2004 IEEE International Symposium on
Computer Aided Control Systems Design
Taipei, Taiwan, September 2-4, 2004

Quantified Set Inversion with Applications to Control

Pau Herrero, Miguel A. Sainz, Josep Vehí and Luc Jaulin

***Abstract*— This paper describes a new reliable method, based on Modal Interval Analysis ($\mathcal{MIA}$) and Set Inversion ($\mathcal{SI}$) techniques, for the characterization of solution sets defined by Quantified Constraints Satisfaction Problems ($\mathcal{QCSP}$) over continuous domains. The presented methodology, called Quantified Set Inversion ($\mathcal{QSI}$), can be used over a wide range of engineering problems involving uncertain nonlinear models. Finally, an application on parameter identification is presented.**

I. INTRODUCTION

Many engineering problems, like in control engineering, can be formulated in a logical form by means of some kind of first order predicate formulas: formulas with the logical quantifiers, universal and existential, a set of real continuous functions, equalities or inequalities and variables ranging over real interval domains. More recently, this formulation has been referenced by different authors under the names: Generalized Constraints Satisfaction Problems [27] or Quantified Constraints Satisfaction Problems ($\mathcal{QCSP}$) [2], [24].

A. State-of-the-Art

Up to now, Cylindrical Algebraic Decomposition [29], [7], [12], for which a practical implementation exists [4], has been the most extended method to solve this type of problems. However, this technique is only well suited for small or middle-size problems because of its computational complexity. Moreover, it often generates huge output consisting on highly complicated algebraic expressions which are not useful for many applications and it does not provide partial information before computing the total result.

Methods that appear lately [11], [2] try to avoid some of these problems restricting oneself to approximate instead of exact solutions, using solvers based on numerical methods. However, these algorithms are also restricted to very special cases (e.g. quantified variables only occur once, only one quantifier,etc.). Recently, some of these deficiencies have been partially removed by Ratschan [24] but, a lot of work remains to be done before obtaining an efficient and general method.

Many practical examples exist on the resolution of $\mathcal{QCSP}$ using the different existing approaches, for example in control engineering [5], [18], [9], [25], [20], electrical engineering [28], mechanical engineering [14], [13], biology [6] and various others [3].

P. Herrero, M. A. Sainz and J. Vehí are with Institut d'Informàtica i Aplicacions, Universitat de Girona, Bldg. P4, Campus de Montilivi. E-17071 Girona, Catalonia, Spain. {pherrero@eia, vehi@eia, sainz@ima}.udg.es

L. Jaulin is with Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers, 62 avenue Notre Dame du Lac 49 000 Angers, France. jaulin@univ-angers.fr

II. PROBLEM STATEMENT

A Quantified Constraint ($\mathcal{QC}$) is an algebraic expression over the reals which contains quantifiers $(\exists, \forall)$, predicate symbols $(e.g., =, <, \leq)$, function symbols $(e.g. +, -, \times, sin, exp)$, rational constants and variables $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ ranging over reals domains $\boldsymbol{D} = \{D_1, \ldots, D_n\}$.

An example of a $\mathcal{QC}$ is the following one,

$$\forall x \in \mathbb{R}\; x^4 + px^2 + qx + r \geq 0, \tag{1}$$

where x is a universally $(\forall)$ quantified variable and p and r are free variable.

As defined in [27], a numerical constraint satisfaction problem, is a triple $\mathcal{CSP} = (\boldsymbol{x}, \boldsymbol{D}, \mathcal{C}(\boldsymbol{x}))$ defined by

(i) a set of numeric variables $\boldsymbol{x} = \{x_1, \ldots, x_n\}$,
(ii) a set of domains $\boldsymbol{D} = \{D_1, \ldots, D_n\}$ where D_i, a set of numeric values, is the domain associated with the variable x_i.
(iii) a set of constraints $\mathcal{C}(\boldsymbol{x}) = \{\mathcal{C}_1(\boldsymbol{x}), \ldots, \mathcal{C}_m(\boldsymbol{x})\}$ where a constraint $\mathcal{C}_i(\boldsymbol{x})$ is determined by any numeric relation (equation, inequality, inclusion, etc.) linking a set of variables under consideration.

A solution to a numeric constraint satisfaction problem $\mathcal{CSP} = (\boldsymbol{x}, \boldsymbol{D}, \mathcal{C}(\boldsymbol{x}))$ is an instantiation of the variables of $\boldsymbol{x}$ for which both inclusion in the associated domains and all the constraints of $\mathcal{C}(\boldsymbol{x})$ are satisfied. All the solutions of a constraint satisfaction problem thus constitute the set

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{D} \mid \mathcal{C}(\boldsymbol{x})\; is\; satisfied\}. \tag{2}$$

Now suppose that the constraints $\mathcal{C}(\boldsymbol{x})$ depend on some parameters $p_1, p_2, \ldots, p_l$ about which we only know that they belong to some intervals $P_1, P_2, \ldots, P_l$. Moreover, these parameters have an associated quantifier $Q \in \{\forall, \exists\}$. Taking into account the dual character of interval uncertainty, the most general definition of the set of solutions to such Quantified Constraint Satisfaction problem $\mathcal{QCSP}$ should have the form

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{D} \mid Q_1(p_{\sigma_1}, P_{\sigma_1}) \ldots Q_l(p_{\sigma_l}, P_{\sigma_l})\mathcal{C}(\boldsymbol{x})\}, \tag{3}$$

where

- Q_i are logical quantifiers $\forall$ or $\exists$ (in this paper, only the case of universal quantifiers preceding the existential ones will be dealt).

- $\{p_1, p_2, \ldots, p_l\}$ is the set of parameters of the constraints system considered,
- $\{P_1, P_2, \ldots, P_l\}$ is a set of intervals containing the possible values of these parameters,
- $\sigma_i \in \Sigma_l$ is a permutation of the numbers $1, \ldots, l$.

The sets of the form (3) will be referred to as quantified solutions sets to the numerical quantified constraints satisfaction problem $\mathcal{QCSP} = (\boldsymbol{x}, \boldsymbol{D}, \mathcal{C}(\boldsymbol{x}))$.

III. Methodology

A. Set Inversion

One way of solving a $\mathcal{CSP}$ is through the characterization of its solution set by means of the Set Inversion ($\mathcal{SI}$) approach.

Let $\mathcal{CSP}$ be a constraint satisfaction problem $\mathcal{CSP} = (\boldsymbol{x}, \boldsymbol{D}, \mathcal{C}(\boldsymbol{x}))$. Set inversion aims at characterizing the set Σ of all $\boldsymbol{x}$ such that $\mathcal{C}$ is satisfied.

Remark: All constraints are considered under the form $\mathcal{C}(\boldsymbol{x}) := \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}$, where $\boldsymbol{f}$ a continuous function from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Given a box $\boldsymbol{X}$ (cartesian product of intervals), an algorithm which does set inversion is based on a branch-and-bound technique and the 3 followings set of rules:

$$Rule\ 1: \ \forall(\boldsymbol{x}, \boldsymbol{X})\mathcal{C}(\boldsymbol{x}) \ \Leftrightarrow \ \boldsymbol{X} \subseteq \Sigma.$$

This logic formula, used to prove that a box $\boldsymbol{X}$ is contained in the solution set, is equivalent to the following interval computation and interval inclusions

$$Out(f(\boldsymbol{X})) \subseteq \boldsymbol{Y},$$

where $\boldsymbol{f}(\boldsymbol{X})$ are the ranges of the function components over the interval vector $\boldsymbol{X}$ and $Out(\boldsymbol{f}(\boldsymbol{X}))$ are outer approximations of $\boldsymbol{f}(\boldsymbol{X})$.

$$Rule\ 2: \ \forall(\boldsymbol{x}, \boldsymbol{X})\neg\mathcal{C}(\boldsymbol{x}) \ \Leftrightarrow \ \boldsymbol{X} \subseteq \overline{\Sigma}.$$

This logic formula, used to prove that a box $\boldsymbol{X}$ does not belongs to the solution set, is easily proved by means of the following interval computation and interval inclusions

$$Out(f(\boldsymbol{X})) \subseteq \overline{\boldsymbol{Y}}.$$

Finally, if *Rule* 1 and *Rule* 2 are not accomplished the position of the box $\boldsymbol{X}$ is undefined.

$$Rule\ 3: \ Otherwise,\ \boldsymbol{X}\ is\ undefined.$$

Fig. 1 shows a two dimensional example of the three possible situations corresponding to the 3 rules.

Then the algorithm which does set inversion is as follows

TABLE I

$\mathcal{SI}$ ALGORITHM

Algorithm $\mathcal{SI}$(In: $\mathcal{C}, X_0, \epsilon$, Out: $\Sigma^-, \Delta\Sigma$)

1) **Initialization**: Stack=X_0;$\Sigma^- := \emptyset$;$\Delta\Sigma := \emptyset$
2) **Repeat**
3) Unstack $\boldsymbol{X}$;
4) **if** $With(\boldsymbol{X}) \leq \epsilon$, **then** $\Delta\Sigma := \Delta\Sigma \cup \boldsymbol{X}$,
5) **else if** *Rule* 1 is satisfied, **then** $\Sigma^- := \Sigma^- \cup \boldsymbol{X}$,
6) **else if** *Rule* 2 is satisfied, **then** $\boldsymbol{X}$ is non solution,
7) **else** Bisect $\boldsymbol{X}$ and stack resulting boxes;
8) **Until** Stack=$\emptyset$;

where

- ϵ: $\mathcal{SI}$ stops the bisecting procedure over $\boldsymbol{X}$ when this precision is reached,
- Σ^-: Subpaving (list of nonoverlapping boxes) representing an inner approximation of the solution set,
- $\Delta\Sigma$: Subpaving representing all the boxes for which nothing could be proved.

These subpavings provide the following bracketing of the solution set:

$$\Sigma^- \subset \Sigma \subset \Sigma^- \cup \Delta\Sigma$$

B. Quantified Set Inversion via Modal Interval Analysis

Classical $\mathcal{SI}$ is well suited characterizing solution sets of the form (2). The problem arises when the sets are of the form (3). Several authors have proposed solutions to this problem using classical interval analysis and constraint propagation approaches [16], [2], [24]. In this section, a new algorithm for the characterization of quantified solution sets based on Modal Interval Analysis ($\mathcal{MIA}$) [10] is presented. This algorithm will be referred to as Quantified Set Inversion ($\mathcal{QSI}$).

Let us consider the case when the constraints are under the form $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) \lesseqgtr 0$, with f a continuous function from $\mathbb{R}^n$ to $\mathbb{R}$.

The main difference between the classical $\mathcal{SI}$ Algorithm and the quantified one lies on the used set of rules. For the proposed algorithm the following rules will be used:

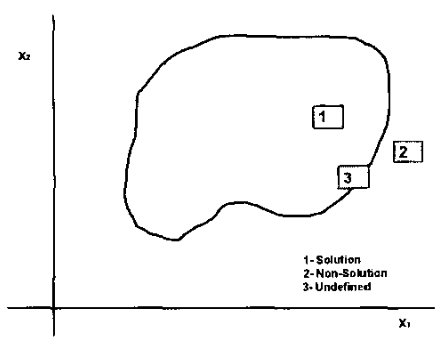

Fig. 1. Solution set

$$Rule\ 1: \quad \forall(\boldsymbol{x}, \boldsymbol{X})\forall(\boldsymbol{p_U}, \boldsymbol{P_U})\exists(\boldsymbol{p_E}, \boldsymbol{P_E})\, \mathcal{C}(\boldsymbol{x}) \Leftrightarrow \boldsymbol{X} \subseteq \Sigma.$$

This logic formula, used to prove that a box $\boldsymbol{X}$ belongs to the solution set, can not be easily proved by means of classical interval computations. For this reason, $\mathcal{MIA}$ is proposed. $\mathcal{MIA}$ is a powerful mathematical tool which allows the evaluation of quantified interval formulas by means of interval computations. Concretely, to evaluate the set of logic formulas, the *-semantic theorem given by $\mathcal{MIA}$ is used to reduce equivalently the logical formula to the interval inclusion

$$Out(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E})) \subseteq Z,$$

where $\boldsymbol{X}, \boldsymbol{P_U}$ are proper intervals, $\boldsymbol{P_E}$ improper one, $Out(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E}))$ is an outer approximation of the the *-semantic extension of the continuous function f and $Z = [0,0]$, $Z = [-\infty, 0]$ or $Z = [0, \infty]$ depending on if the constraints are under the form $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) = 0$, $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) < 0$ or $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) > 0$, respectively.

Remark: A modal interval X is defined as a couple $X = (X', \forall)$ or $X = (X', \exists)$ where X' is its classic interval *domain*, $X' \in I(\mathbb{R})$, and the quantifiers $\forall$ and $\exists$ are a selection *modality*. The modal intervals of the type $X = (X', E)$ are called *proper intervals* or *existential intervals*, the intervals of the type $X = (X', \forall)$ are called *improper intervals* or *universal intervals*. A modal interval can be represented using their canonical coordinates in the form

$$X = [a, b] = \begin{cases} ([a, b]', \exists) & \text{if } a \leq b \\ ([b, a]', \forall) & \text{if } a \geq b \end{cases}$$

For example, the interval $[2, 5]$ is equal to $([2, 5], \exists)$ and the interval $[8, 4]$ is equal to $([4, 8], \forall)$.

In order to obtain the second rule, used to prove that a box $\boldsymbol{X}$ does not belongs to the solution set, the following implication is used:

$$Rule\ 2: \quad \neg(\forall(\boldsymbol{p_U}, \boldsymbol{P_U})\exists(\boldsymbol{p_E}, \boldsymbol{P_E})\exists(\boldsymbol{x}, \boldsymbol{X})\, \mathcal{C}(\boldsymbol{x})) \Rightarrow \boldsymbol{X} \subseteq \overline{\Sigma}.$$

This logical formula is, analogously, equivalent to the following interval exclusion:

$$Inn(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E})) \nsubseteq Z,$$

where $\boldsymbol{P_U}$ is a proper interval, $\boldsymbol{X}, \boldsymbol{P_E}$ improper ones, $Inn(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E}))$ is an inner approximation of the the *-semantic extension of the continuous function f. and $Z = [0,0]$, $Z = [-\infty, 0]$ or $Z = [0, \infty]$ depending on if the constraints are under the form $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) = 0$, $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) < 0$ or $\mathcal{C}(\boldsymbol{x}) := f(\boldsymbol{x}) > 0$, respectively.

Finally, if none of these rules are accomplished, the box $\boldsymbol{X}$ is undefined.

$$Rule\ 3: \ otherwise,\ \boldsymbol{X}\ is\ undefined.$$

Computing the semantic extension of a continuous function f by means of any of their interpretable rational extensions provokes an overestimation of the interval evaluation, due to the multi-occurrences of variable, when the rational computations is not optimal. An algorithm, based on results of $\mathcal{MIA}$ and branch-and-bound techniques which allows to efficiently compute an inner and an outer approximation of f^* has been recently built.

When the constraints are under the form $C(\boldsymbol{x}) := f(\boldsymbol{x}) \lesseqqgtr 0$, with f a continuous function from $\mathbb{R}^n$ to $\mathbb{R}^m$ and each variable existentially quantified appears in only a component function, the problem is reduced to m different problems, one for each component function.

IV. APPLICATION

Interval model based techniques, like robust control [26], [17], [30], [15] or robust fault detection [1], requires from a well knowledge of the process model to be treated. This section describes an application of the $\mathcal{QSI}$ algorithm which consists on identifying on a guaranteed way [31] the parameter of a nonlinear model.

A. Parameter Identification

The problem treated in this section is a well known problem of the literature. It has been taken from [16], which at the same time has been inspired from [23].

B. Problem Statement

The present problem of parameter identification is defined by two main characteristics:

i. **The process model**: A nonlinear process which depends on the variable t and two parameters x_1 and x_2 is used. The theoretical model of the process is:

$$y(\boldsymbol{x}, t) = 20exp(-x_1 t) - 8exp(-x_2 t).$$

ii. **The constraints to be satisfied**: The constraints imposed by the system are:

$$y(\boldsymbol{x}, t_i) \in Y_i, t_i \in T_i, \forall i \in 1, \ldots, 10,$$

where Y_i corresponds to the uncertainty associated to the measure y_i and T_i represents the uncertainty associated to the measurement time t_i.

Table II shows the uncertainty associated to y and t.

Fig. 2 is a graphic representation of the uncertainty rectangles associated to the vectors t and y of the table II.

The accepted parameter set is defined by

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{X} \mid \exists t_1 \in T_1, \exists y_1 \in Y_1, y(\boldsymbol{x}, t_1) - y_1 = 0, \ldots, \exists t_{10} \in T_{10}, \exists y_{10} \in Y_{10}, y(\boldsymbol{x}, t_{10}) - y_{10} = 0\}.$$

TABLE II
UNCERTAINTY ASSOCIATED TO t AND y

i	T_i	Y_i
1	[-0.25,1.75]	[2.7,12.1]
2	[.5,2.5]	[1.04,7.14]
3	[1.25,3.25]	[-0.13,3.61]
4	[2,4]	[-0.95,1.15]
5	[5,7]	[-4.85,-0.29]
6	[8,10]	[-5.06,-0.36]
7	[12,14]	[-4.1,-0.04]
8	[16,18]	[-3.16,0.3]
9	[20,22]	[-2.5,0.51]
10	[24,26]	[-2,0.67]

Grouping the existential quantifiers and expressing it under a vectorial form

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{X} | \exists \mathbf{t} \in \mathbf{T}, \exists \mathbf{y} \in \mathbf{Y}, \mathbf{y}(\boldsymbol{x}, \mathbf{t}) - \mathbf{y} = 0\}.$$

For one sample i ($i = \{1, \dots, 10\}$), the logic formula which fulfils the points belonging to the solution set Σ_i is

$$\forall(x_1, X_1')\forall(x_2, X_2')\exists(t_i, T_i')\exists(y_i, Y_i')y_m(\boldsymbol{x}, t_i) - y_i = 0$$

which is equivalent to the following inclusion test

$$Out(f_i^*(X_1, X_2, T_i, Y_i)) \subseteq [0, 0],$$

with X_1 and X_2 proper intervals and T_i and Y_i improper ones.

The logic formula which fulfils the points not belonging to the solution set Σ_i is

$$\neg(\exists(x_1, X_1')\exists(x_2, X_2')\exists(t_i, T_i')\exists(y_i, Y_i')y_m(\boldsymbol{x}, t_i) - y_i = 0)$$

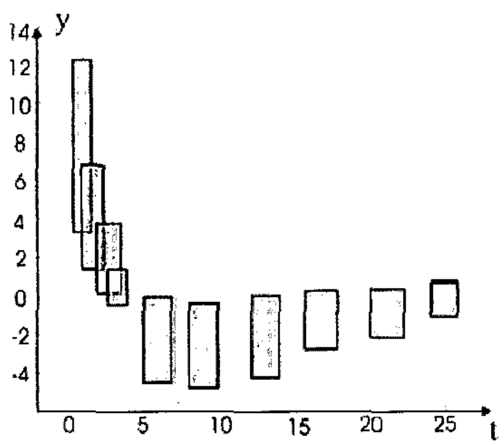

Fig. 2. Graphic representation of the uncertainty associated to the vectors t and y

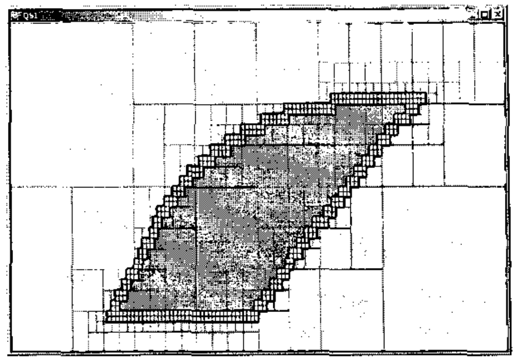

Fig. 3. Pavings generated by $\mathcal{QSI}$ algorithm

which is implied by the following exclusion test

$$Inn(f_i^*(X_1, X_2, T_i, Y_i)) \nsubseteq [0, 0],$$

with X_1, X_2, T_i and Y_i improper intervals.

Then,

$$\Sigma = \Sigma_i \cap \cdots \cap \Sigma_{10}.$$

C. Test Case

For $\boldsymbol{X} = [0, 1.2] \times [0, 0.5]$ and a precision of $\epsilon = 0.01$, $\mathcal{QSI}$ generates in 20 seconds on a Pentium III 1GHz, the paving of fig. 3.

where the darker region corresponds to the solution set Σ, the grey region corresponds to the non solution set $\overline{\Sigma}$ and the white region is undefined.

D. Analysis of the Results

Comparing the obtained results with the ones obtained by other existing algorithms [16], [19], for which an efficient implementation [22] exists for the second one, it can be said that any relevant difference can be observed in terms of the solution and computational performances. However, the method proposed in [19] should be better in terms of computational complexity for a higher order problem (e.g. more parameter to identify) due to the use of constraint propagation techniques [8], [21].

The main difference between the presented algorithm and the mentioned ones does not lie on the computational complexity but on the conceptual complexity. While in the $\mathcal{QSI}$ algorithm the set rules used to prove if a box $\boldsymbol{X}$ is inside or not from the solution set are achieved by means of *simple interval computations provided by* $\mathcal{MIA}$, the other algorithms needs from more complex strategies to carry on the same task.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The contribution of this paper has been to introduce a new algorithm, based on $\mathcal{MIA}$ and $\mathcal{SI}$ techniques, for the characterization of solution sets defined by numerical $\mathcal{QCSP}$. The applicability of the method to engineering problems has been shown by means of a well known problem of the literature on parameter identification. A comparison with other existing techniques has also been carried out concluding that the presented algorithm introduces more simplicity to the problem of characterizing the set defined by a $\mathcal{QCSP}$.

B. Future Works

1) Reducing the complexity via Constraint Propagation: In order to reduce the non polynomial complexity of the $\mathcal{SI}$ algorithm due to the branching, a narrowing operator (a contractor) for quantified constraints will be provided. This contractor, based on constraint propagation techniques and $\mathcal{MIA}$, allows the contraction of an initial box $\boldsymbol{X}$ containing the solution set Σ to another one $\boldsymbol{X}'$ such that $\boldsymbol{X}'$ still contains Σ.

REFERENCES

[1] Joaquim Armengol, Josep Vehí, Miguel Á. Sainz, and Pau Herrero. Fault detection in a pilot plant using interval models and multiple sliding time windows. In *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2003. Washington, D.C., U.S.A.*, page Accepted, 2003.

[2] F. Benhamou and F. Goulard. Universally quantified interval constraints. *Sixth International Conference on Principles of Constraint Programming*, 2000.

[3] Languénou E. Christie M. Benhamou F., Goualard F. Universally quantified interval constraint solving: an application to camera control. *ACM transactions on Computational Logic*, 2003.

[4] Christopher W Brown. *Quantifier Elimination by Partial Cylindrical Algebraic Decomposition*. http://www.cs.usna.edu/~qepcad/B/QEPCAD.html.

[5] W. Yang R. Liska C. Abdallah, P. Dorato and S. Steinberg. Applications of quantifier elimination theory to control system design. *In 4th IEEE Mediterranean Symposium on Control and Automation, Crete, Greece, 1996.*, 1996.

[6] M. Müller C. Chauvin and A. Weber. An application of quantifier elimination to mathematical biology. *In Computer Algebra in Science and Engineering.World Scientific*, pages 287–296, 1994.

[7] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *2nd GI Conf. Automata Theory and Formal Languages*, volume 33, pages 134–189. Springer, 1975.

[8] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32(3):281–331, 1987.

[9] P. Dorato. Quantified multivariate polynomial inequalities. *IEEE Control Systems Magazine*, 20(5):48–58, 2000.

[10] Ernest Gardenyes, Miguel Á. Sainz, Lambert Jorba, Remei Calm, Rosa Estela, Honorino Mielgo, and Albert Trepat. Modal intervals. *Reliable Computing*, 7(2):77–111, April 2001.

[11] Jürgen Garloff and Birgit Graf. Solving strict polynomial inequalities by bernstein expansion. *The Use of Symbolic Methods in Control System Analysis and Design*, pages 339–352, 1999.

[12] H. Hong. Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination. *In Proc. ISSAC'92, International Symposium on Symbolic and Algebraic Computation*, pages 177–188, 1992.

[13] N. I. Ioakimidis. Quantifier elimination in applied mechanics problems with cylindrical algebraic decomposition. *International Journal of Solids and Structures*, 34(30):4037–4070, 1997.

[14] N. I. Ioakimidis. Redlog-aided derivation of feasibility conditions in applied mechanics and engineering problems under simple inequality constraints. *Journal of Mechanical Engineering (Strojnícky Casopis)*, 50(1):58–69, 1999.

[15] J. Rodelllar J. Armengol J. Vehí, N. Luo. Digital control via interval analysis. *Nonlinear Analysis: Theory, Methods and Applications*, 47:203 – 212, 2001.

[16] L. Jaulin and E. Walter. Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors. *Automatica*, 35(5):849–856, 1999.

[17] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer, London, 2001.

[18] M. Jirstrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, 1997.

[19] I. Braems L. Jaulin, M. Kieffer and E. Walter. Guaranteed non-linear estimation using constraint propagation on sets. *BIT*, International Journal of Control:To appear, 2001.

[20] S. Ratschan L. Jaulin and L. Hardouin. Set computation for nonlinear control. *Reliable Computing*, 10:1–26, 2004.

[21] O. Lhomme. Consistency techniques for numerical csps. *In International Joint Conference on Artificial Intelligence (IJCAI)*, 1:232–238, 1993.

[22] X. Baguenard M. Dao and L. Jaulin. Projection d'ensembles pour l'estimation de paramètres, la conception de robot et la commande robuste. *Journées Doctorale d'Automatique - Valenciennes*, 2003.

[23] M. Milanese and A. Vicino. Estimation theory for nonlinear models and set membership uncertainty. *Automatica*, 27(2):403–408, 1991.

[24] Stefan Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *Submitted for publication*, 2003.

[25] Stefan Ratschan and Josep Vehí. Robust pole clustering of parametric uncertain systems using interval methods. In *Proc. of 4th IFAC Symposium on Robust Control Design*, 2003.

[26] M. Milanese S. Malan and M. Taragna. Robust analysis and design of control systems using interval arithmetic. *Automatica*, 33(7):1363–1372, 1997.

[27] Sergey P. Shary. A new technique in systems analysis under interval uncertainty and ambiguity. *Reliable Computing*, 8:321–418, 2002.

[28] T. Sturm. Reasoning over networks by symbolic methods. *Applicable Algebra in Engineering Communication and Computing*, 10(1):79–96, 2000.

[29] A. Tarski. A decision method for elementary algebra and geometry. *Univ. of California Press, Berkeley*, 1951.

[30] J. Vehí, J. Rodellar, M.Á. Sainz, and J. Armengol. Necessary and sufficient conditions for robust stability using modal intervals. *42nd IEEE Midwest Symposium on Circuits and Systems*, 2:673 – 676, 1999.

[31] Éric Walter and Luc Pronzato. *Identification of parametric models from experimental data*. Communications and control engineering series. Springer, Paris, 1997.