

On the Computation of the Direct Kinematics of Parallel Spherical Mechanisms Using Bernstein Polynomials

Carlos Bombín, Lluís Ros, and Federico Thomas

Institut de Robòtica i Informàtica Industrial (CSIC – UPC)
Gran Capità 2-4, 2 planta, 08034 Barcelona. SPAIN
{cbombin, llros, fthomas}@iri.upc.es

Abstract

Solving the direct kinematics of parallel spherical mechanisms with l legs boils down to solving systems of $l-1$ second-order multinomials. This paper presents a recurrent expression for the control points of these multinomials when expressed in Bernstein form. This result allows us to propose a technique for solving the direct kinematics of these mechanisms that takes advantage of the subdivision and convex hull properties of polynomials in Bernstein form. Contrary to other numerical approaches, the one presented here is clearly less involved and, although it can be classified within the same category as interval-based techniques, it does not require any interval arithmetic computation.

1 Introduction

Parallel manipulators are closed-chain mechanisms with one or more loops where only a certain number of pairs are actively controlled. Fully parallel mechanisms, in particular, feature two rigid bodies, termed *base* and *platform*, connected by a set of legs. Position analysis of a parallel manipulator involves a direct and an inverse kinematic problem. In general, the inverse problem is trivial, since it asks for the legs' configurations when the position and orientation of the platform are given with respect to the base. On the contrary, the direct problem, which calls for the position and orientation of the platform when the configurations of the actively controlled pairs are given, is a difficult problem for which no general procedure has been found yet and for which closed-form solutions are only available for certain architectures, sometimes satisfying a number of geometric conditions [8].

The different architectures for parallel mechanisms can be analytically studied by a customized strategy

that can be summarized as follows. First, the configuration of the platform with respect to the base is parameterized, so that a closure system with a reduced number of equations and unknowns can be written. Second, using a suitable elimination procedure a final polynomial equation in only one unknown is obtained. Unfortunately, both steps heavily rely on the geometric intuition of the researcher. The roots of the final polynomial lead to the sought solutions by substitution. Consequently, the order of the polynomial equation does not necessarily represent the number of solutions in the real field and the real roots do not necessarily correspond to configurations within the mechanical limits of the mechanism under consideration. However, despite these drawbacks, this has been the usual approach to the problem and it has triggered the hunt for the lowest order polynomial associated with each architecture.

In this context, current numerical methods have been laid aside because of their proved difficulty to find all solutions. Our group has been working on the application of interval-based techniques to provide a way around this difficulty. In particular, we have applied interval extensions of Newton methods, coupled with bisection to ensure convergence, to solve the inverse kinematic problem of serial manipulators directly from their Denavit-Hartenberg parameters [1, 2]. The rather difficult generalization of our algorithm to multiple loops and its complex implementation — among other drawbacks — led us to explore other alternatives. Here we develop a general technique to numerically solve systems of spherical kinematic constraints directly from the Denavit-Hartenberg parameters of the involved kinematic chains that relies on the properties of polynomials in Bernstein form.

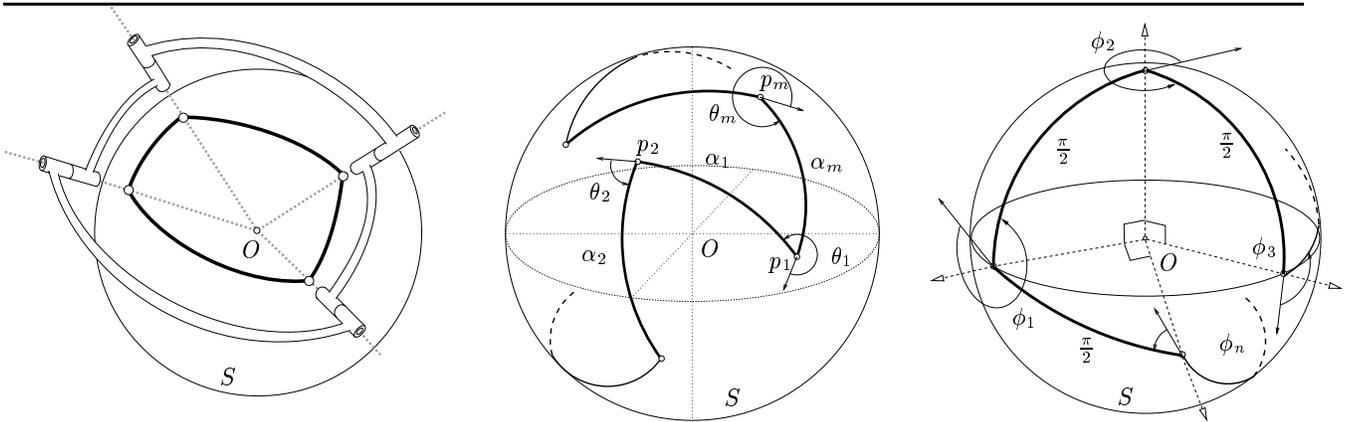


Figure 1: Left: The spherical polygon associated with a spherical 4-bar linkage. Center and right: a spherical polygon before and after normalization.

We begin by introducing, in Section 2, some basic concepts and notations, upon which a standard formulation of spherical problems is developed. Section 3 states our main theoretical results that provide the basis for the application of a subdivision-minimization strategy, described in Section 4, to solve the direct kinematics problem on arbitrary parallel spherical mechanisms. We conclude in Section 5.

2 Spherical polygons and closure equations

A mechanism is called *spherical* if each of its links rotates about a same fixed point, say O . Thus, trajectories of points in each link lie on concentric spheres with O as the center. Only the revolute joint is compatible with this movement and its axis must pass through the fixed point. For convenience, we may think that all revolute joints lie on the same sphere S , and that the links are portions of great circles on S (fig. 1, left). This is how every closed chain of a parallel spherical mechanism defines a spherical polygon on S . For each of these polygons we can derive a closure equation, as explained next.

If we assign a circulating direction to the sides of the polygon, the exterior angle between two adjacent sides is defined as the angle measured from the prolongation beyond the common vertex of the first side to the second side. Next, consider a spherical polygon with vertices p_1, \dots, p_m , sides $\alpha_1, \dots, \alpha_m$, and exterior angles $\theta_1, \dots, \theta_m$ (fig. 1, center). For each side, we define its pole as the point of the sphere lying on a line through the center O , perpendicular to the plane of the side. Now, take a system of coordinate axes centered at O with the x -axis passing through p_1 and

its z -axis passing through the pole of side α_1 . A rotation of α_1 radians about the z -axis will move the x -axis along the side α_1 till vertex p_2 . Next, a rotation of angle θ_2 about the new x -axis will make the z -axis pass through the pole of α_2 . Going on in this way all around the polygon, the x -axis will return to p_1 and the coordinate system will end up being in its original position. This may be expressed by means of a rotation equation stating that the composition of the successive rotations equals the identity transformation, that is,

$$\prod_{i=1}^m \mathbf{R}_x(\theta_i) \mathbf{R}_z(\alpha_i) = \mathbf{I}, \quad (1)$$

where $\mathbf{R}_x(\cdot)$ and $\mathbf{R}_z(\cdot)$ stand for 3×3 orthogonal matrices representing rotations about the x and z axes, respectively, in the amounts given by their arguments. Alternatively to this formulation, we can write

$$\prod_{i=1}^m \mathbf{R}_x(\theta_i + \pi) \mathbf{R}_z(\pi/2) \mathbf{R}_x(\alpha_i + \pi) \mathbf{R}_z(\pi/2) = \mathbf{I},$$

which corresponds to the closure equation of a spherical polygon where all its sides have length $\pi/2$, thus involving variable rotations about the same axis (fig. 1, right). This normalization will simplify our further algebraic manipulations. Then, Equation (1) can be rewritten as:

$$\mathbf{F}_n(\Phi) = \prod_{i=1}^n \mathbf{R}_x(\phi_i) \mathbf{Z} = \mathbf{I}, \quad (2)$$

where $\mathbf{Z} = \mathbf{R}_z(\pi/2)$, $n = 2m$, $\Phi = (\phi_1, \dots, \phi_n)$, and

$$\phi_i = \begin{cases} \theta_{\frac{i+1}{2}} + \pi, & i \text{ even,} \\ \alpha_{\frac{i}{2}} + \pi, & i \text{ odd.} \end{cases}$$

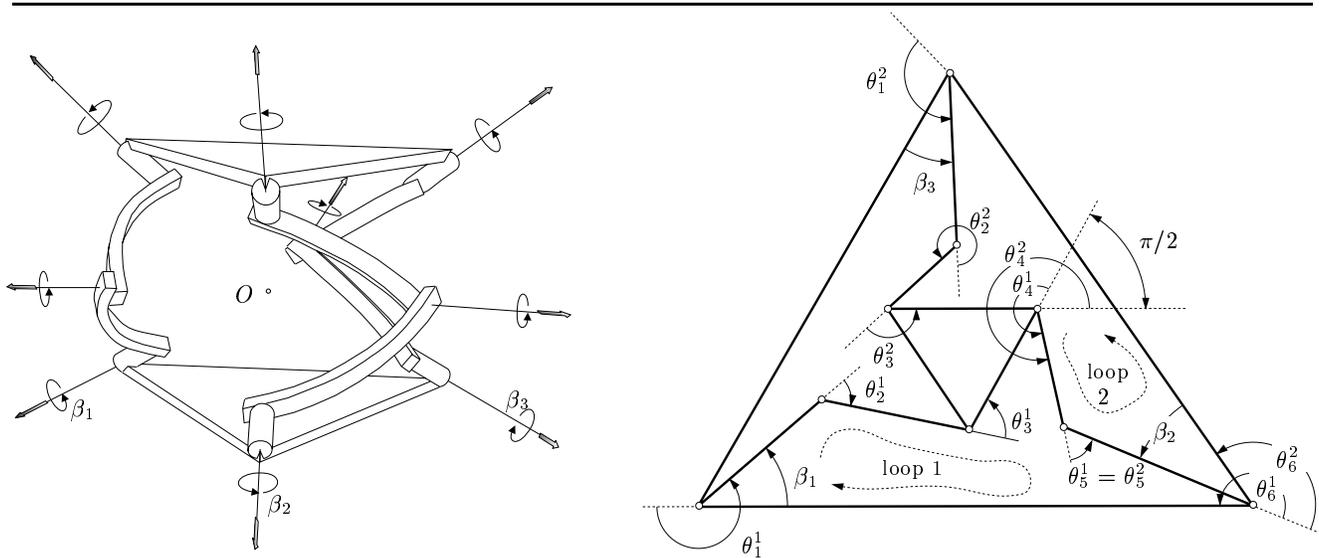


Figure 2: The Gosselin platform (left), and the spherical polygons it defines (right). While the β_i 's are actively controlled, all other variables are passive revolute joints.

The solution set to $\mathbf{F}_n(\Phi) = \mathbf{I}$, a system of nine trigonometric polynomials in n variables, has been geometrically and topologically characterized in [3].

2.1 An example

Consider the spherical parallel manipulator shown in *fig. 2*, sometimes called the *Gosselin platform*. It consists of a platform connected to a fixed base via three kinematic chains, each composed of two intermediate links and three revolute joints. Only the revolute joints connected to the base are actuated. When the link angles and the angles between the revolute axes on the base and on the platform are all set to $\pi/2$, a special geometry is obtained for which a closed-form solution has been derived [6]. It has been shown that the direct kinematic problem of three-degree-of-freedom parallel manipulators has a maximum of 8 solutions [5, Section 3]. The closed-form solution of the Gosselin platform accounts for 8 solutions, as expected, but 4 of them are singular configurations in which the actuators can be moved arbitrarily without affecting the pose of the platform. In practice, such singular configurations should not be inside the workspace of the manipulator, which can be accomplished by limiting the range of motion of the active joints.

Two equations are sufficient to study the kinematics of this manipulator, corresponding to the two loops indicated in *fig. 2*, right. With the indicated angles,

these equations are:

$$\prod_{i=1}^6 \mathbf{R}\mathbf{x}(\theta_i^1) \mathbf{Z} = \mathbf{I}, \quad \text{for loop 1}, \quad (3)$$

$$\prod_{i=1}^6 \mathbf{R}\mathbf{x}(\theta_i^2) \mathbf{Z} = \mathbf{I}, \quad \text{for loop 2}. \quad (4)$$

Note that, due to the special geometry of the manipulator, these are already in the normal form of Equation (2). The angles of the actuators directly determine four variables in the previous equations, because $\theta_1^1 = \pi + \beta_1$, $\theta_6^1 = \frac{\pi}{2} - \beta_2$, $\theta_6^2 = \pi - \beta_2$ and $\theta_4^2 = \frac{3\pi}{2} + \beta_3$. Moreover, from *fig. 2* we see that two variables in one loop are related to others in the second:

$$\theta_4^2 = \theta_4^1 + \pi/2, \quad \theta_5^2 = \theta_5^1. \quad (5)$$

Thus, substituting (5) in Equations (3) and (4), and using the fact that

$$\mathbf{R}\mathbf{x}(\phi_1 + \phi_2) = \mathbf{R}\mathbf{x}(\phi_2 + \pi) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\pi) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\phi_1), \quad (6)$$

we finally get two equations involving six variables,

$$\begin{aligned} & \mathbf{R}\mathbf{x}(\pi + \beta_1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_2^1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_3^1) \mathbf{Z} \cdot \\ & \cdot \mathbf{R}\mathbf{x}(\theta_4^1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_5^1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\frac{\pi}{2} - \beta_2) \mathbf{Z} = \mathbf{I}, \\ & \mathbf{R}\mathbf{x}(\pi - \beta_2) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_2^2) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_3^2) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(-\frac{\pi}{2}) \mathbf{Z} \cdot \\ & \cdot \mathbf{R}\mathbf{x}(\pi) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_4^1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\theta_5^1) \mathbf{Z} \cdot \mathbf{R}\mathbf{x}(\frac{3\pi}{2} + \beta_3) \mathbf{Z} = \mathbf{I}. \end{aligned}$$

2.2 Reduction to scalar equations

Due to the fact that $\mathbf{F}_n(\Phi) = [f_{ij}^n(\Phi)]_{1 \leq i, j \leq 3}$ is a product of orthogonal matrices equated to the identity, it can be reduced to the following single trigonometric equation:

$$f_{11}^n(\Phi) + f_{22}^n(\Phi) - 2 = 0. \quad (7)$$

Now, $\overline{f_{ij}^n}(\Phi)$ can be converted into a rational polynomial $\overline{f_{ij}^n}(\mathbf{t})$ in a new variable, $\mathbf{t} = (t_1, \dots, t_n)$, using the tangent-half-angle substitution, that is, by introducing the substitutions $\sin(\phi_i) = \frac{2t_i}{1+t_i^2}$ and $\cos(\phi_i) = \frac{1-t_i^2}{1+t_i^2}$. Then, if we multiply the resulting rational polynomials by $q_n(\mathbf{t}) = \prod_{i=1}^n (1+t_i^2)$, we obtain the polynomials $\overline{f_{ij}^n}(\mathbf{t}) = q_n(\mathbf{t}) \overline{f_{ij}^n}(\Phi)$ (we adhere to the notation introduced in [7]). Therefore, Equation (2) can be expressed as:

$$\overline{f_{11}^n}(\mathbf{t}) + \overline{f_{22}^n}(\mathbf{t}) - 2q_n(\mathbf{t}) = 0. \quad (8)$$

For an arbitrary function in n variables, say $g(\mathbf{t}) = g(t_1, \dots, t_n)$, we define $g^\sigma(\mathbf{t}) = g(\mathbf{t}^\sigma) = g(t_n, t_1, \dots, t_{n-1})$. Then, if we denote $\mathbf{F}_n^\sigma(\Phi) = [f_{ij}^{n\sigma}(\Phi)]_{1 \leq i, j \leq 3}$, we have that

$$\mathbf{F}_n^\sigma(\Phi) = (\mathbf{R}\mathbf{x}(\phi_n)\mathbf{Z}) \mathbf{F}_n(\Phi) (\mathbf{R}\mathbf{x}(\phi_n)\mathbf{Z})^t.$$

Now, the element (1,1) of \mathbf{F}_n^σ , i.e. $f_{11}^{n\sigma}(\Phi)$, is equal to the element (1,1) of the product $(\mathbf{R}\mathbf{x}(\phi_n)\mathbf{Z}) \mathbf{F}_n(\Phi) (\mathbf{R}\mathbf{x}(\phi_n)\mathbf{Z})^t$, that can be checked to be $f_{22}^n(\Phi)$. In other words, Equation (8) can be rewritten as:

$$\overline{f_{11}^n}(\mathbf{t}) + \overline{f_{11}^n}(\mathbf{t}^\sigma) - 2q_n(\mathbf{t}) = 0. \quad (9)$$

Once the ranges of motions for the passive joints are given, the singularity of the tangent-half-angle substitution at π can be avoided by shifting them a given amount using the relation (6), provided that these ranges are lower than 2π .

3 The closure equations in Bernstein form

Let $M = (m_1, \dots, m_n)$ and $I = (i_1, \dots, i_n)$. The notation $0 \leq I \leq M$ indicates that $0 \leq i_k \leq m_k$, for $k = 1, \dots, n$. It is well known that $\mathbb{R}_M[\mathbf{t}]$, the set of polynomials in the variables t_1, \dots, t_n of degree $\leq m_i$ in t_i , is a vector space. Also, the Bernstein multinomials $\{B_{I,M}(\mathbf{t})\}_{0 \leq I \leq M}$ defined as $B_{I,M}(\mathbf{t}) = b_{i_1, m_1}(t_1) \cdots b_{i_n, m_n}(t_n)$ (where $b_{i,m}$ denotes the i th Bernstein polynomial of degree m) forms a basis of

$\mathbb{R}_M[\mathbf{t}]$, called the multivariate Bernstein basis. Therefore, any polynomial $f(\mathbf{t}) \in \mathbb{R}_M[\mathbf{t}]$ can be written as $f(\mathbf{t}) = \sum_{I=0}^M c_I(f) B_{I,M}(\mathbf{t})$. (For simplicity, we write $B_I(\mathbf{t})$ instead of $B_{I,M}(\mathbf{t})$.) This expression is the Bernstein form of $f(\mathbf{t})$ and the coefficients $c_I(f)$ are called its *control points*.

In our case, we are interested in the Bernstein form of the polynomial $f = \overline{f_{11}^n} + \overline{f_{11}^n}^\sigma - 2q_n \in \mathbb{R}_M[\mathbf{t}]$, where $M = (2, \dots, 2) \in \mathbb{R}^n$. It is easy to prove that $c_I(f) = c_I(\overline{f_{11}^n}) + c_I(\overline{f_{11}^n}^\sigma) - 2c_I(q_n)$.

Proposition 1. *The control points of $q_n(\mathbf{t})$ are $c_I(q_n) = 2^{\chi(I)}$, where $\chi(I)$ is the number of elements of I equal to 2.*

Proof. It can be checked that $q(t_i) = b_{0,2}(t_i) + b_{1,2}(t_i) + 2b_{2,2}(t_i)$, therefore the polynomial q_n can be expressed as: $q_n(\mathbf{t}) = \prod_{i=1}^n [b_{0,2}(t_i) + b_{1,2}(t_i) + 2b_{2,2}(t_i)] = \sum_{I=0}^M 2^{\chi(I)} B_I(\mathbf{t})$. \square

Since it can be easily shown to be $c_I(\overline{f_{11}^n}^\sigma) = c_{I^\sigma}(\overline{f_{11}^n})$, where $I^\sigma = (i_n, i_1, \dots, i_{n-1})$, we only have to calculate the control points $c_I(\overline{f_{11}^n})$, but first we need the following proposition:

Proposition 2. *f_{11}^n and f_{13}^n satisfy the recursion:*

$$\begin{aligned} f_{11}(\phi_1, \dots, \phi_n) &= -f_{11}^{n-2}(\phi_1, \dots, \phi_{n-2}) \cos(\phi_n) + \\ &\quad + f_{13}^{n-1}(\phi_1, \dots, \phi_{n-1}) \sin(\phi_n) \\ f_{13}(\phi_1, \dots, \phi_n) &= f_{11}^{n-2}(\phi_1, \dots, \phi_{n-2}) \sin(\phi_n) + \\ &\quad + f_{13}^{n-1}(\phi_1, \dots, \phi_{n-1}) \cos(\phi_n). \end{aligned}$$

Proof. The proof can be carried out by elementary algebraic manipulations of the matrix products in (2). \square

Corollary 1. *$\overline{f_{11}^n}$ and $\overline{f_{13}^n}$ satisfy the recursion:*

$$\begin{aligned} \overline{f_{11}^n}(t_1, \dots, t_n) &= \overline{f_{11}^{n-2}}(t_1, \dots, t_{n-2}) g_1(t_{n-1}, t_n) + \\ &\quad + \overline{f_{13}^{n-1}}(t_1, \dots, t_{n-1}) h_1(t_n) \\ \overline{f_{13}^n}(t_1, \dots, t_n) &= \overline{f_{11}^{n-2}}(t_1, \dots, t_{n-2}) g_2(t_{n-1}, t_n) - \\ &\quad - \overline{f_{13}^{n-1}}(t_1, \dots, t_{n-1}) h_2(t_n), \end{aligned}$$

where, $h_1(t_n) = 2t_n$, $h_2(t_n) = t_n^2 - 1$, $g_1(t_{n-1}, t_n) = (t_{n-1}^2 + 1)h_2(t_n)$, and $g_2(t_{n-1}, t_n) = (t_{n-1}^2 + 1)h_1(t_n)$.

Corollary 2. *The control points of $\overline{f_{11}^n}$ and $\overline{f_{13}^n}$ satisfy the recursion:*

$$\begin{aligned} c_{(i_1, \dots, i_n)}(\overline{f_{11}^n}) &= c_{(i_1, \dots, i_{n-2})}(\overline{f_{11}^{n-2}}) c_{(i_{n-1}, i_n)}(g_1) + \\ &\quad c_{(i_1, \dots, i_{n-1})}(\overline{f_{13}^{n-1}}) c_{i_n}(h_1) \\ c_{(i_1, \dots, i_n)}(\overline{f_{13}^n}) &= c_{(i_1, \dots, i_{n-2})}(\overline{f_{11}^{n-2}}) c_{(i_{n-1}, i_n)}(g_2) - \\ &\quad c_{(i_1, \dots, i_{n-1})}(\overline{f_{13}^{n-1}}) c_{i_n}(h_2), \end{aligned}$$

where the control points of h_1 , h_2 , g_1 , and g_2 different from zero are:

$$\begin{aligned} c_{(0,0)}(g_1) &= c_{(0,1)}(g_1) = c_{(1,0)}(g_1) = c_{(1,1)}(g_1) = -1, \\ c_{(2,0)}(g_1) &= c_{(2,1)}(g_1) = -2, \\ c_{(0,1)}(g_2) &= c_{(1,1)}(g_2) = 1, \\ c_{(0,2)}(g_2) &= c_{(1,2)}(g_2) = c_{(2,1)}(g_2) = 2, \\ c_{(2,2)}(g_2) &= 4, \\ c_1(h_1) &= 1, \\ c_2(h_1) &= 2, \\ c_0(h_2) &= -1, \\ c_1(h_2) &= -1. \end{aligned}$$

Finally, we have that the Bernstein form of Equation (9) is:

$$\sum_{I=0}^M \left[c_I(\overline{f_{11}^n}) + c_{I\sigma}(\overline{f_{11}^n}) - 2^{\chi(I)+1} \right] B_I(\mathbf{t}) = 0, \quad (10)$$

where $c_I(\overline{f_{11}^n})$ and $c_{I\sigma}(\overline{f_{11}^n})$ can be computed using Corollary 2.

So far, for simplicity, we have treated all ϕ_i as variables, thus yielding 3^n control points in Equation (10). In practice though, many variables correspond to known angles of the mechanism and hence only a reduced set of control points is actually needed. For example, for the Gosselin platform each equation has four variables. Then, we have to compute $3^4 = 81$ control points to write term in the form of Equation (10). That is, altogether, 162 control points will fully describe the mechanism.

4 A subdivision-minimization strategy

Remind that a closure equation is obtained for every couple of legs. Then, note that the number of independent closure equations we can derive is $l - 1$ for a mechanism with l legs. Nevertheless, to simplify the presentation, let us assume that we are working with only one closure equation, i.e., we need to compute the solutions to just one equation of the form given by (9). The generalization will appear obvious at the end.

We are going to apply a method that allows searching for those roots of a Bernstein-form polynomial in n variables that lie in the unit box $[0, 1]^n$ of \mathbb{R}^n . Since the variables t_i in Equation (10) take values in their range, we first apply an affine parameter transformation to it so that the initial box is converted into the unit box. This scaling yields a new polynomial in Bernstein form with a new set of control points [4, Sec. 15.7]. Let us write it as $f(\mathbf{x}) = \sum_{I=0}^M w_I B_I(\mathbf{x})$, where \mathbf{x} stands for (x_1, \dots, x_n) , and

let us construct the function $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n+1}$ defined as $F(\mathbf{x}) = (\mathbf{x}, f(\mathbf{x}))$. Trivially, finding the roots of $f(\mathbf{x})$ is equivalent to detecting all points of the form $(\mathbf{x}, 0)$ in the graph of $F(\mathbf{x})$. However, the latter formulation is advantageous. First, the graph of $F(\mathbf{x})$ is an algebraic variety in \mathbb{R}^{n+1} whose points can be parameterized with polynomials in Bernstein form as

$$F(\mathbf{x}) = \sum_{I=0}^M v_I B_I(\mathbf{x}), \quad (11)$$

where $v_I = (i_1/m_1, i_2/m_2, \dots, i_n/m_n, w_I)$, which are called the control points of $F(\mathbf{x})$ [9].

Now, the root-finding procedure can make use of two important properties of the Bernstein form of $F(\mathbf{x})$. The first one is the so-called *convex hull property*: when $\mathbf{x} \in [0, 1]^n$, $F(\mathbf{x})$ is totally contained within the convex hull of its control points v_I . This follows immediately from the values taken by the Bernstein polynomials B_I in the unit box. They all are non-negative and sum to 1 [4], and hence the linear combination of control points v_I in $\sum_{I=0}^M v_I B_I(\mathbf{x})$ is actually a convex combination when $\mathbf{x} \in [0, 1]^n$. The second property is *subdivision*: if we are interested in the values that $F(\mathbf{x})$ takes within a sub-box of $[0, 1]^n$, say $\mathcal{B} = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$, with $0 \leq a_i, b_i \leq 1$, then it is possible to apply an affine parameter transformation $x_i = a_i + u_i(b_i - a_i)$, $i = 1, \dots, n$, to scale \mathcal{B} to the unit box and rewrite F in Bernstein form in terms of the new parameters u_i , with new control points. The important point here is that, after the scaling, the new control points for F can be directly obtained from the control points of the initial $F(\mathbf{x})$, and they will be closer to the graph of F than the previous ones [4, Sec. 15.7]. These considerations permit the following procedure to find all the roots of Equation (10).

1. Compute the control points v_I of $F(\mathbf{x})$. Start with the box $\mathcal{B} = [0, 1]^n$.
2. Using the convex hull property, find a sub-box \mathcal{B}' of \mathcal{B} that contains all the solutions of $F(\mathbf{x}) = (\mathbf{x}, 0)$ (see the details below). If there is no such sub-box (i.e., \mathcal{B} contains no solution), set $\mathcal{B}' = \emptyset$.
3. If $\mathcal{B}' \neq \emptyset$, see if it is sufficiently small. If so, conclude that there is a root inside and return \mathcal{B}' ; otherwise split \mathcal{B}' into some number of equally sized smaller boxes, scale these boxes back to $[0, 1]^n$ using the subdivision property for F , and recursively call step 2 once for each new smaller box.

It remains to see how step 2 is performed. Let \mathcal{C} denote the convex hull of the control points v_I , and let \mathcal{R} be the region of intersection of \mathcal{C} with the hyperplane $x_{n+1} = 0$. Then, we define \mathcal{B}' as the smallest rectangular box enclosing \mathcal{R} . Although the explicit computation of \mathcal{R} is a complex and time-consuming

task, it is not necessary to carry it out explicitly if all we need is just a bounding box for it. Indeed, \mathcal{R} can be described with a set of linear equalities and inequalities as follows. Since a point in \mathcal{R} is of the form $(\mathbf{x}, 0)$ and it must be a convex combination of the control points v_I , there must be coefficients $c_I \in \mathbb{R}$ such that

$$(\mathbf{x}, 0) = \sum_{I=0}^M c_I v_I, \quad c_I \geq 0 \quad \forall I, \quad \text{and} \quad \sum_{I=0}^M c_I = 1. \quad (12)$$

Now, to obtain the bounds of the box \mathcal{B}' we simply need to maximize and minimize x_i , for $i = 1, \dots, n$ subject to the constraints in Equations (12). These optimizations are linear programming problems and, hence, they can be efficiently solved with the simplex algorithm.

The above algorithm has been proven to terminate in all cases. Moreover, if there is a finite number of roots, then it returns a box enclosing each of them that is smaller than a user-specified tolerance. If the number of roots is infinite, the algorithm also terminates, providing a discretization of the solution space in a number of small boxes enclosing it. Additionally, the algorithm has the good property of being quadratically convergent to the roots. See [9] for details on all these facts.

This process is straightforwardly generalizable to mechanisms with several loops as follows. For each loop j we will have one closure equation $f_j(\mathbf{x}) = 0$, with its corresponding function $F_j(\mathbf{x}) = (x, f_j(\mathbf{x}))$ written as in Equation (11), and an associated convex hull \mathcal{C}_j , wrapping the control points of $F_j(\mathbf{x})$. Note that the common solutions to the whole system of equations necessarily lie in the intersection of all \mathcal{C}_j 's, on $x_{n+1} = 0$. This just introduces a slight modification in step 2 of the previous algorithm: as before, \mathcal{B}' is computed from \mathcal{B} by minimizing each x_i , $i = 1, \dots, n$, but now subject to all linear constraints in (12), associated with all \mathcal{C}_j 's.

5 Conclusions

We have presented an algorithm for solving the direct kinematics of parallel spherical mechanisms using a technique that takes advantage of the subdivision and convex-hull properties of the polynomials in Bernstein form, a technique developed in the context of Computer Graphics applications. This has been possible thanks to the recursion found for the control points of the closure equations.

We have used a tangent-half-angle substitution. This is probably the worst possible algebraic parameterization of the unit circle. Alternatively, it would

be possible to apply the substitutions $x_i = \sin \phi_i$ and $y_i = \cos \phi_i$. Then, the equation $x_i^2 + y_i^2 = 1$ should be included in the resulting system of equations. This would have at least two main advantages: the singularity at $\phi_i = \pi$, inherent to the tangent-half-angle substitution, would be avoided; and the set of control points would be greatly simplified because the closure equations would become linear in all the variables. These points concentrate our current efforts.

Acknowledgments

The authors would like to acknowledge the financial support of the Spanish CICYT under contract TAP99-1086-C03-01.

References

- [1] A. Castellet, *Solving inverse kinematics problems using an interval method*, Ph.D. Thesis, Polytechnical University of Catalonia. Available at <http://www-iri.upc.es/people/thomas/students.html>
- [2] A. Castellet and F. Thomas, "Towards an efficient interval method for solving inverse kinematic problems," *IEEE Int. Conf. on Robotics and Automation*, Vol. IV, pp. 3615-3620, 1997.
- [3] A. Castellet and F. Thomas, "The self-motion manifold of the orthogonal spherical mechanism," *Mechanisms and Machine Theory*, Vol. 34, No. 1, pp. 59-88, 1999.
- [4] G. Farin, *Curves and surfaces for computer-aided geometric design - A practical guide*. 2nd edition. Academic Press, 1990.
- [5] C.M. Gosselin, J. Sefrioui, and M.J. Richard, "On the direct kinematics of spherical three-degree-of-freedom parallel manipulators of general architecture," *ASME Journal of Mechanical Design*, Vol. 116, No. 2, pp. 594-598, 1994.
- [6] C.M. Gosselin, and M. Gagné, "A closed-form solution for the direct kinematics of a special class of spherical three-degree-of-freedom parallel manipulators," in *Computational Kinematics*, J.-P. Merlet and B. Ravani (eds.), pp. 231-240, Kluwer Academic Publishers, 1995.
- [7] P. Kovács and G. Hommel, "On the tangent-half-angle substitution," in *Computational Kinematics*, J. Angeles (editor), pp. 27-39, Kluwer Academic Publishers, 1993.
- [8] J.-P. Merlet, *Parallel robots*, Kluwer Academic Publishers, 2000.
- [9] E.C. Sherbrooke and N.M. Patrikalakis, "Computation of the solution of nonlinear polynomial systems," *Computer Aided Geometric Design*, Vol. 10, pp. 379-405, 1993.