained by means of the proposed in this papers approach, is of the same order of magnitud like the ones found in [7], for similar faulty scenarios, by using an approach non-based on interval calculations. It could be said

the two approach isolation speed are comparable between them and smaller than the speed isolation of other approaches based on adaptive observers.

IV. CONCLUSIONS AND FUTURE WORKS

The proposed isolation approach bring relevant information for the design and reconfiguration of fault tolerant control (FTC), such as fault magnitude and isolation time. Interval calculations bring to the approach independence from assumptions of monotony made in other approaches based on observers.

Through the case study, consistency techniques are shown to be particularly efficient to solve the isolation problem when a fault can be represented as parameter deviations.

The isolation method performance is satisfactory in comparison with other methods based on observers, specially taking into account the isolation speed.

As part of future work, we would like to investigate how the work presented in this paper can be extended for diagnosis of multiple faults (or faults which change more than one parameter simultaneously).

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] 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.
- [2] V. Puig, J. Quevedo, T. Escobet, and S. De las Heras, "Robust fault detection approaches using interval models," in *16th IFAC World Congress*, 2002.
- [3] S. Ploix and C. Follot, "Fault diagnosis reasoning for set-membership approaches and application," in *IEEE International Symposium on Intelligent Control*, 2001.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Z. Li, B. Dahhou, and G. Roux, "A fault isolation method for nonlinear dynamic systems based on monotonous observers," Beijing (China), October 2006.
- [10] 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.
- [11] S. Shary, "A new technique in systems analysis under interval uncertainty and ambiguity," *Reliable Computing*, vol. 8, pp. 321–418, 2002.
- [12] H. Collavizza, F. Delobel, and M. Rueher, "Comparing partial consistencies," *Reliable Computing*, vol. 5, pp. 213–228, 1999.
- [13] 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.
- [14] J. Cruz and P. Barahona, "Maintaining global hull consistency with local search for continuous CSPs." in *COCOS*, 2002, pp. 178–193.
- [15] L. Jaulin, "Consistency techniques for the localization of a satellite," in *COCOS*, 2002, pp. 157–170.
- [16] 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. Theiseider Dupré, Eds., 2001, pp. 9–16.