Efficient and certified algorithms for solving polynomial system of equalities and inequalities

Jean-Charles Faugère[1]
[1] CALFOR (LIP6) and SALSA (INRIA)
Jean-Charles.Faugere@lip6.fr

Fabrice Rouillier[2]
[2] SALSA (INRIA) and CALFOR (LIP6)
Fabrice.Rouillier@inria.fr

11th July 2005

Abstract

This paper is of mostly of expository nature and has to be considered as a support for the author's lecture at JNRR'05. We mainly describe some efficient strategies for studying real roots of zero - dimensional systems (with a finite number of complex roots) as well as parametric systems, with or without inequations or inequalities. As an example we give a new computational proof of the existence of a parallel robot with 40 *real roots in less than 1 sec on a PC. An application of solving parametric systems is given in the paper: the classification of* 3*-revolute-jointed manipulators based on the cuspidal behavior.*

1 Introduction

When dealing with polynomial systems, the mathematical specification of the result of a computation, in particular when the number of solutions is infinite, is itself a difficult problem [1], [2], [13], [15]. Sorting the most frequently asked questions appearing in the applications, one distinguishes several classes of problems which are different either by their mathematical structure or by the significance that one can give to the word *"solving"*.
Some of the following questions have a different meanings in the real case or in the complex case, others are posed only in the real case:

- zero-dimensional systems (with a finite number of complex solutions - which includes the particular case of univariate polynomials); The questions in general are well defined (numerical approximation, number of solutions, etc) and the handled mathematical objects are relatively simple and well-known;
- parametric systems; They are generally zero-dimensional for almost all the parameters' values. The objective consists in characterizing the solutions of the system (number of real solutions, existence of a parameterization, etc.) with respect to parameters' values.
- positive dimensional systems; For a direct application, the first question is the existence of zeros of a particular type (for example real, real positive, in a finite field). The resolution of such systems can be considered as a black box for the study of more general problems (semi-algebraic sets for example) and information to be extracted is generally the computation of a point per connected component in the real case.
- constructible and semi-algebraic sets; As opposed to what occurs numerically, the addition of constraints or inequalities complicates the problem. Even if semi-algebraic sets represent the basic object of the real geometry, their automatic *"and effective study"* remains a major challenge. To date, the state of the art is poor since only two classes of methods are existing :
  - the Cylindrical Algebraic Decomposition which basically computes a partition of the ambient space in cells where the signs of a given set of polynomials are constant;
  - deformations based methods that turn the problem into solving algebraic varieties.

  The first solution is limited in terms of performances (maximum 3 or 4 variables) because of a recursive treatment variable by variable, the second also because of the use of a sophisticated arithmetic (formal infinitesimals).
- quantified formulas; deciding efficiently if a first order formula is valid or not is certainly one of the greatest challenges in |em "effective" real algebraic geometry. However this problem is relatively well encircled since it can always be rewritten as the conjunction of (supposed to be) simpler problems like the computation of a point per connected component of a semi-algebraic set.

In the present document, we focus on zero-dimensional and parametric systems which currently represents the main class of non trivial practical problems which can be solved using recent algorithm from computer algebra.
We denote by $\mathbb{Q}[X_1,\ldots,X_n]$ the ring of polynomials with rational coefficients and unknowns $X_1,\ldots,X_n$ and $S = \{P_1 =,\ldots,P_s\}$ any subset of $\mathbb{Q}[X_1,\ldots,X_n]$. A point $x \in \mathbb{C}^n$ is a zero of S if $P_i(x) = 0 \quad \forall i = 1\ldots s$. The ideal $\mathcal{I} = \langle P_1,\ldots,P_s\rangle$ generated by $P_1,\ldots,P_s$ is the set of polynomials in $\mathbb{Q}[X_1,\ldots,X_n]$ constituted by all the combinations $\sum_{k=1}^{R} P_k U_k$ with $U_k \in \mathbb{Q}[X_1,\ldots,X_n]$. Since

every element of $\mathcal{I}$ vanishes at each zero of S, we denote by $V_{\mathbb{C}}(S) = V_{\mathbb{C}}(I) = \{x \in \mathbb{C}^n \mid p(x) = 0 \ \ \forall p \in \mathcal{I}\}$ (resp. $V_{\mathbb{R}}(S) = V_{\mathbb{R}}(I) = V_{\mathbb{C}}(I) \bigcap \mathbb{R}^n$) the set of complex (resp. real) zeroes of S.

2 Gröbner bases

A Gröbner basis of an ideal $\mathcal{I}$ is a computable generator set of $\mathcal{I}$ with good algorithmical properties (as described below) and defined with respect to a monomial ordering. For instance the *lexicographic" order* (Lex) is defined by $X_1^{\alpha_1} \cdots X_n^{\alpha_n} <_{Lex} X_1^{\beta_1} \cdots X_n^{\beta_n}$ iff $\exists i_0 \leq n$ such that $\alpha_{i_0} < \beta_{i_0}$ and $\alpha_i = \beta_i \ \ , \ \ \forall i = 1 \ldots i_0 - 1$.
Lets define some useful notations :

Definition 1 *For any n-uple $\mu = (\mu_1, \ldots, \mu_n) \in \mathbb{N}^n$, let denote by X^μ the monomial $X_1^{\mu_1} \cdot \ldots \cdot X_n^{\mu_n}$. If $<$ is an admissible (compatible with the multiplication) monomial ordering and $P = \sum_{i=0}^{r} a_i X^{\mu^{(i)}}$ any polynomial in $\mathbb{Q}[X_1, \ldots, X_n]$, we define :*

- $\mathrm{LM}(P, <) = \max_{i=0\ldots r} \ \ , \ _{<} X^{\mu^{(i)}}$ *(leading monomial of P w.r.t. $<$)*
- $\mathrm{LC}(P, <) = a_i$ *with i such that* $\mathrm{LT}(P) = X^{\mu^{(i)}}$ *(leading coefficient of P w.r.t. $<$)*
- $\mathrm{LT}(P, <) = \mathrm{LC}(P, <) * \mathrm{LM}(P, <)$ *(leading term of P w.r.t. $<$)*

Lets define the mathematical object "Gröbner":

Definition 2 *A set of polynomials G is a Gröbner basis of an ideal $\mathcal{I}$ wrt to a monomial ordering $<$ if for all $f \in \mathcal{I}$ there exists $g \in G$ such that $LM(g)$ divides $LM(f)$.*

Given any admissible monomial ordering one can extend the classical Euclidean division to *reduce* a polynomial$p$ by another one or, more generally, by a set of polynomials F. Lets denote by $\mathrm{Reduce}(p, F, <)$ the result of this division. Unlike in the univariate case, the result of such a process is not canonical and depends on the monomial ordering used but also on the order you perform the reductions. One of the main properties of Gröbner basis is to provide an algorithmic method for deciding if a polynomial belongs or not to an ideal :

Theorem 1 *Let G be a Gröbner basis G of an ideal $\mathcal{I} \subset \mathbb{Q}[X_1, \ldots, X_n]$ for any monomial ordering $<$.*

(i) a polynomial $p \in \mathbb{Q}[X_1, \ldots, X_n]$ belongs to $\mathcal{I}$ if and only if Reduce$(p, G, <) = 0$,

(ii) Reduce(p,G,<) does not depend on the order of the polynomials in the list G, thus, this is a canonical reduced expression modulus $\mathcal{I}$.

Gröbner bases are computable objects. The historical method for computing them is Buchberger's algorithm ([7, 6]). It has several variants and it is implemented in most of general computer algebra systems like Maple or Mathematica. Recently, more efficient algorithms have been proposed to compute Gröbner bases:

- the F_4 algorithm [10] is based on the intensive use of linear algebra methods: in short, the arbitrary choices are left to computational strategies related to classical linear algebra problems (mainly the computation of row echelon form).

- In [12] a new criterion (the F_5 criterion) for detecting useless computations has been given; under some regularity conditions on the system, it is proved that the algorithm do never perform useless computations. A new algorithm named F_5 has been built using these two ideas: the F_5 algorithm constructs incrementally the following matrices in degree d:

$$A_d = \begin{array}{c} \\ t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \ldots \end{array} \begin{array}{c} \begin{array}{cccc} m_1 > & m_2 > & m_3 & \ldots \end{array} \\ \begin{bmatrix} \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{bmatrix} \end{array}$$

  where the indices of the columns are monomials sorted for the admissible ordering $<$and the rows are product of some polynomials f_i by some monomials t_j such that $\deg(t_j f_i) \leq d$.For a regular system the matrices $A_d$are full rank. In a second step, row echelon forms of the matrices are computed:

$$A'_d = \begin{array}{c} \\ t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \ldots \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & \ldots \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \ldots \\ 0 & 1 & 0 & \ldots \\ 0 & 0 & 1 & \ldots \\ 0 & 0 & 0 & \ldots \end{bmatrix} \end{array}$$

Even if F_5 still computes the same mathematical object (a Gröbner basis), the gap with existing other algorithms is consequent. In particular, due to the range of examples that become computable, Gröbner basis can be considered as a reasonable computable object in large applications. Important parameters to evaluate the complexity of Gröbner bases with the F_5 are the D the maximal degree d occurring in the computation and the size of the matrix A_d. The overall cost is thus dominated by $(\#A_d)^3$.

We pay a particular attention to Gröbner bases computed for elimination orderings since they provide a way of simplifying the system (an equivalent system with a structured shape). For example, a lexicographic Gröbner basis of a

zero dimensional system has always the following shape :

$$\begin{cases} f(X_1) = 0 \\ f_2(X_1, X_2) = 0 \\ \vdots \\ f_{k_2}(X_1, X_2) = 0 \\ f_{k_2+1}(X_1, X_2, X_3) = 0 \\ \vdots \\ f_{k_{n-1}+1}(X_1, \ldots, X_n) = 0 \\ \vdots \\ f_{k_n}(X_1, \ldots, X_n) = 0 \end{cases}.$$

(when the system is not zero dimensional some of the polynomials may be identically null). A well known property is that the zeros of the smallest (w.r.t. $<$) non null polynomial define the Zariski closure (classical closure in the case of complex coefficients) of the projection on the coordinate's space associated with the smallest variables.
More generally, an admissible ordering $<$ on the monomials depending on variables $[U_1, \ldots, U_d, X_{d+1}, \ldots, X_n]$ which eliminates $X_{d+1}, \ldots, X_n$ is an ordering such that $U_i < X_j \ \forall i = 1 \ldots d, j = d+1 \ldots n$. The lexicographic ordering is a particular elimination ordering. Given two monomial orderings $<_U$(w.r.t. the variables $U_1, \ldots, U_d$) and $<_X$(w.r.t. the variables $X_{d+1}, \ldots, X_n$) one can define an ordering which "eliminates" $X_{d+1}, \ldots, X_n$ by setting the so called block ordering $<_{U,X}$ as follows : given two monomials m and m', $m <_{U,X} m'$ if and only if $m_{|_{U_1=1,\ldots,U_d=1}} <_2 m'_{|_{U_1=1,\ldots,U_d=1}}$ or ($m_{|_{U_1=1,\ldots,U_d=1}} = m'_{|_{U_1=1,\ldots,U_d=1}}$ and $m_{|_{X_{d+1}=1,\ldots,X_n=1}} <_1 m'_{|_{X_{d+1}=1,\ldots,X_n=1}}$).
Two important applications of elimination theory are the "projections" and "localizations". In the following, given any subset $\mathcal{V}$ of $\mathbb{C}^d$ (d is an arbitrary positive integer), $\overline{\mathcal{V}}$ is its Zariski closure, say the smallest subset of $\mathbb{C}^d$ containing $\mathcal{V}$ which is the zero set of a system of polynomial equations.

Proposition 1 *Let G be a Gröbner basis of an ideal $I \subset \mathbb{Q}[U, X]$ w.r.t. $<_{U,X}$, then $G \bigcap \mathbb{Q}[U]$ is a Gröbner basis of $I \bigcap \mathbb{Q}[U]$ w.r.t. $<_U$;*
Let T be a new indeterminate, then $\overline{V(I) \setminus V(f)} = V((I + \langle Tf - 1\rangle) \bigcap \mathbb{Q}[U, X])$. If $G' \subset \mathbb{Q}[U, X, T]$ is a Gröbner basis of $I + \langle Tf - 1\rangle$ with respect to $<_{(U,X),T}$ then $G' \bigcap \mathbb{Q}[U, T]$ is a Gröbner basis of $I : f^\infty := \overline{(I + \langle Tf - 1\rangle) \bigcap \mathbb{Q}[U, X]}$ w.r.t. $<_{(U,X)}$. The variety $\overline{V(I) \setminus V(f)}$ and the ideal $I : f^\infty$ are usually called the localization of $V(I)$ and I by f.

3 Zero-dimensional systems

Zero-dimensional systems are polynomial systems with a finite number of complex solutions. This specific case is fundamental for many engineering applications. The following theorem shows that we can detect easily that a system is zero dimensional or not by computing a Gröbner base for any monomial ordering :

Theorem 2 *Let $G = \{g_1, \ldots, g_l\}$ be a Gröbner basis for any ordering $<$ of any system $S = \{P_1, \ldots, P_s\} \in \mathbb{Q}[X_1, \ldots, X_n]^s$. The two following properties are equivalent :*

- *For all index i, $i = 1 \ldots n$, there exists a polynomial $g_j \in G$ and a positive integer n_j such that $X_i^{n_j} = LM(g_j, <)$;*
- *The system $\{P_1 = 0, \ldots, P_s = 0\}$ has a finite number of solutions in $\mathbb{C}^n$.*

If S is zero-dimensional, then, according to theorem 2, only a finite number of monomials $m \in \mathbb{Q}[X_1, \ldots, X_n]$ are not reducible modulo G, meaning that Reduce(m,G,$<$)$= m$. Mathematically, a system is zero-dimensional if and only if $\mathbb{Q}[X_1, \ldots, X_n]/I$ is a $\mathbb{Q}$-vector space of finite dimension. This vector space can fully be characterized when knowing a Gröbner basis:

Theorem 3 *Let $S = \{p_1, \ldots, p_s\}$ be a set of polynomials with $p_i \in \mathbb{Q}[X_1, \ldots, X_n]$,;$\forall i = 1 \ldots s$, and suppose that G is a Gröbner basis of $\langle S\rangle$ with respect to any monomial ordering $<$. Then :*

- *$\mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I} = \{\mathrm{Reduce}(p, G, <) \ \ p \in \mathcal{I}\}$ is a vector space of finite dimension;*
- *$\mathcal{B} = \{t = X_1^{e_1} \cdot X_n^{e_n}, (e_1, \ldots, e_n) \in \mathbb{N}^n | \ \mathrm{Reduce}(t, G, <) = t\} = \{w_1, \ldots, w_D\}$ is a (vector space) basis of $\mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I}$;*
- *$D = \sharp\mathcal{B}$ is exactly the number of elements of complex zeroes of the system $\{P = 0, \ \ \forall P \in S\}$ counted with multiplicities.*

Thus, when a polynomial system is known to be zero-dimensional, one can switch to linear algebra methods to get informations about its roots. Once a Gröbner basis is known, a basis of $\mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I}$ can easily be computed (Theorem 3) so that linear algebra methods can be applied for doing several computations.
For any polynomial $q \in \mathbb{Q}[X_1, \ldots, X_n]$ the decomposition $\overline{q}$ =Reduce(q,G,$<$)=$\sum_{i=1}^{D} a_i w_i$ is unique (theorem 1) and we denote by $\vec{q} = [a_1, \ldots, a_D]$ the representation of $\overline{q}$ in the basis $\mathcal{B}$. For example, the matrix w.r.t. $\mathcal{B}$ of the linear map

$$\begin{array}{rccc} m_q : & \mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I} & \longrightarrow & \mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I} \\ & \vec{p} & \longmapsto & \overrightarrow{\mathrm{pq}} \end{array}$$

can explicitly be computed (its columns are the vectors $\overrightarrow{qw_i}$) and one can then apply the following well-known theorem:

Theorem 4 *(Stickelberger) The eigenvalues of m_q are exactly the $q(\alpha)$ where $\alpha \in V_{\mathbb{C}}(S)$.*

According to Theorem 4, the i-th coordinate of all $\alpha \in V_{\mathbb{C}}(S)$ can be obtained from M_{X_i} eigenvalues but the issue of finding all the coordinates of all the $\alpha \in V_{\mathbb{C}}(S)$ from

$M_{X_1}, \ldots, M_{X_n}$ eigenvalues is not explicit nor straightforward (see [4] for example). Note also that some authors propose algorithms to compute numerically the matrices $M_{X_1}, \ldots, M_{X_n}$ without computing Gröbner bases (see [16]). Up to our experiments, such computations are not numerically stable for general manipulators and it may be preferable to compute, for example, the characteristic polynomial of the matrix M_{X_i} and then isolate its real roots. Thus one would prefer to follow with exact computations a little bit more, providing exact formulas as explained in the next section.

3.1 The Rational Univariate Representation

The Rational Univariate Representation [19] is, with the end-user point of view, the simplest way for representing symbolically the roots of a zero-dimensional system without loosing information (multiplicities or real roots) since one can get all the information on the roots of the system by solving univariate polynomials.

Given a zero-dimensional system $I =< p_1, \ldots, p_s >$ where the $p_i \in \mathbb{Q}[X_1, \ldots, X_n]$, a Rational Univariate Representation of $\mathrm{V}(I)$ has the following shape : $f_t(T) = 0, X_1 = \frac{g_{t,X_1}(T)}{g_{t,1}(T)}, \ldots, X_n = \frac{g_{t,X_n}(T)}{g_{t,1}(T)}$, where $f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n} \in \mathbb{Q}[T]$ (T is a new variable). It is uniquely defined w.r.t. a given polynomial t which separates $V(I)$ (injective on $V(I)$), the polynomial f_t being necessarily the characteristic polynomial of m_t (see above section) in $\mathbb{Q}[X_1, \ldots, X_n]/I$ [19]. The RUR defines a bijection between the roots of $\mathcal{F}$ and those of f_t preserving the multiplicities and the real roots :

$$\begin{array}{ccc} \mathrm{V}(\mathcal{S})(\cap\mathbb{R}) & \approx & \mathrm{V}(f_t)(\cap\mathbb{R}) \\ \alpha = (\alpha_1, \ldots, \alpha_n) & \rightarrow & t(\alpha) \\ (\frac{g_{t,X_1}(t(\alpha))}{g_{t,1}(t(\alpha))}, \ldots, \frac{g_{t,X_n}(t(\alpha))}{g_{t,1}(t(\alpha))}) & \leftarrow & t(\alpha) \end{array}$$

For computing a RUR one have to solve two problems :

- finding a separating element t
- given any polynomial t, compute a RUR-Candidate $f_t, g_{t,1}, g_{t,X_1}, \ldots, g_{t,X_n}$ such that if t is a separating polynomial, then the RUR-Candidate is a RUR.

According to [19], a RUR-Candidate can explicitly be computed when knowing a suitable representation of $\mathbb{Q}[X_1, \ldots, X_n]/\mathcal{I}$:

- $f_t = \sum_{i=0}^{D} a_i T^i$ is the characteristic polynomial of m_t. Lets denotes by $\overline{f_t}$ its square-free part.
- for any $v \in \mathbb{Q}[X_1, \ldots, X_n]$, $g_{t,v} = g_{t,v}(T) = \sum_{i=0}^{d-1} \mathrm{Tr\,ace}(m_{vt^i}) H_{d-i-1}(T)$, $d = \deg(\overline{f_t})$ and $H_j(T) = \sum_{i=0}^{j} a_i T^{i-j}$

In [19], a strategy is proposed for computing a RUR for any system (a RUR-Candidate and a separating element), but there are special cases where it can be computed differently. When X_1 is separating $\mathrm{V}(I)$ and when I is a radical ideal the system is said to be in *shape position*. In such cases, the shape of the lexicographic Gröbner basis is always the following :

$$\left\{ \begin{array}{l} f(X_1) = 0 \\ X_2 = f_2(X_1) \\ \vdots \\ X_n = f_n(X_1) \end{array} \right. . \qquad (1)$$

As shown in [19], if the system is in shape position, $g_{X_1,1} = f'_{X_1}$ and we have $f_{X_1} = f$ and $f_i(X_1) = g_{X_1,X_i}(X_1) g_{X_1,1}(X_1) mod f$. Thus the RUR associated with X_1 and the lexicographic Gröbner basis are equivalent up to the inversion of $g_{X_1,1} = f'_{X_1}$ modulo f_{X_1}. In the rest of this paper we call this object a RR-Form of the corresponding lexicographic Gröbner basis. The RUR is well known to be smaller than the lexicographic Gröbner basis in general and thus will be our priviligied object. Note that it is easy to check that a system is in shape position once knowing a RUR-Candidate (and so to check that X_1 separates $\mathrm{V}(I)$): it is necessary and sufficient that f_{X_1} is square-free.

These results have many practical drawbacks since, the systems which are often in shape position. We thus can multiply the strategies for computing a RUR : one can compute a "modified" lexicographic Gröbner directly using [10] for example or by change of ordering like in [11] or a RUR using the algorithm from [19].

3.2 From formal to numerical solutions

Computing a RUR reduces the resolution of a zero-dimensional system to solving one polynomial in one variable (f_t) and to evaluating n rational fractions ($\frac{g_{t,X_i}(T)}{g_{t,1}(T)}, i = 1 \ldots n$) at its roots (note that if one simply want to compute the number of real roots of the system there is no need to consider the rational coordinates). Our goal is to compute all the real roots of the system (and only the real roots), providing a numerical approximation with an arbitrary precision (set by the user) of the coordinates.

In practice, the computation of the RUR is not the end point of the work : approximating the roots of f_t is not sufficient to provide accurate numerical approximations of the roots of the initial system and, moreover, not sufficient to guarantee the sign of the coordinates. Also a naive algorithm which would consist in "plugging" numerical approximations of the roots into the $f_1, \ldots, f_s$ will not give, in most cases, any suitable information. If one is only interested in the signs of the f_i one could imagine computing $f_i(\frac{g_{t,X_1}(T)}{g_{t,A}(T)}, \ldots, \frac{g_{t,X_n}(T)}{g_{t,1}(T)})$ and studying the values of these polynomials at the roots of I. Again, this would lead to very hard computations since such a plug induces multiplying large polynomials modulo f_t.

The isolation of the real roots of f_t can be done using the algorithm proposed in [20] : the output will be a list l_{f_t} of intervals with rational bounds such that for each real root α of f_t, there exists a unique interval in l_{f_t} which contains

α. The second step consists in refining each interval in order to ensure that it does not contain any real root of $g_{t,1}$. Since f_t and $g_{t,1}$ are co-prime this computation is easy and we then can ensure that the rational functions can be evaluated using interval arithmetics without any cancellation of the denominator. This last evaluation is performed using multi-precision arithmetics (MPFI package - [18]). As we will see in the experiments, the precision needed for the computations is poor and, moreover, the rational functions defined by the RUR are stable under numerical evaluation, even if their coefficients are huge (rational numbers), and thus this part of the computation is still efficient. For increasing the precision of the result, it is only necessary to decrease the length of the intervals in l_{f_t} which can easily be done by bisection. Note that is is quite simple to certify the sign of the coordinates : one simply have to compute some gcds and split, when necessary the RUR.

3.3 Signs of polynomials at the roots of a system

Computing the sign of given multivariate polynomials at the real roots of a zero-dimensional system may be important for many applications and this problem is not solved by the above method. Instead of *"plugging"* straightforwardly the formal coordinates provided by the RUR into the f_i, we better extend the RUR by computing rational functions which coincide with the f_i at the roots of I. This can theoretically simply be done by using the general formula from [19] : $h_{t,j} = \sum_{i=0}^{D-1} \mathrm{Tr}\, ace(f_j t^i) H_{D-i-1}(T)$. One can directly compute the $\mathrm{Tr}\, ace(f_j t^i)$ reusing the computations already done if the (classical) RUR (without additional constraints) has already been computed and show that as soon as s is small, it is not more costly to compute the extended RUR than the classical one.
The right way for studying the signs of the f_i at the elements of $\mathrm{V}(I)$ consists in first computing the Gcd of each $h_{t,i}$ and f_t to localize the roots where the f_i vanish and then to evaluate the $h_{t,i}$ at the other roots using interval arithmetic.

4 Solving Parametric Systems

The method described in this section is a particular case of the algorithm from [14] : we impose here that the system has as many equations as unknowns, the ideal generated by its equations is radical. This class of example may be considered as being generic in practice since it contains all parametric systems which can be solved by simple versions of Newtons' method for almost all the specializations of the parameters. The following notations will be used:

Notation 1 *Let us consider the basic semi-algebraic set*

$$\mathcal{S} = \{x \in \mathbb{R}^n \quad , \quad \begin{array}{l} p_1(x) = 0, \ldots, p_s(x) = 0, \\ f_1(x) > 0, \ldots f_l(x) > 0 \end{array} \}$$

and the basic constructible set

$$\mathcal{C} = \{x \in \mathbb{C}^n \quad , \quad \begin{array}{l} p_1(x) = 0, \ldots, p_s(x) = 0, \\ f_1(x) \neq 0, \ldots f_l(x) \neq 0 \end{array} \}$$

where p_i, f_j are polynomials with rational coefficients.

- *$[U, X] = [U_1, \ldots U_d, X_{d+1}, \ldots X_n]$ is the set of unknowns or variables, while $U = [U_1, \ldots U_d]$ is the set of parameters and $X = [X_{d+1}, \ldots X_n]$ the set of unknowns;*
- *$\mathcal{E} = \{p_1, \ldots p_s\}$ is the set of polynomials defining the equations;*
- *$\mathcal{F} = \{f_1, \ldots f_l\}$ is the set of polynomials defining the inequations in the complex case or the inequalities in the real case;*
- *For any $u \in C^d$, ϕ_u is the specialization map $U \longrightarrow u$;*
- *$\Pi_U : \mathbb{C}^n \longrightarrow \mathbb{C}^d$ denotes the canonical projection on the parameter's space $(u_1, \ldots, u_d, x_{d+1}, \ldots, x_n) \longrightarrow (u_1, \ldots, u_d)$;*
- *Given any ideal I we denote by $V(I) \subset \mathbb{C}^n$ the associated (algebraic) variety.*
- *for any set $\mathcal{V} \subset \mathbb{C}^n$, $\overline{\mathcal{V}}$ denotes its $\mathbb{C}$-Zariski closure (the smallest algebraic variety containing $\mathcal{V}$).*

Solving $\mathcal{C}$ or $\mathcal{S}$ amounts to compute sub-manifolds $\mathcal{U} \subset \mathbb{C}^d$ such that $(\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C}, \Pi_U)$ is an analytic covering of $\mathcal{U}$ (in that case, we say that $\mathcal{U}$ has the $(\Pi_U, \mathcal{C})$-*covering property*). This guarantees that the cardinal of $\Pi_U^{-1}(\sqcap) \cap \mathcal{C}$ is constant for all $u \in \mathcal{U}$ and that $\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C}$ is a finite collection of sheets which are all homeomorphic to $\mathcal{U}$. Note that the result remains true in restriction to the reals, replacing $\mathcal{C}$ by $\mathcal{S}$, so we focus on the complex case (study of $\mathcal{C}$).
Under our assumption "as many unknowns as equations", $\Pi_U(\mathcal{C})$ is dense in $\mathbb{C}^d$ and all the known algorithms for solving $\mathcal{C}$ or $\mathcal{S}$ compute implicitly or explicitly a Zariski closed subset W such that any sub-manifold of $\mathbb{C}^d \setminus W$ have the $(\Pi_U, \mathcal{C})$-covering property.
In [14], the authors introduce the *discriminant varieties of $\mathcal{C}$ w.r.t.* Π_U which are algebraic sets with the above property (even in the cases where Π_U is not dense in $\mathbb{C}^d$). As one of the main results, they show that the complement in $\mathbb{C}^d$ of the union of the open subsets which have the $(\Pi_U, \mathcal{C})$-covering property is a Zariski closed set which is thus the *minimal discriminant variety of $\mathcal{C}$ w.r.t.* Π_U.
Under the hypothesis $s = n - d$ (as many equations as unknowns), results from [14] shows that this minimal discriminant variety can be decomposed as $W_D = W_\infty \cup W_c \cup_{i=1}^{l} W_{f_i}$, where:

- W_c is the Zariski closure of the set of critical values of Π_U in restricted to the union of the components of dimension d of $\overline{\mathcal{C}}$;
- W_∞ is the set of points $u \in \mathbb{C}^d$ such that $\Pi_U^{-1}(\mathcal{U}) \bigcap \overline{\mathcal{C}}$ is not compact for any compact neighborhood $\mathcal{U}$ of u in Π;

- W_{f_i} is the Zariski closure of the projection of the intersection of $\overline{\mathcal{C}}$ with the hyper-surface defined by f_i=0;

4.1 Computing the minima discriminant variety

For computing the minimal discriminant variety W_D, on need to compute first $\overline{C}$ and then the components W_c, W_∞ and W_{f_i}.
Writing $\overline{\mathcal{C}} = \overline{\mathrm{V}(\mathcal{E}) \setminus \cup_{i=1}^{l} \mathrm{V}(f_i)}$, one can apply proposition 1 to compute a polynomial system (in fact a Gröbner basis) whose zero set is $\overline{\mathcal{C}}$ by localizing $\langle \mathcal{E} \rangle$ iteratively w.r.t. the f_i, say compute $I = (\ldots((\langle \mathcal{E} \rangle : f_1^\infty) : f_2^\infty)\ldots) : f_l^\infty$. Using again proposition 1, one can then compute $I_{f_i} = (I + < f_i >) \cap \mathbb{Q}[U]$ such that $W_{f_i} = \mathrm{V}(f_i)$.
The computation of W_∞ can be done using the theorem from [14]:

Theorem 5 *Let G be a reduced Gröbner basis of any ideal I such that $V(I) = \overline{\mathcal{C}}$ for the product ordering$<_{U,X}$ where $<_X$ is the Degree Reverse Lexicographic ordering s.t. $X_{d+1} < \ldots < X_n$. We define $\mathcal{E}_i^\infty = \{\mathrm{LC}_{<_X}(g) \mid g \in G \ , \ \exists m \geq 0, \ \mathrm{LM}_{<_X}(g) = X_i^m\}$, and $\mathcal{E}_0 = G \bigcap \mathbb{Q}[U]$. Then:*

- *$\mathcal{E}_0$ is a Gröbner basis of $I \bigcap \mathbb{Q}[U]$ w.r.t. $<_U$ and $\mathcal{E}_0 \subset \mathcal{E}_i^\infty$ for $i = d+1 \ldots n$;*
- *$\mathcal{E}_i^\infty$ is a Gröbner basis of some ideal $I_i^\infty \subset \mathbb{Q}[U]$ w.r.t. $<_U$;*
- *$W_\infty = \bigcup_{i=d+1}^{n} V(I_i^\infty)$.*
- *if $I \bigcap \mathbb{Q}[U]$ is prime, then $W_\infty = \Pi$ if and only if $\mathcal{E}_i^\infty = \mathcal{E}_0$ for some i.*

Note that under the hypothesis $s = n - d$, $\mathcal{E}_0 = \emptyset$.
If I is prime, then W_c is the zero set of $(I + \mathrm{Jac}_X^{n-\delta}(I)) \cap \mathbb{Q}[U]$ where $\mathrm{Jac}_X^{n-\delta}(I)$ is the ideal generated by the Jacobian determinant with respect to the variables X of any systems of generators of I. This characterization can be extended to equi-dimensional and radical ideals but not to the general case (consider for example the system $P^2 = 0$ where P is a non constant polynomial in $\mathbb{Q}[U, X]$). Under the hypothesis $s = n - d$ together with the condition I is radical, one has always (according to [14]) $W_c = \mathrm{V}(I + \mathrm{Jac}_X^{n-\delta}(I)) \cap \mathbb{Q}[U]$. In this case, one can compute, using again proposition 1, a system of generators I_c (in practice a Gröbner basis) such that $W_c = \mathrm{V}(I_c)$.
The condition $s = n - d$ can be a priori tested (it is sufficient to count the number of equalities in the system) and the condition "I is radical" can be replaced by "$I + \mathrm{Jac}_X^{n-\delta}(I)$" has dimension less than d. If it is not the case, one would need to compute the so called "radical of I" and run the process again (not developed in this short survey).
At this step, one knows how to compute a set of ideals $I_i^\infty, i = d+1 \ldots n, I_c, I_{f_i}, i = 1 \ldots l$ such that $W_D = \cup_{i=d+1}^{n} \mathrm{V}(I_i^\infty) \cup \mathrm{V}(I_c) \cup_{i=1}^{l} \mathrm{V}(I_{f_i})$.

4.2 Using the discriminant variety

Let us denote by $\mathcal{U}_1, \ldots \mathcal{U}_k$ the connected components of $\mathbb{R}^d \setminus W_D$. If $u_1, \ldots, u_k$ are sample points such that $u_i \in \mathcal{U}_i$ then $\bigcup_{i=1}^{k} \Pi_U^{-1}(u_i)$ intersects each connected component of $\mathcal{V} = \mathrm{V}(\langle \mathcal{E} \rangle) \cap \mathbb{R}^n$ in a finite number of points. Moreover, if $\mathcal{U}$ is a small neighborhood of u_i, then $\Pi_U^{-1}(u_i) \cap \mathcal{V}$ consists in exactly one point in each connected component of $\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{V}$. By removing the points of $\Pi_U^{-1}(u_i) \cap \mathcal{V}$ which do not verify the inequations $(f > 0)_{f \in \mathcal{F}}$, one gets exactly one point on each semi-algebraic connected component of $\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{S}$. Thus, by computing one point on each $\mathcal{U}_\rangle$, one can get the number of real points of $\mathcal{S}$ over any point of $\mathcal{U}_\rangle$, which is constant on $\mathcal{U}_\rangle$. Thus the number of real or complex solutions of $\mathcal{S}$ for parameters' values which do not belong to W_D depends only on the connected component $\mathcal{U}_\rangle$ and is a computable well defined function of the index i.
Obtaining the sample points $u_1, \ldots, u_k$ consists in computing one point on each connected component of $\mathbb{C}^d \setminus W_D$, which may be got with a good theoretical complexity by the algorithms described in [5]. In practice, the end-user often wants to compute the number of real roots of the system as a function of the parameters. Computing at least one point on each $\mathcal{U}_i$ not enough for this: one needs also, at least, an algorithm to test if two points are in the same connected component, or hopefully a comprehensive description of the connected components.
Basically, the CAD algorithm computes a cylindrical decomposition of the ambient space in cells such that the polynomials of a given set have a constant sign on each cell. Precisely :

Definition 3 *A cylindrical algebraic decomposition of $\mathbb{R}^d$ is a sequence $C_1, ..., C_d$, where, for $1 \leq k \leq d$, C_k is a finite partition of $\mathbb{R}^k$ into semi-algebraic subsets (which are called cells), satisfying the following properties:*

- *Each cell $C \in C_1$ is either a point, or an open interval.*
- *For every $k, 1 \leq k < d$, and for every $C \in C_k$, there are finitely many continuous semi-algebraic functions (graphs of semi-algebraic sets) $\xi_{C,1} < \ldots < \xi_{C,l_C}$ $C : C \longrightarrow \mathbb{R}$ such that the cylinder $C \times \mathbb{R} \subset \mathbb{R}^{k+1}$ is the disjoint union of cells of C_{k+1} which are:*
  - *either the graph of one of the functions $\xi_{C,j}$, for $j = 1, ..., l_C$:*

    $$A_{C,j} = \{(x', x_{k+1}) \in C \times \mathbb{R} \ ; \ x_{k+1} = \xi_{C,j}(x')\};$$

  - *or a band of the cylinder bounded from below and from above by the graphs of functions $\xi_{C,j}$ and $\xi_{C,j+1}$, for $j = 0, \ldots, l_C$, where we take $\xi_{C,0} = -\infty$ and $\xi_{i,l_C+1} = +\infty$:*

    $$B_{C,j} = \{(x', x_{k+1}) \in C \times \mathbb{R} \ ; \ \xi_{C,j}(x') < x_{k+1} < \xi_{C,j+1}$$

A CAD adapted to a set $\{P_1, \ldots, P_s\}$ of polynomials of $\mathbb{R}[U_1, \ldots, U_d]$ is a CAD such that each cell C is $(P_1, \ldots, P_s)$-invariant, which means that the P_i have a constant sign in each cell.

In our case, a CAD adapted to the set of the polynomials defining the discriminant variety will provide a partition of $\mathbb{R}^d$ into cells where the signs of these polynomials are constant. In particular, all the cells such that none of these polynomials vanishes are embedded in a $\mathcal{U}_i$ defined above while the others will be embedded in W_D. If we are not interested in decomposing W_D (most practical situations), one can simplify a lot the original algorithm proposed by Collins and compute a *Partial CAD*. In the following, one suppose that $\mathcal{P}_\infty$ is the set of polynomials which appear in the above representation of W_D

PCAD - Projection step
At level k, we have a set $\mathcal{P}_k$ of polynomial of $K[U_k, \ldots, U_d]$. We construct $\mathcal{P}_{k+1} = Proj(\mathcal{P}_k)$ as being the smallest set such that:

- If $p \in \mathcal{P}_k$, $\deg_{U_k}(p) = d \geq 2$, $Proj(\mathcal{P}_k)$ contains all the (non constant))discriminant $Discrim(p, U_k)$.
- If $p \in \mathcal{P}_k$, $q \in \mathcal{P}_k$, $Proj(\mathcal{P}_k)$ contains $Resultant(p, q)$ (if non-constant).
- If $p \in \mathcal{P}_k$, $\deg_{U_k}(p) \geq 1$ and $lc_{U_k}(p)$ non constant, $Proj(\mathcal{P}_k)$ contains $lc_{U_k}(p)$.
- If $p \in \mathcal{P}_k$, $\deg_{U_k}(p) = 0$ and p non constant, $Proj(\mathcal{P}_k)$, contains p.

PCAD - lifting step / effective output
A human readable characterization of a cell in R that is not a point (real algebraic number) could simply be an integer i such that if p_1 denotes the product of all the polynomials of $Proj(\mathcal{P}_1)$,then if $i = l$, the cell is the interval between the l-th and the $l+1$-th root of p_1. By convention, $i = 0$ represents the interval $]-\infty; \alpha_1[$ where α_1 is the smallest real root of p_1, and if d_1 is the number of real roots of p_1, then the integer $i = d_1 + 1$ represents the interval $]\alpha_{d_1}, +\infty[$.
More generally, we can characterize recursively the cells of R^k we need as a k-uple $[i_1, \ldots, i_k]$ such that $[i_1, \ldots, i_{k-1}]$ characterizes a cell of R^{k-1} and i_k is an integer such that if p_k denotes the product of all the polynomials of $Proj(\mathcal{P}_k)$,then if $i_k = l$, the cell is the interval between the l-th and the $l+1$-th root of p_k. Also the final output may consists in a list of d-uples of integers and a triangular set $(p1, p2, p3, \ldots, p_d)$ which provides sufficiently many informations to compute at least one point on each cell and so compute the corresponding sequence of signs realized by the initial set of polynomials $\{\mathcal{P}_1, \ldots \mathcal{P}_s\}$.
In practice, each step of the lifting phase induce the following computations:

- (1) compute real roots of all polynomials of $\mathcal{P}_k$ and sort them;
- (2) take one point on each interval between roots of (1);
- (3) specialize U_k to (2) in $\mathcal{P}_{k-1} \ldots \mathcal{P}_1$.

One can notice that there are no more computations with real algebraic numbers ...
The proof of the correctness of this algorithm comes from the correctness of Collins' algorithm and from the fact that we only removed cells that belong to $\Pi_U^{-1}(W_D)$.

Remark 1 An important additional test is useful in practice: before adding a polynomial in the projection step, we use filters or algorithms to detect if it has no real roots. This may be done by applying CAD algorithm again but also methods such as proposed in [3] or better in [21].

5 Applications

5.1 Parallel robot with 40 real roots

Using a numerical global optimization program, Dietmaier [17] gives explicitly an example of a robot with 40 real roots; we show that using the tools presented in the paper it is very easy to check that the solutions are really real numbers (and not complex number with a very small imaginary part).
Solving the Direct Kinematic Problem (*DKP*) consists in computing the position of the robot (designated by B_i articulation position located on the end-effector moving platform) knowing the configuration of the robot (points A_i located on the base) and the lengths of the actuators $L_i = ||A_iB_i||$. Among the numerous existing algebraic formulations of the DKP problem which are commonly used in computer algebra we used the Displacement based equations: let R_f (resp. R_m) be the base Cartesian reference frame of center O (resp. reference frame of center C relative to the mobile platform). if there exists any mobile platform position $\overrightarrow{\mathrm{OB}}_{i_{|R_f}}$ which meets the constrains , $i = 1 \ldots 6$, then there exists a rotation $\mathcal{R}$ such that :

$$\overrightarrow{\mathrm{OB}}_{i_{|R_f}} = \overrightarrow{\mathrm{OC}}_{|R_f} + \mathcal{R} \cdot \overrightarrow{\mathrm{CB}}_{i_{|R_m}} \quad , \quad i = 1 \ldots 6 \qquad (2)$$

The natural way to set an algebraic equation system from (2) is to straightforwardly use the rotation matrix parameters and the vector $\overrightarrow{\mathrm{OC}}_{|R_f} = [X, Y, Z]$ coordinates as unknowns. Any rotation $\mathcal{R}$ can be expressed using the Cayley transform: if H is any anti-symmetric matrix:

$$H = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$$

then, provided that 1 is not an eigenvalue of H, then $\mathcal{R} = \frac{\mathcal{I}+\mathcal{H}}{\mathcal{I}-\mathcal{H}}$ is a rotation and is given by

$$\frac{1}{\Delta} \begin{bmatrix} 1 + c^2 - a^2 - b^2 & 2\,a - 2\,\mathrm{bc} & 2\,\mathrm{ac} + 2\,b \\ -2\,a - 2\,\mathrm{bc} & -a^2 + 1 + b^2 - c^2 & 2\,c - 2\,\mathrm{ab} \\ 2\,\mathrm{ac} - 2\,b & -2\,c - 2\,\mathrm{ab} & -b^2 - c^2 + 1 + a^2 \end{bmatrix}$$

where $\Delta = 1 + a^2 + b^2 + c^2$.

Conversely, if $\mathcal{R}$ is a rotation then $H = \frac{\mathcal{R}-I}{\mathcal{R}+I}$ is anti-symmetric (again -1 should not be an eigenvalue of $\mathcal{R}$). Expressing relation (2) and removing the denominators, one obtain a system depending on 6 variables $[X, Y, Z, a, b, c]$. In fact, knowing a, b and c it is obvious to recover $[X, Y, Z]$ from a *linear* system. Thus it is enough to compute a Gröbner basis of the corresponding algebraic system for an ordering eliminating $[X, Y, Z]$. As explained before we compute a RR-Form of the lexicographical Gröbner basis (equivalent to a RUR in that case). Isolation and certification of the real coordinates is then computed: we found 40 real roots in approximatively 1.1 sec (PC Intel Xeon 2.8 Ghz).

5.2 Cuspidal robots

We revisit here a ad-hoc computation done in [8]. An extension of this problem (one variable more) can be founded in [9]. The goal was to compute a classification of 3-revolute-jointed manipulators based on the cuspidal behavior. This ability to change posture without meeting a singularity is equivalent to the existence of a point in the workspace, such that a polynomial of degree four depending on the parameters of the manipulator and on the Cartesian coordinates of the effector has a triple root.
The system that characterizes the cuspidal robots depends on 3 parameters d_4, d_3 and r_2 which are the design parameters (supposed to be positive). It is given by:

$$P(t) = at^4 + bt^3 + ct^2 + dt + e = 0,$$
$$\frac{\partial P}{\partial t} = 0, \frac{\partial^2 P}{\partial t^2} = 0, d_4 > 0, d_3 > 0, r_2 > 0$$

with:
$$\begin{cases} a &= m_5 - m_2 + m_0 \\ b &= -2m_3 + 2m_1 \\ c &= -2m_5 + 4m_4 + 2m_0 \\ d &= 2m_3 + 2m_1 \\ e &= m_5 + m_2 + m_0 \\ m_0 &= -r^2 + r_2^2 + \frac{(R+1-L)^2}{4} \\ m_1 &= 2r_2 d_4 + (L - R - 1)d_4 r_2 \\ m_2 &= (L - R - 1)d_4 d_3 \\ m_3 &= 2r_2 d_3 d_4^2 \\ m_4 &= d_4^2(r_2^2 + 1) \\ m_5 &= d_4^2 d_3^2 \\ r^2 &= x^2 + y^2 \\ R &= r^2 + z^2 \\ L &= d_4^2 + d_3^2 + r_2^2 \end{cases}$$

We take $\mathcal{E} = \{P, \frac{\partial P}{\partial t}, \frac{\partial^2 P}{\partial t^2}\}$, $\mathcal{F} = \{d_4, d_3, r_2\}$, $U = [d_4, d_3, r_2]$ and $X = [t, z, r]$. The system has dimension 4 but the only component of dimension 4 is embedded in $\mathrm{V}(d_4) \subset W_{\{\infty}$.
As in most situations, W_∞ is easy to compute. Here, the result is :

- $I_4^\infty = \{1\}, I_5^\infty = \{r2d4 - d3r2 + r2^3 d4\}, I_6^\infty = \{1\}$

Since $< \mathcal{E} > + \mathrm{Jac}_X^{n-d}(\mathcal{E})$ has dimension $< d$ and since the system has 3 equations and depends on 3 parameters, then $W_D = \cup_{i=4\ldots6} \mathrm{V}(I_i^\infty) \cup \mathrm{V}((< \mathcal{E} > + \mathrm{Jac}_X^{n-d}(\mathcal{E})) \cap \mathbb{Q}[U])$, and :

- $I_{\mathrm{crit}} =$
```
{
-d4^2+r2^2+d3^2,
d4^2*r2^6-d4^4*r2^4+2*d4^2*r2^4+3*d4^2*d3^2*r2^4-
2*d4^4*r2^2+d4^2*r2^2-2*d4^4*d3^2*r2^2+
3*d4^2*d3^4*r2^2-d3^2*r2^2-
d4^4*d3^4+d4^2*d3^2+d4^2*d3^6-2*d4^2*d3^4-d4^4+
2*d4^4*d3^2,r2^8+2*d3^2*r2^6+2*r2^6-2*d4^2*r2^6+
d4^4*r2^4-4*d4^2*r2^4-2*d3^2*r2^4-
2*d4^2*d3^2*r2^4+r2^4+d3^4*r2^4-2*d4^2*r2^2+
2*d4^4*r2^2+2*d4^2*d3^2*r2^2+d4^4,
d3^2*r2^2-d4^2+2*d4^2*d3+d3^2-d4^2*d3^2-2*d3^3+d3^4,
d3^2*r2^2-d4^2-2*d4^2*d3+d3^2-d4^2*d3^2+2*d3^3+d3^4
}
```

Removing the polynomials which have no real roots, , one can easily terminate the computations by using a partial CAD and some tools for computing the real roots of a zero-dimensional system. The projection of the discriminant variety on the subspace (d_3, r_2) (obtained after the first partial CAD projection step) appears in following figure.

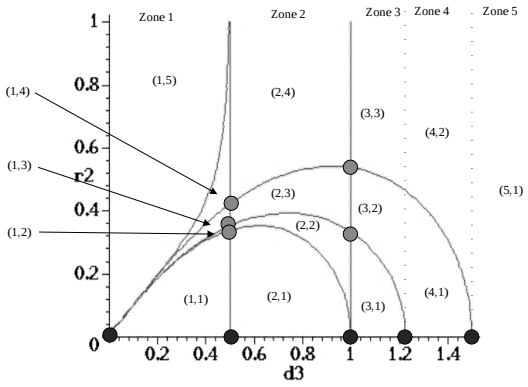

Figure 1: Partition of the parameters'space (d_3, r_2)

Over each open cell, there are exactly six sheets on the discriminant variety, and the following table gives the number of solutions found at a sample point in each of the cells delimited by these sheets (by solving the corresponding zero-dimensional systems):

$(d_3, r_2) \setminus d_4$	1	2	3	4	5	6	7
(1,1)	0	0	4	4	2	0	0
(1,2)	0	4	4	4	2	0	0
(1,3)	0	4	4	4	2	0	0
(1,4)	0	4	4	2	2	0	0
(1,5)	0	4	4	2	0	0	0
(2,1)	0	0	4	4	2	2	0
(2,2)	0	4	4	4	2	2	0
(2,3)	0	4	4	4	2	2	0
(2,4)	0	4	4	2	2	2	0
(3,1)	0	4	4	4	2	2	4
(3,2)	0	4	4	4	2	2	4
(3,3)	0	4	4	2	2	2	4
(4,1)	0	4	4	4	2	2	4
(4,2)	0	4	4	2	2	2	4
(5,1)	0	4	4	2	2	2	4

Table I: Number of real solutions for each cell.

We may consider that the problem is completely solved, even if no precise information is known for parameter's values that belongs to the discriminant variety: it will anyway be impossible to construct, in practice, a robot whose parameters belong to a strict closed subset of the parameter's space.

References

[1] P. Aubry. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques*. PhD thesis, Université Paris 6, France, 1999.

[2] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *Journal of Symbilic Computation*, 28:105–124, 1999.

[3] P. Aubry, F. Rouillier, and M. Safey. Real solving for positive dimensional systems. *Journal of Symbolic Computation*, 34(6):543–560, 2002.

[4] Auzinger and Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. *Int. Series of Numerical Math.*, 86:11–30, 1998.

[5] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computations in Mathematics*. Springer-Verlag, 2003.

[6] B. Buchberger. *Gröbner bases : an algorithmic method in polynomial ideal theory*. Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.

[7] B. Buchberger, G.-E. Collins, and R. Loos. *Computer Algebra Symbolic and Algebraic Computation*. Springer-Verlag, second edition edition, 1982.

[8] S. Corvez and F. Rouillier. Using computer algebra tools to classify serial manipulators. In F. Winkler, editor, *Automated Deduction in Geometry*, volume 2930 of *Lecture Notes in Artificial Intelligence*, pages 31–43. Springer, 2003.

[9] Solen Corvez. *Etude de systèmes polynomiaux : contributions à la classification d'une famille de manipulateurs et au calculs des intersections de courbes A-splines*. PhD thesis, Université de Rennes 1, 2005.

[10] J.-C. Faugère. A new efficient algorithm for computing gröbner bases (f_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[11] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner basis by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, Oct. 1993.

[12] Jean-Charles Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero f_5. In *International Symposium on Symbolic and Algebraic Computation Symposium - ISSAC 2002, Villeneuve d'Ascq, France*, Jul 2002.

[13] D. Lazard. On the specification for solvers of polynomial systems. In *5th Asian Symposium on Computers Mathematics -ASCM 2001*, volume 9 of *Lecture Notes Series in Computing*, pages 66–75. World Scientific, 2001.

[14] D. Lazard and F. Rouillier. Solving parametric polynomial systems. Technical Report RR-5322, INRIA, Oct 2004.

[15] Daniel Lazard. Resolution of polynomial systems. In *4th Asian Symposium on Computer Mathematics - ASCM 2000, Chiang Mai, Thailand*, volume 8 of *Lecture Notes Series on Computing*, pages 1 – 8. World Scientific, Dec 2000.

[16] B. Mourrain. An introduction to linear algebra methods for solving polynomial equations, 1998.

[17] Dietmaier P. The stewart-gough platform of general geometry can have 40 real postures. *Advances in Robot Kinematics: Analysis and Control*, pages 1–10, 1998.

[18] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the mpfi library. *Reliable Computing*, 11:1–16, 2005.

[19] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[20] F. Rouillier and P. Zimmermann. Efficient isolation of polynomial real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2003.

[21] Mohab Safey El Din and Eric Schost. Properness defects of projection functions and computation of at least one point in each connected component of a real algebraic set. *Journal of Discrete and Computational Geometry*, sep 2004.