An Approach for Reasoning about Semiqualitative Models with Explicit Constraints

J.A. Ortega, R.M. Gasca, M. Toro
Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Sevilla
Avda. Reina Mercedes s/n. Sevilla (Spain)
{ortega,gasca,mtoro}@lsi.us.es

Abstract

In our approach, a semiqualitative model of a dynamic system is expressed by means of a set of constraints among variables, parameters and intervals. Applying the methodology (Ortega99) (Ortega00a) these semiqualitative models with constraints were transformed into a family of quantitative models without explicit constraints. In this paper, the methodology is enriched to manage family of quantitative models where may appear explicit constraints.

The main idea of the proposed improvement is as follows: Given a non-causal semiqualitative model with constraints, it is rewritten into an equivalent model by means of the transformation of the qualitative knowledge. This non-causal model is expressed by means of a set of constraints where the states variables may be included. These constraints are ordered and transformed into an equivalent set of constraints in order to apply the whole methodology. This new set represents a family of quantitative models with explicit constraints. This extended methodology is applied to an interconnected tank model with constraints.

Introduction

In real systems studied in science and engineering, knowledge about dynamic systems may be quantitative, qualitative, and semiqualitative. When these models are studied all this knowledge should be taken into account. Different levels of numeric abstraction have been considered: purely qualitative (Kuipers94), semiqualitative (Kay96), (Berleant97) and (Ortega98), and quantitative.

Different approximations have been developed in the literature when qualitative knowledge is taken into account: distributions of probability, transformation of non-linear to piecewise linear relationships, MonteCarlo method, fuzzy sets, causal relations, and combination of all levels of qualitative and quantitative abstraction (Kay96).

In this paper, it is used the idea of causal ordering in a system of structural equations (Simons52), which formal basis was defined in (Iwasaki86). On the other hand, it is also used the idea of automating the qualitative analysis of physical systems (Sacks91) which proposes a way to carry out through a combination of theoretical dynamics, numerical simulation, and geometric reasoning. We use these works to improve the methodology introduced in (Ortega99) (Ortega00a).

This paper is proposed a technique based in our previous works to study non-causal dynamic systems with qualitative and quantitative knowledge. This technique is appropriate to study some systems, however the methodology cannot be always applied. In section *Restrictions and problems* are described some limitations of our approach, however it is necessary to carry out a more detailed study.

The main idea of the methodology is as follows: *a semiqualitative model with explicit constraints is transformed into a family of quantitative models. Every quantitative model has a different quantitative behaviour, however they may have similar qualitative behaviours.*

The methodology applies two transformations to the original model. Let be a non-causal semiqualitative model with constraints. Firstly, it is rewritten into an equivalent model where the qualitative knowledge has been transformed. This obtained non-causal model is expressed by means of a set of constraints where the states variables may be included. Secondly, these constraints are ordered and transformed into an equivalent set of constraints to apply the remaining methodology. This new set represents a family of quantitative models with explicit constraints.

In this paper, these two transformation techniques are studied in detail. It is also proposed a way to compile the transformed model into an equivalent model written like it is required by the commercial simulation environment *Vensim*[1].

The paper is organized as follows: firstly, the concept of semiqualitative model with explicit constraints is defined and the approach is detailed. Secondly the kind of qualitative knowledge we are using is introduced, and its transformation process is described. Thirdly, a tech-

[1] *Vensim* is a register mark by Ventana Systems, Inc.

nique is proposed to order the set of equality constraints. Next, the transformation of the ordering constraints and its implementation details are described. Finally, the approach is applied to an interconnected tank model with explicit constraints.

Semiqualitative models with explicit constraints

In this paper, we focus on those dynamic systems where there may be qualitative knowledge in their parameters, initial conditions and/or vector field. They constitute the semiqualitative differential equations of the system. A semiqualitative model S is represented by means of

$$\Phi(\dot{x}, x, y, q, t), \qquad x(t_0) = x_0, \qquad \Phi_0(q, x_0) \qquad (1)$$

being $x \in I\!R^n$ the state variables, y auxiliary variables, q the parameters, t the time, $\dot{x}$ the derivative of the state variables, Φ constraints among $\dot{x}, x, q, t$, and Φ_0 constraints with the initial conditions. These constraints may be composed of qualitative knowledge, arithmetic and relational operators, intervals, predefined functions, $(ln, exp, sin, \ldots)$ and numbers.

Our approach

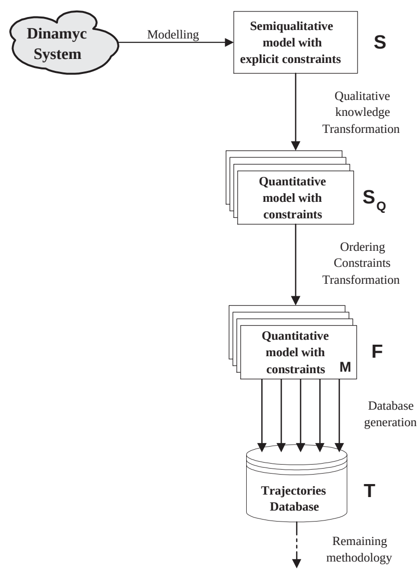

Figure 1: Our approach

The proposed approach in this paper is shown in figure 1. Let S be the semiqualitative model with explicit constraints defined in (1) by means of

$$\Phi(\dot{x}, x, y, q, t), \qquad x(t_0) = x_0, \qquad \Phi_0(q, x_0) \qquad (2)$$

Applying the transformation techniques of the qualitative knowledge to S is obtained S_Q given by:

$$S_Q \equiv \begin{cases} \psi(\dot{x}, x, y, p, t), \\ \Psi_H, \\ x(t_0) = x_0, p \in I_p, x_0 \in I_0 \end{cases} \qquad (3)$$

where p are the parameters of the system, ψ are the new obtained constraints and I_p, I_0 are real intervals. Let Ψ_H be the set of constraints obtained with the transformation of the qualitative continuous functions of the model. Firstly, this transformation process completes the definition of the function. Secondly, this definition is enriched by means of an automatic process which incorporates new landmarks. Finally, the qualitative function is transformed into a set of quantitative functions whose behaviours are in accordance with the definition of the function (in (Ortega99) is explained the transformation process in detail).

The equation (3) denotes a family of semiqualitative models with constraints where may be state variables into the constraints. In order to apply the remaining methodology (Ortega99), it is necessary to transform (3) into an equivalent model given by:

$$F \equiv \begin{cases} \dot{x} = f(x, y, p, t), \\ \varphi(x, y, q, t) = 0, \\ \theta(x, y, p, t) > 0, \\ \Psi_H, \\ x(t_0) = x_0, \quad p \in I_p, \quad x_0 \in I_0 \end{cases} \qquad (4)$$

where φ, θ are constraints among parameters and variables, being φ equalities and θ inequalities. On the other hand, f is the obtained function in accordance with section . Therefore, this second transformation only changes the form and the order of the constraints:

$$\begin{gathered} \psi(\dot{x}, x, y, p, t) \\ \Downarrow \\ \dot{x} = f(x, y, p, t),\ \varphi(x, y, q, t) = 0,\ \theta(x, y, p, t) > 0 \end{gathered} \qquad (5)$$

Stochastic techniques are applied to choose every quantitative model of the family of quantitative models F. The number of different models to obtain of F is discussed in (Ortega00b). Every obtained model is quantitatively simulated obtaining a trajectory. A trajectory contains the values of the parameters and the values of all variables from their initial value until their final value. These quantitative trajectories are stored into a database.

Knowledge Discovery in Databases (KDD) techniques are applied to study the system. It is possible to carry out queries about the qualitative properties of the set of trajectories included in the database, using the language proposed in (Ortega99).

A labeled database is obtained when these trajectories are classified according to some criteria. Qualitative behaviours patterns of the system may be automatically obtained from this database by applying machine learning based on genetic algorithms. These steps of our approach were detailed in (Ortega99) (Ortega00a), and therefore they are obviated in this paper.

Qualitative knowledge

Qualitative knowledge about a model may be composed of qualitative operators, qualitative labels, envelope functions and qualitative continuous functions.

In this section, the representation and transformation techniques of this qualitative knowledge are described in a summarize way. A complete description of these transformation techniques is included in (Ortega99).

Qualitative operators

Every qualitative operator or qualitative label *op* is defined by means of an interval I_{op} which is supplied by the experts.

Unary qualitative operators Every magnitude of the problem with qualitative knowledge has defined its own unary operators.

Let U_x be the unary operators for a variable x, i. e., $U_x = \{VN_x,\ MN_x, LN_x, AP0_x, LP_x, MP_x, VP_x\}$. They denote for x its qualitative labels: *very negative, moderately negative, slightly negative, approximately zero, slightly positive, moderately positive,* and *very positive* respectively.

The transformation rule for a unary operator is

$$op_u(e) \quad \equiv \quad \begin{cases} e - \alpha = 0 \\ \alpha \in I_u \end{cases} \tag{6}$$

being α a new generated variable, and I_u the interval associated with operator op_u which is established in accordance with (Travé-Massuyès97).

Binary qualitative operators Let e_1, e_2 be two arithmetic expressions. A binary qualitative operator $b(e_1, e_2)$ denotes the qualitative order relationship between e_1 and e_2. These operators are classified into

- Operators related to the difference $\geq, =, \leq$. The following transformation rules are applied

$e_1 = e_2$	$\equiv$	$e_1 - e_2 = 0$
$e_1 \leq e_2$	$\equiv$	$\begin{cases} e_1 - e_2 - \alpha = 0 \\ \alpha \in [-\infty, 0] \end{cases}$
$e_1 \geq e_2$	$\equiv$	$\begin{cases} e_1 - e_2 - \alpha = 0 \\ \alpha \in [0, \infty] \end{cases}$

Table 1: Transformation rules

- Operators related to the quotient $\lll, -<, \sim, \approx, \ggg, Vo, Ne, ...$ The applied transformation rule is

$$op_b(e_1, e_2) \quad \equiv \quad \begin{cases} e_1 - e_2 * \alpha = 0 \\ \alpha \in I_b \end{cases} \tag{7}$$

being α a new variable and I_b the interval associated to op_b which is defined in according to (Travé-Massuyès97).

Envelope functions

An envelope function $y = g(x)$ (Figure 2) represents the family of functions included between two defined real functions: a upper one $\overline{g} : \mathbb{R} \rightarrow \mathbb{R}$ and a lower one $\underline{g} : \mathbb{R} \rightarrow \mathbb{R}$. An envelope function is represented by means of

$$y = g(x)\ \langle \underline{g}(x), \overline{g}(x), I \rangle, \quad \forall x \in I : \underline{g}(x) \leq \overline{g}(x) \tag{8}$$

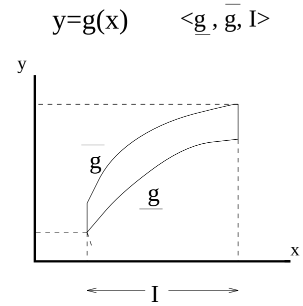

Figure 2: Envelope function

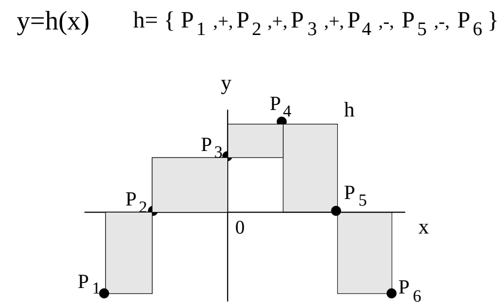

Figure 3: Qualitative continuous function

being I the definition domain of g, and x the independent variable.

The transformation rule applied to (8) is

$$g(x) = \alpha \underline{g}(x) + (1 - \alpha)\overline{g}(x) \qquad \text{with} \quad \alpha \in [0, 1] \tag{9}$$

where α is a new variable. If $\alpha = 0 \Rightarrow g(x) = \overline{g}(x)$ and if $\alpha = 1 \Rightarrow g(x) = \underline{g}(x)$. Any other value of α in (0,1) stands for any included value between $\underline{g}(x)$ and $\overline{g}(x)$.

Qualitative continuous functions

A qualitative continuous function $y = h(x)$ (figure 3) represents a constraint involving the values of y and x according to the properties of h. It is denoted by means of

$$y = h(x), \quad h \equiv \{P_1, s_1, P_2, ...s_{k-1}, P_k\} \tag{10}$$

being P_i the points of the function. Every P_i is defined by a couple (d_i, e_i), being d_i and e_i the qualitative landmarks associated to the variables x and y respectively. These points are separated by the sign s_i of the derivative in the interval between two consecutive points. A monotonous qualitative function is a particular case of these functions where the sign is always the same $s_1 = ... = s_{k-1}$.

The transformation rules of a qualitative continuous function are applied in three steps: normalization, extension and transformation. They are described in (Ortega99). Let Ψ_H be the set of obtained constraints applying the transformation rules to all qualitative continuous function of the model.

Ordering equalities

This section describes an approach to order equalities. It is applied to carry out the transformation describes in section . Let be a set of constraints only with equalities:

$$\varphi(x, u) = 0 \tag{11}$$

being $x_1, ..., x_n$ the variables to solve in the set of constraints $\varphi_1, ..., \varphi_n$, and $u_1, ..., u_m$ are the remaining variables. This set of constraints will be transformed into:

$$\begin{cases} x_1 = f_1(u), \\ x_2 = f_2(u, x_1), \\ x_3 = f_3(u, x_1, x_2), \\ ... \\ y_1 = h_1(u, x_1, ...), \\ ... \\ x_p = f_p(u, x_1, ..., y_1, ...) \end{cases} \tag{12}$$

where x are the variables to solve. Every y variable denotes a set of variables $x_i, ..., x_j$ which will be simultaneously solved by means of symbolic techniques. The way to determine if a set of $x_i, ..., x_j$ variables may be solved simultaneously is below described in section . The functions f, h are obtained taking into account that the variable to solve, x or y depends exclusively on the previous solved variables, and obviously on u variables.

This transformation will be carry out in three steps:

1. Constructing a bipartite weighted graph.
2. Solving the bipartite weighted assignment problem.
3. Topological sorting of the assigned graph.

Constructing a bipartite weighted graph

A graph $G = (V, E)$ is called *bipartite* if its set of vertices can be partitioned into two subsets such that there is no edge connecting two vertices from the same subset.

This first step constructs a bipartite graph starting from (11). Let $G_b = (\varphi, x, E)$ be this bipartite graph, where the vertices of G_b are the constraints φ and the variables to solve x, and the edges E of G_b connect each constraint with its variables.

This graph is also weighted. Let φ_i be a constraint and let x_j be a variable of φ_i, therefore there is an edge between the vertices x_j and φ_i. This edge is weighted with a value p_{ij}, which denotes the *solving degree* of x_j in φ_i. The solving degree is established in accordance with the difficulty to solve that variable x_j in the constraint φ_i. The value of the solving degree is 1 when the variable x_j is easy to solve, and this number becomes bigger if the variable is more difficult to solve. Therefore, the value $+\infty$ indicates that it is impossible to solve the variable. Next table contains some values for the solving degree: The solving degree is automatically defined by means of pattern matching of symbolic mathematical expressions.

Solving degree	How can you give explicit algebraic formula?
1	symbolically polynomial
5	symbolically polynomial with quotient
10	symbolically no polynomial
15	Newton method
$+\infty$	Impossible

Table 2: Solving degree

Solving the assignment problem

The bipartite weighted assignment problem, or, as it is usually called the *assignment problem* is a particular case of the standard matching problem, but with weights.

Let $G_b = (\varphi, x, E)$ be the previous bipartite weighted graph. The assignment problem tries to assign one and only one variable $x_j \in x$ to every constraint $\varphi_i \in \varphi$ minimizing the sum of the solving degrees $p_{ij} \in E$. Let $j = Assig(i)$ be the assignment function of the variable x_j to the constraint φ_i. Therefore, the objective of the algorithm is as follows:

$$min(\sum_{i=1..m,\ j=Assig(i)} p_{ij}) \tag{13}$$

where m is the number of constraints (the cardinality of φ), j is a number indicating that the variable x_j will be solved in the constraint φ_i. This is a minimization problem. The *Hungarian method* (Kuhn55) solves this bipartite weighted assignment problem by means of an efficient algorithm (Papadimitriou82).

When a variable x_j is assigned to a constraint φ_i by means of the function $j = Assig(i)$, it means that this variable is the best candidate to transform $\varphi_i(x_j, x_k, ...) = 0$ into $x_j = f_i(x_k, ...)$.

This function $Assig$ defines a partial order relation among the variables to solve:

$$x_i < x_j \text{ if } \exists k \ : \ x_i, x_j \in var(\varphi_k) \wedge j = Assig(k) \tag{14}$$

being $var(\varphi)$ the set of variables of φ.

Topological sorting of the assigned graph

A directed graph is constructed $G_d = (V, E')$ starting from the previous bipartite graph $G_b = (\varphi, x, E)$. Every vertex $v \in V$ is a constraint $\varphi_i \in G_b$ and its assigned variable $j = Assig(i)$. The edges $e \in E'$ are established in accordance with the partial order relation among variables defined in (14). An edge is defined between two vertices when there is an order relation between their two assigned variables $x_i < x_j$, and the sense of the edge is from x_i to x_j. This is the way to construct the directed graph G_d.

A topological sorting algorithm (Kahn62) is applied to the graph G_d. It works as follows: if there is a path from v_i to v_j then v_j appears after than v_i in the final sorting. This algorithm has been implemented as is described in (Manber89).

The results of the algorithm indicates the order to calculate the variables, and therefore the result of the transformation of the equalities constraints (12).

An example

Let be these equalities constraints

$$\begin{cases} \varphi_1(x_1, x_2, u_1) = 0, \\ \varphi_2(x_2, x_3, u_2) = 0, \\ \varphi_3(x_2, u_3) = 0 \end{cases}$$

where x_1, x_2, x_3 are the variables to solve, and u_1, u_2, u_3 the rest of variables. In figure 4 appears the bipartite weighted graph of this set of constraints. It has been supposed that: the variable x_1 is easy to solve and x_2 is impossible to solve in φ_1; the variable x_3 is easily solved, and if the variable x_2 is solved then a polynomial with a quotient expression is obtained in φ_2; and in a similar way x_2 in φ_3.

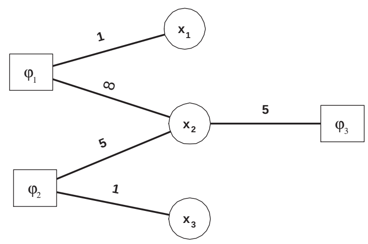

Figure 4: Bipartite weighted graph

The Hungarian method assigns x_1 to φ_1, x_3 to φ_2 and x_2 to φ_3, that is, $1 = Assig(1)$, $3 = Assig(2)$ and $2 = Assig(3)$. The partial order relation defined in accordance with (14) follows:

$$x_2 < x_1 \text{ because } x_1, x_2 \in var(\varphi_1) \ \wedge \ 1 = Assig(1)$$
$$x_2 < x_3 \text{ because } x_2, x_3 \in var(\varphi_2) \ \wedge \ 3 = Assig(2)$$

The obtained directed graph appears in figure 5.

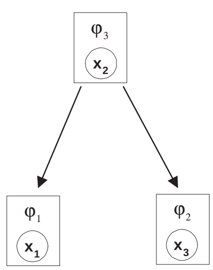

Figure 5: Dependencies graph

Applying the topological sorting algorithm to this graph, the order to calculate the expressions follows:

$$\begin{aligned} x_2 &= f_3(u_3) \\ x_1 &= f_1(x_2, u_1) \\ x_3 &= f_2(x_2, u_2) \end{aligned}$$

Restrictions and problems

Some problems may appear when a set of equalities is ordered.

Related to the assignment problem

- *Multiples results as solution.* Any solution may be choosen. However, it is always appropriate to choose those solution with less loops. A loop is below defined.
- *Assignment problem without solution.* This problem has not been studied yet.

Related to the topological sorting

- *A graph with strong components.* This problem appears when there are loops in the directed graph. Next example explains this problem. Let be the set of equalities:

$$\begin{cases} \varphi_1 \equiv x_1 = f_1(x_2, x, u), \\ \varphi_2 \equiv x_2 = f_2(x_1, x, u) \end{cases} \tag{15}$$

where to calculate x_1 it is necessary to calculate x_2, and in a similar way, to calculate x_2 it is necessary that x_1 has been previously calculated. We suppose that the other variables x have been already calculated. Therefore these two constraints are a loop.

The proposed solution to this problem is to group the loop, and it is computed like a unique constraint. These constraints have been denoted by h_i in the expression (12). However, this grouping will be possible when the Jacobian of the system with all the constraints of the loop will be not equal to 0.

In the previous example, the condition to verify is given by

$$det|D_{x_1,x_2}(f_1, f_2)| \neq 0 \quad \forall x_1, x_2 \tag{16}$$

The values from x_1, x_2 are calculated by means of a fixed point algorithm. The convergence condition to find a solution is that the Eigenvalues are less than 1 (Ortega00b). This way to calculate the values from x_1, x_2 will be improved using other techniques like Newton or *regula falsi*. Therefore, the transformation of the constraints (15) is as follows:

$$y_1 = h_1(u, x) \equiv \langle x_1, x_2 \rangle = h_1(u, x)$$

being

$$h_1(u, x) \equiv \begin{cases} x_1 = f_1(x_2, x, u), \\ x_2 = f_2(x_1, x, u) \end{cases}$$

Ordering constraints

Let S_Q be the obtained model with the qualitative knowledge transformation. It is defined by means of:

$$S_Q \equiv \begin{cases} \psi(\dot{x}, x, y, p, t), \\ \Psi_H, \\ x(t_0) = x_0, p \in I_p, x_0 \in I_0 \end{cases} \tag{17}$$

In order to carry out this new transformation, it is necessary to divide the constraints ψ into two groups:

equalities and inequalities[2]:

$$\psi(\dot{x}, x, y, p, t) \Rightarrow \begin{cases} \psi_1(\dot{x}, x, y, p, t) = 0, \\ \psi_2(\dot{x}, x, y, p, t) > 0 \end{cases} \quad (18)$$

where ψ_1 contains the equality constraints and ψ_2 the inequality constraints.

Substituting in (18) the variables $\dot{x}$ by z, it is obtained:

$$\begin{cases} \psi_1(z, y, x, p, t) = 0, \\ \dot{x} = z, \\ \psi_2(z, y, x, p, t) > 0 \end{cases} \quad (19)$$

The ordering techniques described in section are applied to ψ_1 in (19), choosing z, y as the set of variables to solve and x, p, t as the remaining variables:

$$\begin{cases} \langle z, y \rangle = (f_1(x, p, t) = 0), \\ \dot{x} = z, \\ \psi_2(z, y, x, p, t) > 0 \end{cases} \quad (20)$$

A new variable s is introduced in (20) for the inequality constraints as follows:

$$\begin{cases} \langle z, y \rangle = (f_1(x, p, t) = 0), \\ \dot{x} = z, \\ s = (\psi_2(z, y, x, p, t) > 0) \end{cases} \quad (21)$$

Taking into account these steps in (17), it is obtained:

$$\begin{cases} \langle z, y \rangle = (f_1(x, p, t) = 0), \\ \dot{x} = z, \\ s = (\psi_2(z, y, x, p, t) > 0), \\ \Psi_H, \\ x(t_0) = x_0, p \in I_p, x_0 \in I_0 \end{cases} \quad (22)$$

where the Boolean variable s contains the evaluation of the inequality constraints. This value may be always true, because if it is false then the simulation has no sense because the evolution of the system does not verify the whole definition of the model.

This is the result of this second transformation process. This is a family of quantitative models with explicit constraints where the state variables may appear. Stochastic techniques are applied to choose every quantitative model of this family. Every model is quantitatively simulated.

Implementation details

Automatically, this family of models (22) may be transformed into a program written like it is required by the simulation environment $Vensim$[3]. It is a visual environment to study quantitative models. It permit us to model, to document, to simulate, to analyze and to optimize quantitative dynamic systems. In this paper, it has been used the *Vensim DSS32 Versión 3.0B* (Vensim97).

The $Vensim$ program for the family of quantitative models with explicit constraints (22) is as follows:

- The equations $\dot{x} = z$ are written in $Vensim$ code as follows:

```
X=INTEG(Z,X0)~~|
```

where $X0$ denotes the initial value of the state variable x. This value is given by the initial conditions of the system $x(t_0) = x_0$.

- The equations $\langle z, y \rangle = (f_1(x, p, t) = 0)$ whose pattern is $x_1 = f(x, u)$ are written as follows:

```
X1=F(X,U)~~|
```

- The equations $\langle z, y \rangle = (f_1(x, p, t) = 0)$ whose pattern is $h(x, y, u) = 0$ are written by means of

```
R=SIMULTANEOUS(H(X,U,Y),0.)~~|
```

This predefined $Vensim$ function $SIMULTANEOUS$ evaluates the function h using the fixed point algorithm for the loop which was previously mentioned. This function returns a value R which is not necessary.

- The equations $s = (\psi_2(z, y, x, p, t) > 0)$ are written as follows:

```
S=IF THEN ELSE(F(z,y,x,p,t)>0,1,0)~~|
```

This function evaluates $F(z, y, x, p, t) > 0$, and the variable S takes the value 1 when its result is $true$ and 0 when it is $false$. This value is used by the KDD techniques to study the evolution of the system, for example, to check the consistency of the system, or to determine those values of the parameters which produce that the system is inconsistent, etc.

- Let Ψ_H be the set of constraints of a qualitative continuous function, and let P be the number of points of this function. The generate code is as follows:

```
df:(c1-c2*P) ~~|
v[df]=SAMPLE(H,P) ~~|
h[df]=INITIAL(v[df])~~|
r = F(h[c1],P,E) ~~|
```

This code uses some C routines that we have developed. They are included into Vensim by means of dynamic link libraries. The function $SAMPLE$ obtains the values for the function H in its domain. The function F evaluates the expression E applied to H, that is $h(e)$.

- Let v be a variable or parameter belongs to an interval $v \in I_v = [a, b]$. The generated code is as follows:

```
V=INITIAL(a+RANDOM_0_1()*(b-a)) ~~|
```

Finally, it is also necessary to add the following code:

```
INITIAL TIME = 0 ~~|
FINAL TIME = 100 ~~|
TIME STEP = 0.1 ~~|
SAVEPER = 0.5 ~~|
```

These values control the simulation time. They may be defined depending on the problem.

Interconnected tanks model

Let be an interconnected two-tank system in the open air (Figure 6). It is a well-known model. There are

[2] It is supposed that all the inequalities are written like $\psi > 0$. Those written like $\psi < 0$ will be transformed into $-\psi > 0$.

[3] Vensim is a register mark by Ventana Systems, Inc.

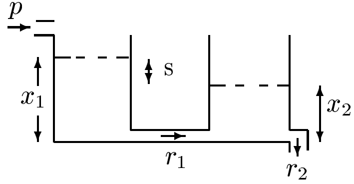

Figure 6: Interconnected tank system

imposed some constraints to this model. Let S be the model, being its constraints Φ given by:

$$\Phi \equiv \begin{cases} \dot{x_1} - p + r_1 = 0, \quad \dot{x_2} - r1 + r2 = 0, \\ r_1 = h_1(x_1 - x_2), \quad r_2 = g_1(x_2), \\ h_1 \equiv \{(-\infty, -\infty), +, (0,0), +, (+\infty, +\infty)\}, \\ g_1 \equiv \langle 0.1x, 5x, [0, 200] \rangle \\ x_1 < 70 - s, \quad x_2 < 70 - s \end{cases}$$

where h_1 is a qualitative continuous function, g_1 is an envelope function. The state variables are x_1, x_2, the parameters are p, s, and the rest of variables r_1, r_2 are intermediate variables.

It is a semiqualitative model with explicit constraints. It is semiqualitative because there is qualitative and quantitative knowledge, and it has explicit constraints, i.e. the heights x_1, x_2 of the tanks may be less than $70 - s$, where s is the diminution of the height due to the system is in the open air.

The initial conditions Φ_0 are defined as follows:

$$\Phi_0 \equiv \begin{cases} LP_x(x_1), \\ MP_x(x_2), \\ LP_x(s), \\ MP_p(p), \end{cases}$$

Initially, the height x_1 of the first tank and the diminution of height s are *slightly positive*, the height x_2 and the in-flow p are *moderately positive*. This qualitative unary operators has been defined by experts by means of:

$$\langle x, 10, 20, 200 \rangle, \quad \langle p, 1, 60, 1000 \rangle$$

where x denotes height magnitude and p is the flow magnitude.

We would like to know:

1. if the height of the tanks is exceeded taking into account the diminution due to s.
2. if an equilibrium is always reached
3. To classify the database in accordance with the height x_1.

Applying the qualitative knowledge transformation to S is obtained S_Q given by:

$$S_Q \equiv \begin{cases} \dot{x_1} - p + r_1 = 0, \quad \dot{x_2} - r1 + r2 = 0, \\ r_1 = \Psi_{H1}(x_1 - x_2), \\ r_2 = G_1(x_2), \\ G_1(k) = \alpha\, 0.1\, k + (1 - \alpha)\, 5\, k, \\ x_2 \in [0, 200], \quad \alpha \in [0, 1], \\ x_1 < 70 - s, \quad x_2 < 70 - s, \\ x_{1_0} \in [0, 10], \quad x_{2_0} \in [10, 20], \\ s \in [0, 10], \quad p \in [1, 60] \end{cases} \tag{23}$$

Applying the ordering constraints transformation to S_Q is obtained F:

$$F \equiv \begin{cases} \dot{x_1} = p + r_1, \quad \dot{x_2} = r1 + r2, \\ r_1 = \Psi_{H1}(x_1 - x_2), \\ r_2 = G_1(x_2), \\ G_1(k) = \alpha\, 0.1\, k + (1 - \alpha)\, 5\, k, \\ x_2 \in [0, 200], \quad \alpha \in [0, 1], \\ x_1 < 70 - s, \quad x_2 < 70 - s, \\ x_{1_0} \in [0, 10], \quad x_{2_0} \in [10, 20], \\ s \in [0, 10], \quad p \in [1, 60] \end{cases} \tag{24}$$

being $\dot{x_1}, \dot{x_2}$ the variables to solve. This model will be transformed into a *Vensim* program as it is explained in section . The simulation of this family of quantitative models generates a trajectories database (Ortega00a), which it is used to obtain the conclusions.

1. if the height of the tanks is exceeded taking into account the diminution due to s.

The answer is *true*. Therefore there are some trajectories exceeding the height of the tanks.

2. if an equilibrium is always reached

In order to answer this question, it is necessary to study those trajectories that always verify the constraints. They are shown in figure 7. The answer to this question is *true*, therefore there are no limit cycles in the system behaviour.

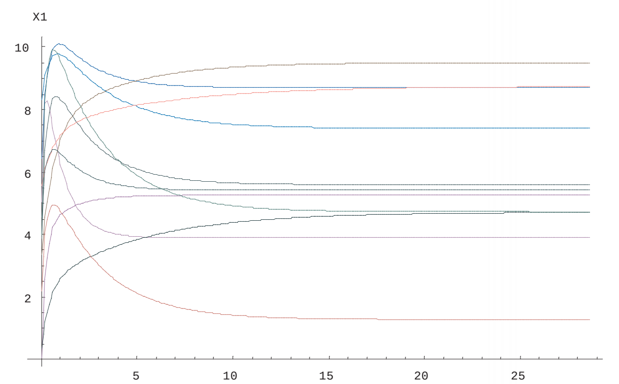

Figure 7: Evolution of X1

3. To classify the database in accordance with the height x_1.

The values of x_1 included in the trajectories database are shown in figure 8. In this figure, we have divided the x_1 axis into four different categories depending on the value of x_1 at the end of the simulation and the value of s. There is a region classified with two labels.

The whole methodology (Ortega99) (Ortega00a) with the improvements described in this paper has been applied to this system.

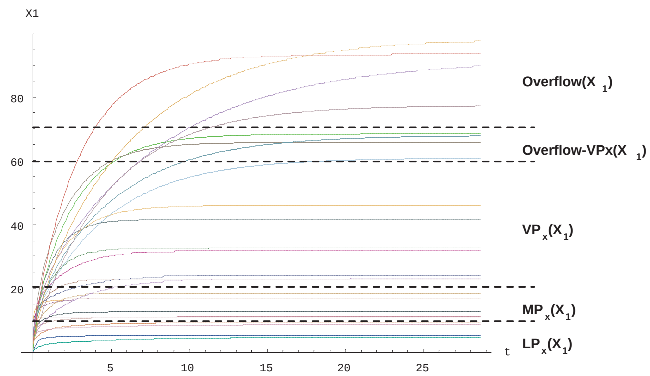

Figure 8: Classification according to x_1

Conclusions and further work

In this paper, we have enriched the methodology introduced in (Ortega99) (Ortega00a). This methodology was appropriate to simulate non-causal semiqualitative models with constraints. In this paper, the methodology is extended to manage family of quantitative models where may be explicit constraints. The idea of this improvement is based in a double-transformation process: *qualitative knowledge transformation* and *ordering constraints transformation*. This extended methodology is applied to an interconnected tank model with constraints.

It is also proposed an automatic way to compile the obtained family of quantitative models into an equivalent model like the commercial simulation environment *Vensim* requires.

In the future, we would like to study the possibility to improve the whole transformation process, modifying the algorithms that we are applying. Dynamic systems with constraints and with multiple scales of time are also one of our future points of interest.

We would like to apply the proposed approach to study a real computer-controlled process. It is a production industrial system. There is a metallurgical Company interested in modifying its steel control production system applying the whole methodology. The production engineers of this company wish to improve the steel quality, and, if it is possible, to reduce the production costs. This collaboration is now developing and in forthcoming papers, we will describe this system in detail and the obtained conclusions.

Acknowledgments

This work was partially supported by the Spanish Interministerial Committee of Science and Technology by means of the programs TIC98-1635-E and DPI2000-0666-C02-02.

We thank the QR reviewers for their constructive criticism and helpful comments.

References

Berleant D., and Kuipers B.J. 1997 Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence* 95: 215–255

Iwasaki Y. and Simon H.A. 1986 Causality in Device Behavior. *Artificil Intelligence* 29: 3-32.

Kahn A.B. "Topological sorting of large networks" Communications of the ACM Vol 5, pp. 558-562.

Kay H. 1996. Refining imprecise models and their behaviors. Ph.D. diss., Texas Univ.

Kuipers B.J. 1994 *Qualitative reasoning. Modeling and simulation with incomplete knowledge*, Cambridge, Massachussetts.: The MIT Press.

Kuhn H.W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2: 83-97.

Manber U. Introduction to algorithms. A creative approach Addison-Wesley Publishing Company Inc., Reading, Massachusetts.

Ortega J.A.; Gasca R.M.; and Toro M. 1998. Including qualitative knowledge in semiqualitative dynamical systems. *Lecture Notes in Artificial Intelligence* 1415: 329–338.

Ortega J.A.; Gasca R.M.; and Toro M. 1999. A semiqualitative methodology for reasoning about dynamic systems. In the 13^{th} International Workshop on Qualitative Reasoning. Loch Awe (Scotland), 169–177.

Ortega J.A.; Gasca R.M.; and Toro M. 1999. Searching for similar semiqualitative temporal patterns in time-series databases. In the 14^{th} International Workshop on Qualitative Reasoning. Morelia(Mexico), 111–122.

Ortega J.A. 2000 Patrones de comportamiento temporal en modelos semicualitativos con restricciones. Ph.D. diss., Dept. of Computer Science, Seville Univ.

Papadimitriou C.H., Steiglitz K. "Combinatorial Optimization. Algorithms and complexity" Prentice-Hall Press, Englewood Cliffs, New Jersey, 1982.

Sacks E.P. 1991. Automatic analysis of one-parameter planar ordinary differential equations by intelligent numeric simulation. *Artificial Intelligence* 48: 27-56.

Simons H.A. 1952. On the definition of causal relation. *The Journal of Philosophy* 49: 517-528.

Travé-Massuyès L.; Dague Ph.; and Guerrin F. 1997. *Le raisonement qualitativ pour les sciences de l'ingénieur*. Paris, France: Hermes Ed.

Vensim "The Ventana Simulation Environment. Vensim Tutorial" VENTANA SYSTEMS INC., 1997.