# Quantified Set Inversion Algorithm

*with Applications to Control*

Pau Herrero Viñas (pherrero@eia.udg.es), Miguel Ángel Sainz Sánchez (sainz@ima.udg.es) and Josep Vehí Caselles (vehi@eia.udg.es)
*Institut d'Informàtica i Aplicacions, Universitat de Girona, Campus Montilivi, Politècnica IV, 17071 Girona, Catalonia, Spain*

Luc Jaulin
*ENSIETA (Ecole Nationale Supérieure des Ingénieurs des Etudes et Techniques d'Armement), Laboratoire E3I2 (Extraction et Exploitation de l'Information en Environnements Incertains), 2 rue François Verny, 29806 Brest Cédex 09, France*

**Abstract.** In this paper, a new algorithm based on Set Inversion techniques and Modal Interval Analysis is presented. This algorithm allows solving problems involving quantified constraints over the reals through the characterization of their solution sets. The presented methodology can be applied over a wide range of problems involving uncertain (non)linear systems. Finally, an advanced application is solved.

**Keywords:** Constraints Satisfaction Problem, Set Inversion, Modal Interval Analysis

## 1. Introduction

Many physical problems can be stated in a logical form by means of some kind of quantified constraints: 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 (Shary, 2002) or Quantified Constraints Satisfaction Problems ($\mathcal{QCSP}$) (Benhamou and Goulard, 2000; Ratschan, 2003).

Cylindrical Algebraic Decomposition (Collins, 1975; Hong, 1992), for which a practical implementation exists (Brown, ), 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 (Garloff and Graf, 1999; Benhamou and Goulard, 2000) 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 (Ratschan, 2003) but, a lot of work remains to be done before obtaining an efficient general method.

Many practical examples exist on the resolution of $\boldsymbol{\mathcal{QCSP}}$ using the different existing approaches, for example in control engineering (Abdallah et al., 1999; Jirstrand., 1997; Dorato., 2000; Ratschan and Vehí, 2004; Jaulin et al., 2004), electrical engineering (Sturm, 2000), mechanical engineering (Ioakimidis, 1999), biology (Chauvin et al., 1994) and many others (Benhamou et al., 2004).

## 2. 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 ($e.g. +, -, \times, sin, exp$), 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 $\boldsymbol{\mathcal{QC}}$ 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 $p$, $q$ and $r$ are free variables.

As defined in (Shary, 2002), a numerical constraint satisfaction problem, is a triple $\boldsymbol{\mathcal{CSP}} = (\boldsymbol{x}, \boldsymbol{D}, \boldsymbol{\mathcal{C}}(\boldsymbol{x}))$ 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 $\boldsymbol{\mathcal{C}}(\boldsymbol{x}) = \{\mathcal{C}_1(\boldsymbol{x}), \ldots, \mathcal{C}_m(\boldsymbol{x})\}$ 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 is an instantiation of the variables of $\boldsymbol{x}$ for which both inclusion in the associated domains and all the constraints of $\boldsymbol{\mathcal{C}}(\boldsymbol{x})$ are satisfied. All the solutions of a constraint satisfaction problem thus constitute the set

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

Let us suppose that the constraints $\boldsymbol{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{p})$ 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 ($\boldsymbol{\mathcal{QCSP}}$) should have the form

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

where

- each $Q_i$ is logical quantifier $\forall$ or $\exists$,
- $\boldsymbol{p} = \{p_1, p_2, \ldots, p_l\}$ is the set of parameters of the constraints system considered,
- $\boldsymbol{P} = \{P_1, P_2, \ldots, P_l\}$ is a set of intervals containing the possible values of $\boldsymbol{p}$,
- $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_l)$ 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 quantified constraints satisfaction problem.

*Remark*: In this paper, only the case of universal quantifiers preceding the existential ones will be dealt. The solution set corresponding to this particular case will be referred to as $UE$-Solution set ($\Sigma_{UE}$) ($AE$-Solution set by other authors (Shary, 2002)).

## 3. Quantified Set Inversion

Set Inversion ($\boldsymbol{\mathcal{SI}}$) (Jaulin and Walter, 1993), a well known paradigm of interval analysis, is well suited approximating solution sets of the form (2) by means of subpavings (sets of nonoverlapping boxes). The problem arises when these sets are of the form (3), because classical Set Inversion is not able to solve this type of problems in a direct way. The problem of characterizing the sets of the form (3) will be referred to as Quantified Set Inversion ($\boldsymbol{\mathcal{QSI}}$).

In this section, a new algorithm based on Modal Interval Analysis ($\mathcal{MIA}$) (Gardenyes et al., 2001) and inspired on the classical Set Inversion algorithm is presented. This algorithm, which will be named Quantified Set Inversion via Modal Interval Analysis ($\mathcal{QSIMIA}$), allows the characterization of UE-solution sets.

### 3.1. Quantified Set Inversion via Modal Interval Analysis

Let $\mathcal{QCSP} = (\boldsymbol{x}, \boldsymbol{D}, \mathcal{C}(\boldsymbol{x}, \boldsymbol{p}))$ be a quantified constraint satisfaction problem. Let us characterize the set $\Sigma_{UE}$ of all $\boldsymbol{x}$ such that $\mathcal{C}(\boldsymbol{x}, \boldsymbol{p})$ is satisfied.

Let us consider the case when the constraints are under the form $\mathcal{C}(\boldsymbol{x}, \boldsymbol{p}) := f(\boldsymbol{x}, \boldsymbol{p}) \lesseqqgtr 0$, with $f$ a continuous function from $\mathbb{R}^n$ to $\mathbb{R}$ and the UE-Solution Set defined by

$$\Sigma_{UE} = \{\boldsymbol{x} \in \mathbb{R} \mid \forall(\boldsymbol{p_U}, \boldsymbol{P_U}')\exists(\boldsymbol{p_E}, \boldsymbol{P_E}')f(\boldsymbol{x}, \boldsymbol{p}) \lesseqqgtr 0\}. \tag{4}$$

*Remark*: $\forall(\boldsymbol{x}, \boldsymbol{X}')$ and $\exists(\boldsymbol{x}, \boldsymbol{X}')$ is another denotation for $(\forall \boldsymbol{x} \in \boldsymbol{X}')$ and $(\exists \boldsymbol{x} \in \boldsymbol{X}')$, respectively, in $\mathcal{MIA}$.

Given a box $\boldsymbol{X}$ (Cartesian product of intervals), an algorithm which does quantified set inversion is based on a branch-and-bound technique and the three following sets of bounding rules:

$$\begin{aligned} \textit{Rule } 1: \quad & \forall(\boldsymbol{x}, \boldsymbol{X}')\forall(\boldsymbol{p_U}, \boldsymbol{P_U}')\exists(\boldsymbol{p_E}, \boldsymbol{P_E}')\ \mathcal{C}(\boldsymbol{x}, \boldsymbol{p}) \Leftrightarrow \\ & \boldsymbol{X} \subseteq \Sigma. \end{aligned}$$

This quantified constraint, used to prove that a box $\boldsymbol{X}$ is contained in the solution set, can not be easily proved by means of classical interval computations. For this reason, $\mathcal{MIA}$ techniques are proposed. $\mathcal{MIA}$ is a powerful mathematical tool which allows the evaluation of quantified constraints over the reals by means of interval computations. Concretely, the *-semantic theorem of $\mathcal{MIA}$ is used. The quantified constraint, corresponding to *Rule* 1, can be checked through the following reasoning

$$\begin{gathered} Out(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E})) \subseteq Z \Rightarrow f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E}) \subseteq Z \\ \Leftrightarrow \forall(\boldsymbol{x}, \boldsymbol{X}')\forall(\boldsymbol{p_U}, \boldsymbol{P_U}')\exists(\boldsymbol{p_E}, \boldsymbol{P_E}')\ f(\boldsymbol{x}, \boldsymbol{p}) \lesseqqgtr 0 \\ \Leftrightarrow \boldsymbol{X} \subseteq \Sigma, \end{gathered}$$

where $\boldsymbol{X}, \boldsymbol{P_U}$ are proper intervals, $\boldsymbol{P_E}$ is an 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 $Z = (0, \infty)$, for

$f(\boldsymbol{x}, \boldsymbol{p}) = 0$, $f(\boldsymbol{x}, \boldsymbol{p}) < 0$ or $f(\boldsymbol{x}, \boldsymbol{p}) > 0$, 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 $([4, 8], \forall)$.*

*The inclusion operator $\subseteq$ defined by $\boldsymbol{\mathcal{MIA}}$ has a different interpretation respect the set based definition used by classical interval analysis. However, using the canonical coordinates $X = [x_1, x_2]$ and $Y = [y_1, y_2]$, the inclusion maintains the traditional* modus operandi*; that is to say,*

$$[x_1, x_2] \subseteq [y_1, y_2] \Leftrightarrow (x_1 \geq y_1, x_2 \leq y_2)\,.$$

*For example, the interval $[1, -1]$ is included inside the interval $[0, 0]$.*

In order to prove the second rule, used to verify that a box $\boldsymbol{X}$ has no intersection with the solution set, the following implication is used:

$$\begin{aligned} \mathit{Rule}\ 2: \ & \neg(\forall(\boldsymbol{p_U}, \boldsymbol{P_U}')\exists(\boldsymbol{p_E}, \boldsymbol{P_E}')\exists(\boldsymbol{x}, \boldsymbol{X}')\ \boldsymbol{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{p})) \Rightarrow \\ & \boldsymbol{X} \subseteq \overline{\Sigma}. \end{aligned}$$

where $\overline{\Sigma}$ is the complementary set of $\Sigma$ defined by

$$\overline{\Sigma} = \{\boldsymbol{x} \in \mathbb{R} \mid \exists(\boldsymbol{p_U}, \boldsymbol{P_U}')\forall(\boldsymbol{p_E}, \boldsymbol{P_E}')\neg(f(\boldsymbol{x}, \boldsymbol{p}) \lesseqgtr 0)\}. \tag{5}$$

This quantified constraint is, analogously, implied by the following interval exclusion:

$$\begin{aligned} &\mathit{Inn}(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E})) \not\subseteq Z \Rightarrow f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E}) \not\subseteq Z \\ &\Leftrightarrow \neg(\forall(\boldsymbol{p_U}, \boldsymbol{P_U}')\exists(\boldsymbol{p_E}, \boldsymbol{P_E}')\exists(\boldsymbol{x}, \boldsymbol{X}')\ f(\boldsymbol{x}, \boldsymbol{p}) \lesseqgtr 0) \\ &\Leftrightarrow \exists(\boldsymbol{p_U}, \boldsymbol{P_U}')\forall(\boldsymbol{p_E}, \boldsymbol{P_E}')\forall(\boldsymbol{x}, \boldsymbol{X}')\neg(f(\boldsymbol{x}, \boldsymbol{p}) \lesseqgtr 0) \\ &\Rightarrow \forall(\boldsymbol{x}, \boldsymbol{X}')\exists(\boldsymbol{p_U}, \boldsymbol{P_U}')\forall(\boldsymbol{p_E}, \boldsymbol{P_E}')\neg(f(\boldsymbol{x}, \boldsymbol{p}) \lesseqgtr 0) \\ &\Leftrightarrow \boldsymbol{X} \subseteq \overline{\Sigma}, \end{aligned}$$

with $\boldsymbol{P_U}$ a proper interval, $\boldsymbol{X}, \boldsymbol{P_E}$ improper ones, $Inn(f^*(\boldsymbol{X}, \boldsymbol{P_U}, \boldsymbol{P_E}))$ an inner approximation of the the *-semantic extension of the continuous function $f$ and $Z = [0,0]$, $Z = (-\infty, 0)$ or $Z = (0, \infty)$, for $f(\boldsymbol{x}, \boldsymbol{p}) = 0$, $f(\boldsymbol{x}, \boldsymbol{p}) < 0$ or $f(\boldsymbol{x}, \boldsymbol{p}) > 0$, respectively.

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

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

*Remark:* When the constraints are under the form $C(\boldsymbol{x}) := \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) \overset{\leq}{\underset{\geq}{=}} 0$, with $\boldsymbol{f}$ a continuous function from $\mathbb{R}^n$ to $\mathbb{R}^m$ and each existentially quantified variable appears in only one function component, the problem is reduced to $m$ different problems, one for each function component. Then, the solution set may be obtained as

$$\Sigma = \Sigma_1 \cap \ldots \cap \Sigma_m.$$

Table I shows the algorithm which does Quantified Set Inversion and figure 3.1 shows a two dimensional example of the three possible situations corresponding to the three rules.

Table I. Quantified Set Inversion Algorithm

| **Algorithm $\boldsymbol{\mathcal{QSI}}$**(In: $\boldsymbol{\mathcal{C}}, \boldsymbol{X}_0, \epsilon$, Out: $\Sigma^-, \triangle\Sigma$) |
|---|
| 1. **Initialization**: Stack=$\boldsymbol{X}_0$;$\Sigma^- := \emptyset$;$\triangle\Sigma := \emptyset$ |
| 2. **Repeat** |
| 3. Unstack $\boldsymbol{X}$; |
| 4. **if** $Width(\boldsymbol{X}) \leq \epsilon$, **then** $\triangle\Sigma := \triangle\Sigma \cup \boldsymbol{X}$, |
| 5. **else if** *Rule* 1 is satisfied, **then** $\Sigma^- := \Sigma^- \cup \boldsymbol{X}$, |
| 6. **else if** *Rule* 2 is satisfied, **then** $\boldsymbol{X}$ has no solutions, |
| 7. **else** Bisect $\boldsymbol{X}$ and stack resulting boxes; |
| 8. **Until** Stack=$\emptyset$; |

where

- $\epsilon$: $\boldsymbol{\mathcal{QSI}}$ stops the bisecting procedure over $\boldsymbol{X}$ when this precision is reached,
- $\Sigma^-$: Subpaving (list of nonoverlapping boxes) representing an inner approximation of the solution set,

– $\triangle\Sigma$: Subpaving representing all the undefined boxes.

These subpavings provide the following bracketing of the solution set:

$$\Sigma^- \subseteq \Sigma \subseteq (\Sigma^- \cup \triangle\Sigma).$$

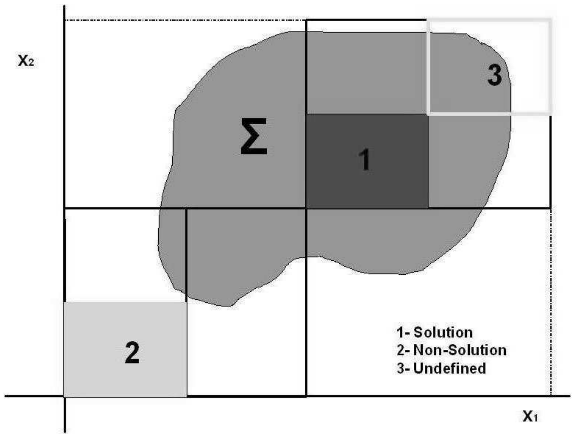


*Figure 1.* Two dimensional example of $\mathcal{QSI}$ algorithm.

### 3.2. $f^*$ Computation

The quantified constraints corresponding to Rules 1 and 2 have been reduced to interval inclusions of the *-semantic extension. However, computing the *-semantic extension of a continuous function $f$ by means of any of the interpretable rational extensions given by $\mathcal{MIA}$, provokes an overestimation of the interval evaluation, due to the possible multi-occurrences of some variable and when the not optimal rational computations. 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 built.

#### 3.2.1. *Twin Arithmetic*

In order to handle simultaneously inner and outer approximations for $f^*$ the set of *twins*, introduced by Gardenyes et al (Gardeñes et al., 1980), will be used.

From the lattice of modal intervals $(I^*(\mathbb{R}), \subseteq)$ a new lattice $(I^*(I^*(\mathbb{R})), \subseteq)$ can be build following the standard process. One element $\mathbb{A} \in (I^*(I^*(\mathbb{R}))$, named *twin*, is defined by

$$\mathbb{A} := |[\underline{A}, \overline{A}]|$$

where $\underline{A} \in I^*(\mathbb{R})$ is the *lower bound* and $\overline{A} \in I^*(\mathbb{R})$ is the *upper bound* of $\mathbb{A}$, and

$$I^*(I^*(\mathbb{R}) := \{\mathbb{A} = |[\underline{A}, \overline{A}]| \mid \underline{A}, \overline{A} \in I^*(\mathbb{R})\}$$

is the set of twins over $I^*(\mathbb{R})$. If $\underline{A} \subseteq \overline{A}$ the twin is called *proper twin*, which can be identified with the set

$$\mathbb{A} = \{X \in I^*(\mathbb{R}) \mid \underline{A} \subseteq X \subseteq \overline{A}\}$$

of which elements are the modal intervals between both bounds $\underline{A}$ and $\overline{A}$.

The inclusion between twins $\mathbb{A} = |[\underline{A}, \overline{A}]|$ and $\mathbb{B} = |[\underline{B}, \overline{B}]|$ is defined by means of the interval inclusion between their bounds

$$\mathbb{A} \subseteq \mathbb{B} \Leftrightarrow (\underline{A} \supseteq \underline{B}\ , \overline{A} \subseteq \overline{B}).$$

The lattice operations *meet* and *join* on $I^*(I^*(\mathbb{R}))$ for a bounded family of twins $\mathbb{A}(I) := \{\mathbb{A}(i) = [\underline{A}(i), \overline{A}(i)] \in I^*(I^*(\mathbb{R})) \mid i \in I\}$ ($I$ is the index's domain) are defined by

$$\bigwedge_{i\in I} \mathbb{A}(i) = [\bigwedge_{i\in I} \underline{A}(i), \bigwedge_{i\in I} \overline{A}(i)],$$
$$\bigvee_{i\in I} \mathbb{A}(i) = [\bigvee_{i\in I} \underline{A}(i), \bigvee_{i\in I} \overline{A}(i)],$$

denoted $\mathbb{A} \wedge \mathbb{B}$ and $\mathbb{A} \vee \mathbb{B}$ for the corresponding two-operands' case. These operators do not have the same set-theoretical meaning than in $I^*(\mathbb{R})$.

Figure 2 shows geometrical representations for a proper twin, the twin inclusion and the twin meet and join operators to illustrate these concepts.

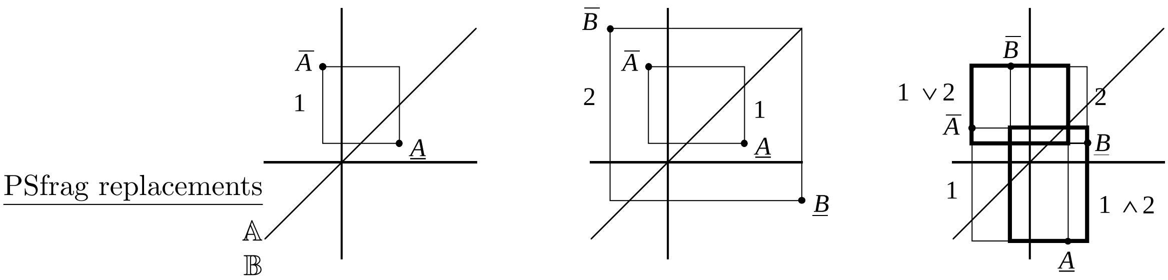


*Figure 2.* Twins, twin inclusion, twin meet and twin join for two proper twins

3.2.2. *Basic Algorithm*

Let $f$ be a $\mathbb{R}^n$ to $\mathbb{R}$ real continuous function and $\boldsymbol{X} = (\boldsymbol{U}, \boldsymbol{V})$ a modal interval vector split into their $\boldsymbol{U}$ proper and $\boldsymbol{V}$ improper components.

Let $\{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_r\}$ be a partition of $\boldsymbol{U}$ and, for every $j = 1, \ldots, r$, let $\{\boldsymbol{V}_{1_j}, \ldots, \boldsymbol{V}_{s_j}\}$ be a partition of $\boldsymbol{V}$. Each interval $\boldsymbol{U}_j \times \boldsymbol{V}_{k_j}$ is called *cell* and each $\boldsymbol{V}$-partition is called *strip*. Figure 3 shows a geometrical representation of an example of these partitions, when $\boldsymbol{X}$ has only one proper component and one improper component.

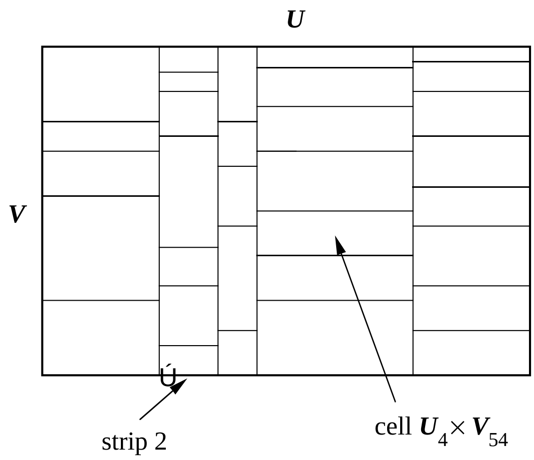


*Figure 3.* Partition, strips and cells

Taking into account the definition of the interval *-semantic extension of $f$ to $\boldsymbol{X}$, then

$$
\begin{aligned}
f^*(\boldsymbol{X}) &:= \bigvee_{\boldsymbol{u} \in \boldsymbol{U}'} \bigwedge_{\boldsymbol{v} \in \boldsymbol{V}'} [f(\boldsymbol{u}, \boldsymbol{v})] \\
&= \bigvee_{j \in \{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j \in \boldsymbol{U}'_j} \bigwedge_{\boldsymbol{v} \in \boldsymbol{V}'} [f(\boldsymbol{u}_j, \boldsymbol{v})] && (6) \\
&= \bigvee_{j \in \{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j \in \boldsymbol{U}'_j} \bigwedge_{k_j \in \{1_j,\ldots,s_j\}} \bigwedge_{\boldsymbol{v}_{k_j} \in \boldsymbol{V}'_{k_j}} [f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})] && (7) \\
&\in \bigvee_{j \in \{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j \in \boldsymbol{U}'_j} \bigwedge_{k_j \in \{1_j,\ldots,s_j\}} \bigwedge_{\boldsymbol{v}_{k_j} \in \boldsymbol{V}'_{k_j}} |[[f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})], [f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})]]| && (8) \\
&\subseteq \bigvee_{j \in \{1,\ldots,r\}} \bigwedge_{k_j \in \{1_j,\ldots,s_j\}} |[f^*(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j}), f^*(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j})]| && (9) \\
&\subseteq \bigvee_{j \in \{1,\ldots,r\}} \bigwedge_{k_j \in \{1_j,\ldots,s_j\}} |[Inn(fR(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j})), Out(fR(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j}))]|, && (10)
\end{aligned}
$$

where $\check{\boldsymbol{u}}_j$ is any fixed point of $\boldsymbol{U}'_j$ $(j = 1, \ldots, r)$ and $\check{\boldsymbol{v}}_{k_j}$ is any fixed point of $\boldsymbol{V}'_{k_j}$ $(k_j = 1_j, \ldots, s_j)$, for example the mid-points or the bounds of the intervals, and $fR$ is the modal rational interval extension of the function $f$ (Gardenyes et al., 2001), because

(6) is true in accordance with the associativity of the join operator,

(7) is true in accordance with the associativity of the meet operator.

(8) is true because the point-wise interval $[f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})]$ obviously belongs to the proper twin $|[[f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})], [f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})]]|$, with equal bounds.

(9) is true since $[f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})] = f^*(\boldsymbol{u}_j, \boldsymbol{v}_{k_j}) \supseteq f^*(\boldsymbol{u}_j, \boldsymbol{V}_{k_j})$ implies

$$\bigvee_{j\in\{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j\in\boldsymbol{U}'_j} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} \bigwedge_{\boldsymbol{v}_{k_j}\in\boldsymbol{V}'_{k_j}} [f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})] \supseteq$$
$$\supseteq \bigvee_{j\in\{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j\in\boldsymbol{U}'_j} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} f^*(\boldsymbol{u}_j, \boldsymbol{V}_{k_j})$$
$$\supseteq \bigvee_{j\in\{1,\ldots,r\}} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} f^*(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j})$$

and, similarly, $[f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})] = f^*(\boldsymbol{u}_j, \boldsymbol{v}_{k_j}) \subseteq f^*(\boldsymbol{U}_j, \boldsymbol{v}_{k_j})$ implies

$$\bigvee_{j\in\{1,\ldots,r\}} \bigvee_{\boldsymbol{u}_j\in\boldsymbol{U}'_j} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} \bigwedge_{\boldsymbol{v}_{k_j}\in\boldsymbol{V}'_{k_j}} [f(\boldsymbol{u}_j, \boldsymbol{v}_{k_j})] \subseteq$$
$$\subseteq \bigvee_{j\in\{1,\ldots,r\}} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} f^*(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j}).$$

(10) is true because

$$f^*(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j}) \supseteq Inn(fR(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j}))$$

and

$$f^*(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j}) \subseteq Out(fR(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j})).$$

The final relation (10) is equivalent to

$$\text{Inner approx.}: \bigvee_{j\in\{1,\ldots,r\}} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} Inn(fR(\check{\boldsymbol{u}}_j, \boldsymbol{V}_{k_j})) \subseteq f^*(\boldsymbol{X}) \quad (11)$$

$$\text{Outer approx.}: \bigvee_{j\in\{1,\ldots,r\}} \bigwedge_{k_j\in\{1_j,\ldots,s_j\}} Out(fR(\boldsymbol{U}_j, \check{\boldsymbol{v}}_{k_j})) \supseteq f^*(\boldsymbol{X}) \quad (12)$$

for any partition of $\boldsymbol{X}$. Moreover, the finer partition, the better approximations.

### 3.2.3. *Improving the $f^*$ Computation*

In order to reduce the run time of the branch-and-bound algorithm, a set of additional criteria, based on the study of the monotony of the function, the syntax tree of the function and theorems from $\mathcal{MIA}$, can be applied. The use of these criteria can drastically improve the computation effort by several orders of magnitude.

*Remark:* The results showed in this paper have been obtained using the improved version of the $f^*$ algorithm, which is not detailed in this paper.

3.2.4. *Example*
Let be the quantified constraint

$$\forall(x, [0,6]')\ \forall(z, [6,8]')\ \exists(y, [2,8]')\ (f(x,y,z) = 0)$$

where $f(x,y,z) := x^2 + y^2 + 2xy - 20x - 20y + 100 - z$.

To prove this quantified constraint, it is sufficient to verify the following interval inclusion

$$Out(f^*([0,6],[6,8],[8,2]) \subseteq [0,0].$$

In less than 0.05 seconds on a Pentium IV the $f^*$ algorithm obtains the following result: inner approximation $[1.01921024, -2.0000001]$ and outer approximation $[0.99040255, -1.99999999]$. As $Out(f^*([0,6],[6,8],[8,2]) \subseteq [0,0]$ fulfills, the quantified constraint is true.

## 4. Application to Control

### 4.1. Advanced Aircraft Application

An important question in advanced aircraft applications (Jirstrand., 1997) is to know what orientation $(\alpha, \beta)$ of an aircraft, with respect to the airflow, can be controlled by the admissible control-surface configurations $(u_1, u_2, u_3)$. See figure 4.

The aerodynamic moments acting over the aircraft $T_L$, $T_M$ and $T_N$, are nonlinear functions of $\alpha, \beta$, which are the angles of attack and sideslip respectively, and the control-surface deflections $(u_1, u_2, u_3)$, which are the aileron, elevator, and rudder deflections respectively. These moments are usually given in tabular form together with some interpolation method. In (Stevens and Lewis, 1993) these tables are listed for an F-16 aircraft and the following are scaled polynomial approximations of the corresponding functions

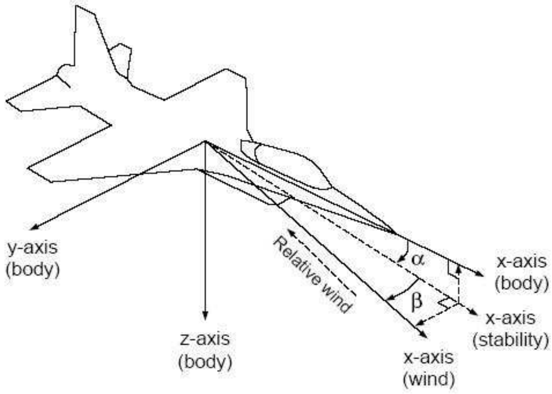


*Figure 4.* The orientation of an aircraft with respect to the airflow.

$$
\begin{aligned}
T_L(\alpha,\beta,u_1,u_3) = \quad & -q_1\beta - q_2\alpha\beta + q_3a^2\beta + q_4\beta^3 + u_1(-q_5 - q_6\alpha + \\
& q_7\alpha^2 - q_8\alpha^3 + q_9\beta^2 + q_{10}\alpha\beta^2) + u_3(q_{11} - q_{12}\alpha + \\
& q_{13}\alpha^2 - q_{14}\alpha^3 + q_{15}\alpha^4 - q_{16}\beta^2 - q_{17}\alpha\beta^2 + \\
& q_{18}\alpha^2\beta^2 + q_{19}\beta^4), \\
T_M(\alpha,u_2) = \quad & -q_{20} - q_{21}u_2 + u_2^2 + q_{22}u_2^3 + q_{23}\alpha - q_{24}u_2\alpha + \\
& q_{25}u_2^2\alpha - q_{26}\alpha^2 + q_{27}\alpha^2 + q_{28}\alpha^3, \\
T_N(\alpha,\beta,u_1,u_3) = \quad & q_{29}\beta - q_{30}\alpha\beta - q_{31}\alpha^2\beta + q_{32}\alpha^3\beta - q_{33}\beta^3 + \\
& q_{34}\alpha\beta^3 + u_1(-q_{35} + q_{36}\alpha - q_{37}\alpha^2 + q_{38}\beta^2 + \\
& q_{39}\alpha^3 - q_{40}\alpha\beta^2) + u_3(-q_{41} + q\alpha - \\
& q_{42}\alpha^2 + q_{43}\beta^2 + q_{44}\alpha^3 + q_{45}\alpha\beta^2 - q_{46}\alpha^4 - \\
& q_{47}\alpha^2\beta^2 - q_{48}\beta^4),
\end{aligned}
$$

where $\{q_1, \cdots, q_{48}\}$ are polynomial coefficients, which in the present work are considered uncertain in contrast with the original work.

This problem can be stated as a $\boldsymbol{\mathcal{QCSP}}$:

- Set of numeric variables $\boldsymbol{x} = \{\alpha, \beta\}$.
- Set of variables' domains $\boldsymbol{X} = \{A, B\}$.
- Set of universally quantified parameters $\boldsymbol{p_U} = \{q_1, \cdots, q_{48}\}$.
- Set of existentially quantified parameters $\boldsymbol{p_E} = \{u_1, u_2, u_3\}$.
- Set of parameters domains $\boldsymbol{P_U} = \{Q_1, \cdots, Q_{48}\}$ and $\boldsymbol{P_E} = \{U_1, U_2, U_3\}$ where $U_i = [-1, 1]$ for $i = 1, 2, 3$.

Table II. Uncertain coefficients

| $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ | $q_7$ |
|---|---|---|---|---|---|---|
| $38 \mp 0.1$ | $170 \mp 0.1$ | $148 \mp 0.1$ | $4 \mp 0.1$ | $52 \mp 0.1$ | $2 \mp 0.1$ | $114 \mp 0.1$ |
| $q_8$ | $q_9$ | $q_{10}$ | $q_{11}$ | $q_{12}$ | $q_{13}$ | $q_{14}$ |
| $79 \mp 0.1$ | $7 \mp 0.1$ | $14 \mp 0.1$ | $14 \mp 0.1$ | $10 \mp 0.1$ | $37 \mp 0.1$ | $48 \mp 0.1$ |
| $q_{15}$ | $q_{16}$ | $q_{17}$ | $q_{18}$ | $q_{19}$ | $q_{20}$ | $q_{21}$ |
| $8 \mp 0.1$ | $13 \mp 0.1$ | $13 \mp 0.1$ | $20 \mp 0.1$ | $11 \mp 0.1$ | $12 \mp 0.1$ | $125 \mp 0.1$ |
| $q_{22}$ | $q_{23}$ | $q_{24}$ | $q_{25}$ | $q_{26}$ | $q_{27}$ | $q_{28}$ |
| $6 \mp 0.1$ | $95 \mp 0.1$ | $21 \mp 0.1$ | $17 \mp 0.1$ | $20 \mp 0.1$ | $81 \mp 0.1$ | $139 \mp 0.1$ |

- Set of constraints $\boldsymbol{\mathcal{C}}(\boldsymbol{x},\boldsymbol{p}) = \{T_L(\alpha,\beta,u_1,u_3,q_1,\cdots,q_{19}) = 0,$ $T_M(\alpha,u_2,q_{20},\cdots,q_{28}) = 0, T_N(\alpha,\beta,u_1,u_3,q_{29},\cdots,q_{48}) = 0\}$.

  And its UE-solution is expressed by

  $$\Sigma_{UE} = \{\alpha \times \beta | \forall(q_1,Q_1')\cdots\forall(q_{48},Q_{48}')\exists(u_1,U_1')\exists(u_2,U_2')\exists(u_3,U_3') \\ (T_L(\alpha,\beta,u_1,u_3,q_1,\cdots,q_{19}) = 0 \wedge T_M(\alpha,u_2,q_{20},\cdots,q_{28}) = 0 \wedge \\ T_N(\alpha,\beta,u_1,u_3,q_{29},\cdots,q_{48}) = 0)\}.$$

### 4.2. Test Case

Consider the problem stated above of finding the admissible set of orientation $(\alpha,\beta)$ of a F-16 aircraft for which the control-surface system $(u_1,u_2,u_3)$ can keep the aircraft stabilized. For the sake of simplicity, suppose that the aerodynamic moment $T_N$ acting over the aircraft is already controlled.

Let us suppose an initial search domain $(\alpha,\beta) \in ([-1,1],[-1,1])$ and the uncertain coefficients of table 4.2.

***QSIMIA*** generates, in 35 seconds on a Pentium IV 1.5GHz, and a precision of $\epsilon = 0.1$ (smallest bisected variable's domain), the paving of figure 4.2. Where the middle dark region corresponds to an inner approximation of the solution set $\Sigma$, the darker region corresponds to an inner approximation of the non solution set $\overline{\Sigma}$ and the lighter region is undefined.

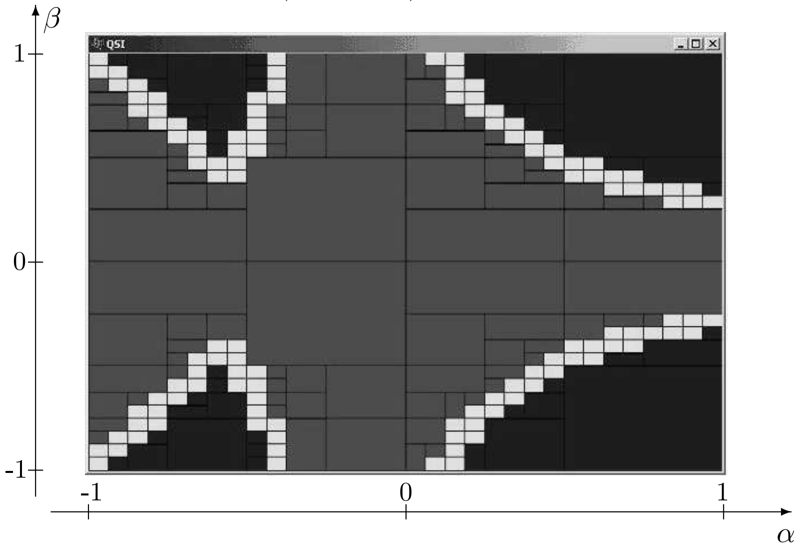


*Figure 5.* Paving generated by $\mathcal{QSI}$ algorithm.

## 5. Conclusions and Work in Progress

The contribution of this paper has been to introduce a new algorithm, named $\mathcal{QSIMIA}$ (Quantified Set Inversion via Modal Interval Analysis), which combines Set Inversion techniques with Modal Interval Analysis in order to solve continuous Quantified Constraint Satisfaction Problems ($\mathcal{QCSP}$) through the characterization of their solution sets. The applicability of the method to engineering problems has been shown by means of solving a control problem on aircraft stabilization.

One of the work in progress consists on solving a $\mathcal{QCSP}$ where one or more existentially quantified variables appears in more than a function component. Another work in progress consists on reducing the non polynomial complexity of the $\mathcal{QSI}$ algorithm due to the branching, introducing a narrowing operator (a contractor) for quantified constraints.

## References


Abdallah, C., Dorato, P., Famularo, D., and Yang, W.: 1999, 'Robust Nonlinear Feedback Design via Quantifier Elimination Theory'. *International Journal of Robust and Non-Linear Control* **9**, 817–822.

Benhamou, F., Goualard, F., Languenou, E., and Christie, M.: 2004, 'Interval constraint solving for camera control and motion planning'. *ACM Trans. Comput. Logic* **5**(4), 732–767.

Benhamou, F. and Goulard, F.: 2000, 'Universally quantified interval constraints'. *Lecture Notes in Computer Science* **5**(4).

Brown, C. W., *Quantifier Elimination by Partial Cylindrical Algebraic Decomposition*, http://www.cs.usna.edu/∼qepcad/B/QEPCAD.html.

Chauvin, C., Muller, M., and Weber, A.: 1994, 'An application of quantifier elimination to mathematical biology'. *In Computer Algebra in Science and Engineering. World Scientific* pp. 287–296.

Collins, G. E.: 1975, 'Quantifier elimination for real closed fields by cylindrical algebraic decomposition'. In: *2nd GI Conf. Automata Theory and Formal Languages*, Vol. 33. pp. 134–189, Springer.

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

Gardeñes, E., Trepat, A., and Janer, J.: 1980, 'Sigla-PL/1: Development and applications'. *K.L.E. Nickel* pp. 301–315.

Gardeñes, E., Sainz, M. Á., Jorba, L., Calm, R., Estela,R., Mielgo,H., and Trepat, A. : 2001, 'Modal Intervals'. *Reliable Computing* **7**(2), 77–111.

Garloff, J. and Graf, B.: 1999, 'Solving strict polynomial inequalities by Bernstein expansion'. *The Use of Symbolic Methods in Control System Analysis and Design* pp. 339–352.

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

Ioakimidis, N. I.: 1999, '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.

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

Jaulin, L. and Walter, E.: 1993, 'Set inversion via interval analysis for nonlinear bounded-error estimation'. *Automatica* **32(8)**, 1053–1064.

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

Ratschan, S.: 2003, 'Solving Existentially Quantified Constraints with One Equality and Arbitrarily Many Inequalities'. In: F. Rossi (ed.): *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming.* pp. 615–633, Springer.

Ratschan, S. and Vehí, J.: 2004, 'Robust Pole Clustering of Parametric Uncertain Systems Using Interval Methods'. In: P. Colaneri (ed.): *Robust Control Design 2003 — Proceedings of the 4th IFAC Symposium.* Elsevier Science.

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

Stevens, B. and Lewis, F.: 1993, 'Aircraft Control and Simulation'. *Wiley.*

Sturm, T.: 2000, 'Reasoning over networks by symbolic methods'. *Applicable Algebra in Engineering Communication and Computing* **10(1)**, 79–96.