Envelope generation for interval systems

Joaquim Armengol[1,2], Louise Travé-Massuyès[2], Josep Lluís de la Rosa[1,2], Josep Vehí[1,2]
[1]Institut d'Informàtica i Aplicacions (IIiA)
Universitat de Girona
Avinguda Lluís Santaló s/n
17071 Girona
[2]Laboratoire Européen Associé – Systèmes Intelligents et Contrôle Avancé (LEA-SICA)
e-mail: armengol@eia.udg.es

Keywords: simulation, interval arithmetic, qualitative reasoning.

ABSTRACT. *Uncertainty of systems can be expressed in some cases with interval models. Simulation of interval models produces envelopes. These envelopes can have properties like completeness, soundness and hence be overbounded or underbounded. To perform the simulation of interval models there are quantitative, qualitative and semiqualitative simulators. A brief description of some simulators and a study of their properties are made. Some enhancements based on Modal Interval Analysis are proposed.*

1.- INTRODUCTION.

Most of nowadays simulators need a mathematical model of the system to be simulated where parameters are real numbers. This means that user must have a totally determinist knowledge of the system. The model of complex systems sometimes can not be defined precisely due to uncertainties. These uncertainties can be unstructured (the equations of the system are not known) or structured (the equations are known but not the values of its parameters). There are different causes:

- The knowledge of the system is not complete because the real system can not be observed (some parts of a nuclear power plant) or does not exist yet (a factory before it is built or a product before its is manufactured) [Bon94] [Ves95].
- The model of the system is known but it is too complex and a simplified one is used [Bon94].
- Parameters of the system can change through time due to unknown, unpredictable or difficult to modelize phenomenons [Ves95].

2.- UNCERTAIN SYSTEMS AND INTERVAL MODELS.

Precise models can not describe the behaviour of these systems and a model space is needed [Kay95]. One special case of these model space is when it is described with known equations (structured uncertainties) where parameters are intervals. For instance, such a model could be a transfer function with interval parameters:

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

In fact, a precise model is an interval model where interval widths are zero. As interval widths decrease, precision increases [Kay96].

3.- SIMULATION OF INTERVAL MODELS.

When a model space or a model family is represented with an interval model and its behaviour is simulated, the result can not be the one given by traditional simulators for precise systems, that is a single curve through time for each variable. The behaviour of such a system can be represented in different ways:

- Envisionment. The behaviour of the system is represented globally using a graph like the one in figure 1 [Cog96]. This graph is a tree where all possible behaviours of a system starting from an initial state are qualitatively represented. It is a tree and not a single path because uncertainty is propagated [Ber92] [Bon94] [Bon96]. There are different types of envisionments depending on the initial state. If it is known then it is a complete envisionment [Kui86] and if it is unknown then it is a total envisionment [For88].

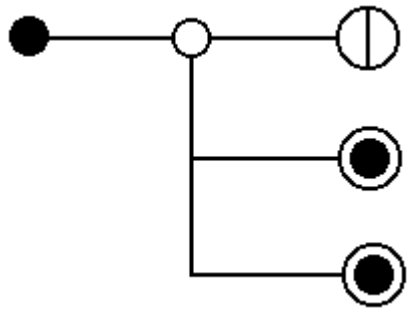

figure 1. Envisionment

- Simulation. An only image for a specific initial state is given. This image is a family of time curves called envelope, like the one in figure 2. Some authors call them attainable envisionments [For88] or partial envisionments [Kui86].

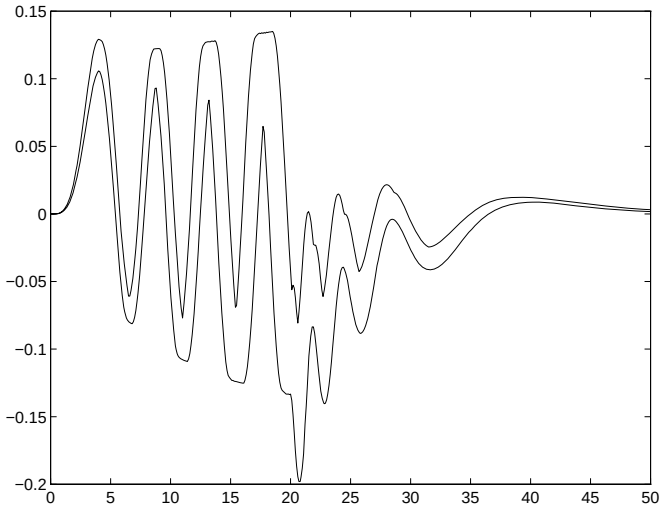

figure 2. The ends of the envelope are two curves

4.- ENVELOPES' PROPERTIES.

This work is centered in envelope generation so envisionment will not be taken into account. Size of an envelope is critical. If it is too tight there are systems belonging to the model family whose output is outside the envelope, hence their behaviour is not represented by the envelope. Such an envelope is not complete. A complete envelope is the one that includes all possible behaviours. On the other hand, if the envelope is too wide it includes zones that can not be reached by any of the systems belonging to the family. Such an envelope is not sound. An envelope is sound if every point inside the envelope belong to the output of at least one of the systems belonging to the family. A complete but not sound envelope is an overbounded envelope. A sound but not complete envelope is an underbounded envelope.

The ideal goal would be a complete and sound envelope, that is the exact envelope. However, a more realistic goal will be a minimum overbounded envelope or a minimum underbounded envelope depending on the use of the envelope. For instance, if the envelope is to be used for fault detection it can be said that the system is faulty if the system output goes off the envelope. Therefore with an overbounded envelope the system can be faulty and it is not detected, and, on the other hand, with an underbounded envelope it can be said that the system is faulty even if it is not. In the latter case unjustified alarms are generated hence it is better to have overbounded envelopes

Another property an envelope can have is stability. If the envelope width grows through time, it is unstable. This is an indesirable property because the envelope becomes useless.

5.- SIMULATORS AS ENVELOPE GENERATORS

In this section, an overview of the simulators that generate envelopes is presented. These simulators have been classified into three groups depending on the information used for the simulation:

- Quantitative or numeric.
- Qualitative.
- Semiqualitative. A combination of the previous two types.

5.1. Quantitative simulation of interval systems

Qualitative or numeric simulation makes numeric predictions of the system states. This implies the prediction of the variables' values at determined time points.

There is a description of the different quantitative methods to simulate the behaviour of interval systems below.

5.1.1. Threshold calculus

One way to consider uncertainty is to have separately a precise model and the uncertainty associated to that model [Kay96]. The behaviour of the precise model, a representative of all the family of models called nominal system, can be simulated using a simulator for precise models. To obtain the envelope a tolerance called threshold is superimposed to the trajectory. Thresholds

can be fixed or variable. If it is variable, its size can depend on the working point, the values of inputs and outputs, etc. Such a threshold adapts to circumstances hence is called adaptive threshold. It has better properties than a fixed threshold, of course, but it is also more difficult to compute: a new value must be computed at every time step so it is not sufficient to compute a single value [Gas96b].

The threshold represents an estimation of the error at each point or the likelihood of the envelopes, that is the confidence degree of the envelopes. It can be computed by probabilistic or statistical methods. These methods are applicable only if the system is linear [Kay96], accessible and measurable [Bon94] and the uncertainty can be modelled as a probability. One disadvantage of these methods is that it is necessary to deal with great amounts of data to have reliable results. Another disadvantage is that neither completeness nor correctness can be assured.

5.1.2. Scalar systems method

Another way to study a family of systems is studying many precise systems (as many as possible) belonging to the family and then extracting conclusions for the whole family. These techniques are slow and sometimes it is not possible to extrapolate the properties of the studied systems to the whole family. There can exist unstudied systems where the extrapolation is not valid [Kay96]. This is the reason to say that these methods have no guaranty. This will be demonstrated with a practical case in the next paragraphs.

In the case of simulation, systems belonging to the family have to be chosen and simulated. Any of the simulators for precise (scalar) systems can be used to simulate the behaviour of these systems, where the parameters are real numbers. These simulators use numerical methods to integrate the differential equations used to describe the system. The envelopes are obtained superimposing all the trajectories obtained.

If the systems to be studied are chosen randomly, it is said that it is a Monte Carlo method.

The systems to be studied can also be chosen in a systematic way by making a grid into the parameter space. In this case, the systems obtained combining the limits of each interval are called extreme systems. For instance, given the system

$$F(s) = \frac{[0.95,1.05]}{[5,20]s+1}$$

the extreme systems are

$$F_1(s) = \frac{0.95}{5s+1},\ F_2(s) = \frac{0.95}{20s+1},\ F_3(s) = \frac{1.05}{5s+1} \text{ and } F_4(s) = \frac{1.05}{20s+1}$$

which are shown in figure 3.

The output of these systems when the input is 1 between t=0 s and t=50 s and 0.5 after this time is shown in figure 4.

In this figure there are represented the outputs of the four systems with dotted lines and the resulting envelope in dashed lines. Moreover, it is represented in solid line the output of the system

$$F_5(s) = \frac{1.05}{14s+1}$$

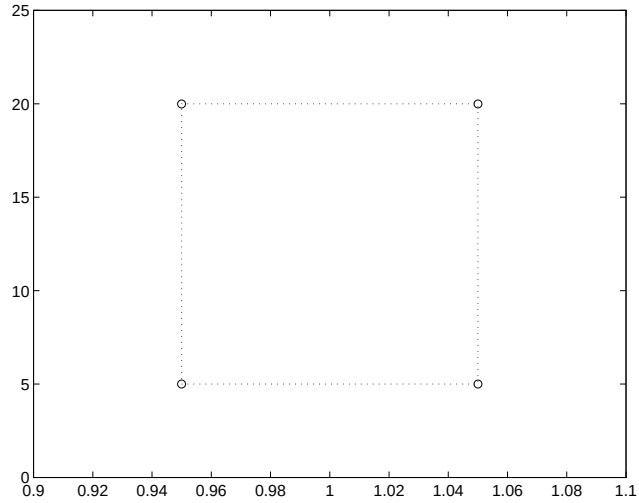

figure 3. Extreme systems

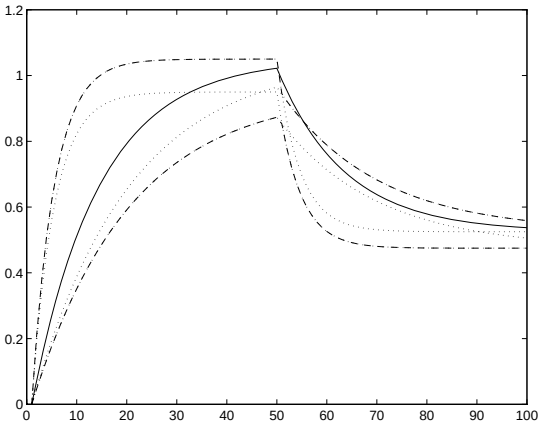

figure 4

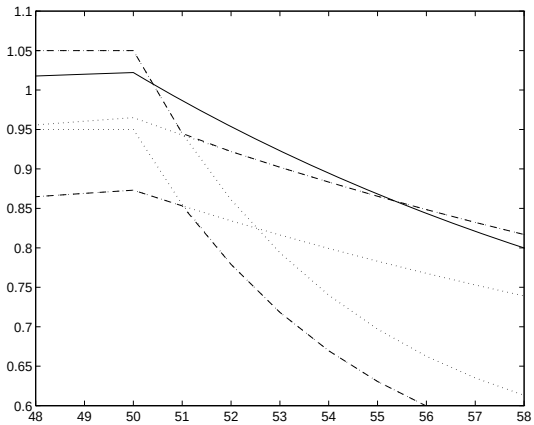

figure 5

This system belongs to the considered family but its output is, for a period of time, outside the envelope obtained simulating extreme systems, as it can be seen in the zoom of figure 5

Therefore, the envelopes obtained simulating extreme systems are underbounded. The envelope can be widened simulating other systems belonging to the family. This additional systems can be chosen in a systematic way, like in figure 6, or randomly, like in figure 7. But the result is ever the same: the widened envelope will be closer to the exact one but it will remain an underbounded envelope. The reason is that it can ever exist an unstudied system whose output is outside the envelope. Hence, this is a method with no guaranty: conclusions can be taken out from the studied systems, but not from the unstudied systems. A property of this method is convergence: as the number of studied systems increases, the envelope closes more and more to the exact envelope. In the limit, when infinite systems belonging to the family are studied, the envelope obtained is the exact one [Ber92].

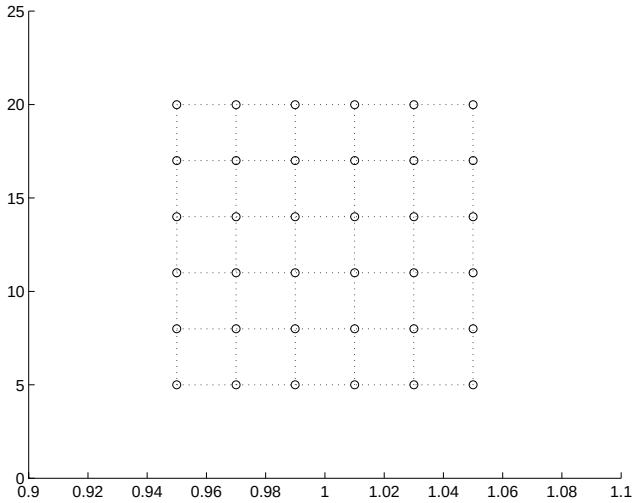

figure 6. Scalar systems chosen systematically.

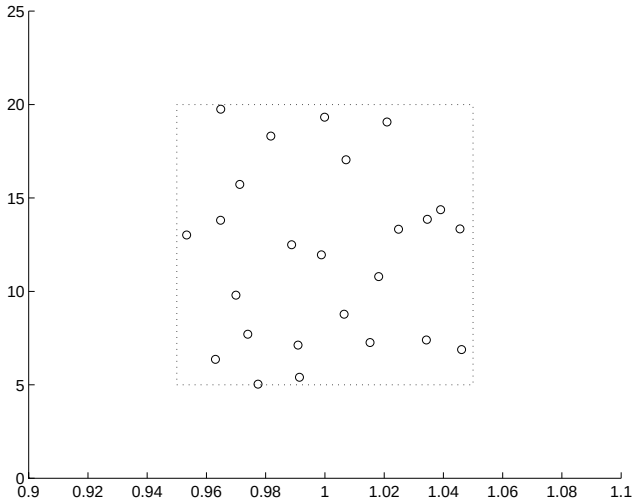

figure 7. Scalar systems chosen randomly.

As a conclusion, the properties of this method are:

- The envelope obtained is underbounded or incomplete.
- It is a convergent method.
- It is valid only for time invariant systems because time variant systems can not be simulated.
- To compute the limits of the envelope at a specific time point it is necessary to simulate the behaviour of all the considered systems from zero to this time and search the maximum and the minimum outputs. Hence it is not recursive and it can not be used in real time.

5.1.3. Optimisation

The recursivity problem can be solved if the discrete representation of the system is used to simulate its behaviour. This representation in differences' equation is like this:

$$y(k+1) = a_0 y(k) + a_1 y(k-1) + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2)$$

In this equation it can be observed that the output of the system at a time point depend on the values of the previous outputs and inputs.

If an interval system is represented by difference equations, parameters a_i and b_i are intervals, outputs $y(i)$ also are intervals because they express the limits of the envelope and the inputs $u(i)$ are intervals if they are imprecise, for instance if there are errors in the sensors (figure 8).

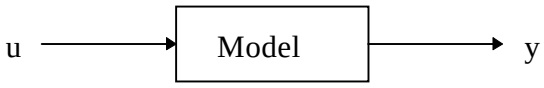

figure 8 If the model or the inputs are intervals, the output will be an interval.

Hence, finding the limits of the envelope at a time point and finding the maximum and the minimum of a function into a parameter space are the same. This is a global optimisation problem. There are many different methods for global optimisation: methods based on gradients, genetic algorithms, etc. Many of them have no guarantee: that is a local optimum can be found when the global optimum is searched. In the case of envelopes, this means an incomplete or underbounded envelope.

5.2. Semiqualitative simulation based on interval arithmetic

It is necessary to study all the possible cases to obtain a guarantee, that is, not underbounded envelopes. This can be done using interval arithmetic. Methods that use interval arithmetic are not only numeric methods but also semiquantitative or semiqualitative methods, because they do not use only numeric knowledge.

5.2.1. Brief introduction to interval arithmetic

An interval is a set of real numbers like

$$X = [\underline{x}, \overline{x}] = \{x \in \Re \mid \underline{x} \le x \le \overline{x}\}$$

where $\underline{x}$ is the lowest end of the interval and $\overline{x}$ is the highest end of the interval.

The function obtained substituting real numbers by intervals and real number operations by interval operations into a real function is called united extension or natural extension of the real function. One of the properties of this extension is monotonic inclusion : if $f(x_i)$ is a function into a parameter space $x_i \in X_i$, $F(X_i)$ is the united extension of $f(x_i)$ and it complies that $f(X_i) \subseteq F(X_i)$.

United extension is very useful to compute range of functions because it gives a guarantee on the result (function can not take values outside the range computed using interval arithmetic).

On the other hand, exact result (complete and sound) can not be obtained in some cases. Interval arithmetic considers that each incidence of a variable in a function is independent of each other. Therefore, when there are multi-incident variables (variables that appear at different places into an only function), interdependent variables (there is a relation among them) or variables that depend on the same variable, the range obtained (an interval) includes the values that can take the function plus other values that the function can never take.

One way to tighten the result when there are multi-incident variables is splitting the parameter space into smaller subspaces.

5.2.2. Use into simulation

There are some semiqualitative simulators based on interval arithmetic and monotonic inclusion. Some of them use integration algorithms adapted to work with intervals.

Moore's interval simulator [Moo66] [Moo79] obtains much overbounded envelopes because ignores multi-incidences. Moreover, there is the wrapping problem. This problem is linked to the use of interval arithmetic and was described for the first time by Moore [Moo66]. If the state of the system is represented by an hypercube at a time point because each variable is an interval, maybe the system will not evolve to another hypercube at the following time point. In figure 9 it is shown an example where there are two state variables and therefore the hypercube into the parameter space is a rectangle. This rectangle evolves to a rhombus in the following time step (it could evolve to any figure in two dimensions). As the value of each variable is expressed with an interval, new state will be represented with a new rectangle, which includes all possible states and impossible states as well. These impossible states are shown shadowed in the figure. Hence, the envelopes obtained will be overbounded and, possibly, they will be unstable. This problem can not be solved simulating with smaller steps, because more steps will be needed.

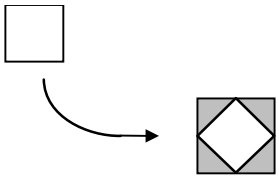

figure 9. The wrapping problem.

Other simulators based on these algorithms are the interval simulator by Markov and Angelov [Mar86] and the simulator AWA [Loh88], which tries to minimise the wrapping effect moving the system of co-ordinates along the trajectory of the system into the parameter space. This goal is achieved in some cases [Kay96].

NIS (Numerical Interval Simulation) [Ves95] uses interval arithmetic to compute, at every time step of the simulation, maximum and minimum values the derivative of each variable can take. These values are used for Euler or Runge-Kutta integration algorithms. The obtained envelopes are overbounded

Other simulators based on interval arithmetic are described below.

5.2.2.1. NSIM

NSIM (Numerical Simulator using Interval Methods) [Kay92] [Kay95] [Kay96] uses the same models than QSIM and is an enhancement of Moore interval simulator when there exists quasi-monotonicity. A system is called quasi-monotonic if the exact envelopes can be obtained through the simulation of extreme systems [Kay96].

Given an interval function (a differential equation where the parameters are intervals), this simulator obtains a pair of functions (extreme functions) which envelop the original function in a certain dominion. Hence, completeness is guaranteed. After this, a conventional simulator is used for the simulation. This is one of its advantages: it is no necessary to develop a specialised simulator and therefore it makes good use of speed and efficiency of existing simulators and exploits their possible enhancements.

The disadvantage is that if there are multi-incident or correlated variables, extreme functions do not belong to the original family of functions and therefore they are outside its dominion. The envelopes obtained in these cases are overbounded and they can diverge and become unstable, that is, width of the envelope grows through time.

Another problem is wrapping. To solve these problems, Kay proposes various ways:

- Wrapping problem. Definition of the intervals with respect to a mobile origin of co-ordinates instead of using a fixed origin.
- Ignored correlations. Build models without ignored correlations. That is, decrease number of multi-incident variables and number of incidences of them.
- Split state space into smaller subspaces. As it was noted above this makes the result of evaluating functions using interval arithmetic closer to the exact result when there are multi-incident variables.

5.2.2.2. Simulator of Gasca *et al.*

[Gas96b] As stated above, one way to avoid the wrapping problem when using interval arithmetic is to define the intervals with respect to a mobile origin of co-ordinates instead of defining them with respect to a fixed origin. This can be done computing the trajectory of one of the systems that belongs to the family under study and then superposing a threshold to it. This simulator is based on this principle. The models are represented by band functions and the parameters are represented by intervals covering all real values corresponding to each qualitative label. Trajectories of the family of systems are represented by polyhedrons in the state space defined from a point, a transformation matrix and an uncertainty vector.

Given that transformation matrix is obtained after some simplifications and approximations, the properties (completeness, soundness, stability) of the resulting envelope are not known.

5.3. Qualitative simulation

In order to perform qualitative reasoning it is necessary to have a model and some information about the variables of this model. The model can be qualitative information on the relations between variables. For instance, if value of variable *a* increases, value of variable *b* also

increases. Information about variables, from a purely qualitative point of view, can be a positive value, a negative value or zero (there is a fourth possibility to be considered: unknown value), hence real axis is divided into three parts. This information is used by an inference engine and some conclusions are obtained [Cog96]. These tools for reasoning are very useful to solve problems where information is poor, like qualitative knowledge, but since the information is poor, the results are poor as well.

Qualitative simulation makes a prediction of the qualitative states where the system will be, using a non-numeric model. This means it distinguishes qualitatively (with labels, not with numerical values) the states where the system will be or the values the variables will have.

Most of the qualitative simulators are not causal and hence do not consider the time. Therefore, they predict which states will exist in the future but not when will they exist. Some qualitative simulators, however, are causal and consider time.

Qualitative simulation consists in two phases [Cog96]. In the first one, called TA (Transition Analysis), many states are generated without using the model of the system. In the second phase, called QA (Qualitative Analysis), these states are filtered using the model, hence states that do not fulfil the constraints of the model are eliminated. This is called constraint propagation.

Qualitative simulators are used when knowledge of the system has important limitations or when it is interesting to have qualitative results. These results are poor because they are obtained from a poor knowledge. If the complexity of the systems to be simulated increases, the predictions become poorer [Bon96] and the computational effort to be made increases too. There are big deficiencies in order to take advantage of the possible numerical knowledge of the system. In the of interval systems, the envelopes obtained are much overbounded.

In [Cog96] a classification of simulators according to their degree of constructivity is done. A numeric simulator is constructive because it uses the model of the system to generate the values of the variables of the system: it solves the equations and if it is a dynamic system it integers them. In other words, it constructs future values of variables starting from the model of the system. On the other hand, a qualitative simulator is not constructive because the model of the system is used only to filter out the impossible states that have been generated before. Semiqualitative simulators are somewhere between these two extreme points, according to the degree of use of the system model. If a simulator is more constructive than another simulator, it is more efficient in order to take advantage of the available information and generates less impossible states [Wie91]. In spite of this, in some cases a non-constructive simulator is very useful. For instance when there are algebraic loops a non-constructive simulator can be used but a constructive simulator can not.

Some qualitative simulators are described below: QSIM, PA and Ca~En. They are only a sample of the existing ones: SQUALE [Mis91], DIAPASON, etc.

5.3.1. QSIM

Most popular qualitative simulator is QSIM (Qualitative Simulator) [Kui86]. Probably it is also the most sophisticated due to its continue evolution since it was first implemented [Cog96]. Since then, new elements are being added continuously.

Each variable is represented by its value and the value of its derivative. The value of the derivative is expressed in a purely qualitative form: *inc* for increasing (positive derivative), *dec* for decreasing (negative derivative) or *std* for steady (zero derivative, stable variable). The value of the variable can be expressed also in a purely qualitative form or, if more information is available, can be expressed using labels, that is defining a value space discretising real axis in more than three parts using landmarks. Since each variable is represented with a pair magnitude/derivative, time function of the value of the variable is approximated with a first order Taylor series.

The model of a system is represented by the relations between its variables: algebraic operations (sum, product, sign change), derivative (one variable is the derivative of another), monotony (if the value of a variable increases, the value of the other increases too).

Algorithm of simulation consists of three parts: transition rules, constraint and Waltz filters, and global filter. Transition rules are based on the Euler numeric integration algorithm [Cog96]. They are applied to each variable individually, as if each variable was independent from each other. In consequence, impossible transitions are generated. In the following phase, some of these impossible transitions are eliminated because they do not fulfil the constraints.

QSIM is not a constructive simulator because it does not use the model of the system to generate the transitions.

The result of the simulation is a representation of the qualitative states (expressed by the variables' values and the values of their derivatives) that will succeed to the present state. The duration of these states is not given.

Original QSIM was complete [Kui86], as it found all possible behaviours of the system. But new filters, which optionally can be used or not, have been added through the years and some of them make QSIM lose this property [Cog96]. On the other hand, it is not sound because it can find behaviours that are not real. One of the reasons is that if a predicted state coincides totally or partially with various labels, it is considered that the variable can take any of the values included in these labels, including the ones that have not been predicted.

5.3.2. PA

PA (Predictive Algorithm) [Wie91] is a qualitative simulation algorithm integrated in PE (Predictive Engine) [Wie89]. It belongs to the same family than QSIM but it is more constructive than QSIM. The problem of QSIM is that in the first phase, many impossible states are generated and in the second phase, some of them can not be eliminated. The solution is to generate less impossible states in the first phase making it more constructive, more similar to a numeric simulator.

In PA, the user chooses the number of successive derivatives to be used for each variable. QSIM uses the value of each variable and the value of its first derivative and if the user wants to use higher order derivatives new variables must be created, which will be considered independent in the first phase of the simulation. This makes PA more flexible for the model of the system, that can be interpreted as a set of differential planes. Hence, there were two planes in QSIM and

there can be the desired number of planes in PA [Cog96]. Such a description of the model implies a set of constraints that are used to generate the states of the variables. The procedure to generate the transitions begins at the plane of the highest order derivatives and decreases the order to end at the plane with no derivatives (zero order derivatives). Therefore, PA already uses the system model in the first phase of the simulation while QSIM uses it only in the second phase for filtering. That is the reason to affirm that PA is more constructive than QSIM.

Another aspect that makes PA more constructive than QSIM is that in QSIM the order of the equations is not important and in the first phase of the simulation, all variables are considered independent. However, in PA equations must be causally ordered so less impossible transitions are generated in the first phase.

In spite of that, PA is not totally constructive because temporal information obtained from the simulation is insufficient.

In the phase of filtering, transitions are verified using the constraints. If a transition does not fulfil all constraints, it is eliminated. If a transition fulfils all constraints the transitions yet unverified are eliminated and the simulation continues hence ignoring other possible transitions. This makes PA incomplete.

5.4. Semiqualitative simulation based on qualitative simulation

To enhance qualitative methods, quantitative information must be added. Resulting methods are semiqualitative methods. Some of the existing semiqualitative simulators are described below.

5.4.1. Semiqualitative simulators based on interval arithmetic

5.4.1.1. Q2

Q2 [Kui88] is an extension of QSIM [Kay92]. The basis of the simulator is QSIM to which an interval arithmetic module has been added. This module works in parallel with QSIM and computes the value of each variable using the constraints. The final value is computed intersecting the values given by this module and by QSIM. Therefore, the value of each variable is no more qualitative (a predefined label). It is an interval value [Cog96].

Sometimes intersection gives nothing. This means the predicted state is impossible so it is eliminated. In conclusion, envelopes are tighter than QSIM ones.

Times are calculated using first order Taylor-Lagrange formula. In [Mis91] there is a demonstration that it does not give significant information near critical points with zero derivatives [Ves95].

5.4.1.2. Q3

[Ber92] Numerical simulators are more precise if simulation step decreases. This principle is applied by Q2 to qualitative simulation. It inserts intermediate states in qualitative simulation

hence number of states increases, number of constraints increases and precision increases. Therefore, it is convergent: as more intermediate states are added, more precise are the results. The problems are rounding errors of computers and computing power needed. Importance of both of these two things increases faster than precision.

5.4.1.3. SQSIM

Kay [Kay96] has combined simulators QSIM, Q2, Q3 and NSIM into an only one called SQSIM (Semiqualitative Simulator), that intersects results given by each of them. If overbounded envelopes are intersected, a new tighter overbounded envelope is obtained. If any of the envelopes intersected is not overbounded, the result will be tighter, but it will not be overbounded. This is the problem of SQSIM: some of the simulators used give envelopes with unknown properties hence envelopes obtained using SQSIM have unknown properties as well.

5.4.1.4. Ca~En

Ca~En (*Causal Engine*) [Bou94] is a reasoning system for real-time applications. It is based on causality and constraints and it includes a qualitative simulator that works with a real-time clock. It works with numeric and symbolic (for instance colour) variables.

Models are represented using two levels: local constraints' level and global constraints' level. In the local constraints' level, influences between variables are made explicit. In the global constraints' level constraints derived from physical laws are indicated. Therefore, Ca~En uses a representation that needs multi-model reasoning.

For the simulation, value of each variable through time is computed using interval arithmetic with Euler integration algorithm. After this, constraints are used to refine the results.

The obtained envelopes are overbounded due to the use of interval arithmetic.

5.4.2. Semiqualitative simulators based on fuzzy logic

Follows a description of some semiqualitative simulators based on fuzzy logic. It will be seen that all of them use interval arithmetic after fuzzy sets are converted into intervals through α-cut.

5.4.2.1. FuSim

FuSim (*Fuzzy Qualitative Simulation*) [She93] is an extension of QSIM. It is not constructive, like QSIM. In FuSim, variables' values are given by intervals instead of qualitative labels. It indicates the time the system will remain in each state. This time is computed using first order Taylor-Lagrange formula, like Q2, and hence it has the same disadvantages as Q2 near the points with zero derivatives.

When the value a variable will have at a time point is predicted, it does not coincide, in general, with one of the possible fuzzy values of the variable. Then the likeliest value is chosen using metric distance and state priorization. This action makes FuSim unsound (it includes impossible states) and incomplete (possible states are not included)

5.4.2.2. Mycroft

Mycroft [Cog96] includes two simulators (a non-constructive simulator and a semi-constructive one) plus other elements. These simulators are based on QSIM, FuSim and PA. It takes advantage of the best characteristics of each one and enhances some of them. For instance: works with *n* derivatives of each variable, like PA; generates time information thanks to the use of fuzzy sets, like FuSim; makes simulation semiconstructive because equations are causally ordered, like in PA. Some of the new features are a new priorization algorithm, the way to deal with intermediate variables, etc.

Semi-constructive simulator is not complete. A proof is that in some simulations there are not predicted future states because all generated states are filtered out.

5.4.2.3. Qua.Si.

[Bon94] [Bon96] They are three simulators (Qua.Si. I, II and III) for continuous dynamic systems. Model is expressed using differential equations where parameters or initial conditions can have fuzzy values.

Qua.Si. I uses the scalar systems method, described above, reconstructing the hypercube at every time step. It simulates characteristic points of the hypercube in the state space: vertices, centre of edges, centre of faces, etc. Therefore, it produces incomplete envelopes. Moreover, the envelopes are unsound, due to wrapping problem, and unstable.

In Qua.Si. II the hypercube is not reconstructed at every time step hence envelopes are underbounded.

Qua.Si. III treats simulation like an optimisation problem. Found optima are not guaranteed because it uses optimisation methods with no guaranty. Computational complexity grows exponentially with time (at each time point the optima are computed with respect to the initial state) and when the order of the system grows (the initial hypercube has more surfaces).

6.- CONCLUSIONS.

Different options to simulate interval systems' behaviour have been presented. Existing simulators can be classified into three groups:

- Quantitative. Numerical techniques are applied:
  - Fixed threshold calculus. Completeness of the envelopes is guaranteed only if threshold is high, but this is not very useful because envelopes are much overbounded.
  - Adaptive threshold calculus. These methods are not adequate for interval systems. They are better when parameters' values are given by probability distributions.

    - Simulation of scalar systems. Obtained envelopes are underbounded.
    - Simulation as an optimisation problem. If there is no guarantee that the optimisation method will find global optima, envelopes will be underbounded.
- Qualitative. Based on qualitative reasoning. They do not take advantage of numerical information contained into an interval model. QSIM brings overbounded envelopes, but there are some optional filters that make it give uncomplete envelopes. PA is incomplete because it finds only one possible behavior, not all of them.
- Semiqualitative. There are two types of semiqualitative simulators:
    - Based on qualitative reasoning plus numerical knowledge. Q2 and Q3 obtain tighter envelopes than QSIM adding numerical techniques to it. Properties of the envelopes have not been studied. SQSIM intersects envelopes produced by QSIM, Q2, Q3 and NSIM. If all the envelopes generated by these simulators were overbounded, the intersection will be overbounded too, but tighter. But as it was stated above, some of these simulators have unknown properties.
    - Not based on qualitative reasoning. Interval methods can be combined with numerical methods thus giving semiqualitative methods. Interval methods have guarantee over the results because united extension of a function has the monotonic inclusion property. Envelopes obtained by NSIM and NIS are much overbounded. Properties of simulator of Gasca *et al* have not been studied yet.

Therefore, existing simulators generate underbounded envelopes, overbounded envelopes or envelopes with none of these properties or with unknown properties. All of them have a common characteristic: even if the properties are known, it is not known the degree of under or overbounding. Hence, if one of these simulators produces overbounded envelopes, for instance, it is not known if these envelopes are much overbounded or only a few overbounded, it is not known the distance between the obtained envelope and the exact envelope. Of course, if this distance could be known, the exact envelope could be known as well.

7.- FUTURE WORK.

The goal is to obtain known-error envelopes. This can be done with a simulator that produces simultaneously an underbounded envelope and an overbounded envelope. The distance between these envelopes is the maximum error.

Another goal is to create an algorithm to adjust the error to a desired value, widening the underbounded envelope or tightening the overbounded one. Modal Interval Analysis [Gar95] seems to be useful for this goal. It is not difficult to predict that if desired error decreases, the computational effort will increase.

If these goals are achieved, the new simulation technique will be included into Ca~En simulator.

Acknowledgements

This work have 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 coordinated research project CICYT TAP96-1114-C03-03, “Plataformas integradas de CAD de Supervisión y Metodologías”.

REFERENCES.

[Ber92] Berleant, Daniel, Benjamin J. Kuipers. Qualitative and Numerical Simulation with Q3. In [Fal92].

[Bon94] Bonarini, Andrea, Gianluca Bontempi. A qualitative simulation approach for fuzzy dynamical models. ACM transactions on modeling and computer simulation (TOMACS). Vol. 4, No. 4, pp. 258-313. 1994.

[Bon96] Bontempi, Gianluca. Qua.Si. III : A software tool for simulation of fuzzy dynamical systems. In Proceedings of the European Simulation Multiconference (ESM 96). Pp. 615-619. A. Javor, A. Lehmann, I. Molnar, Eds. SCS International. Ghent, Belgium. 1996.

[Bou94] Bousson, Kouamana, Louise Travé-Massuyès. Putting more numbers in the qualitative simulator CA-EN. In Proceedings of the second International Conference on Intelligent Systems Engineering (ISE 94). Hamburg-Harburg, Germany. 1994 09 05 - 1994 09 09. Pp. 62-69.

[Cog96] Coghill, George Macleod. Mycroft : a framework for constraint based fuzzy qualitative reasoning. Doctoral thesis. Department of computing and electrical engineering. Heriot-Watt University. Edinburgh, Scotland. 1996 09.

[Fal92] Faltings, Boi, Peter Struss, eds. Recent advances in qualitative physics. The MIT Press. Cambridge, Massachussets. 1992. ISBN 0-262-06142-2.

[For88] Forbus, Kenneth D. Commonsense physics: a review. In Annual Review of Computer Science, vol. 3, pp. 197-932. 1988.

[Gar95] Gardeñes, Ernest, Honorino Mielgo, Miguel Angel Sáinz. Presentation of the research group SIGLA/X. Report de recerca IMA 95-10. Universitat de Girona.

[Gas96b] Gasca, Rafael Martínez, Juan Antonio Ortega, Miguel Toro. Simulación semicualitativa. In Proceedings of Jornades Hispano-Franceses de Sistemes Intel·ligents i Control Avançat. Barcelona, 1996 11 12 - 1996 11 13.

[Kay92] Kay, Herbert, Benjamin J. Kuipers. Numerical behavior envelopes for qualitative models. In Proceedings of the sixth International Workshop on Qualitative Reasoning about Physical Systems (QR 92). Pp. 252-267. Edinburgh, Scotland. 1992 08 24 - 1992 08 27.

[Kay95] Kay, Herbert. Semiquantitative Simulation : Successes, Failures, and Future Directions. 1995.

[Kay96] Kay, Herbert. Refining imprecise models and their behaviors. Ph. D. dissertation. University of Texas at Austin. 1996 12.

[Kui86] Kuipers, Benjamin J. Qualitative Simulation. Artificial Intelligence. Vol. 29, pp. 289-338. 1986 09. Also in [Wel90], pp. 236-260.

[Kui88] Kuipers, Benjamin J., Daniel Berleant. Using Incomplete Quantitative Knowledge in Qualitative Reasoning. In Proceedings of the Seventh National Conference on Artificial Intelligence AAAI-88. American Association for Artificial Intelligence. Pp. 324-329. Morgan Kauffmann Publishers. Los Altos, California. 1988.

[Loh88] Lohner, Rudolf. Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen. Ph. D. thesis. Universität Karlsruhe, Germany. 1988.

[Mar86] Markov, S., R. Angelov. An interval method for systems of ODE. In Lecture Notes in Computer Science #212 - Interval Mathematics 1985, pages 103-108. Springer-Verlag, Berlin, 1986.

[Mis91] Missier, Antoine. Structures Mathematiques pour le Calcul Qualitatif - Contribution a la Simulation Qualitative. Ph. D. thesis, I. N. S. A. Toulouse, France. 1991.

[Moo66] Moore, Ramon Edgar. Interval Analysis. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[Moo79] Moore, Ramon Edgar. Methods and Applications of Interval Analysis. Studies in Applied Mathematics. SIAM, Philadelphia, 1979.

[She93] Shen, Qiang, Roy R. Leitch. Fuzzy qualitative simulation. In IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 4. Pp. 1038-1061. 1993 07 - 1993 08.

[Ves95] Vescovi, Marcos R., Adam Farquhar, Yumi Iwasaki. Numerical Interval Simulation. International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canada. 1995 08 20-1995 08 25. (IJCAI 95). Pp 1806-1813.

[Wel90] Weld, Daniel S., Johan de Kleer, eds. Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers, Inc. San Mateo, California. 1990. ISBN 1-55860-095-7.

[Wie89] Wiegand, M. E., Roy R. Leitch. A Predictive Engine for the qualitative simulation of dynamic systems. In Proceedings of the fourth International Conference on the Application of Artificial Intelligence in Engineering. Cambridge, United Kingdom. 1989.

[Wie91] Wiegand, M. E. Constructive qualitative simulation of continuous dynamic systems. Ph. D. Thesis. Heriot-Watt University. Edinburgh, Scotland, 1991 05.