Modal Interval Analysis for error-bounded semiqualitative simulation

Joaquim Armengol[1], Louise Travé-Massuyès[2], Josep Vehí[1], Miguel Ángel Sáinz[3]
[1]Institut d'Informàtica i Aplicacions. Universitat de Girona & LEA-SICA
Campus Montilivi. E-17071 Girona. E-mail:{armengol,vehi}@eia.udg.es
[2]LAAS-CNRS & LEA-SICA. 7 Avenue du Colonel-Roche. F-31077 Toulouse. E-mail:louise@laas.fr
[3]Departament d'Informàtica i Matemàtica Aplicada. Universitat de Girona.
Campus Montilivi. E-17071 Girona. E-mail:sainz@ima.udg.es

Abstract

Imprecision and uncertainty in systems can often be expressed with interval models. The result of the simulation of these models can be represented in the form of envelope trajectories. These envelopes can be characterised by several properties such as completeness and soundness which lead to the concept of overbounded and underbounded envelopes. Simulation of such interval models can be performed by several means including quantitative, qualitative and semiqualitative techniques. Whereas existing simulators do not provide any information about the "error" with respect to the exact envelope, a method to obtain error-known envelopes is proposed. It is based on the simultaneous computation of an underbounded and an overbounded envelope. Both envelopes are computed by means of Modal Interval Analysis. A way of controlling the error of the envelopes and adjust it to the desired value is presented.

Keywords: semiqualitative simulation, interval analysis.

1 Introduction

Most of the existing simulators need a mathematical model in which the values of the parameters are real numbers. This implies that the user must have a totally deterministic knowledge of the system. However, complex systems are often subjected to uncertainties that make such a model difficult if not impossible to obtain. A precise model cannot represent the behaviour of such systems which require an explicit representation of imprecisions and uncertainties. A special case is when the uncertainties are *structured*: only the parameters undergo imprecisions but the model structure is known. This case can be handled with interval models in which the equation parameter values are allowed to vary within numeric intervals. For instance, such an interval model could be given by the following transfer function with interval parameters which represents a linear differential relation between an input $u(t)$ and an output $y(t)$:

$$F(s) = \frac{Y(s)}{U(s)} = \frac{[2,3]\,s + [1,3]}{[1,2]\,s^2 + [3,5]\,s + [2,4]} \qquad (1)$$

in which s is the Laplace variable and $U(s)$ and $Y(s)$ are the input and output Laplace transforms, respectively. Actually, a precise model can be viewed as an interval model in which the interval widths are zero. As interval widths decrease, precision increases [12].

The results of the simulation of such interval models is in the form of envelope trajectories (or *envelopes* for short)[6] [24]. The most common use is for Fault Detection (FD) as it provides a way to compute automatically and in a model-based sound manner *adaptive alarm thresholds* for every variable [22]. The envelopes can be characterised by several properties, the main ones being completeness and soundness which lead to the concepts of overbounded and underbounded envelopes which have radical consequences on the robustness and sensitivity of the FD system.

The simulation of interval models can be performed by several means including quantitative, qualitative and semiqualitative techniques. Existing simulators do not provide any information about the "error" with respect to the exact envelope. A method to obtain *error-bounded envelopes*

is proposed. It is based on the simultaneous computation of an underbounded and an overbounded envelope. Both envelopes are computed by means of Modal Interval Analysis. A way of controlling the error of the envelopes and adjust it to the desired value is also provided.

The next section defines the envelopes and their properties in relation to the fault detection problem. Section 3 discusses the related work and provides a summary of the existing simulators that can be used to generate envelopes, putting special emphasis on simulators based on interval arithmetic. In section 5 a method to generate error-bounded envelopes is presented. This method is based on Modal Interval Analysis, which is presented in section 4, and applied to the envelope generation problem in section 6. Finally, some conclusions and directions for the future work are discussed.

2 Envelopes and their properties in relation to Model-based Fault Detection

When simulating an interval model, which actually represents a whole set of models, the state space is a compact set which can be represented by an envelope trajectory for every variable. All the possible behaviours starting from a specific initial state are compacted within a unique curve. This envelope hence includes a whole family of temporal curves, like the one displayed in figure 1.

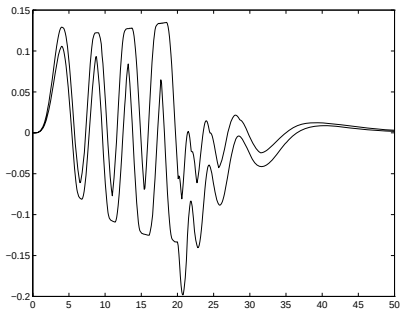

Figure 1: The bounds of the envelopes are two curves

The size of the envelope is critical. If it is too tight there are systems belonging to the model set whose output is outside the envelope. Such an envelope is not *complete*, taking the definition that a complete envelope is one that includes all possible behaviours [21]. On the other hand, if the envelope is too wide it includes zones that cannot be reached by any of the systems belonging to the set. Such an envelope is not *sound*. Our definition of a sound envelope is that every point inside the envelope belongs to the output of at least one of the systems belonging to the set. A complete but not sound envelope is an *overbounded envelope*. A sound but not complete envelope is an *underbounded envelope*.

The ultimate goal is to generate a complete and sound envelope, that is the *exact envelope*. However, more realistic goals are to produce either a minimally overbounded envelope or a minimally underbounded envelope.

3 Related work: envelope simulators

There are many simulators that can be used to generate envelopes. They can be classified into different groups depending on the information used for the simulation: quantitative, qualitative or semiqualitative [1]. A detailed survey of these simulators can be found in [2]. Among the semiqualitative simulators there are methods based on quantitative simulation ([3] [4] [5] [11] [12]), methods based on qualitative simulation ([3] [12] [13]) and methods based on interval arithmetic, which are the ones that are directly related to the approach exposed in this paper.

Interval arithmetic [18] allows to consider the whole continuous range of possible instances represented by an interval model. This is due to the natural extension and one of its properties: monotonic inclusion.

Definition 1 *The natural extension of a rational function is the one obtained by substituting real arguments by intervals and rational operations by their arithmetic interval extensions.*

Theorem 2 (Monotonic inclusion) *Given $f(x_i)$, a real function, and $F(X_i)$, its natural extension, then $x_i \in X_i$ implies $f(x_i) \subseteq F(X_i)$.*

In consequence, the natural extension gives a guarantee on the result: no function in the class can take values outside the range computed using interval arithmetic. Unfortunately, it does not provide the exact result (complete and sound) in the general case. This comes essentially from the two following problems:

- *The multi-incidence problem*: interval arithmetic considers that each incidence of a variable in a function is independent of the other. Similarly, it is unable to take into account other dependencies. The compiled range is hence overbounded.

- *The wrapping problem*: the state, at some time point, of a system with interval parameters may be represented by an hypercube. However, it may be that the system's state does not evolve into another hypercube at the next time point. In figure 2 an example with two state variables is shown: the hypercube is a rectangle. It transforms into a rhombus at the following time step (it could actually evolve into any two dimensional shape). The projection of this rhombus on the variable axis leads to a new rectangle which obviously includes spurious states, as shown in the figure. Therefore, the obtained envelopes are overbounded.

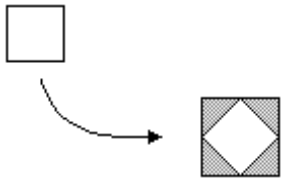

Figure 2: The wrapping problem

Some semiqualitative simulators based on interval arithmetic are presented in [6] [12] [14] [16] [18] and [24].

All these simulators produce overbounded envelopes. Some methods use fuzzy sets: all of them take the α-cut of the fuzzy sets and end up using interval arithmetic. Some of them are presented in [7] and [20].

4 Modal Interval Analysis

Modal Interval Analysis [8] [9] [10] extends real numbers to intervals, identifying the intervals by the predicates that the real numbers fulfil, unlike classical Interval Analysis which identifies the intervals with the set of real numbers they contain.

In the following, some of the properties of modal intervals that are interesting for envelope generation are stated.

Given the set of closed intervals of $\mathbf{R}$, $I(\mathbf{R}) = \{[a,b]' \mid a,b \in \mathbf{R}, \quad a \leq b\}$, and the set of logical existential and universal quantifiers $\{E,U\}$, a modal interval is defined by a pair

$$X := (X', Q) \tag{2}$$

where $X' \in I(\mathbf{R})$ and $Q \in \{E,U\}$. QX is the *modality* and X', called *extension,* is an interval in the classical sense, i.e. a set of real numbers. The canonical notation for modal intervals is:

$$[a_1,a_2] := \left\{ \begin{array}{l} ([a_1,a_2]', E) \text{ if } a_1 \leq a_2 \\ ([a_2,a_1]', U) \text{ if } a_1 \geq a_2 \end{array} \right\} \tag{3}$$

A modal interval $([a_1,a_2]', E)$ (the classical one) is called "existential interval" or "proper interval" and denotes one value within the interval bounds. On the other hand, a modal interval $([a_2,a_1]', U)$ is called "universal interval" or "improper interval" and denotes all the values within the interval bounds. Thus, the proper interval $[1,3]$ points at some real number between 1 and 3 and the improper interval $[3,1]$ points at any real number between 1 and 3.

The rational operations between modal intervals are extensions of classical interval arithmetic with the addition of the dual operator defined by:

$$Dual([a_1,a_2]) = [a_2,a_1] \tag{4}$$

The dual formulation of the modal intervals allows to define two modal interval extensions of a continuous function f: $f^*(X)$ and $f^{**}(X)$. The modal interval extension represented by $f^*(X)$ may be interpreted as

$$\begin{aligned} f^*(X) \quad \subseteq \quad & F(X) \Longleftrightarrow U\left(x_p, X_p^{'}\right) \\ & Q(z, F(X))\, E\left(x_i, X_i^{'}\right) (z = f(x_p, x_i)) \end{aligned} \tag{5}$$

where X_p and X_i are the proper and improper components of X, respectively. This interpretation can be read: "For all elements belonging to the proper intervals there exists at least one element in the improper intervals that fulfil the function".

Example 3 $[10,20] + [20,15] = [30,35]$ *means*

$$\begin{aligned} & U\left(a, [10,20]'\right) E\left(f, [30,35]'\right) \\ & \quad E\left(b, [15,20]'\right) (a + b = f) \end{aligned} \tag{6}$$

Example 4 $[10,20] + [15,20] = [25,40]$ *means*

$$\begin{aligned} & U\left(a, [10,20]'\right) U\left(b, [15,20]'\right) \\ & \quad E\left(f, [25,40]'\right) (a + b = f) \end{aligned} \tag{7}$$

On the other hand, the semantic interpretation of the $f^{**}(X)$ extension is dual:

$$F(X) \subseteq f^{**}(X) \Longleftrightarrow U\left(x_i, X_i^{'}\right) \quad (8)$$
$$Q\left(z, Dual\left(F(X)\right)\right) E\left(x_p, X_p^{'}\right)$$
$$\left(z = f\left(x_p, x_i\right)\right)$$

The computation of $f^{*}(X)$ and $f^{**}(X)$ is not always possible. The usual procedure is to find overbounded computations of $f^{*}(X)$ and underbounded computations of $f^{**}(X)$ which maintain the semantic interpretations. To this respect, an aspect to be taken into consideration is the rounding of computations. Computers work with digital numbers, not with real numbers. In order to maintain the semantic interpretations, direct roundings (up or down) have to be used.

If f is a rational function, there are some theorems in Modal Interval Analysis that allow to obtain the exact range of f in some cases or overbounded computations of $f^{*}(X)$ and underbounded computations of $f^{**}(X)$ in other cases.

Definition 5 *A modal interval extension $fR(X)$ of f in X is optimal if*

$$f^{*}(X) = fR(X) = f^{**}(X) \quad (9)$$

The following theorem provides the conditions and the way to obtain optimal extensions.

Theorem 6 *Given $fR(X)$, a rational interval function defined in a parameter space $Prop(X)$, tree-optimal and totally monotonic for each multi-incident component of X. Let XD be an enlarged vector of X obtained considering each multi-incident component as independent and transforming it into its dual if it is antitonic (the monotonicity of this incidence and the monotonicity of the component have opposite senses). In this case,*

$$f^{*}(X) = fR(XD) = f^{**}(X) \quad (10)$$

Example 7 *Given $f = x^2 - xy$ and the parameter space $x = [2,4]$ and $y = [1,2]$, the range of the function obtained by its natural extension is $f^{*}(X) \subseteq [-4,14]$. The exact range can be computed applying theorem 6 and is $f^{*}(X) = f^{**}(X) = [2,4]^2 - Dual([2,4])[1,2] = [0,12]$*

If the function is not monotonic for each multi-incident component, theorem 6 can be partially applied in order to reduce the complexity of the problem. For instance, given an n variable function, the problem of finding the range of this function in a domain in which the function is monotonic with respect to r variables, can be reduced to evaluate the range of an interval function of $n - r$ variables. Therefore, the problem complexity has a lower order.

Example 8 *Given $f = xy - x^2 - 2y$ and the parameter space $x = [1,2]$ and $y = [3,4]$, the range of the function obtained by its natural extension is $f^{*}(X) \subseteq [-9,1]$. The function is totally monotonic with respect to y*

$$\frac{\partial f}{\partial y} = x - 2 = [-1,0] \leq 0 \quad (11)$$

but it is not totally monotonic with respect to x

$$\frac{\partial f}{\partial x} = y - 2x = [-1,2] \ni 0 \quad (12)$$

Theorem 6 can be applied to y and hence a better approximation of the range of the function is obtained:

$$f^{*}(X) \subseteq xDual(y) - x^2 - 2y = [-8,-1] \quad (13)$$

A way to obtain even a better approximation is by splitting the parameter space. The advantage is that now only the variable x must be split. Moreover, the range in each sub-space can be computed more exactly because the modality of each incidence of the variable y is already known.

As a conclusion, the number of sub-spaces to be made in order to compute an approximation of the range of the function is smaller when modal intervals are used. This is illustrated in [23], in which modal intervals combined with a branch-and-bound algorithm have been applied to the analysis and design of robust controllers.

5 Using modal intervals for generating and controlling error-bounded envelopes

The properties of the simulators presented in section 3 can be exhibited only in a binary manner. For example, all the interval based simulators are complete but not sound, i.e. they provide overbounded envelopes. A step forward would be to produce some kind of measure of the *degree of overbounding,* in other words to be able to evaluate the "error" of the obtained envelope with respect to the

exact one. However, the exact envelope is of course not known. The exact envelope problem is actually highly complex as it requires global optimisation tools for non linear and non convex functions.

The alternative way proposed in this paper is to bound the error, i.e. to determine the maximum distance. This is achieved by computing both an underbounded envelope and an overbounded envelope. The distance between these envelopes indeed gives the maximum error. These envelopes are defined as *error-bounded envelopes*.

Moreover it is shown that the maximum error can be controlled by widening the underbounded envelope or by tightening the overbounded one.

The generation of the underbounded and overbounded envelopes is approached in an original way by using Modal Interval Analysis.

5.1 The multi-incidence problem in the simulation task

Interval based methods are based on the reformulation of the simulation problem into an optimisation problem.

The behaviour of a n-th order dynamic system can be represented by the following difference equation:

$$y_{t+1} = \sum_{i=0}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j u_{t-j} \tag{14}$$

in which it can be observed that the output of the system at any time point (y_{t+1}) depends on the values of the previous outputs (y_{t-i}) and inputs (u_{t-j}). This dependency is given by the parameters of the system (a_i and b_j). Hence, finding the limits of the envelope at a given time point is equivalent to finding the maximum and the minimum of a function into a parameter space. This is a global optimisation problem. Nevertheless, the computational cost of a global optimisation algorithm is too high to use them for this task.

There are many different methods for global optimisation, but many of them have no guarantee of finding the global optimum. In the case of the envelope generation problem, a local optimum results in an incomplete (underbounded) envelope. Moreover, the computational effort needed by these methods is very high because they search the optima trying to minimize the error as much as possible. For most of the applications it is not needed to have a very small error and therefore it is not necessary to make such a computational effort.

Conversely, global optimisation methods based on the interval arithmetic obtain overbounded results as stated by the monotonic inclusion property (see. section 3). This comes from the multi-incidence problem. In the following, two types of multi-incidences are pointed out.

The first type of multi-incidence comes from the fact that, in a difference equation like the one shown above, some parameters may appear several times. For instance, the transfer function representation of a generic first order system is:

$$F(s) = \frac{Y(s)}{U(s)} = \frac{k}{\tau s + 1} \tag{15}$$

in which k is the static gain and τ is the time constant. If this transfer function is discretised by the Euler method, the difference equation representation of this system is:

$$y_t = \left(1 - \frac{T}{\tau}\right) y_{t-1} + \frac{kT}{\tau} u_{t-1} \tag{16}$$

in which T is the sampling period. As it can be seen, τ appears more than once in this equation. If the equation is rewritten, renaming the parameters as follows:

$$a = \left(1 - \frac{T}{\tau}\right) \text{ and } b = \frac{kT}{\tau} \tag{17}$$

the situation is even worse because the multi-incidences do not appear explicitly. This is the reason for avoiding intermediate operations: each time an intermediate operation is performed some information is lost. Going deeper, k and τ themselves include implicit multi-incidences with respect to the physical parameters of the device!

The second type of multi-incidences is particular to the simulation mechanism as the equations are taken at different time points. For instance, the difference equation at the time point $t+1$ of the generic first order system used above is:

$$y_{t+1} = \left(1 - \frac{T}{\tau}\right) y_t + \frac{kT}{\tau} u_t \tag{18}$$

As it can be seen, y_t, τ and k appear both in the difference equation for time point t and $t+1$. It should be noticed that these multi-incidences can be treated as independent variables for systems whose parameters are known to vary in time. On the other hand, they must be treated as so if the physical system is assumed to be invariant.

5.2 The proposed method

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 optimisation is performed. This expression is obtained in a recursive way by substituting y_t within equation 18 down to y_0. For instance, in the case of a first order system the following expressions will be used:

$$
\begin{aligned}
y_1 &= \left(1-\frac{T}{\tau}\right) y_0 + \frac{kT}{\tau} u_0 \qquad (19)\\
y_2 &= \left(1-\frac{T}{\tau}\right)^2 y_0 + \left(1-\frac{T}{\tau}\right) \frac{kT}{\tau} u_0 + \frac{kT}{\tau} u_0\\
y_3 &= \left(1-\frac{T}{\tau}\right)^3 y_0 + \left(1-\frac{T}{\tau}\right)^2 \frac{kT}{\tau} u_0 +\\
&\quad + \left(1-\frac{T}{\tau}\right) \frac{kT}{\tau} u_1 + \frac{kT}{\tau} u_2\\
&\quad \vdots
\end{aligned}
$$

Modal Interval Analysis then provides an efficient tool to perform the optimisation task taking the multi-incidences into account. Moreover, the two semantic interpretations provided by modal intervals are directly applicable to compute the over-bounded and the underbounded envelopes:

- Overbounded envelope. Its semantics is: "-For every (universal quantifier) model parameter, input and initial state, the output belongs to the envelope (existential quantifier)", which corresponds to $f^*(X)$ (see equation 5).
- Underbounded envelope. The semantics is dual: "For every output belonging to the envelope there exist parameter, input and initial state values that produce this output", which corresponds to $f^{**}(X)$ (see equation 8).

Therefore, Modal Interval Analysis can be used to compute both envelopes.

6 Practical implementation and examples

In section 4, some tools of Modal Interval Analysis that are useful for envelope generation have been described. One limitation of these tools is that it is necessary to differentiate the function in order to apply theorem 6, hence restricting the method to differentiable functions. However, the discrete representation of the system used in our simulation problem is differentiable.

A simulator based on these tools has been implemented. It uses Matlab version 5.1 for Unix [17]. Symbolic computations are performed with Maple V r4 [15] through the Symbolic Math Toolbox. Moreover, it uses C++ programs as MEX-files to perform modal interval computations with direct roundings and to accelerate the branch-and-bound algorithm.

The implemented simulation algorithm is the following:

Modal Interval Branch-and-bound Algorithm

```
Given a function in a space
/* The exact range of the function can be computed if
there are not multi-incidences */
Modality of uni-incident variables is not changed
IF all variables are uni-incident THEN
   Exact result
   END
ENDIF
Calculate internal approximation
DO
   External = internal
   DO /* The monotonicity of the function is studied
in order to apply theorem 6*/
      Get subspace
      FOREACH variable with unknown modality
         IF 0∉first derivative THEN
            fix modality
         ENDIF
      ENDFOREACH
      IF all modalities are known THEN
         /* Application of theorem 6 where possible */
         Calculate partial exact
         Internal = internal ∨ partial exact
         External = external ∨ partial exact
      ELSE
         /* Division of subspaces if necessary */
         Calculate partial internal.
         Internal = internal ∨ partial internal.
         Calculate partial external
         External = external ∨ partial external
         Divide subspace
      ENDIF
   WHILE remaining subspaces
   IF error<ε THEN
      final=1
   ENDIF
WHILE final=0
END
```

As an example, figure 3 shows the envelopes obtained for a generic first order system with the following parameters:

- static gain: $k = [0.95, 1.05]$
- time constant: $\tau = [5, 20]$
- initial state: $y_0 = 0$
- sampling time: $T = 1\ s$
- input: steps of different lengths and magnitudes
- maximum error of the envelopes $\varepsilon \leq 0.2$

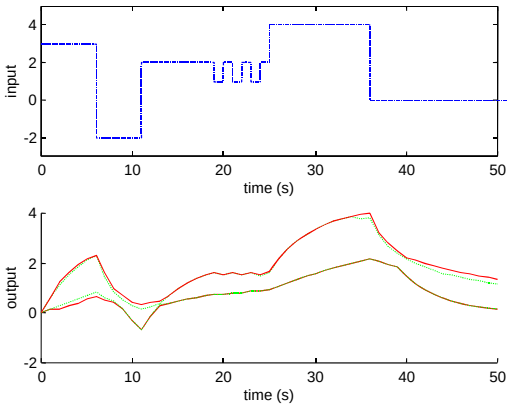

Figure 3: Example of simulation

In this figure, the solid line is the overbounded envelope and the dotted line is the underbounded one. The error between the two envelopes is not very small, but sufficient in most cases, for instance for fault detection. This allows to obtain useful results with a computational effort much lower than the one needed to obtain similar results using global optimisation algorithms.

Another example is the one shown in figure 4, in which the same system as above is excited with a sinusoidal input. On the right, a high frequency white noise has been added to the input, whereas on the left the input is the same but without the noise. The interesting comment about this example is that, as it can be seen on the figures, a significant amount of noise has been "absorbed" by the envelopes. Indeed, the envelopes produced with or without noise have a much higher similarity than the noisy and non noisy inputs. In particular, the envelopes corresponding to the noisy input are not wider than the other ones. The width depends essentially on the imprecision of the model. This means that *semiqualitative simulation acts as a high frequency filter*. This property has not been carefully studied yet but should deserve attention in view of fault detection applications.

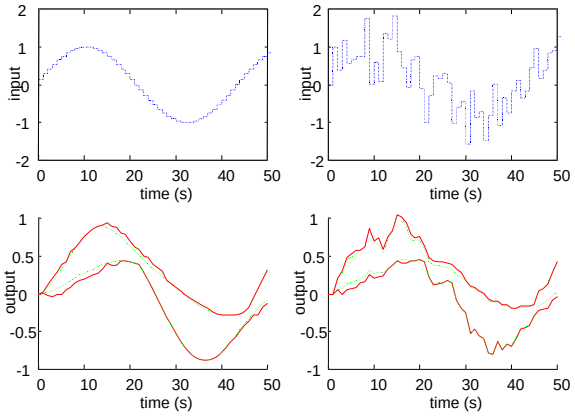

Figure 4: Simulation without (left) and with (right) noise

7 Conclusions and future work

In this paper it has been shown that the existing simulators for systems with structured uncertainties provide envelopes which may or may not have properties like completeness, soundness, stability, etc. or not. Sometimes the properties are not even known.

When the properties are known, the error of the envelopes with respect to the exact one is unknown. A method to obtain error-bounded envelopes is proposed. It is based on the simultaneous computation of an underbounded envelope and an overbounded one. Two ways to control this error are proposed as well: tightening the overbounded envelope or widening the underbounded one. Both can be achieved by means of Modal Interval Analysis. The error of our envelopes can hence be adjusted to the desired value. The computation effort of course increases when the error decreases.

This method has been implemented in Matlab and uses Maple and C routines. This facilitates its future integration into a supervision framework that is being developed based on Matlab and Simulink. It is also planned to be used to improve

the prediction and fault detection algorithms of the Ca~En simulator [22].

An inconvenient of this method is that the computation effort increases at each step of the simulation. The exact envelope at a time point t can only be obtained by computing the range of the function that relates the current time point to the initial one, i.e. all the previous states must be considered in order to obtain the exact envelope. This means that the procedure is not incremental. A solution to overcome this problem is the use of a shorter temporal window. Due to the dynamics of the systems, it has been shown that the influence of the previous states over the current one decreases with time [19]. Hence, it should be possible to obtain close results with a shorter temporal window length. Saludes [19] recommends a window length which essentially depends on the time constant of the system.

A related problem is about the semantics of these results. For instance, given an overbounded envelope and an underbounded one at a time point t, both can be used as the initial state for the function at time point $t + m$. This means that four approximations of the range of the function can be obtained: over and underbounded approximations of the range of the function using over and underbounded approximations of the initial state. Each one of these four approximations has different semantics and the most suitable one, if it exists, has to be decided. This problem is still under investigation. It seems that the solution is simple for overbounded envelopes. This envelope can be obtained computing an overbounded range of the function and using an overbounded initial state. As the length of the temporal window increases, the envelope gets closer to the exact one. The case of underbounded envelopes is more complex and more work remains to be done.

As it has been shown, a very interesting feature of modal intervals is the semantics. A future work is to study whether envelopes with different semantics can be used not only to detect the faults but also to localise the faulty parameter. For instance, if a system has two physical parameters a and b, envelopes with the semantics "for every a there exists b so that..." or "for every b there exists a so that..." can be obtained. If a system is faulty and its output belongs to only one of these two envelopes, it should be possible to determine whether it is a or b that is faulty.

Finally, in [4] it is claimed that it is not necessary to study the evolution of all points belonging to an uncertainty region to know the evolution of the region. The study of the evolution of points belonging to its surface is enough. The possible application of this result to modal interval simulation gives another direction for research.

Acknowledgements

This work has been funded by the autonomous government of Catalonia, Generalitat de Catalunya, through CIRIT 1996 XT 00023 and Grup de Recerca Consolidat ref. 1995GR00474 and by the Spanish government through co-ordinated research project CICYT TAP96-1114-C03-03, "-Plataformas integradas de CAD de Supervisión y Metodologías".

References

[1] Armengol, J., L. Travé-Massuyès, J. Vehí, J. Ll. de la Rosa. On Modal Interval Analysis for envelope determination within the Ca~En qualitative simulator. In Proceedings of the 7th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU'98), pp. 110-117. Paris, France, 1998.

[2] Armengol, J. Simulation of dynamical systems with structured uncertainty: an overview. Report de recerca R1998/0002. Universitat de Girona. 1998.

[3] Berleant, D., B. J. Kuipers. Qualitative and Numerical Simulation with Q3. In Faltings, B., P. Struss, eds. Recent advances in qualitative physics. The MIT Press. Cambridge, Massachusetts. 1992.

[4] Bonarini, A., G. Bontempi. A qualitative simulation approach for fuzzy dynamical models. ACM transactions on modeling and computer simulation, Vol. 4, No. 4, pp. 258-313. 1994.

[5] Bontempi, G. Qua.Si. III : A software tool for simulation of fuzzy dynamical systems. In Proceedings of the European Simulation Multiconference, pp. 615-619. Ghent, Belgium. 1996.

[6] Bousson, K., L. Travé-Massuyès. Putting more numbers in the qualitative simulator CA-EN. In Proceedings of International Conference

on Intelligent Systems Engineering (ISE 94). Hamburg-Harburg, Germany. 1994, pp. 62-69.

[7] Coghill, G. Mycroft: a framework for constraint based fuzzy qualitative reasoning. Ph. D. thesis. Department of computing and electrical engineering. Heriot-Watt University. Edinburgh, Scotland. 1996.

[8] Gardeñes, E, H. Mielgo, A. Trepat Modal Intervals: reasons and ground semantics. In Lecture Notes in Computer Science #212 - Interval Mathematics 1985. Springer-Verlag, Berlin, 1986.

[9] Gardeñes, E., H. Mielgo. Modal intervals: functions. In Proceedings of the Polish Symposium on Interval and Fuzzy Mathematics. Zamenkhov's University of Poznan. Poland, 1986.

[10] Gardeñes, E., H. Mielgo, M. A. Sáinz. Presentation of the research group SIGLA/X. Report de recerca IMA 95-10. Universitat de Girona.

[11] Gasca, R. M., J. A. Ortega, M. Toro. Simulación semicualitativa. In Proceedings of Jornades Hispano-Franceses de Sistemes Intel·ligents i Control Avançat. Barcelona, 1996.

[12] Kay, H. Refining imprecise models and their behaviors. Ph. D. dissertation. University of Texas at Austin. 1996.

[13] Kuipers, B. J., D. Berleant. Using Incomplete Quantitative Knowledge in Qualitative Reasoning. In Proceedings of AAAI-88, pp. 324-329. 1988.

[14] Lohner, R. Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen. Ph. D. thesis. Universität Karlsruhe, Germany. 1988.

[15] Maple V. Waterloo Maple, Inc. 450 Phillip St. Waterloo, ON N2L 5J2, Canada. mailto:info@maplesoft.com, http://www.maplesoft.com.

[16] Markov, S., R. Angelov. An interval method for systems of ODE. In Lecture Notes in Computer Science #212 - Interval Mathematics 1985, pp. 103-108. Springer-Verlag, Berlin, 1986.

[17] Matlab-Simulink. The Math Works, Inc. 24 Prime Park Way. Natick, MA 01760-1500, USA. mailto:info@mathworks.com. http://www.mathworks.com.

[18] Moore, R. E. Interval Analysis. Prentice-Hall, 1966.

[19] Saludes, J., V. Puig, J. Quevedo. Determination of window length for a new algorithm in adaptive threshold generation. In Proceedings of Symposium on Qualitative System Modeling, Qualitative Fault Diagnosis and Fuzzy Logic and Control. TEMPUS S_JEP 07759-94 - MODIFY grant. Budapest, Hungary. 1997.

[20] Shen, Q., R. R. Leitch. Fuzzy qualitative simulation. In IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 4, pp. 1038-1061. 1993.

[21] Struss, P. Problems of interval-based qualitative reasoning. In Weld, D. S., Johan de Kleer, eds. Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers, Inc. San Mateo, California. 1990, pp. 288-305.

[22] Travé-Massuyès, L., R. Milne. TIGER: Gas turbine condition monitoring using qualitative model based diagnosis. IEEE Expert Intelligent Systems and Applications. Vol. May-June.

[23] Vehí, J. Anàlisi i disseny de controladors robustos mitjançant intervals modals. Ph. D. Thesis. Universitat de Girona. 1998.

[24] Vescovi, M. R., A. Farquhar, Y. Iwasaki. Numerical Interval Simulation. International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canada. 1995, pp 1806-1813.