Preprints of the 7th IFAC Symposium on
Fault Detection, Supervision and Safety of Technical Processes
Barcelona, Spain, June 30 - July 3, 2009

Robust Fault Diagnosis of Non-linear Systems using Constraints Satisfaction

Vicenç Puig * **Teresa Escobet** * **Carlos Ocampo-Martinez** **
Sebastián Tornil-Sin *

* *Automatic Control Department*
Universitat Politècnica de Catalunya (UPC)
Rambla de Sant Nebridi, 10, 08222 Terrassa (Spain)
{vicenc.puig, teresa.escobet, sebastian.tornil}@upc.edu
** *Institut de Robòtica i Informàtica Industrial, CSIC-UPC*
C/. Llorens i Artigas, 4-6, 08028 Barcelona (Spain)
cocampo@iri.upc.edu

Abstract: In this paper, the robust fault diagnosis problem for non-linear systems considering both bounded parametric modelling errors and noises is addressed using constraints satisfaction. Combining available measurements with the model of the monitored system, a set of analytical redundancy relations (ARR), relating only known variables, can be derived. These relations will be used in the fault diagnosis procedure to check the consistency between the observed and the predicted system behaviour. When some inconsistency is detected, the fault isolation mechanism will be activated in order to provide an explanation of the possible cause. Finally, a fault estimation procedure is used to estimate the fault size. To illustrate the usefulness of the proposed approach, a case study based on a well-known four tanks system benchmark is used. *Copyright © IFAC 2009*

Keywords: Fault detection, robustness, intervals, set-membership estimation, constraints satisfaction.

1. INTRODUCTION

Model-based fault detection methods rely on the concept of analytical redundancy. The simplest analytical redundancy form consists in the comparison of measurements of a system output with corresponding analytically computed values. These latter values can be obtained from measurements of other variables and/or from previous measurements of the same variable by means of a model. In the general case, different estimations of a same variable, measured or not, can be compared. The resulting differences are called *residuals* and are indicative of faults in the system. Under ideal conditions, residuals are zero in the absence of faults and non-zero when faults are present. However, modelling errors, disturbances and noises in complex engineering systems are inevitable, and hence there is a need to develop robust fault detection algorithms. The robustness of a fault detection system indicates its ability to distinguish between faults and model-reality differences (Chen and Patton, 1999). Classical approaches facing disturbances and modelling errors use the disturbance decoupling principle trying to obtain a residual that is sensitive to faults but not to these errors. Techniques like unknown input observers, eigenstructure assignment (Chen and Patton, 1999) or structured parity equations (Gertler, 1998), among others, can be found in the literature. On the other hand, process and measurement noises are usually stochastically modelled (the typical assumption is a zero mean white noise) and their effect is considered using statistical decision methods (Basseville and Nikiforov, 1993). However, such approaches present several drawbacks. First, decoupling from modelling errors (specially for non-linear models) is difficult to solve because the distribution matrix is normally unknown and time varying. Thus, it should be estimated. Moreover, the number of decoupled disturbances/modelling errors is limited by the degree of freedom in the residual generation procedure (Gertler, 1998). As an alternative strategy, disturbances/model errors are assumed to be bounded and their effects are propagated to the residual using, for example, interval methods (Puig et al., 2002). Second, in many practical situations it is not realistic to assume a statistical distribution law for the noise, being more natural to assume that only bounds on the noise signals are available. In this case, the so called *set-membership approach* (Milanese et al., 1996) can be used in the context of fault detection as suggested by Witczak et al. (2002). In both cases, the advantage of the bounded description of uncertainty is that it does not requires restrictive assumptions (as an small number of unknown disturbances/parameters, known statistical distribution law). However, a limitation is that faults that produce a residual deviation smaller than the residual uncertainty due to model uncertainty will remain undetected (missed alarms).

In this paper, the robust fault diagnosis problem for non-linear systems considering both bounded parametric modelling errors and noises is addressed using *constraints satisfaction*. Combining the available measurements with the model of the monitored system, a set *analytical redundancy relations* (ARR), relating only known variables, can be derived. These relations will be used in the fault diagnosis procedure to check the consistency between the observed and the predicted process behaviour. When some inconsistency is detected, the fault isolation mechanism will be activated in order to provide an explanation of the possible cause. Finally, a fault estimation procedure is used to estimate the fault size. To illustrate the usefulness of the proposed approach, a case study based on a well-known control benchmark based on a four tanks system is used.

The remainder of the paper is organized as follows: *Section* 2 presents the background on constraints satisfaction. In *Section* 3, robust fault detection is formulated as a constraints satisfaction problem. In *Section* 4, the constraints satisfaction approach is applied to fault isolation and estimation tasks. The proposed fault diagnosis approach is applied to the four-tanks system in *Section* 5. Finally, in *Section* 6, the conclusions are drawn.

2. BACKGROUND ON CONSTRAINTS SATISFACTION

2.1 Introduction

A *Constraints Satisfaction Problem* (CSP) on sets can be formulated as a 3-tuple $\mathcal{H} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ (Jaulin et al., 2001), where

- $\mathcal{V} = \{v_1, \cdots, v_n\}$ is a finite set of variables,
- $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_n\}$ is the set of their domains represented by closed sets and
- $\mathcal{C} = \{c_1, \cdots, c_m\}$ is a finite set of constraints relating variables of $\mathcal{V}$.

A solution point of $\mathcal{H}$ is an n-tuple $(\tilde{v}_1, \cdots, \tilde{v}_n) \in \mathcal{D}$ such that all constraints in $\mathcal{C}$ are satisfied. The set of all solution points of $\mathcal{H}$ is denoted by $\mathcal{S}(\mathcal{H})$. This set is called the *global solution set*. The variable $v_i \in \mathcal{V}$ is *consistent* in $\mathcal{H}$ if and only if $\forall \tilde{v}_i \in \mathcal{D}_i$

$$\exists\ (\tilde{v}_1 \in \mathcal{D}_1, \cdots, \tilde{v}_{i-1} \in \mathcal{D}_{i-1}, \tilde{v}_{i+1} \in \mathcal{D}_{i+1}, \cdots, \tilde{v}_n \in \mathcal{D}_n) \mid (\tilde{v}_1, \cdots, \tilde{v}_n) \in \mathcal{S}(\mathcal{H}),$$

The solution of a CSP is said to be *globally consistent*, if and only if every variable is consistent. A variable is *locally consistent* if and only if it is consistent with respect to all directly connected constraints. Thus, the solution of CSP is said to be locally consistent if all variables are locally consistent. An algorithm for finding an approximation of the solution set of a CSP can be found in Jaulin et al. (2001).

2.2 Implementation using Intervals

It is well known that the solution of CSPs involving sets has a high complexity (Jaulin et al., 2001). A first relaxation consists of approximating the variable domains by means of intervals and finding the solution through solving an *Interval Constraints Satisfaction Problem* (ICSP) (Hyvönen, 1992). The determination of the intervals that approximate in a more fitted form the sets that define the variable domains requires global consistency, what demands a high computational cost (Hyvönen, 1992). A second relaxation consists in solving the ICSP by means of local consistency techniques, deriving on conservative intervals. Interval constraint satisfaction algorithms have a polynomial-time worst case complexity that implement local reasonings on constraints to remove inconsistent values from variable domains (Jaulin et al., 2001). In this paper, the ICSP is solved using a tool based on interval constraint propagation, known as *Interval Peeler*. This tool has been designed and developed by research team of Professor Luc Jaulin (Baguenard, 2005). The goal of this software is to determine the solution of ICSP in the case that domains are represented by closed real intervals, and consists in iterating two main operations: *domain contraction* and *propagation*. The solution provides refined interval domains consistent with the set of ICSP constraints.

A. Contractors. A contractor is an operator which reduces domains. Applied to the solution of a CSP $\mathcal{H}$, an operator $\mathcal{C}_{\mathcal{H}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$, where $\mathbb{IR}^n$ is the set of all interval vectors (boxes) in $\mathbb{R}^n$, is a contractor if it satisfies:

$$\forall [v] \in \mathcal{D} : \begin{cases} \mathcal{C}_{\mathcal{H}}([v]) \subset [v] \\ \mathcal{C}_{\mathcal{H}}([v]) \cap \mathcal{S}(\mathcal{H}) = [v] \cap \mathcal{S}(\mathcal{H}). \end{cases} \tag{1}$$

The purpose of a contractor is to reduce any box $[v]$ without loosing any solution point in $\mathcal{S}(\mathcal{H})$. In Jaulin et al. (2001), a number of contractors for a variety of sets are given. They are algorithms of polynomial complexity that reduce the interval domains of variables that comply with a set of constraints.

B. Propagation. When several constraints are involved, the contractions are performed sequentially, until no more significant contraction can be observed. The interval propagation method converges to a box which contains all solution vectors of the constraints set. If this box is empty, it means that there is no solution. It can be shown that the box to which the method converges does not depend on the order to which the contractors are applied (Jaulin et al., 2001), but the computing time is highly sensitive to this order. There is no optimal order in general, but in practice, one of the most efficient is called *forward-backward propagation*.

3. ROBUST FAULT DETECTION AS A CSP

3.1 System Modelling

Let us consider the following discrete-time non-linear system describing the behavior of the system to be monitored:

$$\begin{aligned} x_{k+1} &= g(x_k, u_k, \theta_k) \\ y_k &= h(x_k, u_k, \theta_k), \end{aligned} \tag{2}$$

where

- $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the vector of system states, $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the vector of system inputs and $y \in \mathcal{Y} \subseteq \mathbb{R}^{n_y}$ is the vector of system outputs. Moreover, $\mathcal{X}, \mathcal{U}$ and $\mathcal{Y}$ are the interval boxes
$$\begin{aligned} \mathcal{X} &= \{x \in \mathbb{R}^{n_x} \mid \underline{x}_i \leq x_i \leq \bar{x}_i,\ i = 1, \cdots, n_x\}, \\ \mathcal{U} &= \{u \in \mathbb{R}^{n_u} \mid \underline{u}_i \leq u_i \leq \bar{u}_i,\ i = 1, \cdots, n_u\}, \\ \mathcal{Y} &= \left\{y \in \mathbb{R}^{n_y} \mid \underline{y}_i \leq y_i \leq \bar{y}_i,\ i = 1, \cdots, n_y\right\}. \end{aligned}$$
- $g : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ are the state space and measurement non-linear functions.
- $\theta_k \in \mathbb{R}^{n_\theta}$ is a vector of uncertain time-varying parameters with their values bounded by a compact set $\theta_k \in \Theta$ of box type
$$\Theta = \left\{\theta \in \mathbb{R}^{n_\theta} \mid \underline{\theta} \leq \theta \leq \bar{\theta}\right\}.$$

Alternatively, according to Blanke et al. (2006), the model of the system (2) can be defined by a pair $(\mathcal{C}, \mathcal{V})$, where

$$\mathcal{V} = \{x\} \cup \{u\} \cup \{y\} \cup \{\theta\}$$

is the set of variables that are in the domains $\mathcal{D} = \{\mathcal{X}, \mathcal{U}, \mathcal{Y}, \Theta\}$ and

$$\mathcal{C} = \{g\} \cup \{h\}$$

is the set of constraints.

The system variables and parameters can be decomposed into known $\mathcal{V}_{known}$ and unknown $\mathcal{V}_{unknown}$ ones. When system inputs and outputs are measured, $\mathcal{V}_{known} = \{u\} \cup \{y\} \cup \{\theta\}$.

The design of a model-based diagnosis mechanism is based on utilizing the system model in the construction of the diagnosis tests. Using the structural analysis tool (Blanke et al., 2006), a set of ARRs, $\mathcal{R}$, can be derived from the discrete-time non-linear system (2). ARRs are constraints which only involve the subset of known variables $\mathcal{V}_{known}$. The set of ARRs can be represented as

$$\mathcal{R} = \{r_i = \Psi_i(y_k, u_k, \theta_k) = 0,\ i = 1, \ldots, n_r\}, \tag{3}$$

where Ψ_i is an ARR function and n_r is the number of ARRs obtained.

3.2 Fault Detection using CSP

To be able to diagnose a system, values of the known variables and a set ARRs of the system are needed. The known variables are typically measured sensor and actuator signals. Thus, let us denote the following sequences of input/output measurements and parameters in a time window L from the initial time instant $k - L$ to time a instant k:

$$\tilde{u_k} = (u_j)_{j=k-L}^{k} = (u_{k-L}, u_{k-L+1}, ..., u_k), \tag{4}$$

$$\tilde{y_k} = (y_j)_{j=k-L}^{k} = (y_{k-L}, y_{k-L+1}, ..., y_k), \tag{5}$$

$$\tilde{\theta_k} = (\theta_j)_{j=k-L}^{k} = (\theta_{k-L}, \theta_{k-L+1}, ..., \theta_k). \tag{6}$$

Definition 3.1. (Set-membership estimation). *Given the set of ARRs* (3) *and a sequence of inputs $\tilde{u_k}$ and outputs $\tilde{y_k}$ of the system at time* k*, the set of input/ouputs and parameters that are consistent at time k using the* set-membership *approach is*

$$(\mathcal{U}_k, \mathcal{Y}_k, \Theta_k) = \Big\{(u_k, y_k, \theta_k)\,|\ \exists\,(u_j \in \mathcal{U}_j)_{k-L}^{k}, (y_j \in \mathcal{Y}_j)_{k-L}^{k}, (\theta_j \in \Theta_j)_{k-L}^{k} \text{ such that } \Psi(\tilde{y}_k, \tilde{u}_k, \tilde{\theta}_k) = 0\Big\}.$$

According to this definition, a fault can be detected as follows.

Definition 3.2. (Fault detection). *Given a sequence of measured inputs $\tilde{u_k}$ and outputs $\tilde{y_k}$ of the system, a fault is detected at time* k *if there does not exist a set of sequences $\tilde{u_k}$, $\tilde{y_k}$ and $\tilde{\theta_k}$ which satisfy the set of ARRs* (3) *with inputs, outputs and parameters belonging to $\mathcal{U}$, $\mathcal{Y}$ and Θ, respectively.*

Definition 3.1 and 3.2 suggest a way of implementing the set-membership estimation and fault detection problem as a CSP (Blanke et al., 2006). In this case, when a fault occurs, the solution to the CSP associated to the set-membership estimation problem given in *Definition* 3.1 will be the empty set, detecting the presence of an inconsistency. *Algorithm* 1 resumes the computation procedure when ICSP approach is used.

4. FAULT ISOLATION AND ESTIMATION AS A CSP

4.1 Fault Isolation as a CSP

Once an inconsistency is detected using *Algorithm* 1, fault isolation can be achieved by proving the consistency of the sequence of inputs (4) and outputs (5) with the different sets of ARRs that can be generated for the set of considered faults $f_j, j = 1, ..., n_f$ using a modified set of system equations (2) including the model of each fault:

Algorithm 1 Fault detection using ICSP

> **for** $k = 0$ to N **do**
> $\mathcal{V} \Leftarrow \{u_{k-L}, u_{k-L+1}, \cdots, u_k, y_{k-L}, y_{k-L+1}, \cdots, y_k, \theta_{k-L}, \theta_{k-L+1}, \cdots, \theta_k\}$
> $\mathcal{D} \Leftarrow \{\mathcal{U}_{k-L}, \mathcal{U}_{k-L+1}, \cdots, \mathcal{U}_k, \mathcal{Y}_{k-L}, \mathcal{Y}_{k-L+1}, \cdots, \mathcal{Y}_k, \Theta_{k-L}, \Theta_{k-L+1}, \cdots, \Theta_k\}$
> $\mathcal{C} \Leftarrow \{\Psi(\tilde{y}_k, \tilde{u}_k, \tilde{\theta}_k) = 0\}$
> $\mathcal{H}_k = (\mathcal{V}, \mathcal{D}, \mathcal{C})$
> $\mathcal{S}_k = \mathbf{solve}(\mathcal{H}_k)$
>   **if** $\mathcal{S}_k = \emptyset$ **then**
>   Exit (Fault detected)
>   **end if**
> **end for**

$$\mathcal{R}_{f_j} = \{r_i = \Psi_i(y_k, u_k, \theta_k, f_j) = 0,\ i = 1, \ldots, n_r\}. \tag{7}$$

Let $\mathcal{R}_{f_j}$ be the set of ARRs that is proved to be consistent with the sequence of inputs (4) and outputs (5). Then, f_j is the candidate fault. *Algorithm* 2 shows how the isolation procedure is implemented using ICSP.

Algorithm 2 Fault isolation using ICSP

> $k \Leftarrow$ fault detection time
> $\mathcal{V} \Leftarrow \{u_{k-L}, u_{k-L+1}, \cdots, u_k, y_{k-L}, y_{k-L+1}, \cdots, y_k, \theta_{k-L}, \theta_{k-L+1}, \cdots, \theta_k\}$
> $\mathcal{D} \Leftarrow \{\mathcal{U}_{k-L}, \mathcal{U}_{k-L+1}, \cdots, \mathcal{U}_k, \mathcal{Y}_{k-L}, \mathcal{Y}_{k-L+1}, \cdots, \mathcal{Y}_k, \Theta_{k-L}, \Theta_{k-L+1}, \cdots, \Theta_k\}$
> **for** $j = 1$ to n_f **do**
> $\mathcal{C}_j \Leftarrow \{\Psi_i(\tilde{y}_k, \tilde{u}_k, \tilde{\theta}_k, f_j) = 0\}$
> $\mathcal{H}_j = (\mathcal{V}, \mathcal{D}, \mathcal{C}_j)$
> $\mathcal{S}_j = \mathbf{solve}(\mathcal{H}_j)$
>   **if** $\mathcal{S}_j \neq \emptyset$ **then**
> f_j proposed as candidate fault
>   **end if**
> **end for**

4.2 Fault Isolation using Classical FDI Approach

The fault isolation procedure described in the previous subsection can be made more efficient according to the structural analysis approach to FDI (Blanke et al., 2006). In particular, each individual ARR in (3) is expected or not to be sensitive to a fault, characterizing the binary *Fault Signature Matrix (FSM)*, M. In this matrix, columns correspond to fault signatures and rows represent all possible ARRs R: $m_{ij} = 1$ means that whenever fault f_j occurs, the ARR $r_i \in \mathcal{R}$ is violated.

Given the set of ARRs (3), fault detection and isolation properties can be stated based on the information stored by matrix M. This matrix has the following properties (Blanke et al., 2006):

- *Detectability*: A set of faults are *detectable* if their effects on the system can be detected on the available set of ARRs using *Algorithm* 1. A fault f_k is detectable if at least there is a "*1*" present in the k^{th}-column of M.
- *Isolability*: A set of faults are (fully) *isolable* if their effects can be discriminated one of each other considering the available set of ARRs. Two faults, f_h and f_l, are isolable if the h^{th}-column and the l^{th}-column of M are different.

Taking into account these properties, the fault isolation procedure proceeds as follows. Let

$$\mathcal{R}_1 = \{r_{1,i} = \Psi_{1,i}(y_k, u_k, \theta_k) = 0,\ i = 1, \ldots, n_{1,r}\}$$
$$\mathcal{R}_0 = \{r_{0,i} = \Psi_{0,i}(y_k, u_k, \theta_k) \neq 0,\ i = 1, \ldots, n_{0,r}\}$$

be the set of consistent and inconsistent ARRs, respectively, at time instant k when some inconsistency in the set $\mathcal{R}$ in (3) is detected using *Algorithm* 1. The sets $\mathcal{R}_1(k)$ and $\mathcal{R}_0(k)$ determine the observed fault signature. The fault isolation candidates are given by the faults whose signature in M matches with the observed fault signature.

4.3 Fault Estimation

Once a fault f_j, with $j \in [1, \cdots, n_f]$, has been isolated, the fault size can be estimated by modifying the set-membership approach introduced in Definition 3.1 by including the model of the fault in the system model (2) assuming that is bounded by the set $\mathcal{F}_j$ defined by the interval box

$$\mathcal{F}_j = \left\{ f_j \mid \underline{f_j} \le f_j \le \bar{f}_j \right\}$$

as follows:

$$\begin{aligned}(\mathcal{U}_k, \mathcal{Y}_k, \Theta_k, \mathcal{F}_{j,k}) = \Big\{ & (u_k, y_k, \theta_k) \mid \exists\, (u_l \in \mathcal{U}_l)_{l=k-L}^{k}, \\ & (y_l \in \mathcal{Y}_l)_{l=k-L}^{k}, (\theta_l \in \Theta_l)_{l=k-L}^{k}, (f_{j,l} \in \mathcal{F}_{j,l})_{l=k-L}^{k} \\ & \textit{such that}\ \ \Psi_{f_j}(\tilde{y}_k, \tilde{u}_k, \tilde{\theta}_k, \tilde{f}_{j,k}) = 0 \Big\},\end{aligned}$$

where Ψ_{f_j} are the subsets of ARRs (3) which are affected by the fault f_j

Then, using algorithm *Algorithm* 3 with these new set of ARRs instead of the ones generated with non-faulty model, the fault size can be estimated.

Algorithm 3 Fault estimation using CSP for a given f_j

for $k = 0$ to N **do**
  $\mathcal{V} \Leftarrow \{u_{k-L}, \cdots, u_k, y_{k-L}, \cdots, y_k, \theta_{k-L}, \cdots, \theta_k, \cdots, f_{j,k-L}, \cdots, f_{j,k}\}$
  $\mathcal{D} \Leftarrow \{\mathcal{U}_{k-L}, \cdots, \mathcal{U}_k, \mathcal{Y}_{k-L}, \cdots, \mathcal{Y}_k, \Theta_{k-L}, \cdots, \Theta_k, \mathcal{F}_{j,k-L}, \cdots, \mathcal{F}_{j,k}\}$
  $\mathcal{C} \Leftarrow \{\Psi(\tilde{y}_k, \tilde{u}_k, \tilde{\theta_k}, \tilde{f}_{j,k}) = 0, \}$
  $\mathcal{H}_k = (\mathcal{V}, \mathcal{D}, \mathcal{C})$
  $\mathcal{S}_k = \mathbf{solve}(\mathcal{H}_k)$
  Estimated fault size $\mathcal{F}_k$
end for

5. CASE STUDY

The application example to show the effectiveness of the proposed approach is based on the quadruple-tank process proposed as a control benchmark by Johansson (2000).

5.1 System Description

A schematic diagram of the process is shown in Figure 1. This continuous-time non-linear process has been implemented and simulated in MATLAB/SIMULINK using the parameters described in Johansson (2000). The components of the system are tanks T_1, T_2, T_3 and T_4, by-pass valves V_1 and V_2, and pumps P_1 and P_2. The process inputs variables are u_1 and u_2 (input voltages to the pumps). Finally, the measured variables are the tanks levels h_1, h_2, h_3 and h_4.

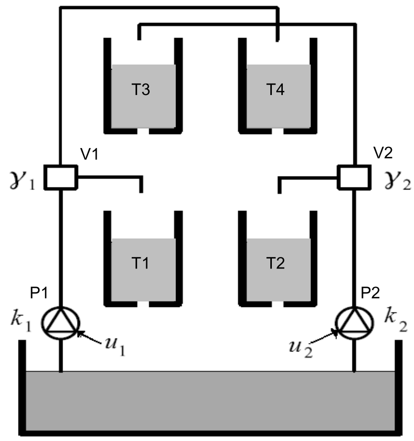

Fig. 1. Quadruple-tank process

From the continuous-time non linear equations presented in Johansson (2000), the following non-linear discrete model can be obtained by using the Euler discretization with a sampling time of 1s:

$$\begin{aligned} h_1(k+1) &= h_1(k) - \frac{a_1}{A_1}\sqrt{2gh_1(k)} \\ &\quad + \frac{a_3}{A_1}\sqrt{2gh_3(k)} + \frac{\gamma_1 k_1}{A_1} u_1(k), \\ h_2(k+1) &= h_2(k) - \frac{a_2}{A_2}\sqrt{2gh_2(k)} \\ &\quad + \frac{a_4}{A_2}\sqrt{2gh_4(k)} + \frac{\gamma_2 k_2}{A_2} u_2(k), \\ h_3(k+1) &= h_3(k) - \frac{a_3}{A_3}\sqrt{2gh_3(k)} \\ &\quad + \frac{(1-\gamma_2)k_2}{A_3} u_2(k), \\ h_4(k+1) &= h_4(k) - \frac{a_4}{A_4}\sqrt{2gh_4(k)} \\ &\quad + \frac{(1-\gamma_1)k_1}{A_4} u_1(k), \end{aligned} \tag{8}$$

where $A_1 = A_3 = 28$ cm^2, $A_2 = A_4 = 32$ cm^2, $a_1 = a_3 = 0.071$ cm^2, $a_2 = a_4 = 0.057$ cm^2 and $g = 981$ cm/s^2. Parameters k_1, k_2, γ_1 and γ_2 are assumed to belong to the intervals $k_1 \in [3.14, 3.33]$, $k_2 \in [3.25, 3.29]$, $\gamma_1 \in [0.43, 0.70]$ and $\gamma_2 \in [0.34, 0.60]$. The objective of by-pass valve, V_1, is to derive a proportional part, γ_1, of the pump flow, Qp_1, to T_1 and the other part, $1-\gamma_1$, to T_4. Similarly, flow Qp_2 is distributed to T_2 and T_3 in function of γ_2 value.

In the simulation benchmark three different types of faults have been considered:

- Abrupt faults f_1 and f_2 in valves V_1 and V_2, respectively. These abrupt faults have been simulated considering $\gamma_1 = 0$ or $\gamma_2 = 0$.
- Abrupt faults f_3 and f_4 in pumps P_1 and P_2, respectively. In these fault situations, pumps reduce a 40% their flow capacity.
- Incipient faults f_5, f_6, f_7 and f_8 have been considered in level sensors h_1, h_2, h_3 and h_4, respectively. Faults are

Table 1. Fault signature matrix (FSM)

ARR	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
r_1	1	0	1	0	1	0	1	0
r_2	0	1	0	1	0	1	0	1
r_3	0	1	0	1	0	0	1	0
r_4	1	0	1	0	0	0	0	1

implemented by introducing a slow derivation of their real value.

5.2 ARRs and Faults

From system constraints given by (8), four ARRs $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$ can be derived by using structural analysis (Blanke et al., 2006)

$$\begin{aligned} r_1 &= \Psi_1(\{\tilde{y}_1, \tilde{y}_3\}, \tilde{u}_1, \{\tilde{\gamma}_1, \tilde{k}_1\}), \\ r_2 &= \Psi_2(\{\tilde{y}_2, \tilde{y}_4\}, \tilde{u}_2, \{\tilde{\gamma}_2, \tilde{k}_2\}), \\ r_3 &= \Psi_3(\tilde{y}_3, \tilde{u}_2, \{\tilde{\gamma}_2, \tilde{k}_2\}, \\ r_4 &= \Psi_4(\tilde{y}_4, \tilde{u}_1, \{\tilde{\gamma}_1, k_1\}), \end{aligned} \tag{9}$$

considering that the set of known variables is:

- inputs: $u = \{u_1, u_2\}$,
- outputs $y = \{y_1, y_2, y_3, y_4\}$ (measured outputs y_i correspond to the voltage of level measurement devices),
- and parameters $\theta = \{\gamma_1, \gamma_2, k_1, k_2\}$.

The FSM associated to this set of ARR is shown in Table 1, where in columns appear the different faults according to the set of fault scenarios considered: faults in valves (f_1, f_2), pumps (f_3, f_4) and level sensors (f_5, f_6, f_7 and f_8).

5.3 Non-faulty Scenario

First, consistency checking results obtained using *Algorithm 1* in a scenario without faults are presented. The initial variables domains are given by: $y_i \in [0, 20]$ for $i = 1, 2, 3, 4$, $\theta_1 \in [0.43, 0.70]$, $\theta_2 \in [0.34, 0.60]$, $\theta_3 \in [3.14, 3.33]$ and $\theta_4 \in [3.29, 3.35]$.

Figure 2 shows in red the predicted output domain, $\mathcal{Y}_i$, of each level h_i using the set of ARR (3) for a given u_1 and u_2. On the other hand, in blue, it is plotted the level measured value y_i plus the measurement uncertainty bound. The consistency is obtained since the intersection is not empty.

5.4 Faulty Scenarios

In the case of a fault scenario, the set of ARRs (9) becomes inconsistent. Figure 3 shows the results of the *Algorithm 1* when a f_1 appears. In this figure, it is plotted in red the predicted value of each $\mathcal{Y}_i$, in black the measured output of h_i in blue and the measured output y_i plus its uncertainty. The fault is introduced at time instant 450 (in samples). Notice that at time instant 451 there is an inconsistency because *Algorithm* 1 provides $\mathcal{Y}_1 = \emptyset$ and $\mathcal{Y}_4 = \emptyset$, that corresponds to the observed fault signature $\{1, 0, 0, 1\}$. Then, according to the theoretical FSM in Table 1, the potential faults are: f_1 or f_3.

These two faults have been implemented in the model as a parametric fault. Fault f_1 changes the domain of the constant valve γ_1 by $\gamma_1 \pm f_{V1}$. In a similar way, fault f_3 changes the

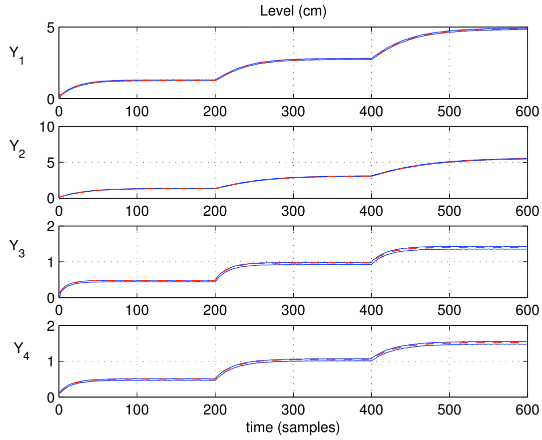

Fig. 2. Temporal evolution of $\mathcal{Y}$ (blue) and measured output variable with uncertainty (red)

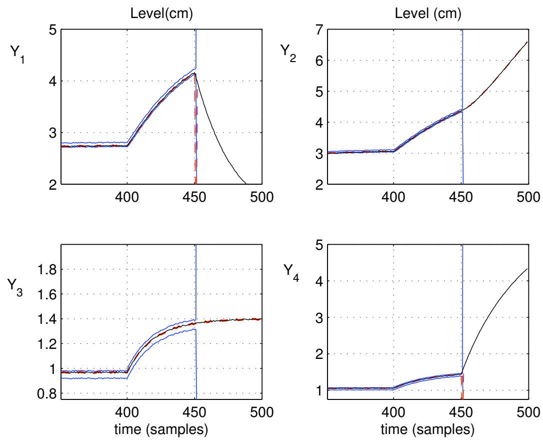

Fig. 3. Temporal evolution of $\mathcal{Y}$ (blue), measured output (black) and measured output with uncertainty (red)

constant pump p_1 by $p_1 \pm f_{P1}$. Taking into account these potential faults, the following new ARRs

$$\begin{aligned} r_{f_{V1,1}} &= \Psi_{f_{V1,1}}(\{\tilde{y}_1, \tilde{y}_3\}, \tilde{u}_1, \{\tilde{\gamma}_1, \tilde{k}_1\}, \tilde{f}_{V1}), \\ r_{f_{V1,4}} &= \Psi_{f_{V1,4}}(\tilde{y}_4, \tilde{u}_1, \{\tilde{\gamma}_1, \tilde{k}_1\}, \tilde{f}_{V1}), \\ r_{f_{P1,1}} &= \Psi_{f_{P1,1}}(\{\tilde{y}_1, \tilde{y}_3\}, \tilde{u}_1, \{\tilde{\gamma}_1, \tilde{k}_1\}, \tilde{f}_{P1}), \\ r_{f_{P1,4}} &= \Psi_{f_{P1,4}}(\tilde{y}_4, \tilde{u}_1, \{\tilde{\gamma}_1, \tilde{k}_1\}, \tilde{f}_{P1}), \end{aligned} \tag{10}$$

have been introduced in the ICSP problem in order to apply *Algorithm 3* [1].

Using these new ARRs in the *Algorithm 3*, the consistency is reached when the fault domain of f_1 is $\mathcal{F}_1 = [0, 0.7]$. Figure 4 shows the $\mathcal{Y}_i$ time evolution and $\mathcal{F}_1$ estimation, respectively. Figure 5 shows the output variables domain evolution and the fault estimation, $\mathcal{F}_4$, when the fault f_4 is considered. In this scenario, the observed fault signature is $\{0, 1, 1, 0\}$, giving as candidate faults: f_2 or f_4. The consistency is reached when the estimated fault domain is $\mathcal{F}_4 = [0, 3]$.

Finally, a scenario with a fault in level sensor y_1 is presented. As described in *Section* 5.1, an incipient fault in this sensor

[1] In the set of ARRs (10) for simplicity the discrete-time instant has been omitted

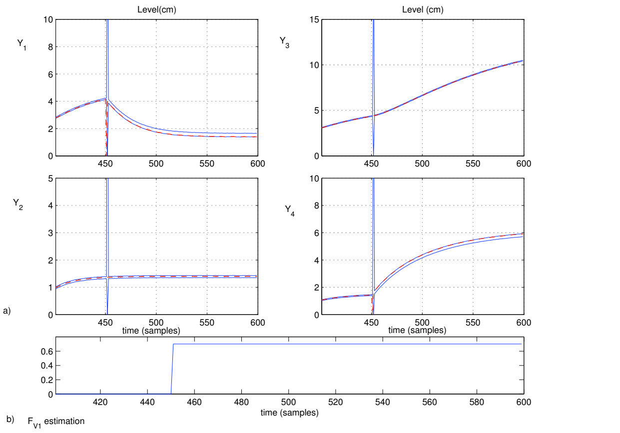

Fig. 4. a) Temporal evolution of $\mathcal{Y}$ (blue) and y (red); b) $\mathcal{F}_{V1}$ estimation

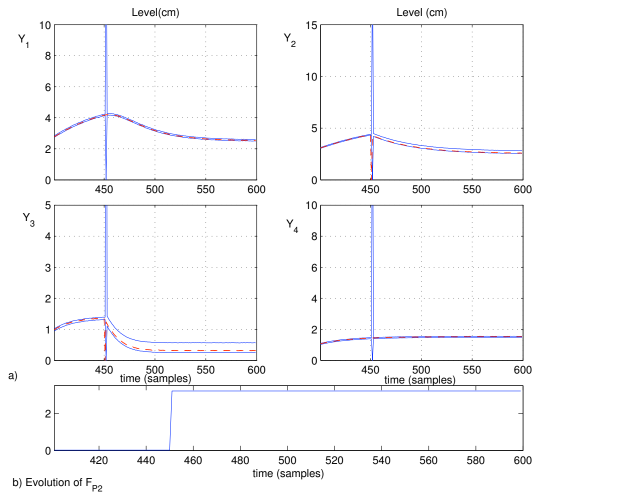

Fig. 5. a) Temporal evolution of $\mathcal{Y}$ (blue) and y (red); b) $\mathcal{F}_{P2}$ estimation

has been introduced starting at sample 500 and producing a constant decrement in the measured value. This fault has been modeled as an additive fault modifying the ARR Ψ_1 with a new parameter f_{y1}. Figure 6a and 6b show, respectively, the output variables domain evolution and the fault estimation, f_{y1}, the detection time is at time instant 581, with a delay of 81 samples. Observed fault signature is $\{1, 0, 0, 0\}$, what according to the theoretical FSM matrix, the fault candidate is: f_5, which corresponds to the fault in level sensor y_1.

6. CONCLUSIONS

In this paper, the robust fault diagnosis problem for non-linear systems considering both bounded parametric modelling errors and noises has been addressed using constraints satisfaction. Combining the available measurements with the model of the monitored system, a set of ARRs, relating only known variables, were derived. These relations were used in the fault diagnosis procedure to check the consistency between the observed and the predicted process behavior. When some inconsistency is detected, the fault isolation algorithm will be activated in order to provide an explanation of the possible cause. Finally, a fault estimation procedure was used to estimate the fault magnitude.

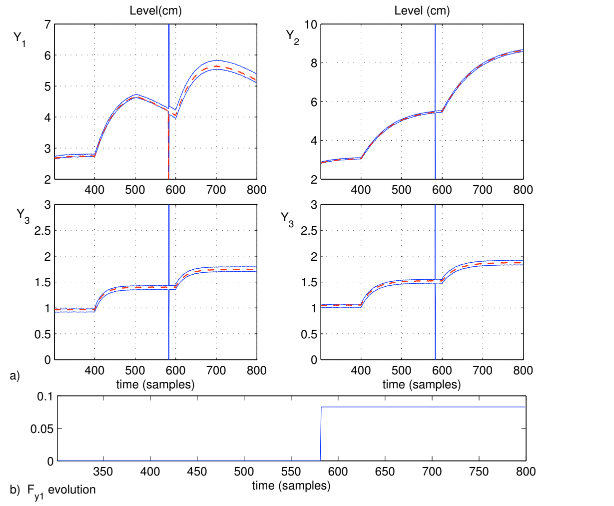

Fig. 6. a) Temporal evolution of $\mathcal{Y}$ (blue) and y (red); b) $\mathcal{F}_{y1}$ estimation

The usefulness of the proposed approach has been illustrated using a case study based on a four-tanks system.

ACKNOWLEDGMENTS

This work has been supported by the CICYT Ref. DPI2006-11944 of the Spanish Science and Technology Ministry, the *Juan de la Cierva* Research Programme (ref. JCI-2008-2438), and the DGR of Generalitat de Catalunya (SAC group Ref. 2005/SGR/00537).

REFERENCES

X. Baguenard. Personal homepage. http://www.istia.univ-angers.fr/∼baguenar/, February 2005.

M. Basseville and I.V. Nikiforov. *Detection of abrupt changes: theory and applications*. Prentice Hall, 1993.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg, 2006. ISBN 2nd Edition.

J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998.

E. Hyvönen. Constraint reasoning based on interval arithmetic: The tolerance approach. *Artificial Intelligence*, 58:71–112, 1992.

L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

K.H. Johansson. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8(3):456–465, 2000.

M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter. *Bounding Approaches to System Identification*. Plenum Press, 1996.

V. Puig, J. Quevedo, T. Escobet, and S. De las Heras. Robust fault detection approaches using interval models. *Proceedings of IFAC World Congress*, 2002.

M. Witczak, J. Korbicz, and R. Patton. A bounder-error approach to designing unknowninput observers. *Proceedings of IFAC World Congress*, 2002.