INTERVAL MODEL-BASED FAULT DETECTION USING MULTIPLE SLIDING TIME WINDOWS

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

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

Abstract: Interval models may be used in many cases to express the imprecisions and uncertainties of the systems. Envelopes are a way to represent the results of the simulation of these models. One of their applications is as reference behaviour for Fault Detection (FD) based on analytical redundancy, so their properties (completeness, soundness) have important consequences on the FD results (missed and false alarms). This paper presents the Modal Interval Simulator (MIS) that approaches this problem by means of error-bounded envelopes. Sliding time windows have to be used for long simulations, depending the adequate window length on the type of the fault. MIS allows to the user to work with several window lengths simultaneously.

Keywords: Envelopes, Fault detection, Intervals, Simulation, Uncertain dynamic systems.

1. INTRODUCTION

The mathematical quantitative models, i.e. models in which the values of the parameters are real numbers, are simplifications of the reality and hence the behaviour obtained by simulation of these models differ from the real one. The use of complex quantitative models is not a solution because in most cases there are uncertainties and imprecisions in the system which can not be represented with this kind of models. A way to represent them is by using qualitative or semiqualitative models. A model of this kind represents a set of models indeed.

The simulation of the behaviour of quantitative models provides a single trajectory across time for each output variable. This can not be the result of the simulation when a set of models is used. In this case, one way to represent the behaviour is by means of envelopes.

2. ENVELOPES

In geometry, an envelope is defined as the curve that is tangent to each member of a system of curves. In this work, the set formed by the upper envelope and the lower one is referred to as envelope and the set of curves is formed by all the possible trajectories across the time of the output of the system. Two properties of this envelope are:

- Completeness: it includes all the possible behaviours of the model.
- Soundness: every point inside the envelope belongs to the output of at least one instance

of the model, i.e. one of the quantitative models that belongs to the set of models.

Notice that this is the terminology used in (Struss, 1990). Some authors use the same words in the opposite sense (Weld and de Kleer, 1990).

3. GENERATION OF ENVELOPES

Assume that a particular system may be modelled by means of a difference equation. For instance, a n-th order dynamic SISO (Single Input, Single Output) system can be modelled by the following generic difference equation:

$$y_k = \sum_{i=1}^{n} a_i y_{k-i} + \sum_{j=1}^{p} b_j u_{k-j} \tag{1}$$

This equation shows that the output of the system at any time point (y_k) depends on the values of the previous outputs (y_{k-i}) and inputs (u_{k-j}). This dependence is given by the parameters of the system model (a_i and b_j).

There may exist some uncertainty in the parameters of the model, in the input or in the initial output y_0, which can be represented by means of intervals. In this case the difference equation can also be seen as the expression of a function into a parameter space that has the shape of a hypercube and its number of dimensions is the sum of interval parameters appearing in the function: inputs, outputs and parameters of the model.

For instance, in the simplest case when $n = 1$ and $p = 1$, i.e. the system is a first order one, the functions of the first steps are:

$$\begin{aligned} y_1 &= a_1 y_0 + b_1 u_0 \\ y_2 &= a_1 y_1 + b_1 u_1 \\ y_3 &= a_1 y_2 + b_1 u_2 \end{aligned} \tag{2}$$

One way to compute the limits of the envelope at a time point t is determining the range of the function of this time point in its parameter space. If the system is considered time variant, these functions can be considered independent. However, if the system is considered time invariant, there are parameters and variables that appear in more than one equation so there are dependencies that must be considered. The proposed approach is to make the multi-incidences explicit by merging the different equations starting from 0 into a unique expression on which the range is computed. This expression is obtained in a recursive way by substituting y_k within the previous equation down to y_0. In this case, the range of the following functions has to be determined:

$$\begin{aligned} y_1 &= a_1 y_0 + b_1 u_0 \\ y_2 &= a_1^2 y_0 + a_1 b_1 u_0 + b_1 u_1 \\ y_3 &= a_1^3 y_0 + a_1^2 b_1 u_0 + a_1 b_1 u_1 + b_1 u_2 \end{aligned} \tag{3}$$

Therefore, the length of the function y_k is larger when k is larger. Moreover, usually the parameters a_i and b_j are functions of physical parameters and there are non-linearities in the model. All these things make the determination of the range of the function a hard task. It may be approached, for instance, using algorithms of global optimisation or consistency checking.

In many cases, only approximations to the range are obtained and hence the result is an approximation to the envelope. Then, in a broader sense, the envelope that is complete and sound is referred to as the exact envelope whereas a complete (resp. sound) approximation to the exact envelope is referred to as a complete (resp. sound) envelope. Furthermore, a complete but not sound envelope is called overbounded and a sound but not complete envelope is called underbounded.

4. ENVELOPES AND FAULT DETECTION

One possible use of the envelopes is for Fault Detection (FD) by means of analytical redundancy. A fault is an unexpected change in a system, such as a component malfunction and variations in the operating condition, that tend to degrade overall system performance (Chen and Patton, 1998). Consequences of degradation are not only economic loss, they can be extremely serious in terms of environmental impact or danger for the population. In order to design a reliable, fault tolerant system, or to maintain a high level of performance for complex systems it is crucial that such changes are detected promptly and diagnosed so that correcting action can be taken to reconfigure the system and accommodate the change.

Redundancy is a widely used technique to detect faults. It consists in comparing the behaviour of the system with a reference one. The discrepancies between these two behaviours indicate a change in the behaviour of the system, i.e. a fault. In the case of analytical redundancy the reference behaviour may be obtained through simulation of a model of the system.

When envelopes are used for FD, the system is guaranteed to be faulty when the measured output of the system is outside of the envelope (Travé-Massuyès *et al.*, 1997). However, the system may also be faulty when the measure is inside the envelope. This is due to the dynamics of the system. The measure can remain inside the envelope for some time after a fault has occurred or even it never goes outside, for instance if the fault only lasts for a short time. In these cases the fault is not detected or, if it is detected, some time, depending on the distance between the actual system parameter values and their nominal values, is needed to detect the fault. If this distance is small, the time is larger.

This is what would happen if the exact envelope was used. If the envelope used for fault detection is overbounded there can be missed alarms: the system is faulty but it is not detected because the measured output remains inside the envelope. If the envelope is underbounded then there can be false alarms: the system is said to be faulty, although it is not, because the measure goes out of the envelope. Therefore, these properties of the envelopes are very important when they are used to detect faults. These properties are assessed for several simulators for uncertain systems in (Armengol *et al.*, 2000).

5. ERROR-BOUNDED ENVELOPES

The approach proposed in this paper is the use of error-bounded envelopes. This consists in the simultaneous computation of an underbounded and an overbounded envelope. These two envelopes determine three zones: the inner zone included in the underbounded envelope, the intermediate zone between the two envelopes and the outer zone outside of the overbounded envelope. If the measure is outside of the overbounded envelope, the system is guaranteed to be faulty. If the measure is inside the underbounded envelope nothing can be said because either the system is not faulty or if the system is faulty it can not be detected using these tools. Finally, if the measure is in the intermediate zone between the two envelopes the situation is one of the two that have been presented above but, to decide which one of them, better approximations to the exact envelope are needed.

An iterative algorithm to compute the two envelopes has been designed. It is a branch and bound algorithm that tightens the overbounded envelope and widens the underbounded one at every iteration. The stopping condition is given by the location of the measured output of the system. If it is in the outer zone, the algorithm stops because it has already been detected that the system is faulty and hence it is not necessary to compute better approximations of the envelopes. If it is in the inner zone, the algorithm also stops because the measurement would remain in this zone even if better approximations would be computed. Finally, the algorithm keeps iterating if the measurement is in the intermediate zone.

This algorithm is based on an interval model, that is a model in which the parameters may take interval values instead of real numbers, and on Modal Interval Analysis (MIA).

MIA (SIGLA/X, 1999) is an extension of classical interval analysis. The main difference is that interval analysis identifies an interval by a set of real numbers, whereas MIA identifies an interval by the set of predicates that are fulfilled by the real numbers. Then, a modal interval is defined by a pair

$$X := (X', QX) \tag{4}$$

In this pair, X' is called the *extension*, $X' \in I(\mathbb{R}) = \{[a,b]' \mid a,b \in \mathbb{R}, \quad a \leq b\}$. QX is the *modality*, $QX \in \{E, U\}$. The existential modality E indicates that at least one element in the interval fulfils a predicate whereas the universal modality U indicates that every element in the interval fulfils it.

The dual formulation of the modal intervals allows the definition of two modal interval extensions of functions, noted by $f^*(X)$ and $f^{**}(X)$ respectively, which provide meaning to the interval computations. In the case of the envelopes, it can be said that the *-extension is complete and the **-extension is sound. Therefore, when both extensions are equal the result is exact.

Unfortunately, the computation of the *- and **-extensions is, in general, a difficult challenge. MIA provides tools to find overbounded computations of $f^*(X)$ and underbounded computations of $f^{**}(X)$ which maintain the semantic interpretations. These computations are made taking into account the multi-incident variables in the functions, which is a source of overbounded results when interval arithmetic is applied. The coercion theorems provide the conditions and the way to obtain optimal extensions when there is monotonicity. If the function is not monotonic for each multi-incident component, these theorems can be partially applied in order to reduce the complexity of the problem. Finally, a way to obtain even better approximations is by splitting the parameter space.

These features have been implemented in the branch and bound algorithm, which is programmed in C++. The simulator MIS (Modal Interval Simulator), which is based on this algorithm, is implemented in Matlab and also uses Maple.

6. AN ACADEMIC EXAMPLE

MIS has been used to simulate the behaviour of a generic first order system

$$y_n = \left(1 - \frac{T}{\tau}\right) y_{n-1} + \frac{kT}{\tau} u_{n-1} \tag{5}$$

with the following parameters:

- static gain: $k = [0.95, 1.05]$
- time constant: $\tau = [5, 20]\ s$
- initial state: $y_0 = [0, 0]$.
- input: steps of different lengths and heights.
- sampling time: $T = 1\ s$.

This simulation is used to detect that the generic first order system

$$y_n = \left(1 - \frac{T}{\tau(t)}\right) y_{n-1} + \frac{k(t)\,T}{\tau(t)} u_{n-1} \quad (6)$$

is faulty when its parameters degrade with time:

- Static gain:

$$k(t) = k_0 + m_k t \quad (7)$$

  with $k_0 = 1.05$ and $m_k = -0.001\ \frac{1}{\mathrm{s}}$.
- Time constant:

$$\tau(t) = \tau_0 + m_\tau t \quad (8)$$

  with $\tau_0 = 5$ s and $m_\tau = -0.04\ \frac{\mathrm{s}}{\mathrm{s}}$.

As the nominal values are $k = [0.95, 1.05]$ and $\tau = [5, 20]$ s, initially the system is not faulty and the time constant gets more and more far from its nominal value with time.

Figure 1 shows the results of the simulation. The upper half of the figure shows the error-bounded envelopes (the overbounded envelope in solid line and the underbounded envelope in dashed line) jointly with the output of the faulty system (crosses). The lower half of the figure shows the FD results indicated in the following way:

- $fault = 1$. The measurement is in the outer zone. The fault is detected.
- $fault = 0$. The measurement is in the inner zone. The fault is not detected.

It may be observed that there are several time points where the overbounded envelope is much

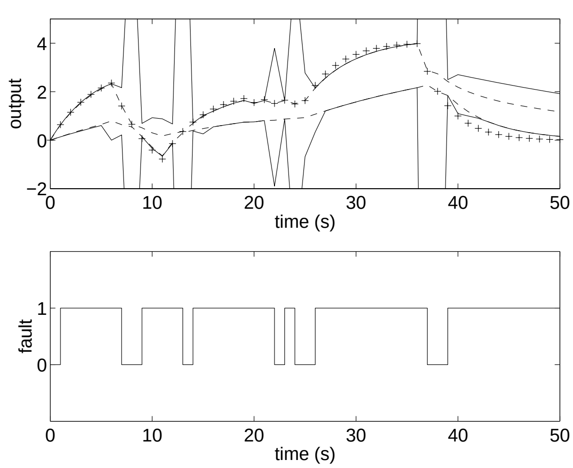

Fig. 1. Fault detection using error-bounded envelopes.

overbounded. In fact, the overbounded envelope is outside the figure at some of these points because of the scale that has been used. At these points the overbounded envelope is so overbounded because at one of the first iterations of the algorithm already has been seen that the measurement is in the inner zone so it is not necessary to obtain better (less overbounded) approximations of it. This means that these results have been obtained with a small computational effort.

7. SLIDING TIME WINDOWS

It has been seen above that if the parameters of the system are assumed to be uncertain but time invariant, the envelopes at each time step have to be computed starting from the initial state. This is the way to assure that each parameter of the model varying within an interval is kept at the same value at every step of the simulation. The drawback of this method is that each step of the simulation needs more and more computations. This makes the algorithm unusable for long simulations or on-line simulations.

An approach to this problem is the use of sliding time windows. In this case, the envelopes at a time point y_t are computed starting from the past state that is at a distance equal to the length of the window y_{t-w}. A consequence of the use of sliding time windows is that the parameters of the model have to maintain their value in the window but may have different values in different windows.

On the other hand, using sliding time windows each measurement is used twice: at the current step to detect faults and at a future step as starting point of the window. This dependence of the envelopes on the measurements may produce missed and false alarms due to the uncertainty associated to the measurements.

Therefore, the detection results may be different for different window lengths. For instance, a system becomes faulty if its performance is slowly degrading across time until it goes under some previously fixed requeriment, like the one in section 6. But it is also faulty if there is a sudden fault, for instance a breakdown. Therefore, there is a relation between the ability to detect a particular type of fault and the window length used.

8. MULTIPLE SLIDING TIME WINDOWS

This is shown in the example of figure 2. This example is taken from a real system: a part of a gas turbine used in the TIGER Esprit project (Milne *et al.*, 1994). The data that have been used are also real. The scenario is formed by approx. 60 s of data collected when there were not faults and about

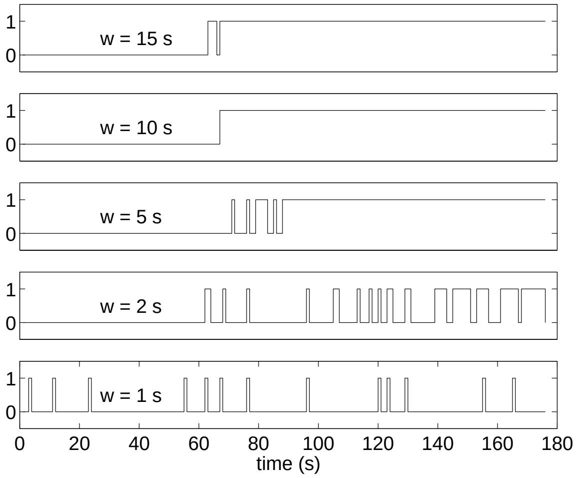

Fig. 2. Fault detection using multiple window lengths.

100 s being the system faulty. The figure shows the indications of fault using different window lengths w. It can be seen that $w = 1$ s gives false and missed alarms and longer windows give better results. When the windows are too short sometimes the faults are not detected because the envelopes "follow" the measurements. The system is considered time variant and it is allowed that the parameters of the model vary at a high speed.

Figure 3 shows another example with the same system known to be faulty between approximately 110 and 160 s. The results are similar to the previous example: shortest windows do not obtain good results. Furthermore, in this case, the figure shows that longest windows do not detect the fault.

These and other examples show that there is not an ideal window length. The length that neither produces false alarms nor misses them depends on the particular system and also on the type of fault. MIS helps the user to deal with this question allowing the use of several window lengths simultaneously.

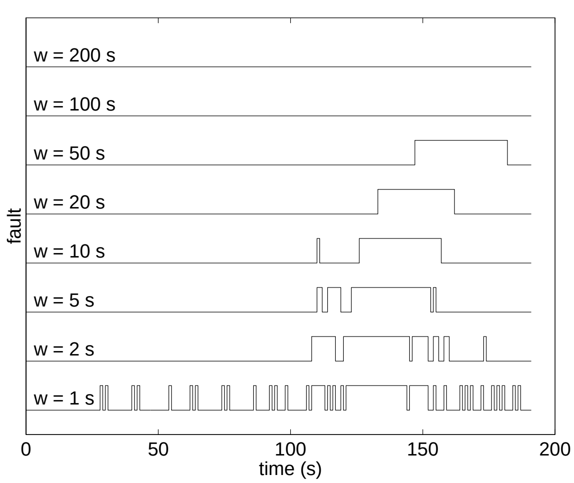

Fig. 3. Detection of a short duration fault.

9. CONCLUSIONS

The semiqualitative simulator MIS has been developed. It converts the simulation problem into a range computation one by means of the discrete model of the system. Two approximations to the range of a function are computed at each step of the simulation by means of an iterating branch and bound algorithm based on Modal Interval Analysis. The results of the simulation are error-bounded envelopes, i.e. an underbounded envelope and an overbounded one.

The error-bounded envelopes have been applied to Fault Detection as reference behaviours for analytical redundancy. The system is guaranteed to be faulty if the measurement is outside of the overbounded envelope. This is the reason to use the location of the measurement to stop the iterating algorithm. When it is in that zone or inside the underbounded envelope the algorithm stops. Sometimes this happens when the error between the envelopes is big, but these envelopes are enough for FD so it is not necessary to perform more computations to obtain better approximations to the exact envelope.

When the system is considered time invariant, each step of the simulation needs a computational effort greater than the previous one. This makes this method unusable for long simulations. To deal with this problem, sliding time windows have been implemented. The measured output of the system is used as the initial state of each window. The adequate length of the window depends on the particular system and on the type of fault to be detected. To help the user to deal with this question, MIS allows the simulation using several window lengths simultaneously.

MIS has been used to detect faults in several systems, both academic and real. The results show that the indications of fault using multiple window lengths can be used not only to detect the faults but to identify the type of fault. A possible future work consists in the design of a higher level system to perform this task.

10. 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 TAP98-0955-C03-02 and TAP99-1079-C03-03.

11. REFERENCES

Armengol, J., L. Travé-Massuyès, J. Vehí and J. Ll. de la Rosa (2000). A survey on interval model simulators and their properties related to fault detection. *Annual Reviews in Control* **24**(1), To appear.

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

Milne, R., Ch. Nicol, M. Ghallab, L. Travé-Massuyès, K. Bousson, C. Dousson, J. Quevedo, J. Aguilar and A. Guasch (1994). Tiger: real-time situation assessment of dynamic systems. *Intelligent Systems Engineering* **3**(3), 103–124.

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

Struss, P. (1990). *Readings in qualitative reasoning about physical systems.* Chap. Problems of interval-based qualitative reasoning, pp. 288–305. Morgan Kaufmann Publishers.

Travé-Massuyès, L., Ph. Dague and F. Guerrin (1997). *Le raisonnement qualitatif pour les sciences de l'ingénieur.* Editions Hermes.

Weld, D. S. and J. de Kleer (1990). *Readings in qualitative reasoning about physical systems.* Morgan Kaufmann Publishers.