ROBUST FAULT DETECTION USING INTERVAL MODELS

S. Tornil[♣], T. Escobet[♣], L. Travé-Massuyès[♦]

[♣]Automatic Control Dept., Technical University of Catalonia
Rambla Sant Nebridi, 10 - 08222 Terrassa - Spain

[♦]Laboratory for Analysis and Architecture of Systems
7, Avenue du Colonel Roche - 31077 Toulouse Cedex 4 - France

E-mail : sebastian.tornil@upc.es
Fax : (+34) 93 739 86 28

Keywords: Robust fault detection, structured uncertainty, interval models, optimisation.

Abstract

The problem of model-based fault detection in the presence of both parametric uncertainty and noise is addressed in this paper. Intervals are used to represent the uncertainty in the system parameters and interval extensions of parity equations are used as adaptive threshold selectors. A proper combination in time of different (interval) parity equations, together with a robust indicator, is used to maintain a good sensitivity to faults whilst avoiding the effect of noise. The results in the application to the detection of different faults in a DC motor are used to show the good properties of the proposed method.

1 Introduction

In model-based FDI (Fault Detection and Isolation) methods, the actual behaviour of the system is checked for consistency with a mathematical model describing its non-faulty operation. But in most practical applications, it is not possible to obtain an accurate and complete model of the system. When the values of the system parameters are not exactly known but bounded, intervals appear as a natural framework to represent this uncertainty. Using this modelling approach, the model provides the set of feasible behaviours in normal system operation and a fault can be reported when the observed behaviour lies outside this set. Within this context, different strategies are proposed in the literature ([1], [2], [5], [8]) to attach the FDI problem.

This paper focuses on the FD (Fault Detection) problem for continuous-time LTI systems. The described methodology considers both structured uncertainty, due to component tolerances, and measurement noise. The goals are to maximize fault sensitivity and to provide persistent fault indicators, whilst being insensitive to noise.

The paper has the following structure. Analytical model-based fault detection is briefly reviewed in section 2, focusing in implementation using parity equations and robustness issues. In section 3, extensions of parity equations to interval models are presented and compared. The proposed FD methodology, which combines in time different interval parity equations and uses a robust indicator, is presented in section 4. Finally, a simple DC motor is used as a case study in section 5, before concluding with conclusions and future work in section 6.

2 Model-Based Fault Detection

2.1 Analytical redundancy

Model-based FD methods rely on the concept of *analytical redundancy*. The simplest analytical redundancy schema (cf figure 1) consists in the comparison of measured values for system outputs ($y_m(k)$) with corresponding analytically computed values ($y_e(k)$), obtained from measures of other variables and/or from previous measures of the same variable by means of a model. The resulting differences are called *residuals* ($r(k)$) and are indicative of faults in the system. Under ideal conditions, residuals are zero in the absence of faults and non-zero when a fault is present. However, in practical situations disturbances, noise and unavoidable modelling errors lead the residuals to non-zero values even in the absence of faults. Thus, the *residual generation* stage is followed by a *decision making* stage. The decision generally relies in comparing each residual with a fixed threshold chosen, according to empirical and/or theoretical considerations, to avoid false alarms and minimise hidden faults.

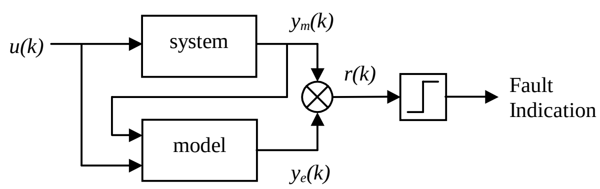

Figure 1. Analytical redundancy schema.

2.2 Parity equations

Parity equations ([3]) are the most straightforward method to implement residual generators. A time-domain derivation of parity equations for SISO systems is presented here.

Given a n^th^ order continuous-time SISO LTI system, the step-equivalent discrete-time approximation is given by a difference equation in the form:

$$y(k)+\sum_{i=1}^{n}a_i*y(k-i)=\sum_{j=1}^{m}b_j*u(k-j) \tag{1}$$

Using equation (1) L times in a recursive way, $y(k)$ can be expressed as function of $y(k-L),...,y(k-n-L)$ and $u(k),...,u(k-m-L)$. Using the controllable canonical form of (1), the following general expression can be obtained:

$$y(k)=c\left(A^L x(k-L)+A^{L-1}bv(k-L)+\cdots+Abv(k-2)+bv(k-1)\right) \tag{2}$$

where:

$$A=\begin{bmatrix}-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & 1 & 0\end{bmatrix}, b=\begin{bmatrix}b_1\\ ...\\ b_m\end{bmatrix}, c=\begin{bmatrix}1\\ 0\\ ...\\ 0\end{bmatrix}^T \tag{3}$$

$$x(k)=\begin{bmatrix}y(k)\\ ...\\ y(k-n)\end{bmatrix}, v(k)=\begin{bmatrix}u(k)\\ ...\\ u(k-m)\end{bmatrix}$$

Starting from (2)-(3), two alternatives can be considered to obtain an on-line estimation $y_e(k)$ of the output real value $y(k)$: (a) *prediction*: use of measured values for prior system outputs $y_m(k-L),..,y_m(k-n-L)$; (b) *simulation*: use of previously estimated values $y_e(k-L),.., y_e(k-n-L)$. The obtained equations, called *MA* (Moving Average) and *ARMA* (Auto-Regressive MA) L^{th} *order parity equations* ([5]) respectively, are:

$$y_p(k)=c\left(A^L\begin{bmatrix}y_m(k-L)\\ ...\\ y_m(k-n-L)\end{bmatrix}+A^{L-1}bv(k-L)+\cdots\right) \tag{4}$$

$$y_s(k)=c\left(A^L\begin{bmatrix}y_s(k-L)\\ ...\\ y_s(k-n-L)\end{bmatrix}+A^{L-1}bv(k-L)+\cdots\right) \tag{5}$$

2.3 Robustness

Robustness in model-based FD systems is the property required to operate in the presence of disturbances, noise and modelling errors whilst maintaining sensitivity to faults. The most critical point in FD is robustness against modelling errors ([3]) and it can be considered at the residual generation stage or at the decision making stage ([6]).

Active robustness techniques are decoupling techniques that try to eliminate the effects of modelling errors and disturbances in the obtained residuals. But decoupling from modelling errors presents two main drawbacks ([6]). The first and main problem is that decoupling resolves the problem of modelling errors by avoiding rather than accounting for them. Obtained residuals are insensitive to uncertain parameters and, consequently, they are also insensitive to faults in these parameters. The second problem is that there is no residual generation algorithm which is robust under arbitrary model error conditions ([5]). Perfect decoupling is only possible under certain restrictive conditions and only approximate decoupling can be obtained in most cases.

When perfect robust residual generation can't be obtained, residuals are non-zero in the absence of faults and their magnitude is function of the system operating conditions (inputs and state). Hence, the use of fixed thresholds, for instance large enough to avoid false alarms when the input is high, can result in missing faults when the input is low. In general, there is no fixed threshold value that gives tolerable false and missing alarm rates across the whole operation range. In this situation, passive robustness techniques can be applied to enhance the robustness of the FD system. One way to enhance robustness against modelling errors is based on the use of *adaptive thresholds* which vary according to the state and the inputs of the system. The advantages of using adaptive thresholds is shown in figure 2.

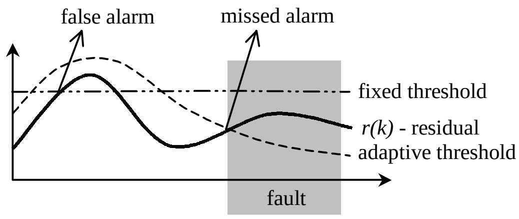

Figure 2. Fixed vs. adaptive threshold.

Different approaches for adaptive thresholding can be found in the literature (see [6] and references therein). The use of interval models was proposed by Adrot [1], Armengol ([2]) and Puig ([5]).

3 Interval Parity Equations

3.1 Fault Detection using Interval Models

When the model structure of a system is known but only bounds for the values of its physical parameters are known, for instance when nominal values and tolerances are given, intervals appear as a natural framework to represent and include this uncertainty in the system model. The obtained model is said to be an *interval model*.

The time response of an interval model is given by two curves, called *envelopes* ([2],[5]) or *enclosures* ([1]), which define at each time instant the minimum and maximum achievable values for the system output. At each time instant, an interval $\left[y_e^-(k), y_e^+(k)\right]$ representing the maximum effect of the uncertainty in the system output is obtained. Hence, the envelopes define the optimal adaptive thresholds, i.e. the minimal thresholds that avoids false alarms.

In this situation, a fault can be reported when the output goes outside the envelopes (cf figure 3). On the other hand, a normal situation is assumed when the output lies inside the envelopes, although for some faulty conditions the output can remain eventually or even definitively inside them ([7]). This is the price to pay for the lack of knowledge about the exact systems parameters.

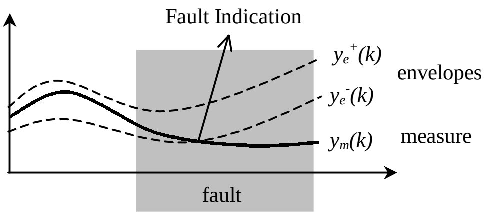

Figure 3. Interval Model-Based FD.

3.2 Obtaining Interval Parity Equations

The extension of parity equations to interval models is derived here. The extension of other classical residual generation methods can be found in the literature, i.e. the parity space approach in [1] and the observer approach in [6].

It was omitted in section 2.2 that the coefficients in equation (1), and hence the matrix A and vectors b and c in (2), are function of the physical system parameters. Let $\varpi = \left[\varpi_1,...,\varpi_p\right]$ be the vector of values for these parameters, then equation (2) can be rewritten as:

$$y(k) = c(\varpi)\left(A(\varpi)^L x\left(k-L\right) + A(\varpi)^{L-1} b(\varpi) v\left(k-L\right) + \cdots\right) \quad (6)$$

Suppose that the exact values for the physical parameters are unknown but bounds are known for each of them $\varpi_i \in \Theta_i = \left[\varpi_i^-, \varpi_i^+\right]$ and let Θ a vector containing these uncertain values $\Theta = \left[\Theta_1,...,\Theta_p\right]$. In this situation, equation (6) must be interpreted as:

$$y(k) \in \underset{\varpi\in\Theta}{range}\left\{c(\varpi)\left(A(\varpi)^L x\left(k-L\right) + A(\varpi)^{L-1} b(\varpi) v\left(k-L\right) + \cdots\right)\right\} \quad (7)$$

Hence, the interval version of L^{th} order MA parity equation is given by the following expressions:

$$y_p^{-/+}(k) = \underset{\varpi\in\Theta}{\min/\max}\left\{c(\varpi)\left(A(\varpi)^L \begin{bmatrix} y_m(k-L) \\ ... \\ y_m(k-n-L) \end{bmatrix} + \cdots\right)\right\} \quad (8)$$

And the interval version of L^{th} ARMA parity equation is given by:

$$y_s^{-/+}(k) = \underset{\substack{\varpi\in\Theta \\ y_s(k')\in\left[y_s^-(k'),y_s^+(k')\right]}}{\min/\max}\left\{c(\varpi)\left(A(\varpi)^L \begin{bmatrix} y_s(k-L) \\ ... \\ y_s(k-n-L) \end{bmatrix} + \cdots\right)\right\} \quad (9)$$

These two options are also called *interval sliding window prediction* and *simulation*, respectively ([8]). For a given *window length L*, we will refer to *L-prediction* and *L-simulation*. Note that for estimating the output value at instant k, L-prediction uses previous measured values whereas L-simulation uses previous estimations in form of intervals.

In both cases, the envelope generation problem at a given time instant is formalised as two global optimisation problems. Modern methods based on interval arithmetic and 'branch and bound' strategies ([4]) can be used to guaranty global convergence, guarantying complete envelopes ([2],[5]).

3.3 MA vs. ARMA Interval Parity Equations

Interval parity equations of different type (MA/ARMA) or with different window length produce different envelopes. This is due to the fact they use different information and because the are affected in a different way by two known problems ([2],[5]): temporal multi-incidence and wrapping effect.

The temporal multi-incidence problem arises from the fact that multi-incidences exist for some parameters and variables in the relation used for estimating *y(k)* and *y(k+1)*. Because these multi-incidences are not into account, the time-invariace assumption is not used and some faulty time-varying behaviours lie inside the envelopes and can't be detected. The only way to thwart this problem is by considering, at any discrete instant k, k^{th} order parity relations (a constantly increasing window length). However, this is not applicable in practice, due to the increasing computational complexity.

The wrapping problem arises because the state vector is multidimensional. The initial state is specified as a parallelotope, but within a prediction step, this volume suffers a transformation which does not necessarily ends as another parallelotope. However, it is approximated as so, which introduces spurious regions. If this overbounded region is used in subsequent calculations, there is a chain effect known as the wrapping problem.

Fault detection using interval simulation is proposed by Puig et al. in [5]. Interval simulation is summited to both temporal multi-incidence and wrapping problems. The longer the length of the window, the lower the effect of these problems. However, it has been proved that the gain decreases drastically when the window lentgh is longer than the response time of the system ([5]); hence, this time defines the optimal window length for interval simulation.

Interval prediction is proposed by Armengol et al. in [2]. Interval prediction, due to the fact that it starts with a real initial state, avoids wrapping and produces envelopes which are always included into the simulation envelopes. This provides the interval prediction method with a better sensitivity to faults and a lower detection time. The effect of window length is the opposite: the longer the window length, wider are the obtained envelopes; hence, small window length should be used. However, a different problem exists. When there is a fault in the system which leads the output *y(k)* outside the envelopes, the fault is detected. But the prediction algorithm uses this output value, which is know to be generated by a faulty system, in next prediction steps. The effect is that the envelopes will follow the faulty output and if noise is present, the system output can lie again inside the envelopes giving a non persistent indicator.

To avoid this problem, a hybrid prediction-simulation approach, called *Semi-Closed Loop* (SCL) strategy, is used in the detection module of the Ca~En FDI system ([8]). This SCL strategy and some improvements are presented in the next section.

4 Mixed MA/ARMA Strategy

4.1 Semi-Closed Loop (SCL) Strategy

Under fault-free conditions, 1-prediction provides the most restrictive envelopes. The use of this envelopes for fault detection is expected to provide the highest fault detectability and the lowest detection time. But, once the output goes outside the envelopes and a fault is detected, it seems reasonable to switch to interval simulation to avoid the use of output measures that are known to be generated in a faulty situation (see figure 5). Hence, the "output following" problem is avoided and the envelopes are insensitive to noise, providing persistent fault indicators.

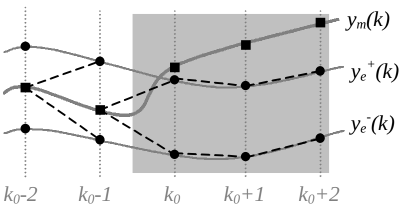

Figure 4. SCL strategy.

But noise is also present in the absence of faults and must be considered. Due to noise, a local incompatibility between prediction and observation at some instant k does not necessarily mean that the system is faulty. Hence, a more robust indicator than just a simple comparison of the system output with the envelopes should be used. When the output goes outside the envelopes, it is said to be *alarming*. But only when the output remains alarming during ν time instants the variable is said to be *misbehaving* (cf. figure 4) and the fault is reported. This temporal threshold ν introduces a delay but provides a more robust indicator.

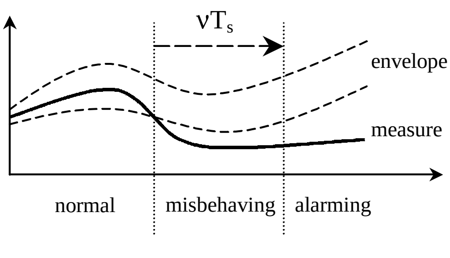

Figure 5. Robust fault indication.

The SCL, as it is implemented in Ca~En works as follows ([7]):

- while the system output remains inside the envelopes, 1-prediction is used and the output is labeled as normal.
- when the system output goes outside the envelopes, there is a switch to 1-simulation and the output is labeled as alarming.
- if the system output remains outside the envelopes for a whole interval time $T=\nu Ts$, the output is labeled as misbehaving and a fault is reported; if not, the local incompatibilities are considered as effect of noise and the output is labeled again as normal.

4.2 Improved SCL (ISCL) Strategy

Some improvements can be made to the described strategy. 1-simulation presents a low computational cost but can provide divergent envelopes for many models ([5]). In these cases, after some period of time, the envelopes will reach the output and it will be considered as normal. Then there will be a switch to interval prediction an the faulty output will probably go outside the envelopes again, reporting correctly the fault. This process will be repeated providing non persistent fault indications.

When the system output goes outside the envelope, the switch to 1-simulation is not the best option that can be used. If the output goes outside the envelopes at instant k_0, then the output is labeled as alarming, but output prediction can still be used. At time k_0+1, 2-prediction, which uses $y_m(k_0-1)$, can be used to avoid the use of the "suspicious" measure $y_m(k_0)$. This will provide a most restrictive interval for $y(k_0+1)$ than the obtained using 1-simulation. If the output remains outside the envelope at time k_0+1, then 3-prediction will be used at time k_0+2. This process is repeated using predictions which use the last measure inside the envelopes (see figure 6).

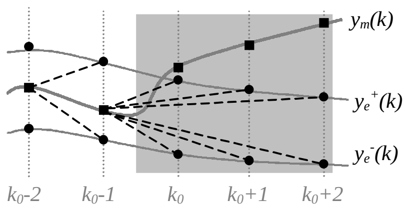

Figure 6. Improved SCL strategy.

If the output remains outside the envelopes after ν samples, then the output is labeled as *alarming*. Remaining the output outside the envelopes, when the window length equals the system response time there is a switch to interval simulation, using the optimal window length defined by this response time ([5]), to avoid an unnecessary computational effort.

For simplicity, explanations and figures in this section correspond to first order systems. But the explained strategy can be easily generalized to n^{th} order systems.

5 Application Example

5.1 DC motor description

Consider the schema of a DC motor with a mechanical load shown in figure 7.

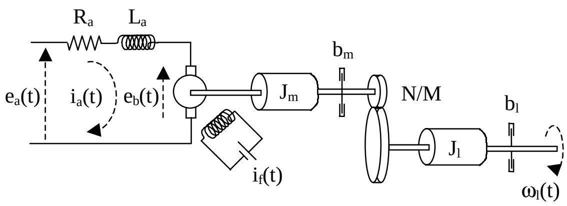

Figure 7. DC motor with mechanical load.

The physical meaning of the parameters and the values used in the example (considering uncertain load parameters) are the following:

- R_a: armature resistance; 1 Ω
- L_a: armature inductance; 0.5 H
- K_b: back emf constant; 0.01 V/rad/sec
- K_i: torque constant; 0.01 (N-m/amp)
- J_m: moment of inertia of the rotor; 0.01 kg-m^2
- b_m: damping ratio of the rotor; 0.1 N-m/rad/sec
- N/M: reduction constant; 1/10
- J_l: moment of inertia of the load; [20,30] kg-m^2
- b_l: damping ratio of the load; [10,15] N-m/rad/sec

In the armature controlled configuration (i_f generates a constant magnetic flow and the speed is controlled by the armature voltage), the transfer function, where the rotation speed ($w_l(t)$) is the output and the armature voltage ($e_a(t)$) is the input, is the following:

$$G(s)=\frac{\dfrac{K_i\left(N/M\right)}{L_aJ_{eq}}}{s^2+\dfrac{L_ab_{eq}+R_aJ_{eq}}{L_aJ_{eq}}+\dfrac{R_ab_{eq}}{L_aJ_{eq}}+K_iK_b} \tag{10}$$

where:

$$\begin{aligned}J_{eq}&=J_m+\left(N/M\right)^2J_l\\ b_{eq}&=b_m+\left(N/M\right)^2b_l\end{aligned} \tag{11}$$

5.2 Comparison of envelope generation methods

The behaviour of the different envelope generation options in a fault-free situation is studied in this section. In all the scenarios, the used input is a step of amplitude 5 volts at time 1 s. and sample time is T_s=0.5 s.

The results obtained using L-prediction using window lengths 1 (continuous), 3 (dashed) and 5 (dotted) are shown in figure 8. L-prediction envelopes grow when L increases, due to the fact that the uncertainty in the output increases across computation time.

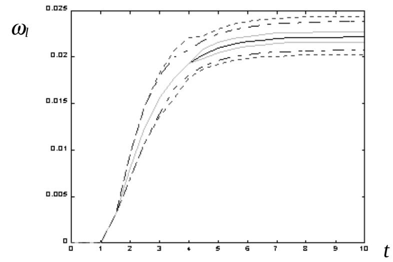

Figure 8. Interval prediction with L=1,3,5.

Results obtained using L-simulation with L equal to 5 (continuous), 3 (dashed) and 1 (dotted) are shown in figure 9. L-simulation envelopes grow when L decreases, due to the effect of temporal multi-incidence and wrapping problems; when L is l, L-simulation results in divergent envelopes.

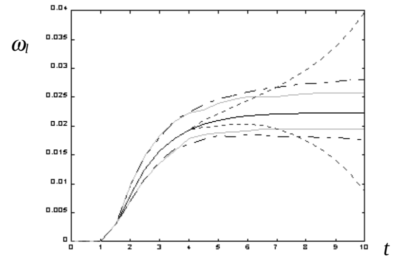

Figure 9. Interval simulation with L=1,3,5.

To study the effect of noise, white noise is added to the output. The results using interval prediction with L equal to 1 (continuous), 3 (dashed) and 5 (dotted) are shown in figure 10. The three small subplots show the fault indicator: 0 when no detection, 1 when a fault is detected. Note that 1-prediction provides many false alarms.

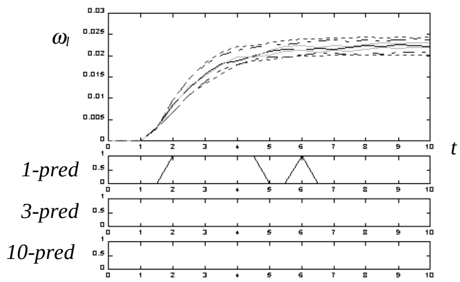

Figure 10. Interval prediction with L=1,3,5 (noise).

5.3 Fault detection

Three fault locations have been considered:

- actuator: bias equal to 1 volt (additive).
- process: 20% increment in torque constant (multiplic.).
- sensor: bias equal to $5*10^{-3}$ (additive).

For each location, two fault patterns have been considered:

- abrupt fault.
- drift fault: the fault reaches its size after 5 seconds.

In all these fault scenarios, the following conventions have been made:

- step input of amplitude 5 activated in t=1 s.
- the fault appears in t=2 s.
- single fault condition is assumed.

In figures 11 and 12, the proposed ISCL strategy (continuous), using ν=2, is compared with 1-prediction (dashed) and 10-simulation (dotted). It can be observed that 1-prediction provides a fast but non persistent fault indication. On the other hand, 10-simulation provides a persistent indication but with high detection time. The proposed strategy appears as a good compromise between these two, providing persistent indications with an acceptable detection time.

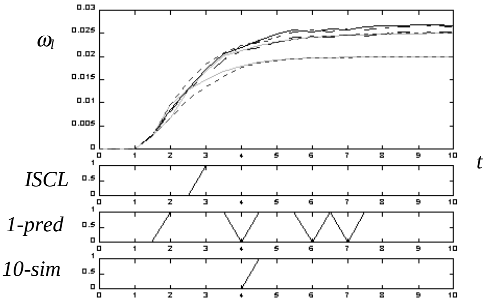

Figure 11. Abrupt actuator fault.

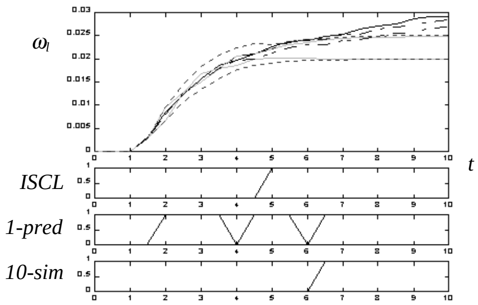

Figure 12. Drift actuator fault.

Persistency of the fault indicator can be taken into account using the definition for the detection time proposed in [9]: peridod of time the from fault time up to the moment of the last leading edge of the fault indicator (when the fault indicator is stabilised).

The results, using this criterium, in all fault scenarios are summarised in the following table:

Fault		**Detection time (sec.)**		
Location	**Pattern**	**1-Pred**	**10-Sim**	**ISCL**
actuator	abrupt	5.5	2.5	**1.0**
	drift	4.5	4.5	**3.0**
process	abrupt	4.5	**1.0**	**1.0**
	drift	4.5	**3.0**	**3.0**
sensor	abrupt	4.5	**0.5**	1.0
	drift	2.5	3.0	**1.0**

Table 1. Fault detection results.

6 Conclusions and Future Work

The proposed SCL-strategy has shown to be effective in the detection of faults when both uncertainty in the model and noise are present. This strategy is more robust against noise than the L-prediction and it avoids the fault-following problem. Moreover, it provides higher fault detectability and smaller detection times than the L-simulation strategy.

Envelope generation is obtained as the result of an optimization problem. The used algorithms, based on interval arithmetic combined with branch and bound strategies present a high computational cost, limiting their application for real-time applications. Alternative solutions should be studied in the near future.

Acknowledgments

This paper is partially supported by the Spanish CICYT under contract TAP99-0748, the Catalan CIRIT, (ref. 1999SGR00134) and by DAMADICS FP5 European Research Training Network (ref. ECC-TRN1-1999-00392).

References

[1] O. Adrot, D.Maquin and J. Ragot. “Bounding Approaches to Fault Detection of Uncertain Dynamic Systems”, *4th IFAC Safeprocess*, **2**, pp.1168-1173, (2000).

[2] J. Armengol, J. Vehi, L. Travé-Massuyès and M.A. Sainz. “Semiqualitative simulation using Modal Interval Analysis”, *14th IFAC World Congress*, **O**, pp.521-526, (1999).

[3] J.J. Gertler. “Fault Detection and Diagnosis in Engineering Systems”, Marcel Dekker, (1998).

[4] E. Hansen. “Global Optimization using Interval Analysis”, Marcel Dekker, (1992).

[5] V. Puig, J. Saludes and J. Quevedo. “A new algorithm for adaptive threshold generation in robust fault detection based on a sliding window and global optimisation”. *6th ECC*, (1999).

[6] V. Puig, J. Quevedo, T. Escobet and S. de las Heras “Passive Robust Fault Detection Approaches using Interval Models”. *15th IFAC World Congress*, (2002).

[7] S. Tornil, T. Escobet and V. Puig. “Fault Detection using Interval Models”, *4th IFAC Safeprocess*, **2**, pp.1180-1186, (2000).

[8] L. Travé-Massuyes, T. Escobet, R. Pons and S. Tornil. “The Ca-En Diagnosis System and its Automatic Modelling Method”, *Computación y Sistemas*, 5, 2, pp. 128-143 (2001).

[9] DAMADICS Team. “Benchmark Definition”. In http://diag.mchtr.pw.edu.pl/damadics/