Solving some overdetermined polynomial systems

Marc Giusti, Éric Schost
UMS MEDICIS, Laboratoire GAGE, École polytechnique
F-91128 Palaiseau, France

Abstract

We propose a strategy to obtain approximate solutions of an overdetermined consistent polynomial system of which we are only given an approximation.

These systems will be supposed to be instantiations of some polynomials $F_1, \ldots, F_{n+1} \in k[\lambda, x]$ on the λ variables, where $k \subset \mathbb{C}$ is an effective field, $x = (x_1, \ldots, x_n)$ are the unknowns and λ is to be seen as a set of parameters.

For an arbitrary choice of λ, this system is generally inconsistent. We first propose hypotheses under which the set of λ where the system is consistent is an hypersurface of the space of parameters. In the second part, we use the algorithm for geometric resolution given in [7] in our particular setting, to give a theoretical polynomial-time resolution algorithm.

Finally, we apply our strategy to the example of an overconstrained parallel manipulator, where an extra measure is adjoined. A resolution is computed in `Magma` that demonstrates the feasibility of the method.

1 Introduction

Many physical phenomenons can be modelized as polynomial systems with parameters, for instance in the fields of robotics or computer vision. Even if the system is overconstrained, physical evidence tells that for some set of parameters, the system has a solution, which is generally unique. As an example, see the case of an overconstrained parallel manipulator with seven measures, whereas this system has six degrees of freedom.

Consider $f = (f_1, \ldots, f_{n+1})$ an overdetermined consistent polynomial system in $\mathbb{C}[x_1, \ldots, x_n]$. It is is supposed to be an instantiation of some polynomials $F_1(\lambda, x), \ldots, F_{n+1}(\lambda, x)$ on an unknown set of parameters λ. If we slightly perturb the parameters, chances are that the new system $\tilde{f}$ becomes generically inconsistent. In many practical situations, we are only given this approximation $\tilde{f}$, the exact set of parameters insuring consistency remains unknown. Typically, noise in a data-acquiring system yields such cases. The issue is thus to recover approximate solutions of the overdetermined consistent system f from the only data of an inconsistent nearby approximation $\tilde{f}$.

Existing methods in this field rely on eigenvectors computations ([2]) or Newton approximations mixed with deformations of the system ([4]).

Our idea is to explicitly work with the polynomials F_i that depend on the parameters $\lambda = (\lambda_1, \ldots, \lambda_p)$ and the unknowns $x = (x_1, \ldots, x_n)$. The parameters lie in the affine space $\mathbb{C}^p$ and x in $\mathbb{C}^n$, so that $F_i \in \mathbb{C}[\lambda_1, \ldots, \lambda_p, x_1, \ldots, x_n]$. We will suppose that the instantiation on $\lambda = 0$ yields the inconsistent system $\tilde{f}$, whereas the "original" parameters λ_0 are unknown. Starting from $\lambda = 0$, we seek a set of parameters such that the system is consistent.

We will work under the following hypotheses :

- the system $F_1(0, x), \ldots, F_n(0, x)$ is a reduced complete intersection, and admits a finite number k of solutions ; the system $F_1(0, x), \ldots, F_{n+1}(0, x)$ defines the empty set. This is the perturbed data.
- there is an unknown set of parameters λ_0 such that $F_1(\lambda_0, x), \ldots, F_{n+1}(\lambda_0, x)$, defines only one point in $\mathbb{C}^n$.
- there is a neighbourhood $\mathcal{V}$ of both 0 and λ_0 in $\mathbb{C}^p$ such that for all λ in $\mathcal{V}$, the system $F_1(\lambda, x), \ldots, F_n(\lambda, x)$ is complete intersection of constant degree k. The zero-set of this system then forms a reduced non-ramified covering of $\mathcal{V}$ of degree k.

In the first part, we show that under these hypotheses, the set of parameters for which the system is consistent is an hypersurface of $\mathbb{C}^p$. This is a purely geometrical description of the situation, and it is followed by a brief discussion on the physical accuracy of this modelization.

With the help of an algorithm for geometric resolution given in [7], we then propose an algorithm that outputs a representation of this hypersurface, and a way to recover solutions above this hypersurface. In the case where the coefficients are rational, this algorithm's complexity is polynomial in some suitably defined quantities. We then propose a strategy to obtain approximate solutions starting from the point $\lambda = 0$.

The last part is devoted to the example of an overconstrained parallel manipulator, for which a resolution is computed with the `Magma` computer algebra system. It is only intended to demonstrate the pertinence of the physical modelization, for we do not yet have a generalist software for geometric resolution.

2 A point of view on approximate consistent systems

In this first part, we give a geometric sketch of the situation we are dealing with. There is no computational consideration here.

For the sake of short writing, we will call Φ_n, Φ_{n+1} the polynomial mappings

$$\begin{array}{cccll} \Phi_n & : & \mathbb{C}^p \times \mathbb{C}^n & \to & \mathbb{C}^n \\ & & (\lambda, x) & \mapsto & F_i(\lambda, x) \qquad i = 1, \ldots, n \\ \Phi_{n+1} & : & \mathbb{C}^p \times \mathbb{C}^n & \to & \mathbb{C}^{n+1} \\ & & (\lambda, x) & \mapsto & F_i(\lambda, x) \qquad i = 1, \ldots, n+1 \end{array}$$

Of course, we cannot say much about the dimension of the varieties $\Phi_n^{-1}(0)$ and $\Phi_{n+1}^{-1}(0)$ in all generality, for there might be places of degeneracy. Thus we have to work locally.

Let π be the projection

$$\begin{array}{cccc} \pi : & \mathbb{C}^p \times \mathbb{C}^n & \to & \mathbb{C}^p \\ & (\lambda, x) & \mapsto & \lambda \end{array}$$

Call Δ the projection $\pi(\Phi_{n+1}^{-1}(0))$ on the space of parameters $\mathbb{C}^p$. This object is our main point of interest. It is to be seen as the locus of consistency : above a point λ of Δ, there is a (generically unique) solution to the system F. The main feature here is the following proposition.

Proposition 1 *There is an open subset $\mathcal{U}$ of $\mathbb{C}^p$ such that the set $\Delta \cap \mathcal{U}$ is of codimension 1 and the restriction of π to $\Phi_{n+1}^{-1}(0)$ is one-to-one above $\Delta \cap \mathcal{U}$.*

Proof. Note first that Δ is generally not an affine algebraic subvariety of $\mathbb{C}^p$, since π needs not be proper. So when speaking of equations describing Δ, we will refer to these of its algebraic Zariski-closure.

Since the system $F_1(0, x), \ldots, F_n(0, x)$ is reduced and defines a complete intersection, the associated jacobian matrix has full rank at every point of the fiber above 0. Then there exists an open subset $\mathcal{U}$ containing 0 in $\mathbb{C}^p$ where this remains valid. From now on, when speaking of Φ_n or Φ_{n+1}, we will refer to their restriction to $\mathcal{U} \times \mathbb{C}^n$.

The fiber of the morphism Φ_n above 0 is a smooth variety, consisting of k irreducible components that form a non-ramified covering of $\mathcal{U}$. By the third assumption, λ_0 can be supposed to belong to this neighbourhood $\mathcal{U}$.

Now let us add the last polynomial F_{n+1}. It cannot vanish identically on any of the previous irreducible sheets, since this would contradict the inconsistency assumption over $\lambda = 0$ contained in the first hypothesis. Then by Krull's theorem, the variety $\Phi_{n+1}^{-1}(0)$ is either empty or of dimension $p - 1$. The first cannot occur, since contrary to the second hypothesis.

The projection π restricted to $\mathcal{U} \times \mathbb{C}^n$ is finite. This implies that the projection of the subvariety $\Phi_{n+1}^{-1}(0) \subset \Phi_n^{-1}(0)$ is a subvariety of the same dimension, that is, $p - 1$.

For the second part of the proposition, we know that the projection is one-to-one on each of the components of the covering. As $\Phi_{n+1}^{-1}(0)$ consists of a single point above λ_0, the result follows, up to the price of perhaps restricting $\mathcal{U}$.

This proposition yields a way of dealing with such systems. The point is to try and reach the hypersurface Δ, with starting point $\lambda = 0$, and to track along the way some values for the unknowns x. What these values should be is not yet clear, for above a parameter λ that is not on Δ, the overconstrained system is inconsistent. We will see later that there is a way to have an estimate for the unknowns x that yields the correct value when the system is consistent.

It must be stressed here that the previous proposition is valid in all generality under the two hypotheses.

Now, if we are given a physical modelization that fulfills these criterions, the quality of the solution we output depends of the quality of the approximation we are given. If 0 is not close enough to Δ, there is no way to guarantee what the output will be.

We can thus reduce the problem of "solving" an inconsistent overdetermined system to the search for a point on an hypersurface for which we have a good initial guess.

The main computational task lies in the elimination process that yields a representation of Δ and $\Phi_{n+1}^{-1}(0)$ above.

The form under which it is given needs not be a list of monomials, all the more as the equation of Δ, as an eliminating polynomial, is bound to show a high complexity in this representation.

The following section exhibits an object that is well suited to our frame, the so-called *geometric resolution* of an affine variety. We will expose an algorithm for geometric resolutions that doesn't rely on the monomial representation.

3 A polynomial-time solving algorithm

This part shows an algorithm of polynomial complexity that outputs a resolution of the system F. The complexity of this algorithm depends on intrinsic arithmetic quantities ; we will then restrict ourselves in this part to polynomials with rational coefficients.

We omit the proofs of the already-published theorems here, they can be found in [7] and [18].

3.1 Geometric resolution of an affine variety

We first recall the definition of a geometric resolution of an affine variety. We forget approximate overconstrained systems for a while, for the notions and results are valid in the general case.

Let $f_1, \ldots, f_s$ be a regular sequence in $\mathbb{Q}[x_1, \ldots, x_n]$, which defines a variety V of dimension $r = n - s$.

The variables $x_1, \ldots, x_r$ are said to be in Noether position with respect to V if the application $\mathbb{Q}[x_1, \ldots x_r] \to \mathbb{Q}[x_1, \ldots, x_n]/(f_1, \ldots, f_s)$ is injective and an integral ring extension. With a geometric point of view, it means that the projection over the first r variables - the *free* variables - is surjective and finite.

The regular sequence $f_1, \ldots, f_s$ is then said to be reduced if there is a change of coordinates $x \mapsto y$ such that for all i in $1, \ldots, s$ the variables $y_1, \ldots, y_{n-i}$ are in Noether position with respect to the system $(f_1, \ldots, f_i)$ and the determinant of the jacobian matrix $J(f_1, \ldots, f_i)$ is non-zero divisor modulo $(f_1, \ldots, f_i)$.

To work locally (outside some $g = 0$) we have the following equivalent geometric condition : $f_1, \ldots, f_s$ form a reduced secant family, i.e. $f_1, \ldots, f_i$ $(i = 1, \ldots, s)$ define a reduced complete intersection of codimension i outside $g = 0$.

It this setting, a geometric resolution consists in :

- a linear change of coordinates $(x_1,\ldots,x_n) \mapsto (y_1,\ldots,y_n)$ such that the polynomials are in Noether position with respect to the new set of variables.
- a *primitive* element u of $\mathrm{Q}[y_1,\ldots y_r] \to \mathrm{Q}[y_1,\ldots,y_n]/(f_1,\ldots,f_s)$, with a minimum polynomial $q_u \in \mathrm{Z}[U]$. A primitive element will also be called *separating*. This is an extension of the definition in the zero-dimensional case, where u is a primitive element if and only if $u(P) \neq u(Q)$ for all P and Q distinct points of V. The degree of q_u is the rank of $\mathrm{Q}[y_1,\ldots,y_n]/(f_1,\ldots,f_s)$ as a $\mathrm{Q}[y_1,\ldots y_r]$-module.

  It is sufficient to consider linear primitive elements.
- a set of parameterizations $\rho_i y_i - v_i(u)$ for $i = r+1,\ldots,n$. The ρ_i (resp. $v_i(u)$) are polynomials in $\mathrm{Z}[y_1,\ldots y_r]$ (resp. $\mathrm{Z}[y_1,\ldots y_r][U]$), and depend on the choice of u. With $\rho = \Pi\rho_i$, the set of points x of V outside $\rho^{-1}(0)$ is given by $q_u(u) = 0, \rho_i y_i - v_i(u) = 0$.

This object is well adapted to our problem. The minimum polynomial q_u of the primitive element is the eliminating object we are looking for. Above a point v that is not a zero of this polynomial, the parametrizations give values for the dependent variables. Suppose that v moves toward v_0, where $q_u(v_0) = 0$. Then the values of the parametrizations tend to the coordinates of the solution above v_0. These values can then be seen as approximations of some solution of the system.

This type of presentation of a variety is first to be found in the work of Kronecker in the end of the 19th century [15]. In the computer algebra community, this idea first appeared under the name "shape lemma" in [6] (see also [3]).

In the zero-dimensional case, the first practical implementation of such ideas is in [21] (see also [22]) where it bears the name Rationate Univariate Representation. These algorithms depend on the precomputation of a Gröbner basis.

Alternatively, a long series of paper has brought a algorithm with better theoretical complexity to compute incrementally a geometric resolution : [8], [14], [10], [9], [7]. The underlying idea is to follow an iterative intersection process instead of the rewriting procedures. This algorithm shows a polynomial complexity in both the *complexity of evaluation* of the input polynomials and some *geometric invariants* of the variety they define. Below, we briefly explain these two features.

3.2 Encoding the polynomials

As has been said, it is easier to obtain a geometric resolution for systems of polynomials with good "evaluation properties". Let's see what this is about.

Consider for instance the square matrix $A = [a_{ij}]$ of size n. Its determinant is a polynomial in the a_{ij}'s having $n!$ monomials. On another hand, there are polynomial-time algorithms that evaluate its value on any instantiation of the a_{ij}'s without using the developed form : a determinant has good evaluation properties.

The point is to think about polynomials as *functions* from C^n to C rather than elements of the vector-space $\mathrm{C}[x]$.

On a practical point of view, this means that we won't stock a polynomial as the list of its coefficients on the monomial basis, but as a function by means of Straight-Line Programs or SLP's. Informally speaking, these are programs that compute the value of a polynomial on any point of the source-space, this space being a power of an effective field k.

We will not recall here precisely the now well-known definitions and theorem about Straight-Line Programs.

One has to know that the size of a SLP is measured by two quantities : its length L and its non-scalar depth ℓ. The first one is simply the number of its nodes ; the non-scalar depth is the length of the longest path in the graph, counting only non-scalar operations.

Equality testing is based on the theorem of Heintz-Schnorr ([13]), we the usual drawback that our algorithms are probabilistic with bounded failure probability. We will also have to differentiate SLP's. Given a SLP of size L that codes a polynomial P, the theorem of Baur and Strassen [1] states that there is a SLP of size $5L$ that computes P and its gradient.

3.3 Geometric invariants

We now turn to the description of some intrinsic combinatorial quantities of the variety defined by a list of polynomials $(f_1,\ldots,f_s)$ in $\mathrm{Q}[x_1,\ldots,x_n]$. We will suppose here that these polynomials form a regular, reduced sequence, their degree being at most d. They are supposed to be coded by Straight-Line Programs of size at most L and non-scalar depth ℓ. Attached to this equidimensional context, we present two invariants, its (affine) degree and its height.

Let's first see the degree of an affine algebraic variety. We will use only the geometric definition which is given in [11], since no multiplicities occur. For a zero-dimensional variety, the degree is equal to the number of points. The general definition is based on this first case. In the worst case, this degree is equal to the classical Bézout's bound d^s, but can be strictly lower.

The degree of the system (f_i) is the maximum of the degrees of the intermediate varieties $V_i = V(f_1,\ldots,f_i)$. It will be noted δ.

The second intrinsic parameter we consider is the height, and is of diophantine nature. It will be noted η. The precise definition is technical, and out of the scope of this paper. It is detailed is [18] and [7].

The definition is based on the definition of the height of an integer $n \in \mathrm{Z}$ as $\max(\log_2(|n|)), 1)$. The process of definition is similar to the degree : first for a zero-dimensional variety, then in the general case. For a zero-dimensional variety, the height is linked to the height of the polynomials in a geometric resolution with integer coefficients. For a positive-dimensional variety $V = V(f_1,\ldots,f_i)$ in Noether position with projection $\pi : V \to \mathrm{C}^{n-i}$, the height is computed from the height of the zero-dimensional varieties $V_a = \pi^{-1}(a)$, for suitable $a \in \mathrm{C}^{n-i}$.

From [14], we have by means of an arithmetic Bézout theorem that η is less than $h\ell d^{An}$, for an universal constant A.

3.4 The geometric resolution algorithm

In this part, we recall a theorem by Giusti, Haegele, Heintz, Montaña, Morais, Pardo which gives an algorithm for geometric resolution, the complexity of which depends on the aforesaid quantities. We will not establish any proof here, but only sketch its behavior.

Theorem 1 *in [7] Let g and $f_1, \ldots, f_s$ in $\mathbb{Q}[x_1, \ldots, x_n]$. Suppose that the polynomials $f_1, \ldots, f_s$ define a regular, reduced sequence in the open set $g \neq 0 \subset \mathbb{C}^n$ and are of degree less than d, height less than h and are coded by Straight-Line Programs of size less than L. There is a bounded error probabilistic Turing machine that outputs a geometric resolution of $V_s = V(f_1, \ldots, f_s)$. The time complexity of the execution is $L(ndh\delta\eta)^{O(1)}$.*

Let's make a few comments about this algorithm.

- The process is incremental : we successively compute geometric resolutions of the varieties $V_i = V(f_1, \ldots, f_i)$. The i^{th} inductive step consists in putting the $i+1^{th}$ polynomial in Noether position and to find a new primitive element together with its minimal polynomial.
- The algorithm could also rely on the precomputation of a correct-test sequence, and thus become deterministic.
- We can test at each step if the system is reduced. If this is not the case, we stop the computation.
- The possibility of local computations outside some hypersurface is likely to lower the affine degree ; for instance, the points at infinity do not alter the complexity of the affine resolution.
- The output polynomials are given as Straight-Line Programs.

This version of the algorithm relies on the notion of *magic point* ; this feature enables the lifting from a zero-dimensional setting to a positive-dimensional one.

Assume that we have reached step i with a geometric resolution of the variety V_{i-1}, and that V_i is in Noether position. Let $\pi_i : V_i \to \mathbb{C}^{n-i}$ be the projection on the first $n-i$ coordinates. With the previous notations, a point $P_i \in \mathbb{Q}^{n-i}$ is a *magic point* for V_i with respect to π_i if the fiber $V_{P_i} = \pi_i^{-1}(P_i)$ has $deg(V_i)$ points.

It can be shown that P_i is such a point if and only if the determinant of the jacobian of the polynomials $f_1, \ldots, f_i$ with respect to the free variables is non-zero above P_i.

These magic points share the property that the knowledge of a geometric resolution of the fiber V_{P_i} is enough to construct a resolution for the whole variety V_i. This is done through a symbolic version of the Newton-Hensel iteration. For a proof, more detailed explanation and extensions, we refer to [18], [7] and [12].

Having this tool in mind, the iterative step of the algorithm splits in two parts :

- Obtain a geometric resolution of V_i from P_i, the Noether normalization of V_i and the geometric resolution of V_{P_i} - this part uses the symbolic Newton-Hensel procedure.
- Normalize V_{i+1}, find a new magic point P_{i+1}, and compute a geometric resolution of the fiber above this point - which is a zero-dimensional problem.

3.5 A polynomial-time existence theorem

Let's get back to overconstrained systems. In this part, we apply the former results to this particular setting. To this effect, we will use the following notations :

The system we study is noted $F_1, \ldots, F_{n+1}$, where $F_i \in \mathbb{Q}[\lambda_1, \ldots, \lambda_p, x_1, \ldots, x_n]$. The degrees of the polynomials are bounded by d, their height by h and they are supposed to be coded by SLP's of size at most L.

Recall that the specialization in $\lambda = 0$ satisfies the hypotheses introduced in the introduction, and that the set of λ such that the system is consistent is an hypersurface of $\mathbb{C}^p$. The objective is to get a representation of the eliminating object Δ and the parametrizations of the unknowns x above a point λ in $\mathbb{C}^p$. This will be done as an application of the theorem 3.4.

The geometric result in the first part is local. We then have to specify in the input an hypersurface $V(g)$ to avoid ; this hypersurface contains the places where degeneracies occur. We will furthermore suppose that the system is regular and reduced out of this surface. As the hypotheses of [7] are now fulfilled, we can apply their result and state the main theoretical theorem of this paper :

Theorem 1 *Let $(F_1, \ldots, F_{n+1}, g) \in \mathbb{Q}[\lambda_1, \ldots, \lambda_p, x_1, \ldots, x_n]$ such that the system F satisfies the two conditions of the introduction. The degrees of these polynomials are bounded by d, their height by h and their are coded by Straight-Line Programs of size at most L. Let V be the variety defined by the system $F_1, \ldots, F_{n+1}$ in $\mathbb{C}^{n+p}$ and call δ and η its affine degree and height. Assume that outside the set $V(g)$ the sequence F_i is regular, reduced.*

There exists a bounded error probabilistic Turing machine that outputs

- *a geometric resolution of the system F consisting of an eliminating polynomial $q(\lambda_p) \in \mathbb{Q}[\lambda_1, \ldots, \lambda_{p-1}][U]$ and the parametrizations $\rho_i x_i = v_i(\lambda_p)$, where $\rho_i \in \mathbb{Q}[\lambda_1, \ldots, \lambda_{p-1}]$ and $v_i \in \mathbb{Q}[\lambda_1, \ldots, \lambda_{p-1}][U]$,*
- *the gradient of q with respect to the variables $\lambda_1, \ldots, \lambda_p$.*

The hypersurface $q^{-1}(0) \subset \mathbb{C}^p$ is locally the consistency locus. The running-time of this algorithm is $L(ndh\delta\eta)^{O(1)}$.

Proof. We apply the theorem of 3.4. The geometric resolution algorithm successively puts the intermediate varieties $V_i = V(F_1, \ldots, F_i)$ in Noether position. As we assume that the zero-set of the n first polynomials is a covering of $\mathbb{C}^p$, it is possible to ensure that the first n changes of variables only touch the variables x.

The last step performs a change of variable in the space of parameters $(\lambda_1, \ldots, \lambda_p) \mapsto (\mu_1, \ldots, \mu_p)$. The output is a polynomial in $\mathbb{Q}[\mu]$ that is monic in μ_1 and the variables x as rational fractions in the variables μ. The application of the inverse change of variables yields an eliminating polynomial q in the variables λ and the parametrizations as rational fractions in the same variables. The theorem of Baur and Strassen then yields the gradient of q in size less than fives times the size of q.

Outside the discriminant locus where a denominator in the parametrizations is zero, the zeroes of q are the points above which the system is consistent, that is the hypersurface Δ.

Our resolution algorithm is now the following :

- Compute a representation q of Δ, its gradient and parametrizations of the unknowns.
- Starting from $\lambda = 0$, perform numerical iterations to move towards Δ.
  Evaluate the unknowns x along the way.

In this sense, this algorithm can be called *semi-numerical* : we first perform a formal elimination, then successively improve the numerical approximations of the solutions.

4 An example : the parallel manipulator

To test the feasibility of the method, we chose the well-known example of the parallel manipulator or Stewart platform. After a brief overview of the problem and its history, we expose the results of our own approach.

The aim of this part is mainly to illustrate our point of view on overdetermined systems, we are not yet ready to compare our software with existing systems.

4.1 The overconstrained parallel manipulator

D. Stewart proposed in 1965 the construction of a parallel manipulator, with a view to design flight simulators. A parallel manipulator consists in :

- two rigid bodies (the platforms) A and B, one of which is moving with respect to the other,
- two set of points $A_1, \ldots, A_6$ (resp. $B_1, \ldots, B_6$) that are fixed with respect to A (resp. B), so that A_i is linked to B_i by a straight leg.

We are interested in the so-called *direct* problem : given the geometries of the two platforms A and B and the lengths of the segments A_iB_i, find the possible positions of B with respect to A. This can be written as a polynomial system, the unknown of which is the displacement taking the coordinate frame of one platform to the other. In the general case, six measures are necessary to obtain a zero-dimensional system.

This has been widely studied. On one hand, numerical algorithms "à la Newton" work as long as the platform is not too tilted, see [17]. On another hand, the computer algebra community has taken interest in the maximum number of positions (real or complex) of a parallel manipulator ([16], [17],[19],[20]). Their works were based on the computation of Gröbner bases, and/or geometric simplifications of the construction. We propose here yet another point of view, adjoining a seventh measure, so as to get back to the framework we gave above.

We have chosen a particular type of platform, which displays more symmetries than the most general case. It is borrowed from S. Egner's modelization in [5], and some real Stewart platforms are built upon this model. In this case, the construction of the platform depends of some parameters : the radii of the circles and the half-angles of adjacent joints. These will be fixed throughout our computations.

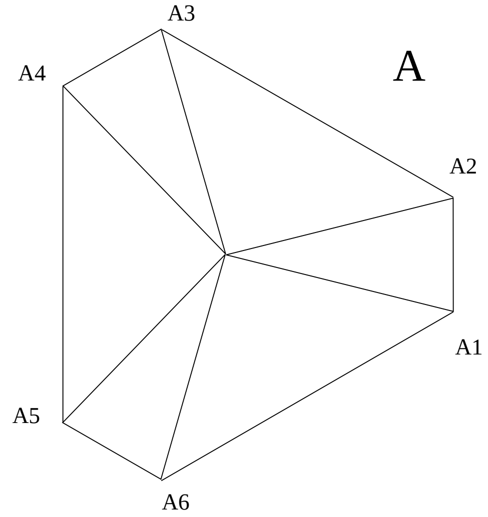

Figure 1: The lower platform

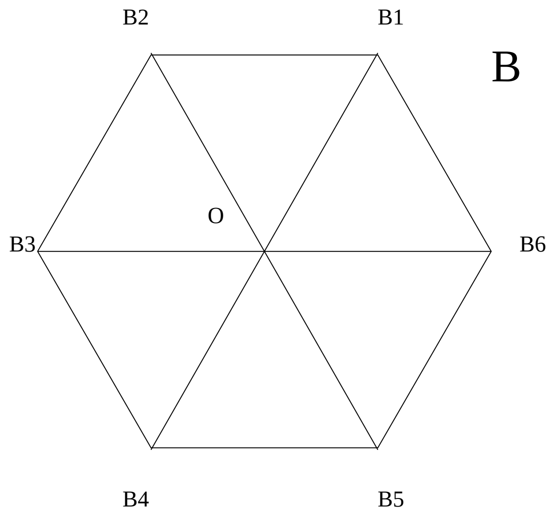

Figure 2: The upper platform

The angles $\widehat{A_2OA_1}$ and $\widehat{B_6O'B_1}$ are respectively of $\pi/6$ and $\pi/3$ rad.

The seventh measure is the height of the point B_1. In the experiments we have led, it was considered as the only parameter : that is, the first six measures were considered as actual constants. This is a simplification of a potentially more complex setting, where every measure could be considered as a parameter.

4.2 Writing down the equations

The system of equations we use is borrowed from S. Egner's article [5]. The idea is straightforward : express the lengths of the six segments and the height of point B_1 as polynomial functions of the input.

The data is the set of $6+1$ measures $\ell_1, \ldots, \ell_6, \ell_7$. Recall that only the last one is a parameter ; the other ones are fixed trough the computation.

The unknown is a displacement $(\mathbf{R}, \mathcal{T})$, where $\mathbf{R} \in SO_3(\mathrm{R})$ and $\mathcal{T} = [x, y, z]$ is a translation.

Rather than a set of variables R_{ij} together with the constraint ${}^t\mathbf{R}\mathbf{R} = \mathbf{I}$, we parameterize the rotation matrix by quaternions. Four real quantities (e_0, e_1, e_2, e_3) such that $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$ yield the rotation matrix

$$\mathbf{R} = \begin{bmatrix} e_0^2+e_1^2-e_2^2-e_3^2 & 2(e_1e_2+e_0e_3) & 2(e_1e_3-e_0e_2) \\ 2(e_1e_2-e_0e_3) & e_0^2-e_1^2+e_2^2-e_3^2 & 2(e_2e_3+e_0e_1) \\ 2(e_1e_3+e_0e_2) & 2(e_2e_3-e_0e_1) & e_0^2-e_1^2-e_2^2+e_3^2 \end{bmatrix}$$

This parametrization has the drawback of introducing second-order terms in the unknowns, but this quantities are highly symmetric. It turns out that the resolution with these variables is easy to compute. To sum up, our variables are $x, y, z, e_0, e_1, e_2, e_3$, and the parametrization adds the equation $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$ to the system.

Let now a_i and b_i be the coordinates of A_i and B_i in the corresponding frame. These quantities are known and depend on the construction parameters. As the half-angles between two consecutive joints are of magnitude $\pi/6$ and $\pi/12$, these coordinates are not rational, but in $\mathrm{Q}(\sqrt{2}, \sqrt{3})$.

We denote by $h(M)$ the third coordinate of point a M. With this convention, the equations are :

$$|\mathbf{R}b_i + \mathcal{T} - a_i|^2 = \ell_i^2 \quad i = 1, \ldots, 6$$
$$e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$$
$$h(\mathbf{R}b_1 + \mathcal{T})^2 = \ell_7^2$$

Some remarks can be made prior to any computation.

- Let's first look at the first six equations. The matrix $\mathbf{R}(e_0, e_1, e_2, e_3)$ is invariant by the transformation $(e_0, e_1, e_2, e_3) \mapsto (-e_0, -e_1, -e_2, -e_3)$, as the parametrization is homogeneous of degree two. This is why we expect an eliminating polynomial involving only even powers of its variable. Then, as the base is in the plane $h = 0$, the knowledge of the lengths isn't enough to determine in what half-space $h > 0$ or $h < 0$ the platform is. As a consequence, the set of solutions is globally invariant by the transformation $(x, y, z, e_0, e_1, e_2, e_3) \mapsto (x, y, -z, -e_0, e_1, e_2, -e_3)$.
- The last equation we have chosen does not eliminate these symmetries, for we have taken squares. It can then be argued that we are not here exactly in the geometric conditions we chose in the preamble.

These equations are in a raw format and we can simplify their writing. Some linear combinations lead to an equivalent system which is much easier to handle. Once again, this part of the work is taken from S. Egner's paper. We obtain the following system :

$$\begin{bmatrix} 4e_0e_3 \\ x - h_{11}(2e_1^2 - 2e_2^2) \\ (e_0^2+e_1^2-e_2^2-e_3^2)x + 2(e_1e_2-e_0e_3)y + 2(e_1e_3+e_0e_2)z - 2h_{12}(e_1^2-e_2^2) \\ y + 4h_{11}e_1e_2 \\ 2(e_1e_2+e_0e_3)x + (e_0^2-e_1^2+e_2^2-e_3^2)y + 2(e_2e_3-e_0e_1)z + 4h_{12}e_1e_2 \\ x^2+y^2+z^2-4g_{10}(e_0^2-e_3^2)+4g_0 \end{bmatrix}$$

$$= \frac{1}{12\sqrt{3}g_1} M \begin{bmatrix} \ell_1^2 \\ \ell_3^2 \\ \ell_4^2 \\ \ell_5^2 \\ \ell_6^2 \\ \ell_2^2 \end{bmatrix}$$

$$e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$$

$$(10(\sqrt{3}(e_1e_3 + e_0e_2) + e_2e_3 - e_0e_1) + z)^2 = \ell_7^2$$

The matrix M is a 6×6 matrix with entries in $\mathrm{Q}(\sqrt{2}, \sqrt{3})$:

$$\begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 \\ -5 & 10 & -5 & -5 & 10 & -5 \\ -5/2\alpha & 5/2\beta & -5\sqrt{2} & -5\sqrt{2} & 5/2\beta & -5/2\beta \\ 5\sqrt{3} & 0 & -5\sqrt{3} & 5\sqrt{3} & 0 & -5\sqrt{3} \\ 5/2\beta & 5/2\alpha & -5\sqrt{2} & 5\sqrt{2} & -5/2\beta & 5/2\alpha \\ 25\alpha & 25\alpha & 25\alpha & 25\alpha & 25\alpha & 25\alpha \end{bmatrix}$$

$$\alpha = \sqrt{6} - \sqrt{2}$$
$$\beta = \sqrt{6} + \sqrt{2}$$

The parameters g_i are :

$$\begin{cases} g_0 & = & 50 \\ g_1 & = & 25/2\,\sqrt{2} - 25/6\,\sqrt{6} \\ g_{10} & = & 25/2\,\sqrt{6} + 25/2\,\sqrt{2} \\ g_{11} & = & 125/2\,\sqrt{2} - 125/6\,\sqrt{6} \\ g_{12} & = & -125 \\ h_{11} & = & g_{11}/g_1 \\ h_{12} & = & g_{12}/g_1 \end{cases}$$

Let's repeat that these value are due to the geometry of the platform that involves angles of $\pi/6$ and $\pi/12$. The set of equations will from now on be called $F_1, \ldots, F_8$.

Geometrically speaking, the space of parameters is C, above which the space of variables is C^7. As the first 7 equations do not involve the parameter, they define a cylinder in $\mathrm{C} \times \mathrm{C}^7$, which is one-dimensional if the 6 lengths are generic enough. The last hypersurface cuts this cylinder on a finite set of points that projects on $\Delta \subset \mathrm{C}$. The seventh measure ℓ_7^{init} yields an initial guess ; we have to find the point of Δ closest to ℓ_7^{init}.

4.3 Computing the geometric resolution

The experiments were based on an home-made cardboard Stewart platform, for which we measured the seven lengths. Expressed in centimers, here is the available data :

$$\begin{array}{lclclcl} \ell_1 & = & 17.9 & & \ell_4 & = & 13.4 \\ \ell_2 & = & 13.5 & & \ell_5 & = & 15.1 \\ \ell_3 & = & 12.05 & & \ell_6 & = & 18.2 \\ \ell_7^{init} & \simeq & 16.7 & & & & \end{array}$$

This data is converted to rational expressions. The precision of measure is about 1/2 mm for the first six lengths. The last one is less precise ; the precision is only of 1 mm.

Next follows a description of a resolution of this system. It is intended to show the soundness of our approach on overdetermined systems, rather than be a comparison between two computational methods, for we cannot compare between a general-purpose software and ad-hoc procedures.

As we are actually in a zero-dimensional situation, a Gröbner basis computation for a pure lexicographic order on the zero-dimensional system of eight equations will give a parametrization of the unknowns, if the coordinates are generic enough. We could also obtain a geometric resolution from the multiplication table of the quotient algebra as was said in part 3.1 using F. Rouillier's work. Unfortunately, we didn't work this possibility out.

On another hand lies the geometric resolution algorithm we have presented. The implementation by G. Lecerf of this algorithm is a work in progress at Lab. Gage so that we are not able to plug the overconstrained system in a procedure and examine the result.

As a consequence, we computed the resolution using `Magma`-procedures that follow the geometric resolution algorithm but do not make use of the Hensel lifting tool. This approach is similar to the algorithm proposed in [10], see there section 4.1 that describes the inductive step. The main computational task consists in evaluating the new equation on the previous parametrizations and then computing the resultant with the previous eliminating polynomial. The implementation uses an idea due to Kronecker which yields both the resultant and some parametrizations in a single step through a generic linear change of variables.

We did not represent all the polynomials as actual Straight-Line Programs. Still, the complexity of evaluation is crucial in the execution-time, for we evaluate equations on the available parametrizations. We then adopted a mixed approach : the substitutions are performed on polynomials coded by SLP's.

It must be stressed that we did not implement here any general-purpose resolution algorithm : our implementation is devoted to the particular case of the parallel manipulator. The order in which we perform the computations, some linear simplifications, as well as the SLP's we use are part of the code itself. The comparison between the running-times of the two approaches then has little meaning.

Both computations were done on a DEC Alpha EV56 400 Mhz with the software `Magma` on the UMS MEDICIS cluster of servers. Here follows a brief description of the experiments. To simplify the writing, we will abusively call *height* of a polynomial in $Q(\sqrt{2}, \sqrt{3})$ the maximum number of digits of its coefficients, written on the form $n_0 + n_1\sqrt{2} + n_2\sqrt{3} + n_3\sqrt{6}$, $n_i \in Q$.

- The first approach was the Gröbner basis computation for the whole system of eight equations for an eliminating order. This process took 95 minutes. It yields an eliminating polynomial p that is of degree 28 and involves even powers of l_7 only. The coefficients have an height of about 2000 digits.

  For any other variable, there is in the basis a monic equation linear in this variable or in the square of this variable. This is a consequence of the symmetry we have mentioned : some signs of variables cannot be determined above the roots of the eliminating polynomial. The coefficients of these equations have an height of about 65000 digits.

  The hypersurface Δ is the finite set of the roots of the polynomial p in C. We expect to find a root of p at approximately 16.7. The root of p closest to 16.7 is app. 16.48. It is the only zero of p in a radius of 9 around 16.7, so that no doubt is allowed regarding the choice of the correct answer. The error is about 2 mm, which is of the order of magnitude of the input precision. We lift the values of the other indeterminates using the other polynomials in the base.

- The second process we followed is similar to the geometric resolution algorithm presented earlier, with the difference that we do not use the Hensel lifting feature.

  Recall that only the last measure is considered a parameter, whereas the other ones are fixed. This last measure does not appear in the first equations. In this sense, the first six eliminations are equivalent to the resolution of the $6-6$ Stewart platform problem. Contrary to the Gröbner basis computation, we have natural access to this intermediate result as a by-product.

  We successively eliminate the variables x, y, z, e_0, e_3 and e_1 using the first six equations, keeping track of the parametrizations of each variable with respect to the remaining ones.

  As a consequence of the symmetry we have mentioned, the elimination process only gives the parametrization of e_3^2 ; this is the same phenomenon as in the Gröbner basis computation.

  We are left with an eliminating polynomial $q(e_2)$ of degree 28, which has an height of about 200. This polynomial only involves even powers of e_2, which too results from the symmetries. The parametrizations have approximately the same number of digits.

  This part takes 13 minutes to compute. Note that a Gröbner basis computation for this sub-system takes 19 minutes and has coefficients of about 16000 digits.

  Up to now, we have left the last equation F_8 untouched. Substituting the parametrizations modulo $q(e_2)$ in F_8 yields a polynomial of the form $n(e_2) - d(e_2)\ell_7^2$. The two equations only involve even powers of e_2, so that their resultant is the square of the polynomial p of degree 28 computed earlier.

  The substitution of the parametrizations in F_8 takes 21 minutes. The importance of a good complexity of evaluation arises here naturally : the time necessary to perform the substitution is proportional to the complexity of evaluation of the equation.

  There is no need to explicitly compute the polynomial p ; we are only interested in finding a numerical approximation of its roots, so that a program evaluating it and the parametrizations on any value of ℓ_7 is enough. The last elimination can thus be done on floating-point numbers. The quality of the solution is then a function of the precision we use, this phenomenon being characteristic of any numerical algorithm. We automatically fixed a decent minimal precision by successive refinements.

  With 1500 significant digits, the computation of the resultant and its derivative on a value of ℓ_7 takes 2 seconds. We use a Newton iteration that converges in 3 iterations from our estimate 16.7 to app. 16.48. We then lift estimates of the variables trough the parametrizations - with only the sign condition to fix.

We finish this section with the display of the solution above $\ell_7 = 16.48$:

$$\mathbf{R} \simeq \begin{bmatrix} 0.95 & 0.012 & -0.32 \\ 0.016 & 0.99 & 0.083 \\ 0.32 & -0.064 & 0.94 \end{bmatrix},$$

$$\mathcal{T} \simeq [-4.1, 1.8, 14].$$

5 Conclusion

In this paper, we present an original approach on overconstrained systems with parameters, that finally amounts to looking for a point on an hypersurface close to an initial estimate. This hypersurface is the projection of the set of solutions of the system on the parameter space ; it is the locus of consistency.

The resolution algorithm yields a representation of this hypersurface and the parametrizations above it by means of Straight-Line Programs. This representation is then used to perform numerical iterations, that give approximations of solutions.

Our first example of the parallel manipulator shows that this approach is feasible in an elementary case. Starting from an approximation of the seventh measure and computing a geometric resolution of the system, we can determine the unique solution to the system with no ambiguity. This process mixes both exact and numerical computations.

References

[1] Baur, W., and Strassen, V. The complexity of partial derivatives. *Theo. Comp. Sci. 22* (1983), 317–330.

[2] Bondyfalat, D., Mourrain, B., and Pan, V. Controlled iterative methods for solving polynomial systems. In *ISSAC'98* (1998), ACM Press., pp. 252–259.

[3] Chistov, A., and Grigoriev, D. Subexponential time solving systems of algebraic equations. LOMI preprint, 1983.

[4] Dedieu, J., and Shub, M. Newton and predictor-corrector methods for overdetermined systems of equations. Tech. rep., IBM Research Division, 1998.

[5] Egner, S. Semi-numerical solution to 6/6-Stewart platform kinematics based on symmetry. *Applicable Algebra in Engineering, Communication and Computing 7* (1996), 449–468.

[6] Gianni, P., and Mora, T. Algebraic solution of systems of polynomials equations. In *AAECC-5* (1989), Springer, pp. 2470–257.

[7] Giusti, M., Haegele, K., Heintz, J., Montana, J., Morais, J., and Pardo, L. Lower bounds for diophantine approximations. In *Journal of Pure and App. Algebra (Proceedings of MEGA'96)* (1997), vol. 117, pp. 277–317.

[8] Giusti, M., and Heintz, J. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In *Computational Algebraic Geometry and Commutative Algebra* (1993), E. D. and R. L., Eds., vol. XXXIV of *Symposia Matematica*, Cambridge University Press, pp. 216–256.

[9] Giusti, M., Heintz, J., Morais, J., Morgenstern, J., and Pardo, L. Straight-line programs in geometric elimination theory. *Journal of Pure and App. Algebra 124* (1998), 101–146.

[10] Giusti, M., Heintz, J., Morais, J., and Pardo, L. When polynomial equations can be "solved" fast ? In *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings AAECC-11* (1995), C. G., G. M., and M. T., Eds., vol. 948 of *LNCS*, Springer, pp. 205–231.

[11] Heintz, J. Definability and fast quantifier elimination in algebraically closed fields. *Theor. Comput. Sci. 24*, 3 (1983), 239–277.

[12] Heintz, J., Krick, T., Puddu, S., Sabia, J., and Waissbein, A. Deformation techniques for efficient polynomial equation solving. Submitted.

[13] Heintz, J., and Schnorr, C. Testing polynomials which are easy to compute. In *Logic and Algorithmic*, vol. 30 of *Monographie de l'enseignement mathématique*. 1980, pp. 237–254.

[14] Krick, T., and Pardo, L. A computational method for diophantine approximation. In *Algorithms in Algebraic Geometry and Applications (Proceedings MEGA'94)* (1996), G. V. L. and R. T., Eds., vol. 143 of *Progress in Mathematics*, Birkäuser Verlag, pp. 193–254.

[15] Kronecker, L. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *Journal für die reine und andewandte Mathematik* (1882).

[16] Lazard, D. Stewart platform and Gröbner bases. In *Proceedings of the third international workshop on advances of robot kinematics* (1992), P.-C. V. and L. J., Eds., Ferrare, pp. 136–142.

[17] Merlet, J. Parallel manipulators - 3rd part. research rep. no. 1003. Tech. rep., INRIA Sophia-Antipolis, 1988.

[18] Morais, J. *Resolución eficaz de sistemas de ecuaciones polinomiales.* PhD thesis, University of Santander, Spain, 1998.

[19] Mourrain, B. Enumeration problems in Geometry, Robotics and Vision. In *Algorithms in Algebraic Geometry and Applications* (1996), L. González and T. Recio, Eds., vol. 143 of *Prog. in Math.*, Birkhäuser, Basel, pp. 285–306.

[20] Ronga, F., and Vust, T. Stewart platforms without computer ? In *International Conference on Real Analytic and Algebraic Geometry* (1995), F. Broglia, M. Galbiati, and A. Tognoli, Eds., W. de Gruyter.

[21] Rouillier, F. *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux.* PhD thesis, Université Rennes I, 1996.

[22] Rouillier, F. Solving zero-dimensional polynomial systems trough the rational univariate representation. Tech. Rep. 3426, INRIA, Projet Polka, 1998.