FAULT DETECTION AND ISOLATION USING MULTIPLE SLIDING TIME WINDOWS AND INTERVAL MODELS

Joaquim Armengol * **Louise Travé-Massuyès** **
Josep Vehí * **Miguel Ángel Sainz** ***

* *Institut d'Informàtica i Aplicacions. Universitat de Girona & LEA-SICA. Campus Montilivi. E-17071 Girona, Spain.*
Email:armengol@eia.udg.es
** *LAAS-CNRS & LEA-SICA. Toulouse, France.*
*** *Departament d'Informàtica i Matemàtica Aplicada. Universitat de Girona. Girona, Spain.*

Abstract: The uncertainty in the systems and in the measurements has to be considered somehow to detect faults by analytical redundancy. This paper considers them by means of interval models and interval measurements. The consistency between them is checked and a fault is detected when there is an inconsistency. The used technique is based on Modal Interval Analysis and saves much computational effort compared to other techniques based on global optimization algorithms. When a fault is detected, the same technique is used to isolate the fault. Even when the fault is not clearly isolated, interesting information about the direction of the changes in the system is obtained.

Keywords: Difference equations, Envelopes, Fault detection, Global optimization, Intervals, Uncertain dynamic systems.

1. INTRODUCTION

A fault is a malfunction in a system, that may have consequences like economical losses derived from lower efficiency of the system or danger for the people or the environment. Many different techniques have been developed in the recent years which intend to detect the faults (Chen and Patton, 1998; Frank *et al.*, 2000). Among these techniques, there are heuristic approaches, which are based on rules or cognitive methods, and analytical approaches, which are based on a model of the system. In the latter case tools like identification or estimation are used.

One way to detect faults is comparing the behaviour of the real system and a reference one. A fault is detected when there are discrepancies between them (Reiter, 1987):

$$\begin{aligned} y(t) &\neq y_r(t) \\ y(t) - y_r(t) &\neq 0 \\ |y(t) - y_r(t)| &> 0 \end{aligned} \tag{1}$$

where $y(t)$ is the value of a variable of the system at time t and $y_r(t)$ is the corresponding reference one. This is a sufficient condition, but not a necessary one. Being the system faulty, possibly there are time points in which the two systems behave the same.

This reference behaviour can be obtained from another system. This is physical redundancy. It can also be obtained from a model of the system. This is analytical redundancy. The results of the fault detection are highly dependent on the model in this case. The main problem is that these two behaviours are never exactly the same because the model is inaccurate, i.e. it is an approximate representation of the system. This is the consequence of the uncertainties of the system and the procedure of systems modeling, which usually involves hypotheses, assumptions, simplifications, linearizations, etc.

Therefore, the uncertainty of the system has to be considered. It can be taken into account when the comparison between the behaviour of the real system and the one of its model is performed. In this case, a fault is indicated when the difference is larger than a threshold:

$$|y(t) - y_r(t)| > \epsilon \tag{2}$$

The difficult thing then is to determine the size of the threshold. If it is too small, faults are indicated even when they do not exist. These are false alarms. On the other side, if the threshold is too large, there can be faults which are not indicated. These are missed alarms.

Another way to take the uncertainty into account is considering it in the modeling procedure. Interval models, which are introduced in section 2, can represent the uncertainties associated to the systems. There is also uncertainty in the measurements. It can be considered by using interval measurements, which are also described in section 2.

The reference behaviour for fault detection is obtained by simulation of the interval model. This problem of simulation is reformulated as a global optimization problem (Tibken and Hofer, 1995). This is a hard problem but can be softened using error-bounded estimations and time windows, which are introduced in sections 3 and 4, respectively. This way the necessary computational effort is much lower and the fault detection results are even better.

Once the fault is detected, the important task is to isolate the fault, i.e to indicate which component is faulty. The technique that has been used is described in section 5. Section 6 shows fault detection and isolation results in an academic example. Finally, section 7 summarizes this work and provides some conclusions.

2. INTERVAL MODELS AND INTERVAL MEASUREMENTS

An usual model consisting of functions with real-valued parameters is precise but inaccurate as it does not include uncertainty. Another option is an interval model, where the values of the parameters are intervals, which is imprecise but may be accurate. An interval model is a set of models indeed. For instance, assume that the behaviour of a n-th order dynamic SISO (Single Input, Single Output) system is represented by the following difference equation:

$$y_t = \sum_{i=1}^{m+1} a_i y_{t-iT} + \sum_{j=1}^{p+1} b_j u_{t-jT} \tag{3}$$

This equation shows that the output of the system at any time point (y_t) depends on the values of the previous outputs (y_{t-iT}) and inputs (u_{t-jT}), being T the sampling time. This dependency is given by the parameters of the system model (a_i and b_j), which can be expressed by means of intervals if they are uncertain.

The simulation of a real-valued model produces a trajectory for each output variable which is a curve representing the evolution of the variable of the system across time: $y_r(t)$. In the case of an interval model, as it is a set of models indeed, a set of curves represents the evolution of each variable: $Y_r(t) = [\min(y_r(t)), \max(y_r(t))]$. This set of curves is called envelope. Then, there is a fault when

$$\begin{gathered} y(t) \not\subset Y_r(t) \\ y(t) \cap Y_r(t) = \emptyset \end{gathered} \tag{4}$$

In fact, the value of the variable $y(t)$ is not available. If it can be measured, the measurement has an inaccuracy due to the uncertainties associated to the measuring procedure:

$$y_m(t) \neq y(t) \tag{5}$$

If this inaccuracy is not considered, false alarms can be generated. One option to take this inaccuracy into account is including the associated uncertainties in the model. Another option is to use the knowledge about these uncertainties to eliminate the inaccuracy by introducing imprecision, for instance using interval measurements:

$$Y_m(t) \supseteq y(t) \tag{6}$$

The disadvantage of the first option is that the model becomes more imprecise and the imprecision of the model is propagated across the time. The problem of the second option is that a source of uncertainty in the measurements is noise, which usually has a gaussian probability distribution. This implies that in some cases, depending on the relation between the width of the interval and the standard deviation of the noise, the real value of the variable is not included in the interval measurement:

$$Y_m(t) \not\supset y(t) \tag{7}$$

Using interval measurements, a fault is detected when

$$Y_m(t) \cap Y_r(t) = \emptyset \quad (8)$$

To compute the envelope is necessary to compute the range of a function in a parameter space at each simulation step, and range computation is a task related to global optimization, which usually needs an important computation effort. Moreover, if the system is considered time invariant, the function corresponding to a simulation step is larger than the function corresponding to the previous one, that is, the number of parameters increases and hence the dimensions of the parameter space increases too (Armengol *et al.*, 2000).

Therefore, the simulation of interval models is a very hard task in most of the cases because of these two problems: the computation of the envelope at one simulation step and the computation of the envelope at every simulation step. However, simulation is not the main goal. It is only a step in the fault detection procedure. In the following, it will be shown that similar results can be obtained at a lower cost.

3. ERROR-BOUNDED ESTIMATIONS

A fault is detected when

$$Y_m(t) \cap Y_r(t) = \emptyset \quad (9)$$

It is also detected if

$$Y_m(t) \cap Y_{rex}(t) = \emptyset \quad (10)$$

being $Y_{rex}(t)$ an external estimation of $Y_r(t)$, i.e.

$$Y_{rex}(t) \supseteq Y_r(t) \quad (11)$$

The differences between using $Y_{rex}(t)$ and $Y_r(t)$ are the following ones:

- $Y_{rex}(t)$ usually is much easier to obtain than $Y_r(t)$. For instance, $Y_{rex}(t) = [-\infty, +\infty] \supseteq Y_r(t)$ is always true.
- $Y_{rex}(t)$ detects less faults than $Y_r(t)$. If $Y_m(t) \cap Y_r(t) = \emptyset$ and $Y_m(t) \cap Y_{rex}(t) \neq \emptyset$, there is a missed alarm. For instance, $Y_{rex}(t) = [-\infty, +\infty]$ does not detect faults, so is useless.

A useful tool to obtain external estimations of the range of a function in a parameter space is the interval arithmetic (Moore, 1966), due to its monotonic inclusion property. An interval function is inclusive monotonic if $\boldsymbol{X} \subset \boldsymbol{Y}$ implies $F(\boldsymbol{X}) \subset F(\boldsymbol{Y})$, being $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$. Interval arithmetic operations are inclusive monotonic and so is the natural extension of a rational function, i.e. the one obtained by substituting each real variable by the corresponding interval one and the rational operations by the corresponding interval ones (Moore, 1979):

$$R_f(\boldsymbol{X}) \subseteq FR(\boldsymbol{X}) \quad (12)$$

where $R_f(\boldsymbol{X})$ is the range of the function f and $FR(\boldsymbol{X})$ is its natural extension.

Moreover, better approximations can be obtained splitting the domain:

$$R_f(\boldsymbol{X}) \subseteq FR(\boldsymbol{X}_1) \cup FR(\boldsymbol{X}_2) \subseteq FR(\boldsymbol{X}) \quad (13)$$

being $\boldsymbol{X}_1 \cup \boldsymbol{X}_2 = \boldsymbol{X}$

This can be used in an iterative procedure:

(1) DO Calculate external estimation $Y_{rex}(t)$
(2) IF $Y_m(t) \cap Y_{rex}(t) = \emptyset$ THEN
(3) Fault is detected
(4) ELSE
(5) Split parameter space
(6) ENDIF
(7) WHILE Fault is not detected

This iterative procedure calculates closer external estimations of the range of the function at each iteration. After infinite iterations it would calculate the exact range, but it stops when the estimation is sufficiently close to detect the fault, thus saving much computational effort in case that $Y_m(t) \cap Y_r(t) = \emptyset$. However, if there is not a fault or $Y_m(t) \cap Y_r(t) \neq \emptyset$, it never stops. This drawback can be overcome by using an internal estimation of $Y_r(t)$:

$$Y_{rin}(t) \subseteq Y_r(t) \quad (14)$$

because if $Y_m(t) \cap Y_{rin}(t) \neq \emptyset$ then $Y_m(t) \cap Y_r(t) \neq \emptyset$ and consequently it is known that the fault, if it exists, will never be detected.

Therefore, the simultaneous use of $Y_{rex}(t)$ and $Y_{rin}(t)$ obtains the same fault detection results that $Y_r(t)$ but with a much lower computation effort. $Y_{rex}(t)$ and $Y_{rin}(t)$ form an error-bounded estimation of $Y_r(t)$ because although $Y_r(t)$ is not known, it is known that

$$Y_{rin}(t) \subseteq Y_r(t) \subseteq Y_{rex}(t) \quad (15)$$

A useful tool to compute error-bounded estimations of the range of a function in a parameter space is Modal Interval Analysis (Gardeñes *et al.*, 1986; Gardeñes and Mielgo, 1986; SIGLA/X, 1999), which is an extension of the interval arithmetic. The Modal Interval Analysis allows the iterative computation of these estimations by using efficient branch-and-bound algorithms in which more and larger subspaces are eliminated compared to other techniques (Armengol *et al.*, 2001).

4. TIME WINDOWS

In simulation, the goal is to predict the future states of a system knowing some initial one and the inputs to the system. Therefore, as the simulation goes on, the time distance between the time point which is being predicted and the initial one is continuously increasing. In the case of interval models, this means that the computing effort is also increasing and, at some time point, the problem becomes intractable.

In fault detection, data from the system are needed to compare the real system behaviour and the reference one obtained analytically. Therefore, any time point can be considered as an initial one and the prediction of the value of a variable at a time point t ($Y_r(t)$) can be calculated starting from the initial time point $t_0 = 0$ ($Y_r(t|t_0)$) or from any other time point $0 < t_i < t$ ($Y_r(t|t_i)$). So, the necessary computing effort can be limited by fixing a maximum distance $w = t - t_i$. If this distance is given a constant value, it is said that a sliding time window of length w is being used: $Y_r(t|t-w)$, $Y_r(t+1|t+1-w)$, ...

The necessary computing effort depends on the value of w: if w is larger, this effort is larger too. The results are also different: tighter or wider envelopes depending on the window length, the measurements, etc. So it can happen that $Y_m(t) \cap Y_{rex}(t|t-w_1) = \emptyset$ and $Y_m(t) \cap Y_{rex}(t|t-w_2) \neq \emptyset$, i.e. w_1 detects a fault and w_2 does not. In this case it can be affirmed that there is a fault because $Y_m(t) \cap Y_{rex}(t) = \emptyset$ is a sufficient condition to do so. Consequently, there is a fault if

$$Y_m(t) \cap Y_{rex}(t|t-1) \cap Y_{rex}(t|t-2) \cap \ldots \quad (16)$$
$$\ldots \cap Y_{rex}(t|0) = \emptyset$$

The fault detection results obtained using several window lengths are obviously 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.

As the necessary computing effort to calculate $Y_{rex}(t|w_1)$ is larger than the one to calculate $Y_{rex}(t|w_2)$ when $w_1 > w_2$, at each time point the fault detection algorithm starts using the shortest window length and stops when a fault is detected, thus saving computing effort and minimizing the rate of missed alarms. The maximum used window length depends on the available computing power and the complexity of the model. The basic algorithm, in a more formal way, is the following one:

(1) $A_0 = Y_m(t)$
(2) FOR w FROM 1 TO w_{max}
(3) $\quad A_w = A_{w-1} \cap Y_{rex}(t|t-w)$
(4) $\quad$ IF $A_w = \emptyset$ THEN
(5) $\quad\quad$ Fault is detected
(6) $\quad\quad$ $w = w_{max}$
(7) $\quad$ ENDIF
(8) ENDFOR

This method has been used in academic and real examples. There are not false alarms if the interval model represents adequately the uncertain system and the interval measurements represent adequately the uncertain measurements, except in the spurious case of equation 7. If in any other case an indication of fault is known to be a false alarm, it can be used to refine the interval model or the interval measurements.

5. FAULT ISOLATION

Fault isolation consists in determining the exact location of the fault. Many detection systems have a detection task running permanently and trigger the diagnostic task when a fault is detected (Gertler, 1998). This diagnostic task includes very often a set of models describing the behaviour of the system when different faults are present and the goal of the task is to select a model, or a set of models, that behaves like the faulty model does.

This technique has been applied in this work. The detection task uses the interval model of the system as reference. If the behaviour of the actual system is not consistent with the behaviour of the model, then a fault is detected and the isolation task is triggered. The isolation task uses the same algorithms that are used by the detection task, but the interval models are different. In this case, the discrepancies between the system and the model mean that the model is not representing the behaviour of the system so is not a model of the present fault. The fault is clearly isolated when only a model is consistent with the system. If several models are consistent with the system, then the isolation is not clear. Finally, if a fault that has not been taken into account appears, possibly none of the models will be consistent with the system.

The next section shows an academic example where a fault appears and is clearly detected and isolated.

6. ACADEMIC EXAMPLE

Consider a system that can be modeled using a generic first order model:

$$y_n = \left(1 - \frac{T}{\tau}\right) y_{n-1} + \frac{kT}{\tau} u_{n-1} \quad (17)$$

with the following parameters, which are intervals in some cases:

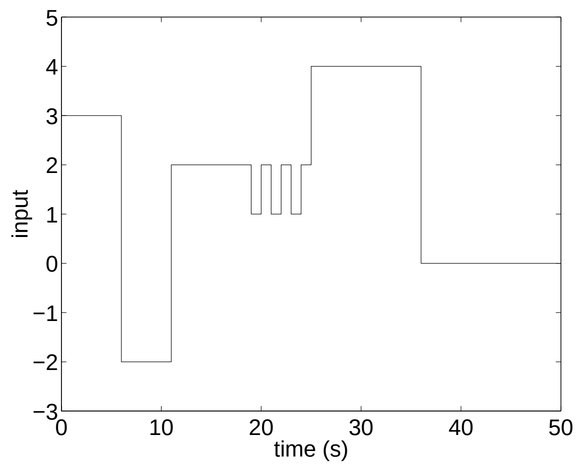

Fig. 1. Input to the first order system.

- static gain: $k = [0.95, 1.05]$.
- time constant: $\tau = [10, 20]$ s.
- initial state: $y_0 = [0, 0]$.
- sampling time: $T = 1$ s.
- input: a sequence of steps of different lengths and heights shown in figure 1.

Assume that the actual system is faulty:

- static gain: $k = 1.15$ ($1.15 \not\subseteq [0.95,\ 1.05]$).
- time constant: $\tau = 5$ s ($5 \not\subseteq [10,\ 20]$).

It is considered that the measurements of the output of the system have an associated uncertainty (noise, analog to digital conversion errors, etc.). To avoid problems of the kind of equation 7, it is assumed that this uncertainty can be generated adding a random number between -0.1 and 0.1 to the exact value of the output variable:

$$y_m(t) = y(t) + rnd([-0.1, 0.1]) \qquad (18)$$

This uncertainty is taken into account converting the real-valued measurements into interval measurements. In this case, as the difference between the measurements and the actual values of the variables is known to be in the interval [-0.1, 0.1], the interval measurements are:

$$Y_m(t) = y_m(t) + [-0.1, 0.1] \qquad (19)$$

The detection task using a window length $w = 5$ s obtains the error-bounded estimations of the envelope shown in figure 2, where $Y_{rex}(t|5)$ is represented in solid line and $Y_{rin}(t|5)$ is represented in dashed line.

It may be observed that there are several time points where the distance between the external estimation and the internal one is large. The reason is that at one of the first iterations of the algorithm already has been seen that $Y_m(t) \cap Y_{rin}(t) \neq \emptyset$ so it is not necessary to obtain better (closer) estimations of $Y_{rex}(t)$ and $Y_{rin}(t)$. In these cases, the detection results are obtained with a small computation effort.

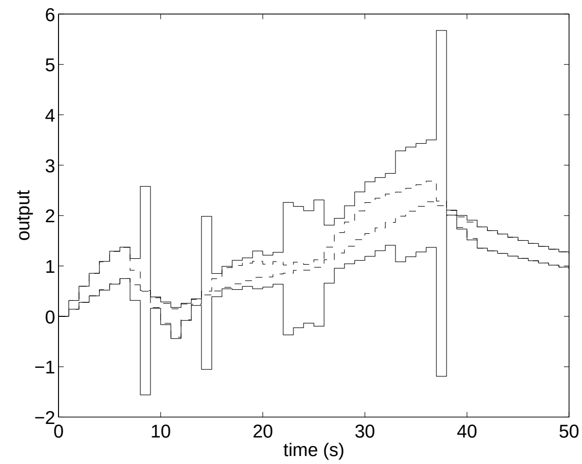

Fig. 2. Error-bounded envelopes for window length $w = 5$ s.

Figure 3 represents the fault detection results for different window lengths alone. In this figure *1* means that there are discrepancies and hence a fault has been detected, while *0* means that there are not discrepancies so there is not fault or, if it exists, it can not be detected, at least using these tools. This figure shows that, although the system is faulty during the whole time interval, it is detected only at some time points. It shows also that, as stated in section 4, the results using different window lengths alone are different. When different sliding time windows are used, the result at a time point is *1* if any of the windows indicates *1* and *0* if all windows indicate *0*, so the use of several sliding time windows enhance the overall results and decreases the amount of missed alarms.

Once the fault has been detected, the isolation task is triggered. In this case, the models that have been introduced in the isolation task are interval models around the nominal one in the two-dimension space determined by the two interval parameters k and τ. Table 1 represents the results of the isolation task by indicating the amount of detected inconsistencies in 50 s between the interval models and the measurements using

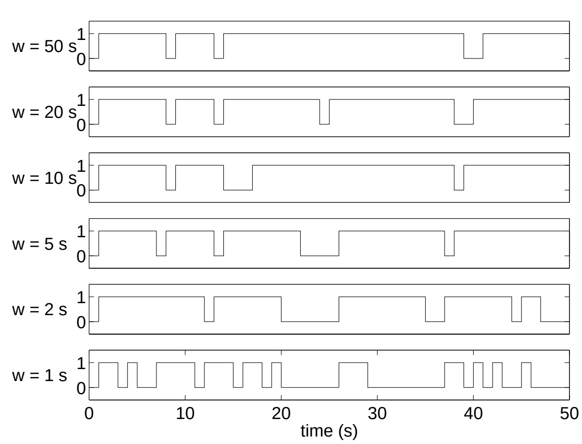

Fig. 3. Fault detection results.

		k		
		[0.5, 0.95]	[0.95, 1.05]	[1.05, 1.5]
τ	[2, 10]	13	2	0
	[10, 20]	103	99	62
	[20, 40]	109	107	99

Table 1. Inconsistencies between the interval models and the measurements.

sliding time windows of lengths 1, 2 and 5 s. As the model of the non-faulty system ($k = [0.95, 1.05]$ and $\tau = [10, 20]$ s) is not consistent with the measurements, the fault is detected. In this case, all the models of the system faults, except one, also are not consistent with the measurement. The exception is the isolation of the fault: the only model that is consistent with the measurements is $k = [1.05, 1.5]$ and $\tau = [2, 10]$ s, which is a correct isolation of the fault because the faulty system is $k = 1.15$ and $\tau = 5$ s: $1.15 \subseteq [1.05, 1.5]$ and $5 \subseteq [2, 10]$).

The table also shows the direction of the change in the value of the parameters: the values of both k and τ have decreased. This is a useful information to refine the isolation in case it is not clear or the fault consists in a drift in some parameter, for instance.

7. CONCLUSIONS

A method for fault detection and isolation for systems with parametric uncertainties is presented. The uncertainty of the systems is represented using interval models and the uncertainty associated to the measurements is also represented by means of intervals. The consistency between the model and the interval measurements is checked using a branch-and-bound algorithm based on Modal Interval Analysis, which increases its efficiency. A fault is detected when there is any inconsistency in any time window. A false alarm indicates that either the interval model of the system or the interval measurements do not represent adequately the corresponding uncertainties. If the representation is adequate, there are not false alarms except in some spurious cases.

Once the fault is detected, the same method is used to isolate the fault by using interval models of the system with different faults. The fault is isolated when one, or more than one, of these models is consistent with the measurements. If the isolation is not clear, the amount of discrepancies between each of these models and the measurements indicates the direction of the changes, which is also a useful information.

8. ACKNOWLEDGEMENTS

LEA-SICA is an associated European laboratory constituted by three Spanish and two French research groups.

This work has been funded by the autonomous government of Catalonia, Generalitat de Catalunya, through Grup de Recerca Consolidat ref. 1999SGR00161 and by the Spanish government through co-ordinated research projects CICYT TAP99-1079-C03-03 and DPI2000-0666-C02-01.

9. REFERENCES

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

Armengol, J., J. Vehí, L. Travé-Massuyès and M. Á. Sainz (2001). Application of modal intervals to the generation of error-bounded envelopes. *Reliable Computing* **7**(2), 171–185.

Chen, J. and R.J. Patton (1998). *Robust model-based fault diagnosis for dynamic systems.* Kluwer.

Frank, P. M., S. X. Ding and B. Köppen-Seliger (2000). Current developments in the theory of FDI. *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2000). Budapest, Hungary* pp. 16–27.

Gardeñes, E. and H. Mielgo (1986). Modal intervals: functions. *Polish Symposium on Interval and Fuzzy Mathematics. Poznan, Poland.*

Gardeñes, E., H. Mielgo and A. Trepat (1986). *Interval Mathematics.* Chap. Modal intervals: reasons and ground semantics, pp. 27–35. Vol. 212 of *Lecture Notes in Computer Sciences.* Springer. Heidelberg.

Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker.

Moore, R. E. (1966). *Interval analysis.* Prentice-Hall.

Moore, R. E. (1979). *Methods and applications of interval analysis.* Studies in Applied Mathematics (SIAM).

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence* **32**(1), 57–95.

SIGLA/X (1999). *Applications of Interval Analysis to Systems and Control. Proceedings of MISC 1999.* Chap. Modal Intervals, pp. 157–227. Universitat de Girona.

Tibken, B. and E. P. Hofer (1995). Simulation of controlled uncertain nonlinear systems. *Applied Mathematics and Computation* **70**(2-3), 329–338.