SQualTrack: A Tool for Robust Fault Detection

Joaquim Armengol, Josep Vehí, *Member, IEEE*, Miguel Ángel Sainz, Pau Herrero, and Esteban R. Gelso

***Abstract*—One of the techniques used to detect faults in dynamic systems is analytical redundancy. An important difficulty in applying this technique to real systems is dealing with the uncertainties associated with the system itself and with the measurements. In this paper, this uncertainty is taken into account by the use of intervals for the parameters of the model and for the measurements. The method that is proposed in this paper checks the consistency between the system's behavior, obtained from the measurements, and the model's behavior; if they are inconsistent, then there is a fault. The problem of detecting faults is stated as a quantified real constraint satisfaction problem, which can be solved using the modal interval analysis (MIA). MIA is used because it provides powerful tools to extend the calculations over real functions to intervals. To improve the results of the detection of the faults, the simultaneous use of several sliding time windows is proposed. The result of implementing this method is SemiQUALitative TRACKing (SQualTrack), a fault-detection tool that is robust in the sense that it does not generate false alarms, i.e., if there are false alarms, they indicate either that the interval model does not represent the system adequately or that the interval measurements do not represent the true values of the variables adequately. SQualTrack is currently being used to detect faults in real processes. Some of these applications using real data have been developed within the European project Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes and are also described in this paper.**

***Index Terms*—Constraint satisfaction problem, fault detection, modal intervals, processes, redundancy, uncertain dynamic systems.**

I. Introduction

A FAULT is a malfunction in a system, which may have consequences such as economic losses derived from lower efficiency of the system or danger to the people or to the environment. Many different techniques have been developed in recent years, which 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. Two research communities work on model-based techniques, namely, the fault-detection and isolation (FDI) community, formed by researchers with a background in control systems engineering, and the principles of diagnosis (DX) community, formed by researchers with a background in computer science and intelligent systems. The collaboration between these two communities to develop more powerful tools for fault detection and diagnosis has been one of the goals of the European Network of Excellence on Model Based Systems and Qualitative Reasoning [3], particularly of the Bridge task group.

Among the techniques developed by the FDI research community, there are classical methods such as state observers, parity equations, and parameter estimation [4]–[7].

One of the methods to detect faults consists in comparing the behavior of an actual system with the behavior of a model of the system. This principle is called *analytical redundancy* and is presented in Section II. Many techniques based on this principle either do not take uncertainty into account or have difficulties in considering uncertainty. In both cases, uncertainty is a source of false alarms, i.e., due to uncertainty, alarms indicating the presence of faults are activated when there are no faults. In this paper, the uncertainties associated with the system itself and with the measurements are taken into account by means of intervals, thus eliminating this source of false alarms.

In Section III, the problem of detecting faults is stated as a quantified real constraint satisfaction problem, which can be solved using the modal interval analysis (MIA) [8], [9]. The sliding time windows, which are introduced in the same section, reduce the necessary computational effort and improve the fault-detection results. Finally, the proposed fault-detection algorithm is presented.

This method is implemented in SemiQUALitative TRACKing (SQualTrack), a software package to robustly detect faults in uncertain systems, which is described in Section IV and which has been applied in the European project Advanced Decision Support System (DSS) for Chemical/Petrochemical Manufacturing Processes (CHEM) [10] to detect faults in real processes. Some of the results are shown in Sections V–VII. Finally, Section VIII summarizes this paper and provides some conclusions.

II. Analytical Redundancy

An actual system or a part of it can be represented by a model described by the following nonlinear discrete-time equation:

$$\boldsymbol{y}(k)=\boldsymbol{f}\left(\boldsymbol{y}(k-1),\ldots,\boldsymbol{y}(k-n),\boldsymbol{u}(k-1),\ldots,\boldsymbol{u}(k-m),\boldsymbol{p}\right) \tag{1}$$

where $\boldsymbol{y}(k)\in\mathbb{R}^{n_y},\ldots,\boldsymbol{y}(k-n)\in\mathbb{R}^{n_y}$ represents the outputs of the system at instants $k,\ldots,k-n$; $\boldsymbol{f}$ is a vector of continuous functions; $\boldsymbol{u}(k-1)\in\mathbb{R}^{n_u},\ldots,\boldsymbol{u}(k-m)\in\mathbb{R}^{n_u}$

Manuscript received October 2, 2007; revised February 15, 2008, May 22, 2008 and August 13, 2008. Current version published March 19, 2009. This work was supported in part by the Spanish Government (Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica, Ministerio de Educación y Ciencia) under Coordinated Research Project DPI2006-15476-C02-02 and in part by the government of Catalonia under Grant SGR00296. This paper was recommended by Associate Editor P. Shi.

J. Armengol, J. Vehí, M. Á. Sainz, and E. R. Gelso are with the Institut d'Informàtica i Aplicacions, Universitat de Girona, 17071 Girona, Spain.

P. Herrero is with the Hospital de la Santa Creu i Sant Pau, 08025 Barcelona, Spain.

Digital Object Identifier 10.1109/TSMCB.2008.2006909

1083-4419/$25.00 © 2008 IEEE

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

represents the inputs at instants $k-1,\ldots,k-m$; and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is a vector of parameters.

An analytical redundancy relation (ARR) is an algebraic constraint deduced from the system model which contains only measured variables. An ARR for (1) is

$$\tilde{\boldsymbol{y}}(k) = \hat{\boldsymbol{y}}(k) \tag{2}$$

where $\tilde{\boldsymbol{y}}(k)$ is the measured output of the system at instant k and $\hat{\boldsymbol{y}}(k)$ is the analytical output of the model at instant k

$$\hat{\boldsymbol{y}}(k) := \boldsymbol{f}\left(\tilde{\boldsymbol{y}}(k-1),\ldots,\tilde{\boldsymbol{y}}(k-n),\tilde{\boldsymbol{u}}(k-1),\ldots,\tilde{\boldsymbol{u}}(k-m),\boldsymbol{p}\right) \tag{3}$$

where $\tilde{\boldsymbol{u}}(k)$ is the measurement of the input at instant k.

An ARR is used to check the consistency of the observations with respect to the system model. Therefore, a fault is detected when

$$\tilde{\boldsymbol{y}}(k) \neq \hat{\boldsymbol{y}}(k) \tag{4}$$

or, equivalently

$$\boldsymbol{r}(k) \neq \boldsymbol{0} \tag{5}$$

where

$$\boldsymbol{r}(k) = \hat{\boldsymbol{y}}(k) - \tilde{\boldsymbol{y}}(k) \tag{6}$$

is the so-called *residual* of the ARR.

The main problem is that the measured output $\tilde{\boldsymbol{y}}(k)$ and the computed output $\hat{\boldsymbol{y}}(k)$ are seldom the same because the model is, by definition, inaccurate, i.e., it is an approximated representation of the system. This is the consequence of the uncertainties of the system and the procedure of systems' modeling. Moreover, the measurements $\tilde{\boldsymbol{y}}(k)$ and $\tilde{\boldsymbol{u}}(k)$ are approximations to the real values of the variables $\boldsymbol{y}(k)$ and $\boldsymbol{u}(k)$, respectively, due to uncertainties in sensors, namely, noise, errors in the analog-to-digital conversion, bias, drift, nonlinearities, inaccuracies due to calibration, etc.

Therefore, the uncertainty has to be considered. In some methods, the uncertainty is taken into account when the behavior of the actual system is compared with the behavior of the model; hence, a fault is indicated when the difference is larger than a threshold ϵ

$$\|\boldsymbol{r}(k)\| > \boldsymbol{\epsilon}. \tag{7}$$

An important difficulty is to determine the size of ϵ. If it is too small, faults are indicated even when none exists, i.e., there are false alarms. On the other side, if the threshold is too large, the amount of missed alarms or undetected faults increases. Hence, there is a need to develop robust fault-diagnosis algorithms.

The robustness problem in FDI is thus defined as the maximization of the detectability and isolability of faults, together with the minimization of the effect of uncertainty and disturbances, on the FDI procedure [11]. Two main approaches have been proposed to solve the robustness problem. One, based on an attempt to account for uncertainty in designing the residual, is known as *active robustness* in FDI. The second approach is called *passive robustness* in FDI and consists in enhancing the robustness of the fault-detection procedure at the decision-making stage [5]. In this latter case, robustness can be achieved by finding and using the most effective threshold. In this paper, effective thresholds are found using intervals to bound the uncertainty in parameters and measurements [12]–[15].

III. Consistency Test

Applying the principle of analytical redundancy but taking into account the uncertainty of measurements and parameters, a fault is detected when

$$\left(\forall \tilde{\boldsymbol{y}}(k) \in \tilde{\boldsymbol{Y}}(k)\right)\left(\forall \tilde{\boldsymbol{y}}(k) \in \hat{\boldsymbol{Y}}(k)\right) \quad \boldsymbol{r}(k) \neq \boldsymbol{0} \tag{8}$$

where $\tilde{\boldsymbol{Y}}(k)$ is the range of possible values that the output $\boldsymbol{y}(k)$ can take (it is obtained from the measurement $\tilde{\boldsymbol{y}}(k)$ and from the properties of the sensor) and $\hat{\boldsymbol{Y}}(k)$ is the range of possible values of the output of the model, which is an interval because the output of the model depends on interval inputs and parameters.

Therefore, at each time step, the range of a function in a parameter space has to be computed. Range computation is a task related to global optimization [16], [17], which usually requires considerable computational effort. The computational cost can be lowered by computing this range in an iterative way so that at each iteration, an outer approximation, closer to the exact range than the one obtained in the previous iteration, is obtained. This iterative procedure can stop when (8) is fulfilled because the fault has already been detected.

The problem is that this iterative procedure never stops either when there is no fault or when the fault cannot be detected using this method, for instance, because there is a fault but

$$\left(\exists \tilde{\boldsymbol{y}}(k) \in \tilde{\boldsymbol{Y}}(k)\right)\left(\exists \hat{\boldsymbol{y}}(k) \in \hat{\boldsymbol{Y}}(k)\right) \quad \boldsymbol{r}(k) = \boldsymbol{0} \tag{9}$$

due to dynamics.

A way to stop this procedure in this case is by computing also an inner approximation to $\hat{\boldsymbol{Y}}(k)$. This is described in the following.

Both approximations, the outer and the inner one, can be computed using MIA. Considering the modal interval $*$-extensions $\boldsymbol{r}^*(k)$ of the residual functions $\boldsymbol{r}(k)$ to the proper intervals $\hat{\boldsymbol{Y}}(k)$ and $\tilde{\boldsymbol{Y}}(k)$, (8) is equivalent, in accordance with the semantic theorems, to the interval relation

$$[0,0] \not\subseteq \boldsymbol{r}^*(k) \tag{10}$$

and (9) is equivalent to the interval relation

$$[0,0] \subseteq \boldsymbol{r}^*(k). \tag{11}$$

Remark that, in this case, $\boldsymbol{r}^*(k)$ is the range of $\boldsymbol{r}(k)$ for the domains $\hat{\boldsymbol{Y}}(k)$ and $\tilde{\boldsymbol{Y}}(k)$. Using the necessary inner and outer

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

roundings of $\boldsymbol{r}^*(k)$, (8) is true if

$$[0,0] \not\subseteq \text{Outer}\left(\boldsymbol{r}^*(k)\right) \tag{12}$$

and (9) is true if

$$[0,0] \subseteq \text{Inner}\left(\boldsymbol{r}^*(k)\right). \tag{13}$$

In this paper, the f^* algorithm, which will be described in Section IV-C, is used to compute these approximations in an iterative way. The execution is stopped either when (12) is satisfied, hence a fault has been detected, or when (13) is true, in which case the ARR is consistent and a faulty or healthy behavior cannot be assured.

A. Window Consistency

Consider a discrete first-order model

$$y(k) = a\, y(k-1) + b\, u(k-1) \tag{14}$$

where a and b are uncertain model parameters such that a and b belong to the intervals A and B, respectively.

At step k, the range of $\hat{\boldsymbol{Y}}(k)$ has to be computed. Moreover, at step $k+1$, the range to be computed is $\hat{\boldsymbol{Y}}(k+1)$, where

$$y(k+1) = a\, y(k) + b\, u(k). \tag{15}$$

Two situations must be distinguished in this case. A first case is when the system is considered to be time variant, in which the values of a and b at steps k and $k+1$ can be considered different.

1) $a_k \neq a_{k+1}$.
2) $b_k \neq b_{k+1}$.

In this case, the ranges of $\hat{\boldsymbol{Y}}(k)$ and $\hat{\boldsymbol{Y}}(k+1)$ can be computed independently.

A second case is when the system is considered time invariant, i.e., there is uncertainty in the values of a and b but it is known that these values are the same at steps k and $k+1$. In this case, there is a dependence between (14) and (15), which is made explicit by substituting $y(k)$ into $y(k+1)$

$$y(k+1) = a\left(a\, y(k-1) + b\, u(k-1)\right) + b\, u(k). \tag{16}$$

In order to distinguish $y(k+1)$ of (16) from $y(k+1)$ of (15), the concept of *window length* (w) is introduced

$$y(k+1|k-1) = a\left(a\, y(k-1) + b\, u(k-1)\right) + b\, u(k) \tag{17}$$

or, in general

$$\boldsymbol{y}(k|k-w) = \boldsymbol{f}_w\left(\boldsymbol{y}(k-w), \boldsymbol{u}(k-w), \ldots, \boldsymbol{u}(k-1), \boldsymbol{p}\right) \tag{18}$$

which constitutes a *sliding time window*.

This second case, time invariant systems, is more difficult to handle than the first one and is the case considered in this paper.

The window consistency allows one to determine the consistency of a set of system measurements (inputs and output) between two sampling times with respect to the predicted behavior in the same time interval. The distance between the two considered sampling times is called the window length. Then, for a window length w, in accordance with (8), a fault is detected when

$$\begin{aligned}&\left(\forall\tilde{\boldsymbol{y}}(k)\in\tilde{\boldsymbol{Y}}(k)\right)\left(\forall\hat{\boldsymbol{y}}(k|k-w)\in\hat{\boldsymbol{Y}}(k|k-w)\right)\\&\left(\forall\tilde{\boldsymbol{u}}(k-w)\in\tilde{\boldsymbol{U}}(k-w)\right),\ldots,\\&\left(\forall\tilde{\boldsymbol{u}}(k-1)\in\tilde{\boldsymbol{U}}(k-1)\right)(\forall\boldsymbol{p}\in\boldsymbol{P})\boldsymbol{r}_w(k)\neq\boldsymbol{0}\end{aligned} \tag{19}$$

where $\tilde{\boldsymbol{U}}(k)$ is defined similar to $\tilde{\boldsymbol{Y}}(k)$ and is the range of possible values that the input $\boldsymbol{u}(k)$ can take (it is obtained from the measurement $\tilde{\boldsymbol{u}}(k)$ and from the properties of the sensor)

$$\boldsymbol{r}_w(k) = \hat{\boldsymbol{y}}(k|k-w) - \tilde{\boldsymbol{y}}(k) \tag{20}$$

$$\hat{\boldsymbol{y}}(k|k-w) = \boldsymbol{f}_w\left(\tilde{\boldsymbol{y}}(k-w), \tilde{\boldsymbol{u}}(k-w), \ldots, \tilde{\boldsymbol{u}}(k-1), \boldsymbol{p}\right) \tag{21}$$

where (21) is the corresponding predicted behavior for a window length of w.

Notice that for two different window lengths w_1 and w_2, it can happen that with w_1, the fault is not detected but that with w_2, the fault is detected. In this case, it can be assured that there is a fault because detecting it with one window length is a sufficient condition to assure it. Consequently, a fault is detected if

$$\begin{aligned}&\left(\forall\tilde{\boldsymbol{y}}(k)\in\tilde{\boldsymbol{Y}}(k)\right)\left(\forall\tilde{\boldsymbol{y}}(k|k-w_1)\in\tilde{\boldsymbol{Y}}(k|k-w_1)\right)\\&\left(\forall\tilde{\boldsymbol{u}}(k-w_1)\in\tilde{\boldsymbol{U}}(k-w_1)\right),\ldots,\\&\left(\forall\tilde{\boldsymbol{u}}(k-1)\in\tilde{\boldsymbol{U}}(k-1)\right)(\forall\boldsymbol{p}\in\boldsymbol{P})\boldsymbol{r}_{w_1}(k)\neq\boldsymbol{0}\quad\vee,\ldots,\vee\\&\left(\forall\tilde{\boldsymbol{y}}(k)\in\tilde{\boldsymbol{Y}}(k)\right)\left(\forall\hat{\boldsymbol{y}}(k|k-w_n)\in\hat{\boldsymbol{Y}}(k|k-w_n)\right)\\&\left(\forall\tilde{\boldsymbol{u}}(k-w_n)\in\tilde{\boldsymbol{U}}(k-w_n)\right),\ldots,\\&\left(\forall\tilde{\boldsymbol{u}}(k-1)\in\tilde{\boldsymbol{U}}(k-1)\right)(\forall\boldsymbol{p}\in\boldsymbol{P})\boldsymbol{r}_{w_n}(k)\neq\boldsymbol{0}.\end{aligned} \tag{22}$$

The fault-detection results obtained using several window lengths are generally better, i.e., there are less missed alarms, than the ones obtained using a single window length, whatever is the length in the latter case. The best window length not only depends on the dynamics of the system but also on the type of fault to be detected.

B. Fault-Detection Algorithm

The proposed fault-detection algorithm requires the following from the user: a process model, the process data, the uncertainty associated to the process model and to the process data, the window lengths $\{w_1, \ldots, w_n\}$, and a time (TimeOut) to limit the computations carried out between two sample times.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

Require: Process model, process data, uncertainties, window lengths $\{w_1, \ldots w_n\}$ and TimeOut
Ensure: Faulty
1: Faulty:=Perhaps
2: Build the ARRs for each window length $\{\boldsymbol{r}_{w_1} = 0, \ldots \boldsymbol{r}_{w_n} = 0\}$
3: **while** Available process data **do**
4: Read process data and assign it to the corresponding interval ARRs
5: **for** i=1 to i=n **do**
6: Approximate $\boldsymbol{r}_{w_i}$ till $[0,0] \nsubseteq Outer(\boldsymbol{r}_{w_i})$ or $[0,0] \subseteq Inner(\boldsymbol{r}_{w_i})$ or TimeOut reached
7: **if** $[0,0] \nsubseteq Outer(\boldsymbol{r}_{w_i})$ **then**
8: Faulty:=true
9: Break
10: **end if**
11: **end for**
12: **end while**
13: **return** Faulty

Fig. 1. Fault-detection algorithm.

First of all, the algorithm builds the corresponding ARRs for each window length $\{\boldsymbol{r}_{w_1} = 0, \ldots, \boldsymbol{r}_{w_n} = 0\}$. At each step, internal and external approximations to the range of each ARR have to be computed.

As the necessary computing effort to deal with a larger window length w_n is bigger than for a shorter one w_1, when $w_n > w_1$, the algorithm starts, at each time point, using the shortest window length. If it detects a fault, the algorithm returns "Faulty" and stops, thus saving computing effort. If this window does not detect fault, the algorithm proceeds with the next one and so on. If none of the ARRs is proven to be inconsistent, the algorithm returns "Perhaps."

This algorithm, which is shown in Fig. 1, also does not report a fault when it is inconclusive because the timeout is reached. This is because the proposed approach prioritizes to avoid false alarms to missed alarms.

IV. SQualTrack

This technique for fault detection has been implemented as a software tool called SQualTrack.

The inputs of this tool are an Input File describing the model of the system to be analyzed and input data coming from the process.

The input data are the values of the variables at each sampling time collected by the sensors. These values are precise but inaccurate, as the uncertainties of the measurements are not considered. These values are converted to intervals, which are imprecise but should be accurate, by using information coming from the manufacturer of the sensor and from the operators of the process.

The modeling procedure usually involves approximations, simplifications, linearizations, etc. These uncertainties are taken into account by giving interval values to the parameters of the model.

The model, the interval values of its parameters, and other parameters (such as the vector of window lengths) are introduced in the Input File.

SQualTrack processes the Input File and the input data and provides numerical and graphical results. Different modules can be distinguished in the implementation of SQualTrack:

1) a parser;
2) a symbolic algorithm, which builds the analytical redundancy relations (ARRs) for each window length;
3) a communication process interface (CPI) which can read input data from a text file or from a real process;
4) an algorithm to compute approximations of the semantic extensions of a continuous function (f^*);
5) the fault-detection algorithm sketched in Section III-B.

These modules are described in the following and shown in Fig. 2.

SQualTrack has been completely developed in C++ to optimize the numerous computations that have to be performed.

A. Parser

In order to provide SQualTrack with a friendly user interface, which allows one to introduce the problems in an easy way, a parser has been developed.

In computer science, parsing (formally named syntax analysis) is the process of analyzing an input sequence (read from a file or a keyboard, for example) in order to determine its grammatical structure with respect to a given formal grammar. A parser is a computer program that carries out this task.

In SQualTrack, the parser interprets the Input File, where the model of the system is described and the interval values of the parameters of the model are introduced. It is based on the Spirit framework [18].

B. ARR Construction

The output of the parser is a binary tree corresponding to the model of the system with a window having a length of one. Then, it is necessary to automatically generate the binary tree for the ARRs corresponding to each one of the introduced window lengths. This task is carried out through the manipulation of the binary tree obtained with the parser. The implemented algorithm crosses over the binary tree and looks for the different occurrences of the output variable. Then, it recursively substitutes these occurrences by the same binary tree but with the corresponding indexes for the new introduced variables and occurrences. For example, given the following generic first-order model

$$y(k) = a\, y(k-1) + b\, u(k-1) \tag{23}$$

where $y(k)$ and $y(k-1)$ are the output variable at time instants k and $k-1$, respectively, $u(k-1)$ is the input variable, and a and b are model parameters, the graphical representation of the binary tree corresponding to the right-hand-side part of the model equation is shown in Fig. 3, where the subindexes represent the occurrences of the variables. Then, the corresponding binary tree for a window having a length of four is shown in Fig. 4.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

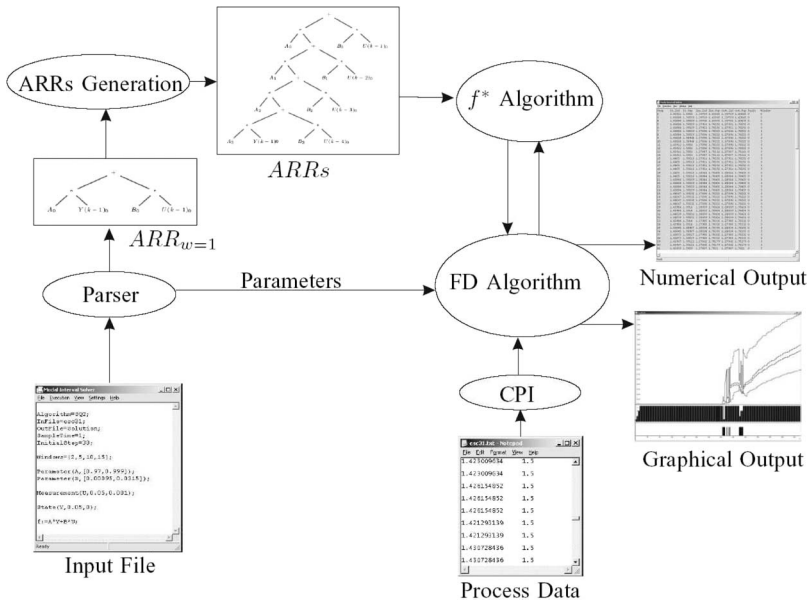

Fig. 2. SQualTrack modules.

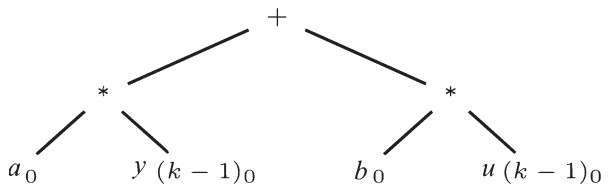

Fig. 3. First-order model representation.

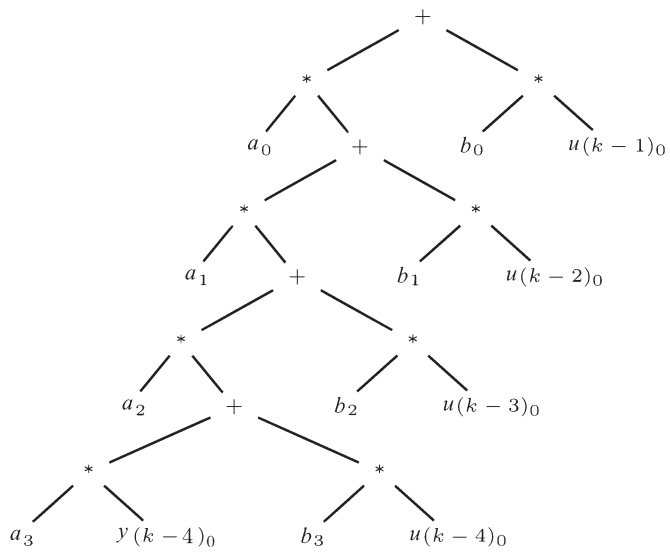

Fig. 4. First-order model representation for a window length of four.

C. f^* Algorithm

Computing the range of a continuous function involving interval variables is a difficult task due to the overestimation of the interval evaluation caused by the multiincidences of variables in a certain condition of monotony.

One way to deal with this problem is combining MIA and branch-and-bound techniques [16]. Even if classical interval analysis (IA) could be used for such a purpose, MIA performs better, since it provides a more powerful mathematical framework for computing, in an efficient way, an inner and an outer approximation to the range of a continuous function.

MIA (for a complete introduction, see [8] and [19]) is a completion of the IA, not only in a lattice and an arithmetic sense, as the extended intervals of Kaucher [20], but a logical completion as well. This way, it provides tools to verify logical formulas such as the ones in (8) and (9) and, therefore, to tackle problems which can be stated by means of similar logical formulas.

The starting point is to associate a logical quantifier to a classic interval. Thus, a modal interval X is defined as a couple $X = (X', \forall)$ or $X = (X', \exists)$, where X' is its classic interval *domain*, $X' \in I(\mathbb{R})$, and the quantifiers $\forall$ and $\exists$ are a selection *modality*. The set of the modal intervals is represented by $I^*(\mathbb{R})$. The modal intervals of the type $X = (X', \exists)$ are called *proper intervals* or *existential intervals*, and the intervals of the type $X = (X', \forall)$ are called *improper* or *universal intervals*. A modal interval can be represented using its canonical coordinates in the form

$$X = [a, b] = \begin{cases} ([a, b]', \exists), & \text{if } a \leq b \\ ([b, a]', \forall), & \text{if } a \geq b. \end{cases} \tag{24}$$

For example, the interval [2, 5] is equal to $([2, 5]', \exists)$, and the interval [8, 4] is equal to $([4, 8]', \forall)$.

Lattice operations (meet and join), arithmetic operators, and modal interval extensions f^* and f^{**} for a continuous function f, which are a generalization of the range of f over a classic interval, can be defined over the modal intervals.

Two key results, named semantic theorems, give logical interpretation to these semantic extensions.

Theorem 4.1 ($*$—Semantic Theorem): Let be $X \in I^*(\mathbb{R}^n)$ and $Z \in I^*(\mathbb{R})$; then

$$f^*(X) \subseteq Z \Leftrightarrow \forall \left(x_p, X_p'\right)\ Q(z, Z)\ \exists \left(x_i, X_i'\right)\ z = f(x_p, x_i). \tag{25}$$

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

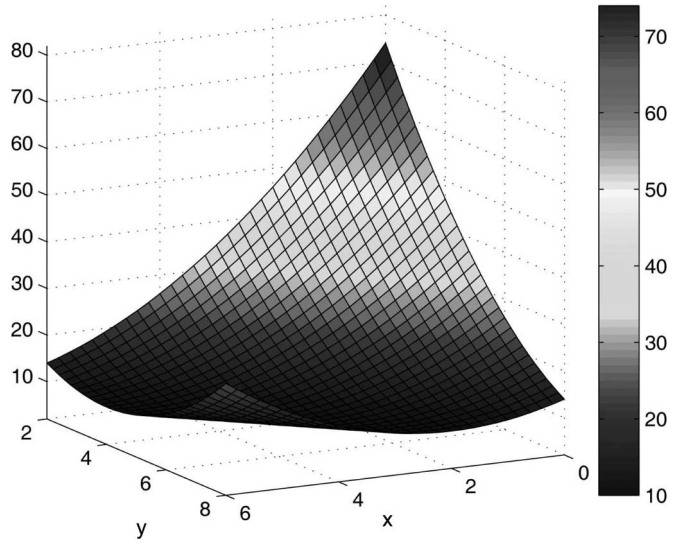

Fig. 5. Objective function.

Theorem 4.2 (∗∗—Semantic Theorem): Let be $X \in I^*(\mathbb{R}^n)$ and $Z \in I^*(\mathbb{R})$; then

$$f^{**}(X) \supseteq Z \Leftrightarrow \forall\,(x_i, X_i')\, Q\,(z, \text{Dual}(Z)) \\ \exists\,(x_p, X_p')\; z = f(x_p, x_i) \quad (26)$$

where $\text{Dual}([a,b]) = [b,a]$.

Both semantic theorems make equivalent a logical formula, with intervals and functional predicates where the universal quantifiers precede the existential ones, to an interval inclusion.

Although functions f^* and f^{**} are optimal in the semantic sense, these theorems do not explicit the process of computation of the interval Z which fulfills $f^*(A) \subseteq Z$ or $f^{**}(A) \supseteq Z$, i.e., intervals which are an outer estimate of f^* or an inner estimate of f^{**}. To find these estimates, an algorithm has been developed [21], [22].

The basic concept consists of bisecting, in a intelligent manner, the variable space by studying the monotonic nature of the function with respect to each of its variables. Then, by applying some theorems from MIA, which are also based on the monotony study, it is possible to obtain better results in the evaluation.

For example, consider the continuous function

$$f(x,y) = x^2 + y^2 + 2xy - 20x - 20y + 110 \quad (27)$$

with its real variables ranging in the domains $x \in [0,6]$ and $y \in [2,8]$, which is shown in Fig. 5.

The maximum value of f is 74, and the minimum value, ten, is taken in all the points of the domain belonging to the line $x + y = 10$. Therefore, the exact range of this function is $f([0,6],[2,8]) = [10,74]$, whereas the range obtained by interval natural extension, which means replacing real variables by interval variables and real operators by their interval counterparts, is $[-166, 266]$. It can be seen that a huge overestimate is produced due to the multiincidences of the variables and the nonmonotonic nature of the function with respect to them. By applying the proposed algorithm [23] (with a PENTIUM M 760, 2 GHz), the results after a specific time are the ones of Table I. These results show seven significative digits conveniently rounded to maintain the corresponding semantics of the inner and the outer approximations.

TABLE I
RESULTS OF THE f^* ALGORITHM

Time	Bisections	Inner approx.	Outer approx.
0.1 s	68	[10.00099, 73.99999]	[-4.750001, 74.00001]
0.2 s	154	[10.00025, 73.99999]	[4.375976, 74.00001]
0.5 s	497	[10.00003, 73.99999]	[8.125976, 74.00001]
1.0 s	1036	[10.00001, 73.99999]	[9.063476, 74.00001]
2.0 s	1751	[10.00001, 73.99999]	[9.531310, 74.00001]
5.0 s	3123	[10.00001, 73.99999]	[9.648623, 74.00001]
10.0 s	4881	[10.00001, 73.99999]	[9.824222, 74.00001]

The run time of the f^* algorithm increases with the number of variables and the required precision. However, for fault detection, a close approximation to the range of the residual r is not necessary because the execution can finish when $[0,0] \subseteq \text{Inner}(r)$ or $[0,0] \not\subseteq \text{Outer}(r)$.

D. CPI

SQualTrack is presented in two different versions, one for working with offline data and another for working with online data. The CPI of these versions is different.

1) Offline CPI. The offline version of the SQualTrack solver obtains the required process data from a text file. This version is part of the Modal Intervals Software and can be freely downloaded [23].
2) Online CPI. In the context of the CHEM project [10], an online version of the SQualTrack solver has been implemented. This version of the solver is provided with a CPI, developed within the CHEM project, which allows the communication with the process and with other software tools devoted to solving other supervision tasks (e.g., diagnosis and rescheduling). This version has been used in the applications described in this paper.

E. Numerical and Graphical Outputs

In order to provide a friendly output to the user, SQualTrack represents the results at the graphical user interface.

The graphical output of the software is shown in different figures in the following sections, for instance, in Fig. 7. The upper graph shows the approximations (inner and outer) for the output variable together with the measured output, which is plotted with brighter colors than the approximations in order to distinguish them. Note that often, the inner and outer approximations cannot be distinguished in the graphic because they are very close.

As all these signals are represented by intervals, it is easy to see if the outer approximation intersects or not with the measured output which is equivalent to testing (12). Then, it is possible to visually detect faulty behavior and, in this case, see if variables exceed or fall below their expected values.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

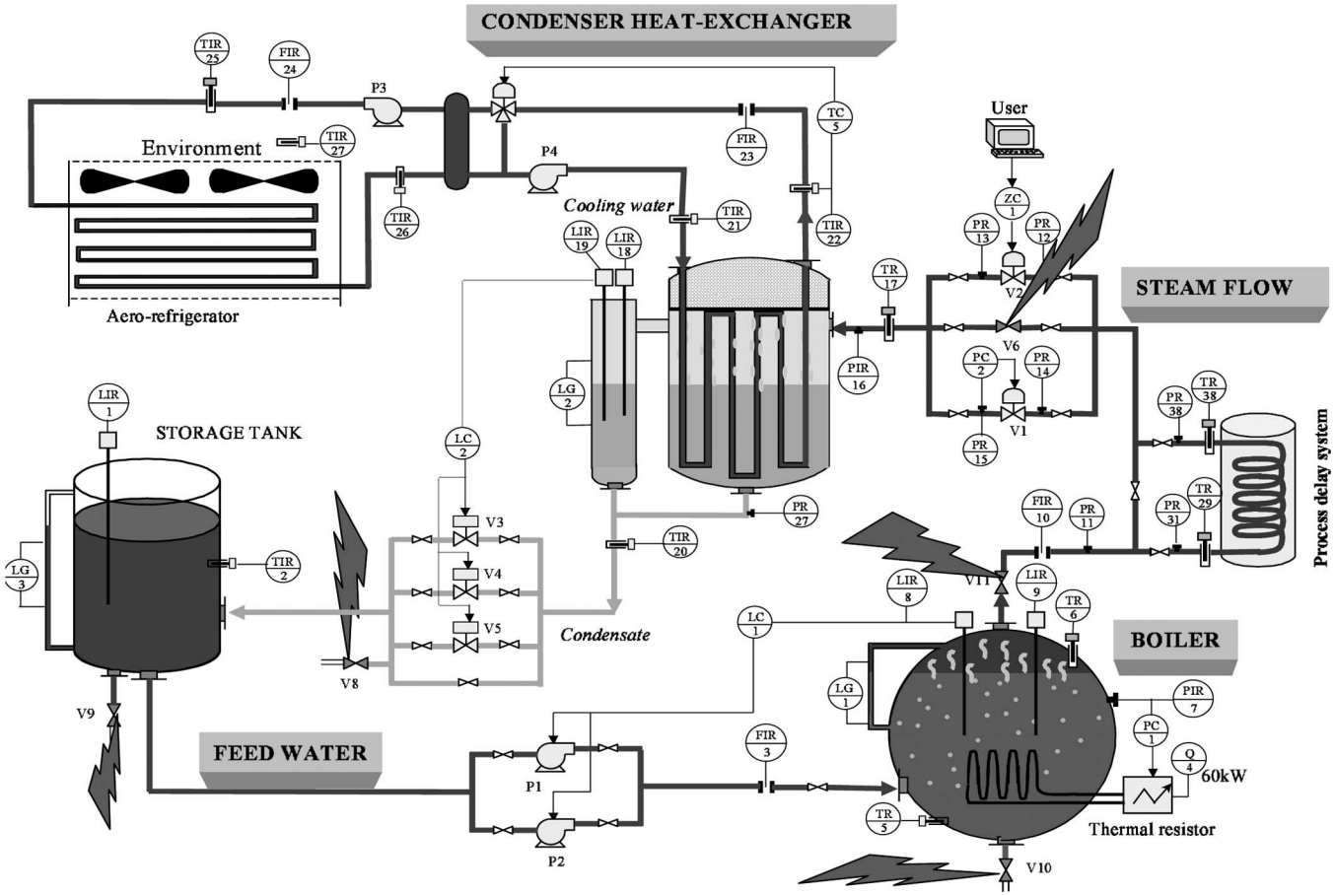

Fig. 6. Flowsheet of the steam-generator pilot plant.

The graph in the center of Fig. 7 shows a bar when a fault is detected, i.e., when the intersection between the interval measurement and the external approximation is void. Finally, the lower graph indicates the longest window length that has been used at each time step to compute the approximations. Therefore, although the approximations at each time step may be computed using different window lengths, they are represented in a single figure. Due to this, the approximations sometimes seem to widen or tighten significantly from step to step.

SQualTrack also generates a numerical output, which consists of a text file containing the time step, the output variable, the inner and outer approximations of the f^* computation, a Boolean variable representing the detection of the fault, and the longest length of window which has been used at each time step.

V. Applications

SQualTrack is being applied to academic and to real processes. The application procedure follows these steps.

1) Select the critical components to be monitored.
2) Identify which subsystems are more affected by the critical components.
3) Verify that the input and output variables of the selected subsystems are measured. If not, additional sensors may be required.

TABLE II
Steam-Generator Uncertainty

Sensor	Relative error	Absolute error
Flow $F3$	0.016	0.0001085
Flow $F10$	0.01	0.0244
Level $L8$	0.027	0.000073
Pressure $P7$	0.005	0.0039

4) Obtain models of the selected subsystems from the physical equations or by identification methods.
5) Process the models to put them in a suitable form to be included in the software tool.

Once these steps have been completed, the tool is ready to be used.

This paper describes some applications within the European project CHEM [10]. The aim of this project is to develop and implement an advanced DSS for process monitoring, data and event analysis, and operational support in industrial processes. The system integrates innovative software tools in a synergistic way, thereby improving safety, product quality, and operational reliability, as well as reducing the economic losses due to faulty states, mainly in chemical and petrochemical processes.

One of the main tasks of this project is situation assessment of uncertain processes, which includes fault detection. This task is undertaken from different points of view. One is the model-based approach, which is the approach used by SQualTrack.

To test the different techniques that have been developed, several partners of the project provide industrial processes

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

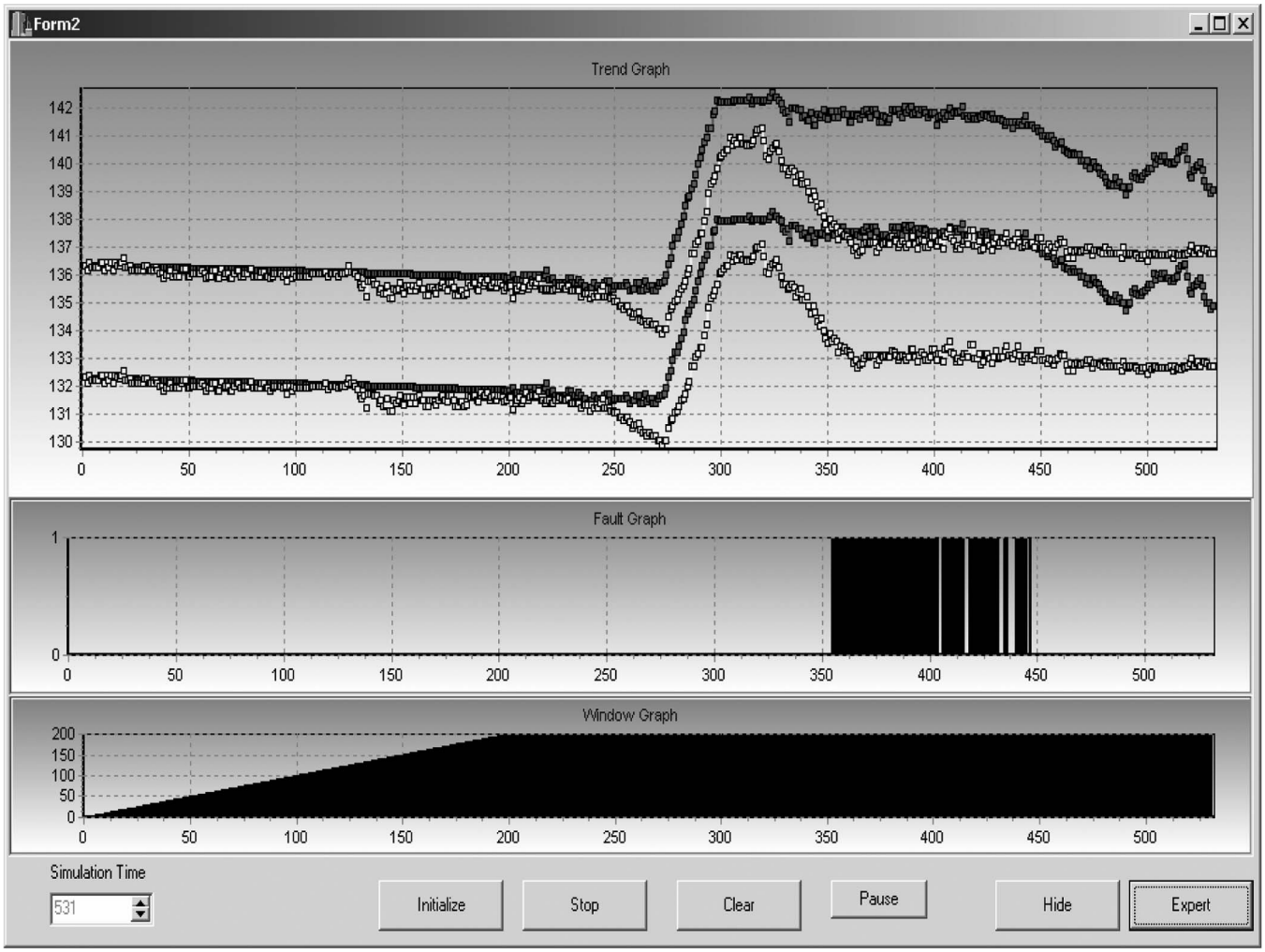

Fig. 7. Boiler leakage.

and pilot plants, including the Fluid Catalytic Cracking plant owned by the French Institute of Petroleum and located in Lyon (France); the steam-generator pilot plant owned by the Laboratoire d'Automatique et d'Informatique Industrielle de Lille (France); and the flexible chemical pilot plant PROCEL, owned by the Universitat Politècnica de Catalunya and located in Barcelona (Catalonia, Spain). Some of the results obtained by applying SQualTrack to fault detection in some of these processes are presented in the following.

VI. Steam-Generator Pilot Plant

A. Process Description

The steam generator is a scale model of a power plant. It is a complex nonlinear system that reproduces the same thermodynamic phenomena as the real industrial process. As shown in the flowsheet of Fig. 6, this installation consists of four main subsystems: a receiver with feedwater supply, a boiler heated by a 60-kW resistor, a steam flow system, and a complex condenser coupled with a heat exchanger.

The feedwater flow is circulated via two feed pumps connected in parallel. Each pump is controlled by an on–off controller to maintain a constant water level in the steam generator. The heat control depends on the pressure in the boiler; when this pressure falls below a minimum value, the heater resistor delivers maximum power, and when the accumulator reaches a maximum pressure, the electrical feed to the heater resistor is switched off. The expansion of the generated steam is realized by three valves connected in parallel: V6 is a manual bypass valve, simulating the bypassing of the steam flow to the condenser; V2 is a controlled position valve; and V1 is automatically controlled to maintain proper pressure to the condenser. In an industrial plant, the steam flows to the turbine to generate power; however, at the test rig, the steam is condensed, stored in a receiver tank, and then returned to the steam generator.

B. Testing Scenario

The values of the measured variables are collected once per second. These data are grouped in "scenarios," which include data from normal and abnormal situations, such as unusual events or incidents. Among these, there are scenarios with problems in the sensors (poor calibration and failures), in the actuators, or in the process itself.

The faulty scenario that is considered consists of opening valve V10 for a time, causing leakage in the boiler.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

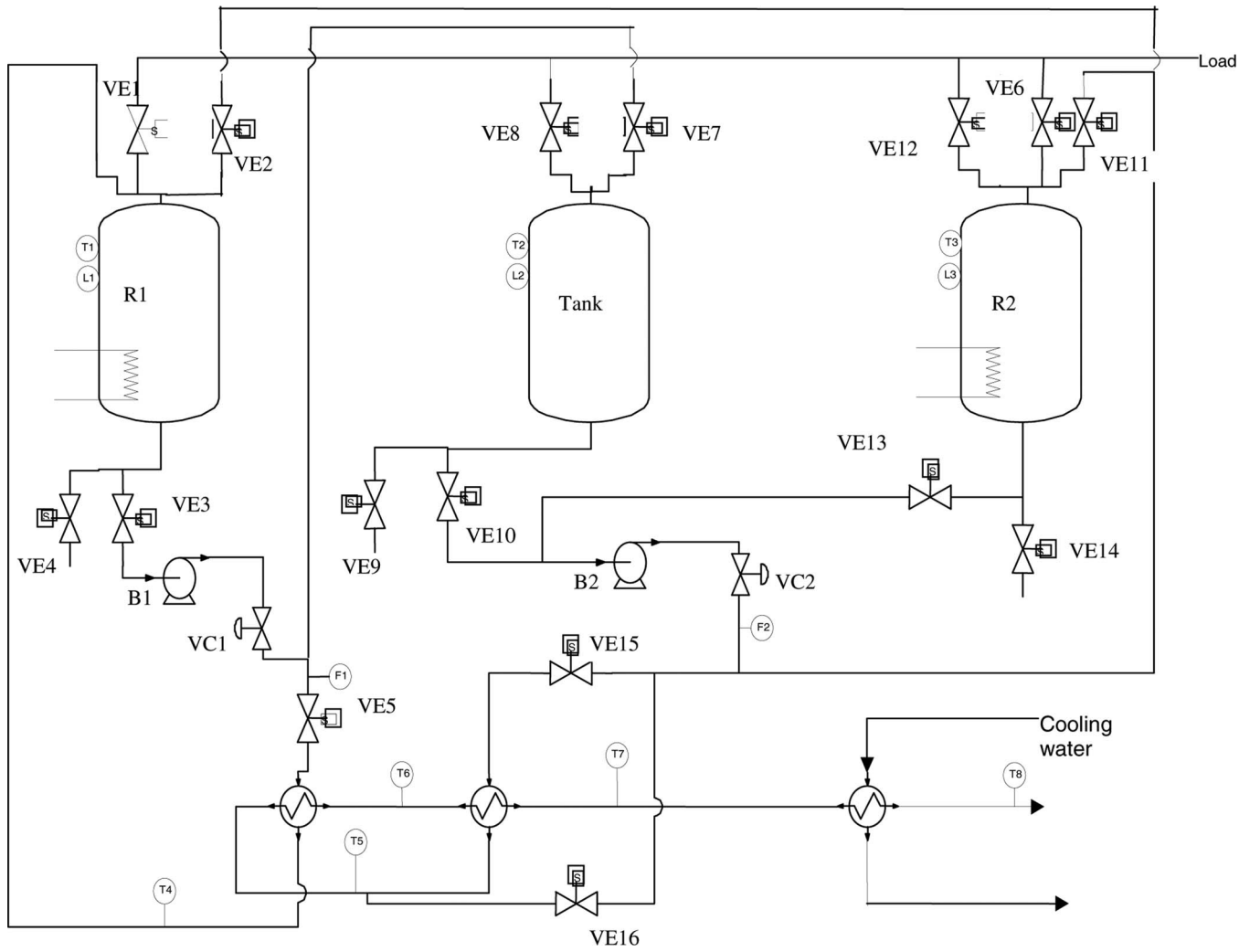

Fig. 8. Flowsheet of PROCEL.

C. Model From Mass Balance

A model of the boiler is obtained from a mass balance

$$\frac{dM}{dt} = F3 - F10 \tag{28}$$

where M is the mass inside the boiler, $F10$ is the output mass flow, and $F3$ is the input mass flow.

The corresponding difference equation obtained using the explicit Euler discretization is

$$M(k+1) = M(k) + (F3(k) - F10(k))\, T_s. \tag{29}$$

As the monitored variable M is not directly measurable, it is estimated using the following:

$$M = V_{\text{steam}}\rho_{\text{steam}} + V_{\text{liquid}}\rho_{\text{liquid}} \tag{30}$$

where V_{liquid} is the volume of liquid calculated from the level which is highly uncertain due to the bubbles in the boiling water, ρ_{liquid} is the liquid water density, which is a function of the pressure inside the boiler and obtained from tables of steam properties, V_{steam} is the volume of steam (given by $V_{\text{steam}} = V_{\text{boiler}} - V_{\text{liquid}}$), and ρ_{steam} is the steam density (also given by the steam tables).

TABLE III
PROCEL UNCERTAINTY

Sensor	Relative error	Absolute error
$F1, F2, F3$	0.05	-
$F4$	0.05	-
V_{R1}	0.03	-
$T1, T2, T3$	0.05	-
$T4 = T_{R1}$	0.05	-
P_H	0.1	-
Parameter	**Minimum**	**Maximum**
ρ_{R1}	1000	1000
T_{amb}	18	20
c_{pR1}	4186	4186
G_{R1}	0.01	0.01

Table II shows the uncertainty associated with each measurement. The relative error e_r corresponds to the sensor precision, and the absolute error e_a corresponds to the error introduced by the truncation of the digital scale which is used. To obtain the interval X associated with a measurement x, the following is used:

$$X = [x(1 - e_r) - e_a, x(1 + e_r) + e_a]\,. \tag{31}$$

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

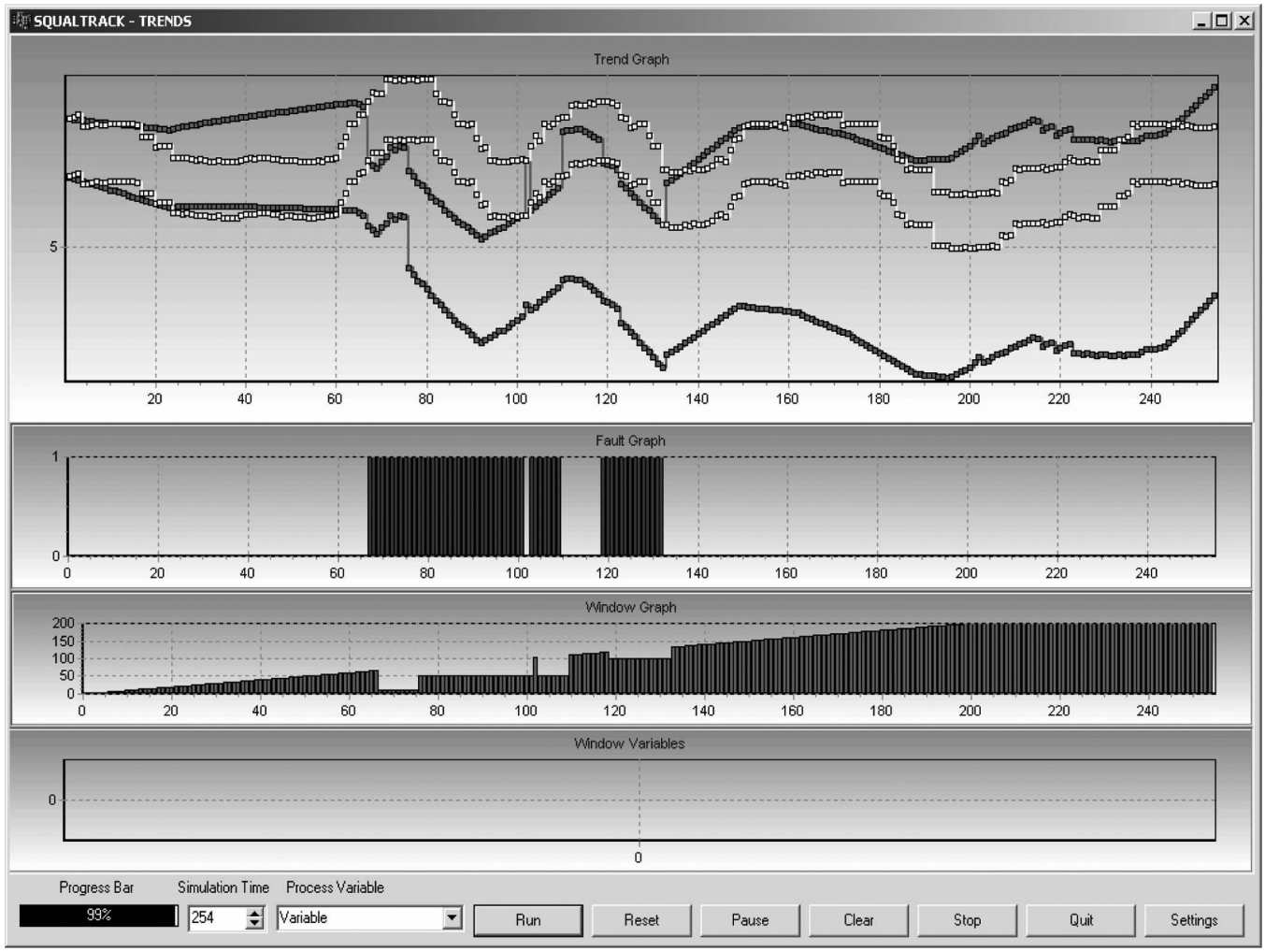

Fig. 9. Additional input flow to reactor 1.

D. Test Results

Fig. 7 shows the main window of the fault-detection tool for the test scenario. In this case, the sample time is 1 s, and the time windows, of lengths 10, 50, 100, and 200 s, are used.

The fault began at 240 s and ended at 360 s and is detected from 353 to 448 s using a window of 200 s. Due to dynamics, the differences between the behaviors of the healthy and the faulty systems when there are small faults are small and undistinguishable from the usual differences due to uncertainty. To amplify these differences, long windows are needed. In this case, shorter windows do not detect the fault, and only the 200-s one does it.

On the other hand, it is not necessary to use long windows to detect large faults. Short windows also detect the faults, needing a much lower computing effort.

An added difficulty in this case is that the level in the boiler is controlled so that a pump is switched on when the level is below a predefined level. Fig. 7 shows that when the fault appears, the level decreases, then increases for a while because the pump is on, and then decreases again due to the leakage, and finally, the fault is detected because the level is too low, taking into account that the pump has been on.

VII. PROCEL Pilot Plant

A. Process Description

PROCEL comprises three tank reactors, three heat exchangers, and the necessary pumps and valves to allow for changes in the configuration. The PROCEL equipment is fully connected, and the associated instrumentation allows for changing of the configuration by software. Fig. 8 shows a flowsheet of PROCEL.

The example shown in this paper is based on a part of the process, the temperature in reactor 1.

B. Testing Scenarios

Three faulty scenarios affecting reactor 1 are considered.

1) An additional input flow. The fault consists of opening valve *VE*2 during a time period to simulate an additional input flow that is not taken into account, i.e., a perturbation.
2) A leakage. Valve *VE*4 is opened during a time period.
3) Heating resistor 1 shutdown.

For scenarios 1) and 2), a model obtained from the mass balance of reactor 1 is sufficient. The monitored variable is the

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

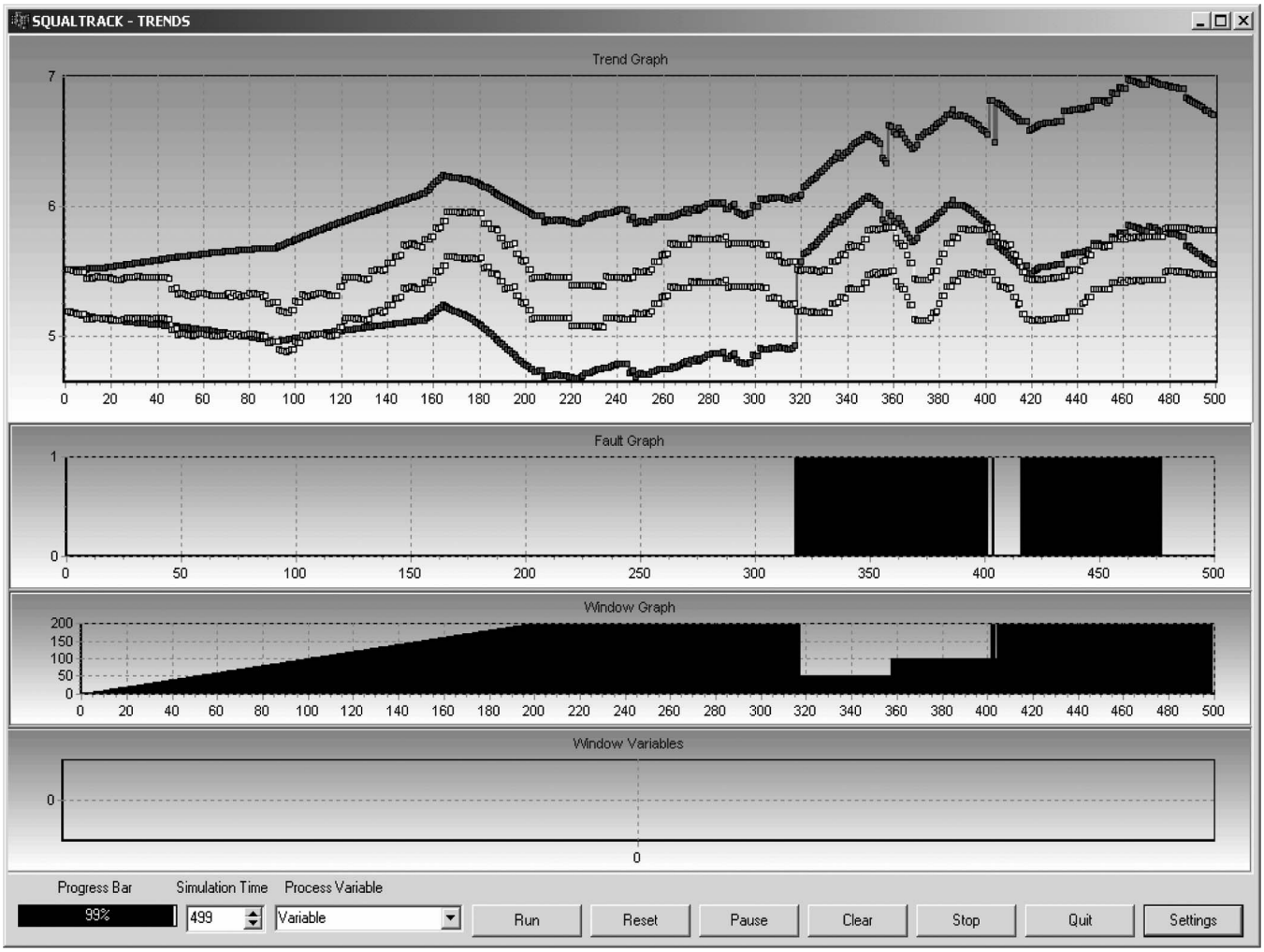

Fig. 10. Reactor 1 leakage.

volume of liquid in reactor 1. For scenario 3, a model obtained from the energy balance is required. The monitored variable is the temperature of reactor 1.

C. Model From Mass Balance

The volume variation inside reactor 1 is

$$\frac{dV_{R1}}{dt} = (F1 + F2 + F3) - F4 \tag{32}$$

where V_{R1} is the volume of liquid in the reactor; $F1$, $F2$, and $F3$ are the volumetric input flows; and $F4$ is the volumetric output flow. It is assumed that there is a relative error of 3% for V_{R1} and 5% for $F1$, $F2$, $F3$, and $F4$ (Table III summarizes these uncertainties). The corresponding difference equation is

$$V_{R1}(k+1) = V_{R1}(k) + (F1(k) + F2(k) + F3(k) - F4(k))\, T_s \tag{33}$$

where T_s is the sample time.

As V_{R1} is easily obtainable from the measurements of the level, the ARR consists of comparing the value of V_{R1} computed from (33), which is an interval, with the value computed from the measurements of the level, which also is an interval.

D. Model From Energy Balance

The temperature variation inside reactor 1 is

$$\frac{dT_{R1}}{dt} = \frac{F1(T_1 - T_{R1})}{V_{R1}} + \frac{F2(T_2 - T_{R1})}{V_{R1}} + \frac{F3(T_3 - T_{R1})}{V_{R1}} + \frac{P_H - G_{R1}(T_{R1} - T_{\mathrm{amb}})}{\rho_{R1} c_{pR1} V_{R1}} \tag{34}$$

where T_{R1} is the temperature of the liquid in the reactor and, therefore, the output of this subsystem. The inputs are the flows $F1$, $F2$, and $F3$ and the corresponding temperatures $T1$, $T2$, and $T3$. It is assumed that there is a relative error of 5% for T_{R1}, $T1$, $T2$, and $T3$.

The parameters of the model are the volume of liquid V_{R1}, the ambient temperature T_{amb} ($T_{\mathrm{amb}} = [18, 20] + 273.15$ K), the heater power P_H (with a relative error of 10%), the thermal conductance of the wall G_{R1} (approximately 0.01 W/K), the density of the liquid ρ_{R1} (water), and its specific heat c_{pR1}. These uncertainties are summarized in Table III. Moreover, a digital variable indicating if the heater is on or off is required. This variable is already included in the data sets.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

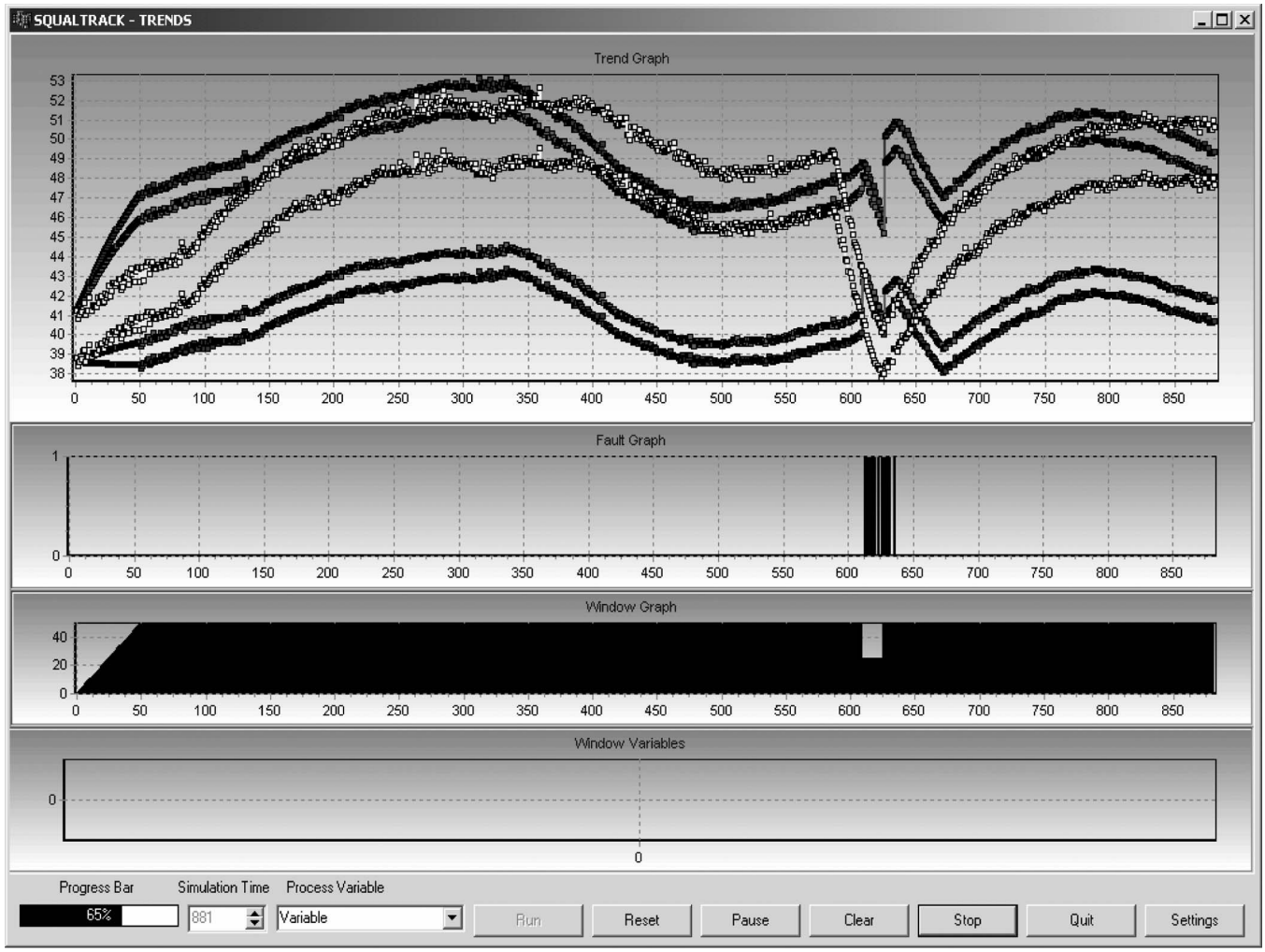

Fig. 11. Reactor 1 heater shutdown.

The corresponding difference equation is

$$T_{R1}(k+1) = T_{R1}(k) + \frac{T_s F1(k)\,[T_1(k) - T_{R1}(k)]}{V_{R1}(k)} + \frac{T_s F2(k)\,[T_2(k) - T_{R1}(k)]}{V_{R1}(k)} + \frac{T_s F3(k)\,[T_3(k) - T_{R1}(k)]}{V_{R1}(k)} + \frac{T_s\,[P_H(k) - G_{R1}(T_{R1}(k) - T_{\mathrm{amb}})]}{\rho_{R1} c_{pR1} V_{R1}(k)}. \quad (35)$$

As $T_{R1}(k)$ is directly measurable, the ARR consists of comparing the computed value of $T_{R1}(k)$ with its measurement.

E. Test Results

Figs. 9–11 show the main window of the fault-detection software when it is used to detect faults for each of the test scenarios.

In all the graphs, note that the horizontal axes indicate samples, not time, and that the sample time is 3 s.

For scenario 1 (additional input flow), the fault began at sample 55 and ended at sample 73. It is detected from sample 63 until sample 112.

For scenario 2 (leakage) the fault began at sample 272 and ended at sample 343. It is detected from sample 318 until sample 477.

For scenario 3 (heater shutdown) the fault began at sample 581 and ended at sample 617. It is detected from sample 610 until sample 640.

VIII. Conclusion and Future Work

This paper has presented a method based on analytical redundancy to detect faults in dynamic systems with parametric uncertainties. The systems' uncertainty is represented using interval models, and the uncertainty associated with the measurements is also represented by intervals. The consistency between the interval model and the interval measurements is checked by error-bounded estimations of the value of a variable; therefore, a fault is detected when the interval measurement does not intersect with the external estimation. The number

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

of missed alarms is reduced by using several window lengths simultaneously, considering that a fault is detected when there is any inconsistency in any time window. Therefore, this method increases the detection of faults compared with the use of a single window length and, at the same time, minimizes the computations required.

The method uses a branch-and-bound algorithm, which computes error-bounded estimations. The use of MIA significantly contributes to the overall algorithm efficiency. It guarantees the absence of false alarms due to the fault-detection scheme itself. If there are false alarms, they indicate either that the interval model does not represent the system adequately or that the interval measurements do not represent the true values of the variables adequately.

This method has been implemented in SQualTrack and has been applied to several real processes within the European project CHEM. SQualTrack can be used either off- or online. This paper describes some of these applications.

SQualTrack can also be used for fault isolation. The most classical diagnosis technique consists in using several ARRs and fault signatures. SQualTrack can be used to evaluate these ARRs by simultaneously running several instances of SQualTrack, one per ARR. The binary vector defined by the fault-detection results is the observed fault signature of the system (set of symptoms) to be compared with the theoretical fault signature matrix (in which each cell represents if a given fault has an effect or not over a given ARR) [24].

More advanced techniques for fault diagnosis use other information, apart from the binary one, such as the sign of the symptom, the order of the symptom appearance, or the size of the residual [25]–[27]. SQualTrack provides all this information so it can be used in combination with these techniques. This use of SQualTrack is a research topic for the future.

REFERENCES

[1] K. Balakrishnan and V. Honavar, "Intelligent diagnosis systems," *J. Intell. Syst.*, 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," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, Mar. 2003.

[3] MONET, *European Network of Excellence on Model Based Systems and Qualitative Reasoning*, 1998. [Online]. Available: http://monet.aber.ac.uk:8080/monet/index.html

[4] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. New York: Springer-Verlag, 2003.

[5] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Norwell, MA: Kluwer, 1998.

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

[7] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. New York: Springer-Verlag, 2000.

[8] E. Gardeñes, M. Á. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepat, "Modal intervals," *Reliab. Comput.*, vol. 7, no. 2, pp. 77–111, Apr. 2001.

[9] M. Á. Sainz, P. Herrero, J. Armengol, and J. Vehí, "Continuous minimax optimization using modal intervals," *J. Math. Anal. Appl.*, vol. 339, no. 1, pp. 18–30, 2008.

[10] CHEM Consortium, *Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes*, 2000.

[11] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications," *Control Eng. Pract.*, vol. 5, no. 5, pp. 671–682, May 1997.

[12] V. Puig, J. Quevedo, T. Escobet, and S. De las Heras, "Passive robust fault detection approaches using interval models," in *Proc. 15th IFAC World Congr.*, 2002. CD-ROM. [Online]. Available: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2002/data/content/02019/2019.pdf

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

[14] I. Fagarasan, S. Ploix, and S. Gentil, "Causal fault detection and isolation based on a set-membership approach," *Automatica*, vol. 40, no. 12, pp. 2099–2110, Dec. 2004.

[15] S. Ploix and C. Follot, "Fault diagnosis reasoning for set-membership approaches and application," in *Proc. IEEE Int. Symp. Intell. Control*, 2001, pp. 85–90.

[16] E. Hansen, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker, 1992.

[17] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*. Norwell, MA: Kluwer, 1996.

[18] J. de Guzmàn, *Spirit Framework*, 2006. [Online]. Available: http://spirit.sourceforge.net/

[19] "Modal intervals. Basic tutorial," *Applications of Interval Analysis to Systems and Control. Proceedings of MISC 99*, pp. 157–227, 1999, Catalonia, Spain: Universitat de Girona.

[20] E. Kaucher, "Interval analysis in the extended interval space IR," *Computing, Suppl.*, vol. 2, pp. 33–49, 1979.

[21] P. Herrero, M. Á. Sainz, J. Vehí, and L. Jaulin, "Quantified set inversion algorithm with applications to control," *Reliab. Comput.*, vol. 11, no. 5, pp. 369–382, Oct. 2005.

[22] P. Herrero, "Quantified real constraint solving using modal intervals with applications to control," Ph.D. dissertation, Universitat de Girona, Girona, Spain, 2006.

[23] Modal Intervals and Control Engineering Research Group, *MISO: Modal Interval SOlver (FSTAR Solver)*, 2006. [Online]. Available: http://pau.herrero.googlepages.com/software

[24] M.-O. Cordier, P. Dague, F. Levy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès, "Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2163–2177, Oct. 2004.

[25] P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 6, pp. 554–565, Nov. 1999.

[26] V. Puig, J. Quevedo, T. Escobet, and B. Pulido, "On the integration of fault detection and isolation in model-based fault diagnosis," in *Proc. 15th Int. Workshop Principles Diagnosis DX*, 2004, pp. 227–232.

[27] M. Daigle, X. Koutsoukos, and G. Biswas, "Fault diagnosis of continuous systems using discrete-event methods," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 2626–2632.

Joaquim Armengol was born in Girona, (Catalonia, Spain), in 1967. He received the degree in industrial engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1992 and the Ph.D. degree in industrial engineering from the Universitat de Girona, Girona, Spain, in 2000.

He is currently with the Universitat de Girona, as an Associate Professor and working on his research at the Institut d'Informàtica i Aplicacions. His research interests include application of modal intervals to the assessment of uncertain dynamics systems with parametric uncertainties. Specifically, his main works are about fault detection and diagnosis.

Prof. Armengol has participated in several national and international research projects (Real-Time Situation Assessment of Dynamic, Hard to Measure Systems, Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes) and has been a member of several Spanish and European (European Network of Excellence on Model Based Systems and Qualitative Reasoning, European Network of Excellence on Intelligent Technologies for Smart Adaptive Systems) scientific networks.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.

Josep Vehí (M'99) was born in Girona, Spain, on January 9, 1964. He received the Ph.D. degree in electrical engineering from the Universitat de Girona, Girona, Spain in 1998.

He was a Teaching Assistant, from 1987 to 1992, with the Technical University of Catalonia, Barcelona, Spain. Since 1992, he has been with the Universitat de Girona, where he was first an Assistant Professor, from 1992 to 1999, and then an Associate Professor of electrical engineering in 1999. Since 2006, he has been the Director of the Institut d'Informàtica i Aplicacions (Research Institute on Computer Engineering and Automation), Universitat de Girona. His current research interests include interval methods for control, modeling and control of biomedical systems, robust and nonlinear control, fault detection and diagnostics for complex dynamic systems and intelligent methods for structural assessment.

Miguel Ángel Sainz was born in La Rioja, Spain, in 1942. He received the degree in mathematics from the Universidad de Zaragoza, Zaragoza, Spain, in 1964 and the Ph.D. degree from the Universidad de Madrid, Madrid, Spain, in 1970.

Since 1971, he has been a Professor with the Universitat Politècnica de Catalunya, Barcelona, Spain, with the Universitat Autònoma de Barcelona, Barcelona, Spain, and also with the Universitat de Girona, Girona, Spain, where he is currently a member of the Institut d'Informàtica i Aplicacions, developing his research in modal interval analysis and its applications to problems of simulation and control. He presented several papers in international meetings (International Federation of Automatic Control World Congress, SAFEPROCESS) and published in scientific journals (*Journal of Process Control* and *Reliable Computing*).

Pau Herrero received the Ph.D. degree in control engineering from the Universitat de Girona, Girona, Spain, in 2006 and from the Université d'Angers, Angers, France.

In 2007, he was a Postdoctoral Fellow, for one year, with the University of California, Santa Barbara, doing research on the development of an artificial pancreas. He is currently with the Hospital de la Santa Creu i Sant Pau, Barcelona, Spain, which belongs to the Centro de Investigación Biomédica en Red en Bioingeniería, Biomateriales, y Nanomedicina network. His current research interests include the development and application of new technologies applied to the treatment of diabetes.

Esteban R. Gelso was born in Olavarría, Buenos Aires, Argentina, in 1977. He received the degree in electromechanical engineering from the National University of the Center of Buenos Aires Province, Olavarría, in 2000. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering, Electronics and Automation, Universitat de Girona, Girona, Spain.

He is currently working on his research at the Institut d'Informàtica i Aplicacions, Universitat de Girona. His research interests include the aspects of model-based fault detection and diagnosis-combining elements from the FDI and DX research communities.

Authorized licensed use limited to: UNIVERSITAT DE GIRONA. Downloaded on May 8, 2009 at 06:49 from IEEE Xplore. Restrictions apply.