

# A UNIFIED METHOD FOR COMPUTING POSITION AND ORIENTATION WORKSPACES OF GENERAL STEWART PLATFORMS

**Oriol Bohigas, Lluís Ros, and Montserrat Manubens**  
Institut de Robòtica i Informàtica Industrial (CSIC-UPC)  
08028 Barcelona, Catalonia  
Email: {obohigas,llros,mmanuben}@iri.upc.edu

## ABSTRACT

*The workspace of a Stewart platform is a complex six-dimensional volume embedded in the Cartesian space defined by six pose parameters. Because of its large dimension and complex shape, such workspace is difficult to compute and represent, so that comprehension on its structure is being gained by studying its three-dimensional slices. While successful methods have been given to determine the constant-orientation slice, the computation and appropriate visualization of the constant-position slice (also known as the orientation workspace) has proved to be a challenging task. This paper presents a unified method for computing both of such slices, and any other ones defined by fixing three pose parameters, on general Stewart platforms involving mechanical limits on the active and passive joints. Additional advantages over previous methods include the ability to determine all connected components of the workspace, and any motion barriers present in its interior.*

## INTRODUCTION

Due to their advantages in terms of dynamic properties, load-carrying capacity, high accuracy, and stiffness, Stewart platforms are widely used as flight simulators, high-precision positioning devices, mining machines, or surgical robots [1–7]. The assembly constraints imposed by their kinematic design, however, substantially reduce the set of poses that such platforms can attain, leading to highly-constrained workspaces in most of the cases. The availability of proper tools to accurately compute and represent such workspaces is thus of utmost importance, not only to assist the robot designer during the conception of the platform, but also to be able to implement trajectory planners more efficiently [8], once an adequate design has been chosen for a particular application.

The workspace of the Stewart platform is hard to compute and visualize. Its large dimension and complex shape, which may encompass several connected components, difficult any attempt of computing it exhaustively, due to the curse of dimensionality. In many situations, fortunately, the platform either operates with a fixed orientation or rotates about a fixed point, so that it can be assumed that three of the six pose parameters are held constant, leading to three-dimensional workspaces that are easier to obtain and represent. The constant-orientation workspace, in particular, is clearly understood, and fast geometric algorithms exist for computing its boundary [9], even in the presence of joint limits in the passive joints, or potential link-link interferences [10]. Interval analysis methods have been given too, to compute the interior of such workspace [11]. The constant-position workspace, also known as the orientation workspace, has also been studied, but its computation and visualization turn out to be more problematic, due to the complexity of the intervening equations, and to the difficulty of representing orientations in an intuitive way. Previous methods either assume one of the orientation angles held fixed [12, 13], thus producing two-dimensional sections of the workspace only, or let the three angles vary [14, 15], but all methods rely on some sort of discretization, which leads to incomplete or less accurate output in some situations. Recently, a fast method providing appealing visualizations of the orientation workspace has been given [16], but mechanical limits in the passive joints are neglected, so that the computed workspace is, actually, an overestimation of the real workspace. Some of the methods, finally, are only devised to compute the connected component of the workspace that is achievable from a pre-defined configuration [16–18], which limits the ability to characterize the movement of the platform under alternative assembly modes.

Despite the literature on the topic is rich, three important requirements are not fully met by previous approaches. First of all,

a method should ideally be *complete*, i.e., it should be able to obtain *all* connected components of the workspace, rather than just one achievable from a given configuration. Such ability is useful to the robot designer, to choose the appropriate component on which to assemble the manipulator, according to the operational volume desired, or to motion restrictions imposed by the application environment. Moreover, the method should be *accurate*, not only providing a precise representation of the workspace volume, but also of any motion barriers interior to the volume. Such barriers constitute true obstacles within the workspace, and may be encountered on Stewart platforms of special geometry, as shown in the paper. Finally, the method should be as *general* as possible, allowing to determine the constant-orientation and constant-position workspaces, and any other of the twenty workspaces that can be defined by fixing three of the six pose parameters of the platform, in order to understand the motion capability of the platform under any of the corresponding working modes.

The continuation approach in [17] and references therein is perhaps the closest to satisfy the previous requirements simultaneously. This approach uses ray-shooting techniques in conjunction with continuation methods to track the configurations in which the moving platform loses some of its instantaneous degrees of freedom, which yield the boundary of the workspace and its interior barriers when properly analyzed. While general and accurate in favorable situations, however, this approach loses boundary segments in several situations, as noted by the authors themselves in [19], and therefore it is not complete.

A unified method satisfying all previous requirements is given in this paper, valid for Stewart platforms of arbitrary geometry, involving mechanical limits both on the active and passive joints. The method entails formulating a system of equations defining the boundary of the workspace (Section “Formulation”), and then using an iterative procedure based on linear relaxations to isolate slices of such boundary exhaustively at the required resolution (Section “Boundary Isolation”). It is worth noting that while no consensus has been reached as to how should platform orientations be parameterized in order to yield intuitive workspace representations, the method we propose does not adopt any particular choice on this respect, and can obtain the orientation workspace under any possible parameterization, including those based on conventional Euler angles [16], tilt-and-torsion angles [14, 20], or Euler-Rodrigues parameters [15]. The approach has been implemented and validated successfully on several test cases (Section “Illustrative Examples”) and points requiring further attention have also been identified (Section “Conclusion”).

## FORMULATION

A Stewart platform consists of a body (the platform) which is linked to the ground (the base) by means of six legs, where each leg is a universal-prismatic-spherical chain. The most gen-

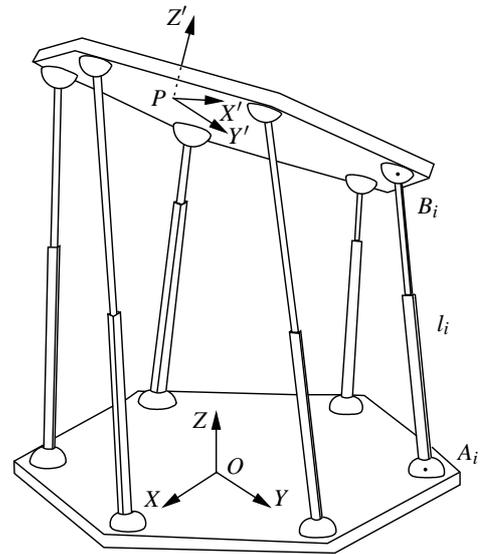


FIGURE 1. A Stewart platform.

eral version of such platform follows the so-called 6-6 UPS design, where the leg anchor points are all different (Fig. 1), though not necessarily coplanar [21]. The six prismatic joints are active, i.e. actuated, allowing to control the six degrees of freedom of the platform, and the remaining joints are passive.

Let  $OXYZ$  and  $PX'Y'Z'$  be fixed and moving reference frames attached to the base and the platform, respectively (Fig. 1). Any pose of the platform can be uniquely represented by a  $3 \times 3$  rotation matrix  $\mathbf{R}$  that provides the orientation of  $PX'Y'Z'$  relative to  $OXYZ$ , and the position vector  $\mathbf{p} = [x, y, z]^T$  of point  $P$  in the absolute frame. Not all values for  $\mathbf{R}$  and  $\mathbf{p}$  are allowed, however, because the mechanical limits present in the active and passive joints constrain the possible poses of the platform. We next define the equations modelling these constraints, so as to obtain a system of equations describing the workspace  $\mathcal{W}$  of the manipulator. This system is then extended with an additional constraint, in order to select only the points that lie on the boundary of  $\mathcal{W}$ .

## Workspace Equations

Let  $\mathbf{a}_i$  and  $\mathbf{b}_i$  denote the position vectors of the anchor points  $A_i$  and  $B_i$  of the  $i$ th leg, expressed in the fixed and moving frames, respectively. The leg lengths can then be written as

$$l_i^2 = |\mathbf{q}_i|^2, \quad (1)$$

for  $i = 1, \dots, 6$ , where  $\mathbf{q}_i = \mathbf{p} + \mathbf{R}\mathbf{b}_i - \mathbf{a}_i$  is a vector aligned with the  $i$ th leg expressed in the fixed frame.

Although the entries of  $\mathbf{R}$  are assumed to be variables here, note that they are not independent, as they must define an orthogonal matrix of positive determinant. Such a constraint can be defined in a variety of ways, e.g. by establishing appropriate dot- and cross-product equations on the columns of  $\mathbf{R}$ , but more intuitive representations of the orientation are obtained when three-parameter expressions for  $\mathbf{R}$  are introduced. For ease of comparison with [16], we will here adopt the parametrization provided by roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) angles, for which

$$\mathbf{R} = \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi),$$

or, in columnwise form,

$$\begin{bmatrix} r_{1,x} \\ r_{1,y} \\ r_{1,z} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi \\ \cos \theta \sin \psi \\ -\sin \theta \end{bmatrix}, \quad (2)$$

$$\begin{bmatrix} r_{2,x} \\ r_{2,y} \\ r_{2,z} \end{bmatrix} = \begin{bmatrix} \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi \\ \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi \\ \sin \phi \cos \theta \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} r_{3,x} \\ r_{3,y} \\ r_{3,z} \end{bmatrix} = \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix}, \quad (4)$$

but the presented method is applicable to other parametrizations as well, including those based on tilt-and-torsion angles or Euler-Rodrigues parameters.

To see the constraints introduced by the mechanical limits on all joints, note first that the prismatic joints have a range of actuation that force the lengths  $l_i$  to take values within some intervals  $[l_i^{\min}, l_i^{\max}]$ . These constraints can be modelled as equalities by defining  $m_i = (l_i^{\max} + l_i^{\min})/2$  and  $h_i = (l_i^{\max} - l_i^{\min})/2$ , and imposing

$$(l_i - m_i)^2 + d_i^2 = h_i^2, \quad (5)$$

for  $i = 1, \dots, 6$ , where the  $d_i$  are newly-defined auxiliary variables. The passive joints on the base and on the platform also have mechanical limits constraining the allowable positions of each leg. To describe these constraints, let  $\mathbf{j}_{A_i}$  be a unit vector given in the base frame, aligned with the axis of symmetry of the joint at  $A_i$ . The constraint imposed by a base joint can then be expressed as

$$\mathbf{j}_{A_i} \mathbf{q}_i \geq l_i \cos \alpha_i,$$

which simply restricts the maximum allowed misalignment between the axis of the joint and the leg to be of angle  $\alpha_i$ . These

inequalities can be transformed into equalities by introducing a new variable  $t_i$  and writing

$$\mathbf{j}_{A_i} \mathbf{q}_i - l_i \cos \alpha_i = t_i^2 \quad (6)$$

for  $i = 1, \dots, 6$ . Note that  $\mathbf{j}_{A_i} \mathbf{q}_i \geq l_i \cos \alpha_i$  is satisfied if, and only if, Eq. (6) is satisfied for some value of  $t_i$ . Similarly, for each passive joint on the platform we define a new variable  $g_i$  and impose

$$\mathbf{j}_{B_i} (\mathbf{R}^T \mathbf{q}_i) - l_i \cos \beta_i = g_i^2, \quad (7)$$

where  $\mathbf{j}_{B_i}$  is a unit vector along the axis of symmetry of the joint at  $B_i$ , expressed in the moving frame, and  $\beta_i$  is the maximum allowed misalignment in this joint.

In conclusion, the workspace  $\mathcal{W}$  of the platform is the set of all possible tuples

$$(x, y, z, \phi, \theta, \psi)$$

that satisfy Eqs. (1)-(7) for some value of

$$(l_1, \dots, l_6, r_{1,x}, \dots, r_{3,z}, d_1, \dots, d_6, t_1, \dots, t_6, g_1, \dots, g_6).$$

Since Eqs. (1)-(7) form a system of 33 equations in 39 variables,  $\mathcal{W}$  will be a six-dimensional set in general, which is in agreement with the fact that the Stewart platform has six degrees of freedom.

## Boundary Equations

While we could try to find  $\mathcal{W}$  by solving Eqs. (1)-(7) directly, it is preferable to compute the boundary of  $\mathcal{W}$  instead, because such boundary is a set of lower dimension. A point lies on the boundary of  $\mathcal{W}$  whenever any of the active or passive joints reaches a mechanical limit.

In order to select only the points on the boundary, note that the  $i$ th leg reaches its maximal or minimal length, or a limit angle on its passive joints, whenever  $d_i$ ,  $t_i$ , or  $g_i$  vanish. Thus, a solution to Eqs. (1)-(7) corresponds to a boundary point if, and only if,

$$\prod_{i=1}^6 d_i t_i g_i = 0. \quad (8)$$

In sum, the boundary of  $\mathcal{W}$  is formed by the points  $(x, y, z, \phi, \theta, \psi)$  that satisfy Eqs. (1)-(8) for some value of the remaining variables. Such points will form a five-dimensional set in general because just one equation, and no extra variable, have been added to Eqs. (1)-(7). Despite the boundary of  $\mathcal{W}$

has one dimension less than  $\mathcal{W}$ , it is still hard to compute it exhaustively using the current hardware technology. However, by setting three pose variables to a constant value, one can obtain two-dimensional slices of the boundary, which are much easier to compute and yet provide useful representations. Typically, these slices are those of the constant-orientation or constant-position workspaces, but any other slice of interest, obtained by fixing any three of the six pose parameters, should also be available to the designer. The system characterizing such slices, thus, is formed by Eqs. (1)-(8) with the corresponding three pose variables held constant. The case of a planar workspace, where the platform moves on a plane and is only free to rotate about an angle, as well as the constant-orientation and the constant-position workspaces will be given later as examples of such slices.

## BOUNDARY ISOLATION

A numerical method able to solve the system of equations just described is next provided, based on the linear relaxation paradigm proposed in [22]. The approach entails algebraizing the equations into a canonical form, then computing an initial box that contains all solutions, and finally using a branch-and-prune method exploiting such canonical form to isolate all boundary points at the desired resolution.

## Equation algebraization

In order to algebraize the system of equations, two different changes of variables need to be introduced. First, all trigonometric terms of Eqs. (2)-(4) are eliminated by introducing the changes of variables

$$\begin{aligned} c_\tau &= \cos \tau, \\ s_\tau &= \sin \tau, \end{aligned}$$

for  $\tau \in \{\phi, \theta, \psi\}$ . Since the new variables  $c_\tau$  and  $s_\tau$  represent the cosine and sine of  $\tau$ , they must satisfy the circle equations

$$c_\tau^2 + s_\tau^2 = 1, \quad (9)$$

which thus need to be introduced into the system for  $\tau \in \{\phi, \theta, \psi\}$ .

After applying such changes, note that the system formed by Eqs. (1)-(9) is already polynomial. Let  $\mathbf{y}$  be a vector containing all of the variables of this system, and let  $y_i$  and  $y_j$  refer to any two of such variables. A second change of variables

$$p_k = y_i^2, \quad (10)$$

$$b_k = y_i y_j, \quad (11)$$

is introduced for all  $y_i y_j$  and  $y_i^2$  monomials intervening in the equations, in order to allow transforming the system into the expanded form

$$\left. \begin{aligned} \mathbf{\Lambda}(\mathbf{x}) &= 0 \\ \mathbf{\Omega}(\mathbf{x}) &= 0 \end{aligned} \right\}, \quad (12)$$

where  $\mathbf{x}$  is an  $n_x$ -dimensional vector including the original  $\mathbf{y}$  variables and the newly-introduced  $p_k$  and  $b_k$  ones,  $\mathbf{\Lambda}(\mathbf{x}) = 0$  is a collection of linear equations in  $\mathbf{x}$ , and  $\mathbf{\Omega}(\mathbf{x}) = 0$  is a collection of quadratic equations, each of which adopts one of the two forms

$$\begin{aligned} x_k &= x_i^2, \\ x_k &= x_i x_j, \end{aligned}$$

which correspond to the variable changes of Eqs. (10) and (11), respectively. Note that in some cases, as in Eq. (8), the change relative to Eq. (11) needs to be applied recursively in order to arrive at the form assumed in Eq. (12).

## Initial Bounding Box

It can be seen that each variable  $x_i$  of  $\mathbf{x}$  can only take values within a prescribed interval, so that from the Cartesian product of all such intervals one can define the initial  $n_x$ -dimensional box  $\mathcal{B}$  that bounds all solutions of Eq. (12). Some intervals can be automatically computed from the intervals of other variables; namely, those of the variables  $p_k$  or  $b_k$  that appear due to the change of Eqs. (10) and (11). Thus, only the intervals of the variables in  $\mathbf{y}$  need to be determined.

The lengths of the legs are bound within maximal and minimal values, so that

$$l_i \in [l_i^{\min}, l_i^{\max}],$$

for  $i = 1, \dots, 6$ . Since the columns of  $\mathbf{R}$  are orthonormal vectors, its entries must satisfy

$$r_{j,x}, r_{j,y}, r_{j,z} \in [-1, 1],$$

for  $j = 1, 2, 3$ . Similarly, for  $\tau \in \{\phi, \theta, \psi\}$ , it must be

$$c_\tau, s_\tau \in [-1, 1],$$

since these variables refer to the cosine and sine of  $\tau$ .

For the variables  $d_i$  we note that Eq. (5) simply constrains such variables to take values in  $[-h_i, h_i]$ . Similarly, Eqs. (6) and (7) constrain the left-hand side of the equation and the  $t_i$

and  $g_i$  variables to take values on a parabola of vertical axis with its minimum point at the origin. Note that, in any case, the maximum value that the left-hand side of these equations can achieve is  $l_i^{max} - l_i^{max} \cos \alpha_i$  or  $l_i^{max} - l_i^{max} \cos \beta_i$ , respectively. In sum, we obtain the following bounds for  $i = 1, \dots, 6$ :

$$\begin{aligned} d_i &\in [-h_i, h_i], \\ t_i &\in \left[ -\sqrt{l_i^{max} - l_i^{max} \cos \alpha_i}, \sqrt{l_i^{max} - l_i^{max} \cos \alpha_i} \right], \\ g_i &\in \left[ -\sqrt{l_i^{max} - l_i^{max} \cos \beta_i}, \sqrt{l_i^{max} - l_i^{max} \cos \beta_i} \right]. \end{aligned}$$

Finally, it only remains to find the ranges for the  $x$ ,  $y$ , and  $z$  components of  $\mathbf{p}$ . While tight bounds could be computed to accurately confine these variables, this is not necessary under the presented approach. The method is not significantly sensitive to the volume of the initial box because parts with no solution are rapidly discarded by the numerical procedure. Thus, we can use any easy-to-compute upper bound to establish the ranges of these position variables. Here, we shall use the fact that, for any valid configuration of the platform, point  $P$  will always lie inside a sphere of radius  $l_i^{max} + |\mathbf{b}_i|$  centered at  $A_i$ , which is valid for any leg  $i = 1, \dots, 6$ . By choosing any leg, for instance leg one, we can take the ranges of the smallest box containing the sphere, that is

$$\begin{aligned} x &\in [a_{1,x} - l_1^{max} - |\mathbf{b}_1|, a_{1,x} + l_1^{max} + |\mathbf{b}_1|], \\ y &\in [a_{1,y} - l_1^{max} - |\mathbf{b}_1|, a_{1,y} + l_1^{max} + |\mathbf{b}_1|], \\ z &\in [a_{1,z} - l_1^{max} - |\mathbf{b}_1|, a_{1,z} + l_1^{max} + |\mathbf{b}_1|]. \end{aligned}$$

## Numerical Solution

The algorithm for solving Eq. (12) recursively applies two operations on  $\mathcal{B}$ : box *shrinking* and box *splitting*. Using box shrinking, portions of  $\mathcal{B}$  containing no solution are eliminated by narrowing some of the box intervals. This process is repeated until either the box is reduced to an empty set, in which case it contains no solution, or the box is “sufficiently” small, in which case it is considered a *solution* box, or the box cannot be “significantly” reduced, in which case it is bisected into two sub-boxes via box splitting (which simply bisects its largest interval). To converge to all solutions, the whole process is recursively applied to the new sub-boxes, until one obtains a collection of solution boxes whose side lengths are below a given threshold  $\sigma$ .

The crucial operation in this scheme is box shrinking, which is implemented as follows. Note first that the solutions falling in some sub-box  $\mathcal{B}_c \subseteq \mathcal{B}$  must lie in the linear variety defined by  $\mathbf{\Lambda}(\mathbf{x}) = 0$ . Thus, we may shrink  $\mathcal{B}_c$  to the smallest possible box bounding this variety inside  $\mathcal{B}_c$ . The limits of the shrunk box along, say, dimension  $x_i$  can be found by solving the following

two linear programs:

$$\begin{aligned} \text{LP1: Minimize } &x_i, \\ &\text{subject to: } \mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c, \\ \text{LP2: Maximize } &x_i, \\ &\text{subject to: } \mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c. \end{aligned}$$

However, observe that  $\mathcal{B}_c$  can be further reduced, because the solutions must also satisfy all equations  $x_k = x_i^2$  and  $x_k = x_i x_j$  in  $\mathbf{\Omega}(\mathbf{x}) = 0$ . These equations can be taken into account by noting that, if  $[v_i, u_i]$  denotes the interval of  $\mathcal{B}_c$  along dimension  $x_i$ , then:

1. The portion of the parabola  $x_k = x_i^2$  lying inside  $\mathcal{B}_c$  is bound by the triangle  $A_1 A_2 A_3$ , where  $A_1$  and  $A_2$  are the points where the parabola intercepts the lines  $x_i = v_i$  and  $x_i = u_i$ , and  $A_3$  is the point where the tangent lines at  $A_1$  and  $A_2$  meet (Fig. 2a).
2. The portion of the hyperbolic paraboloid  $x_k = x_i x_j$  lying inside  $\mathcal{B}_c$  is bound by the tetrahedron  $B_1 B_2 B_3 B_4$ , where the points  $B_1, \dots, B_4$  are obtained by lifting the corners of the rectangle  $[v_i, u_i] \times [v_j, u_j]$  vertically to the paraboloid (Fig. 2b).

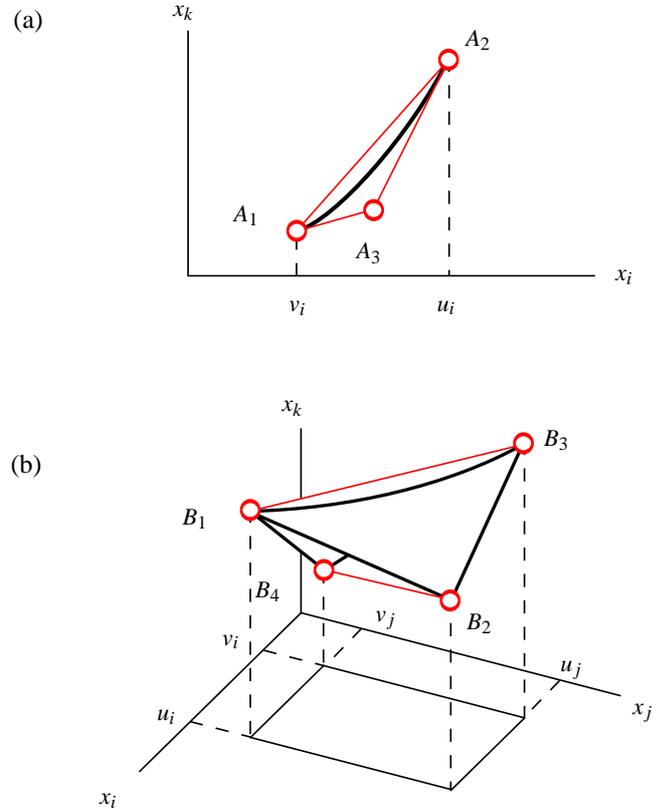


FIGURE 2. Polytope bounds within box  $\mathcal{B}_c$ .

Thus, linear inequalities corresponding to these bounds can be added to  $LP1$  and  $LP2$ , which usually produces a much larger reduction of  $\mathcal{B}_c$ , or its complete elimination if one of the linear programs is found unfeasible.

As it turns out, the previous algorithm explores a binary tree of boxes whose internal nodes correspond to boxes that have been split at some time, and whose leaves are either solution or empty boxes. The collection  $B$  of all solution boxes, which is returned as output upon termination, is said to form a *box approximation* of the solution set of Eq. (12), because the boxes in  $B$  form a discrete envelop of such set, whose accuracy can be adjusted through the  $\sigma$  parameter. Notice that the algorithm is complete, in the sense that it will succeed in isolating all solution points of Eq. (12) accurately, provided that a small-enough value for  $\sigma$  is used. Detailed properties of the algorithm, including an analysis of its completeness, correctness, and convergence order, are given in [22].

## ILLUSTRATIVE EXAMPLES

We next illustrate the performance of the method by computing the boundaries of several workspaces of the Stewart platform. To emphasize the generality of the method, we first obtain the boundaries corresponding to the constant-orientation, constant-position and planar workspaces of a standard platform. We then analyze a special platform to show that the method performs well on situations that hinder the application of previous methods. The method is able to completely determine the boundaries of the workspace in all situations.

The geometric parameters of the analyzed platforms are indicated in Table 1, and the ranges for  $l_i$  are assumed to be  $[1.2, 1.8]$  for all legs. All results reported have been obtained by using a parallelized version of the method, implemented in C using the libraries of the CUIK platform [22], and executed on a grid computer with four PC units equipped with two Intel Quadcore Xeon E5310 processors and 4 Gb of RAM each one. Table 2 provides, for each experiment, the amount of CPU time required to solve it ( $t_s$ ) and the number of solution boxes returned ( $n_s$ ) using  $\sigma = 0.1$ .

### A Standard Platform

The standard platform analyzed here corresponds to the one studied in [16], where the authors compute the constant-position workspace obtained by fixing  $\mathbf{p}$  to a constant value, but neglecting the mechanical limits on the passive joints. To see that such limits really reduce the workspace, we have computed such workspace twice, first neglecting the limits on the passive joints, and then taking them into account. Fig. 3 shows the obtained results for the roll-pitch-yaw angles, assuming  $\mathbf{p} = [0, \frac{2}{3}\sqrt{3}, \frac{5}{4}]^T$ . As it can be seen, the workspace contains three different connected components, while in [16] the authors are only able to

TABLE 1. Stewart platform parameters.

| $i$      | 1         | 2       | 3       | 4       | 5      | 6       |         |
|----------|-----------|---------|---------|---------|--------|---------|---------|
| Standard | $a_{i,x}$ | 0       | 0       | 0.7598  | 0.7598 | -0.7598 | -0.7598 |
|          | $a_{i,y}$ | 0       | 0       | 1.3161  | 1.3161 | 1.3161  | 1.3161  |
|          | $a_{i,z}$ | 0       | 0       | 0       | 0      | 0       | 0       |
|          | $b_{i,x}$ | -0.4559 | 0.4559  | 0.4599  | 0      | 0       | -0.4559 |
|          | $b_{i,y}$ | -0.2632 | -0.2632 | -0.2632 | 0.5264 | 0.5264  | -0.2632 |
|          | $b_{i,z}$ | 0       | 0       | 0       | 0      | 0       | 0       |
| Special  | $a_{i,x}$ | 0.8165  | -0.4082 | -0.4082 | 0.8165 | -0.4082 | -0.4082 |
|          | $a_{i,y}$ | 0       | 0.7071  | -0.7071 | 0      | 0.7071  | -0.7071 |
|          | $a_{i,z}$ | 0       | 0       | 0       | 0      | 0       | 0       |
|          | $b_{i,x}$ | 0       | 0       | 0       | 0.8165 | -0.4082 | -0.4082 |
|          | $b_{i,y}$ | 0       | 0       | 0       | 0      | 0.7071  | -0.7071 |
|          | $b_{i,z}$ | 0       | 0       | 0       | 0.5774 | 0.5774  | 0.5774  |

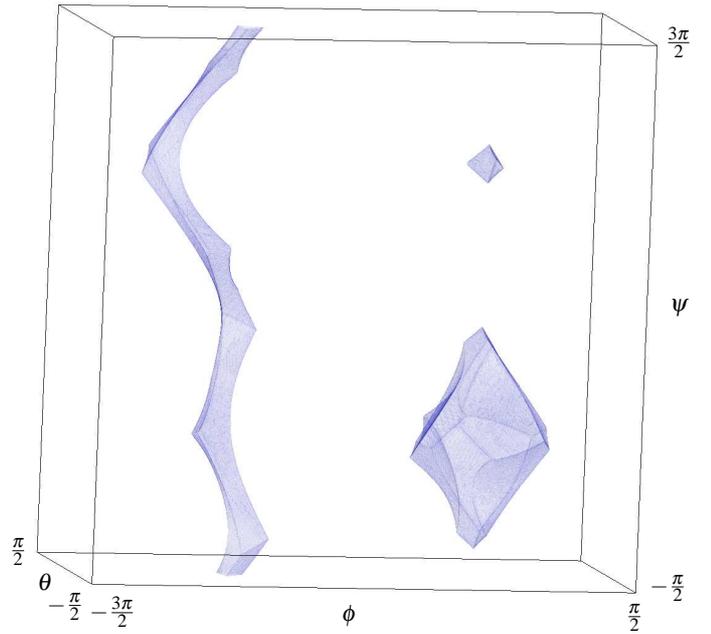
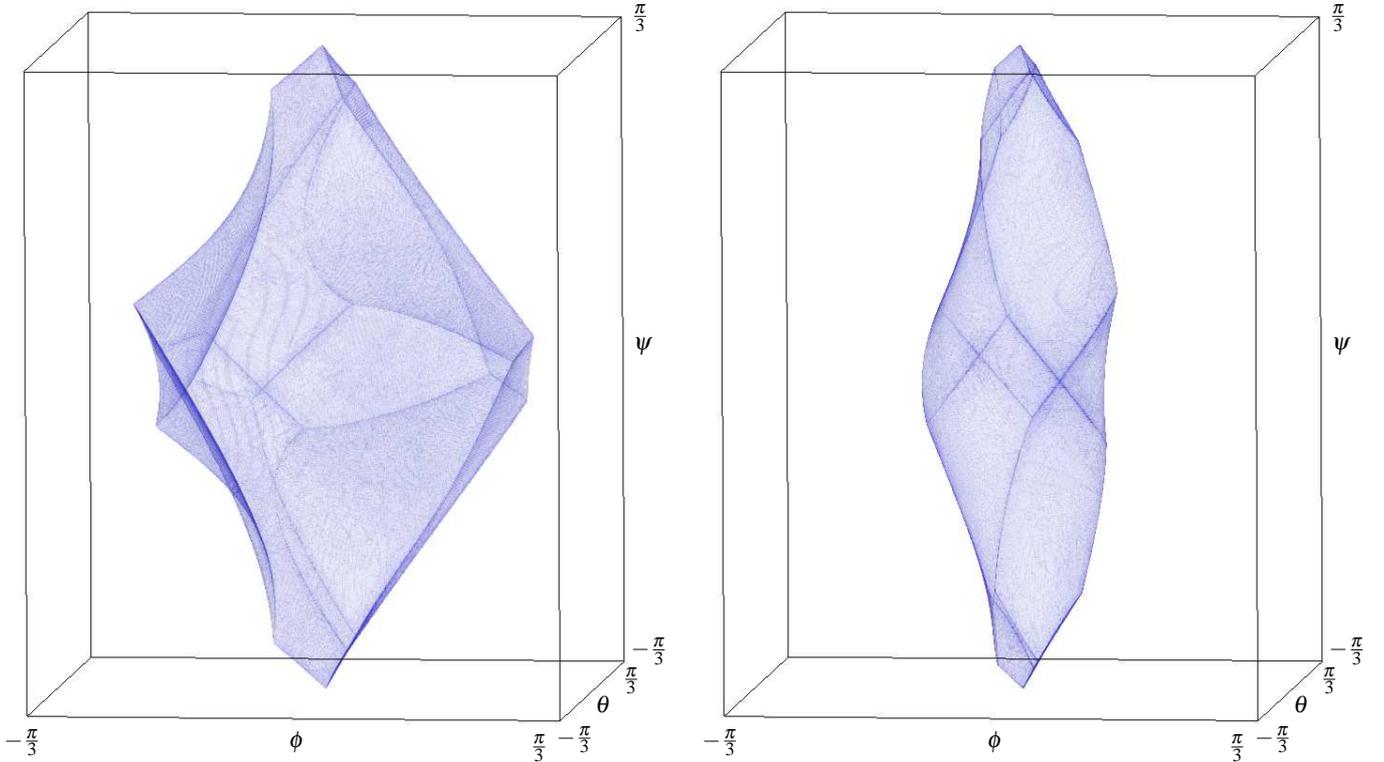


FIGURE 3. Boundaries of the constant-position workspace.

describe the one accessible from the origin (the lower-right component in Fig. 3). Detecting all connected components is necessary though, to let the designer choose the most appropriate assembly mode for the platform, depending on the specific task to be performed. Now, by taking into account the limit angles  $\alpha_i = \beta_i = 50^\circ$  on all passive joints, some of the components are no longer accessible and the workspace reduces to just one connected component that corresponds to the one around the origin in Fig. 3. As expected, the enclosed volume is much smaller, as



**FIGURE 4.** Only the lower-right component in Fig. 3 is partially achievable after taking passive joint limits into account.

noted in Fig. 4, where such component (right) is compared to the original one (left).

Instead of fixing  $\mathbf{p}$ , we can fix the orientation of the platform, and hence obtain the constant-orientation workspace. Setting  $\phi = \theta = \psi = 0$ , which is equivalent to letting  $\mathbf{R} = \mathbf{I}$ , we obtain the workspace shown in Fig. 5. The figure compares the resulting workspaces taking into account only active constraints (left) or active and passive constraints with  $\alpha_i = \beta_i = 47.16^\circ$  (right). As before, the enclosed volume is smaller in the second case, meaning that the attainable positions of the platform are reduced because some of the passive joints achieve their mechanical limit. In fact, in both cases, the workspace has an additional connected component symmetric to the one shown, which corresponds to the assembly mode of the platform where  $P$  sweeps a similar volume for  $z < 0$ .

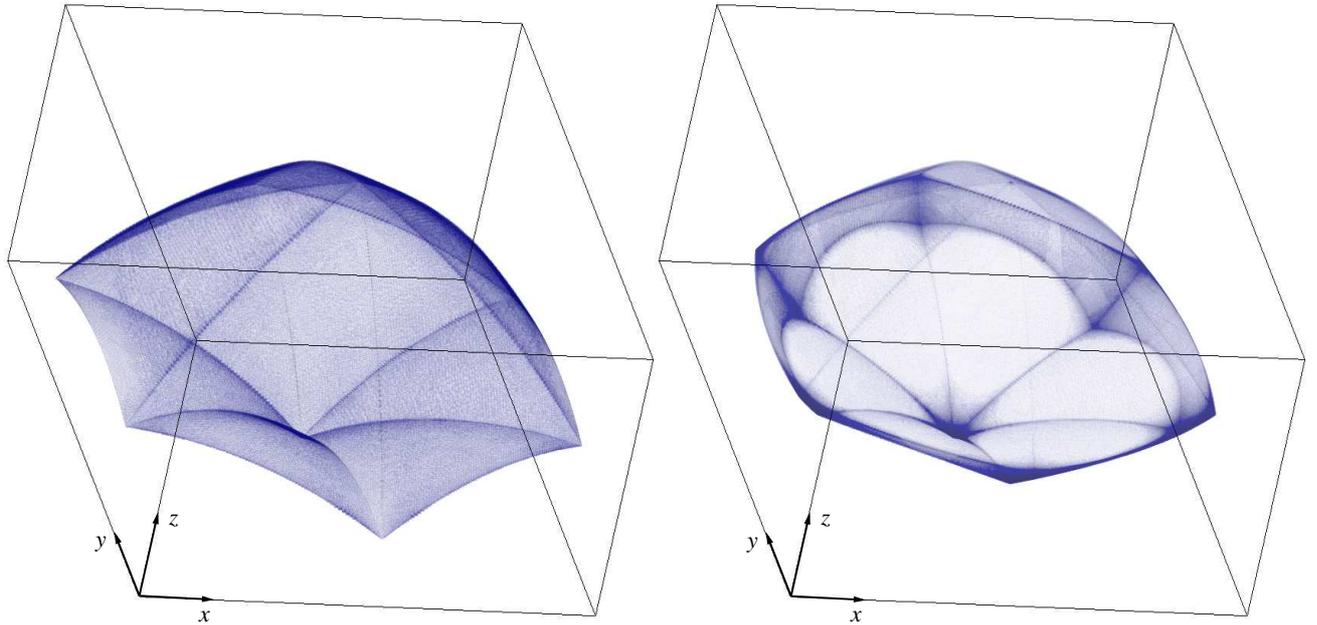
In general, previous methods in the literature only compute the constant-position or the constant-orientation slices of the complete six-dimensional workspace, i.e., those shown so far in Figs. 3-5. But the method herein described can also be used to derive any other slice. As an example, Fig. 6 presents the slices obtained when fixing  $z = \frac{5}{4}$  and  $\phi = \theta = 0$ , before and after the introduction of the passive joint limits with  $\alpha_i = \beta_i = 41.41^\circ$ . Note that in this case the platform is equivalent to a planar positioning device, where  $P$  moves on the  $z = \frac{5}{4}$  plane, and only

rotation about the  $Z'$  axis is allowed. As shown in Fig. 6, when passive joint constraints are considered, both the attainable positions of the platform and its orientational capability get substantially reduced.

### A Special Platform

We next show results on computing the constant-position workspace of a special platform that yields interior barriers in such workspace. Its geometric parameters are shown in Table 1 and correspond to a design where three of the legs are anchored at a same point  $P$  on the platform, with the base joints coincident in pairs. This design may seem difficult to construct, but if  $P$  is the point we keep fixed when computing the constant-position workspace, then the platform is equivalent to the 3-UPS/S design shown in Fig. 7, which allows orientational capability only, and is in fact equivalent to the *Agile Eye* design [20].

While in general platforms the constant-position workspace has a two-dimensional boundary, such boundary degenerates into one-dimensional sets in this case, as shown in Fig. 8, and these sets are impossible to obtain by previous numerical methods. The method in [17], for example, would proceed by first intersecting the boundaries through planes, and then computing the portion of the boundaries on such planes using ray-shooting techniques combined with continuation. However, note that the



**FIGURE 5.** Boundaries of the constant-orientation workspace for  $z > 0$ .

planes would only contain isolated points in this case, which would be hit by ray-shooting with probability zero, as explained in [23]. Similarly, the method in [16] would perform a discretization on angle  $\theta$  and then on angle  $\phi$  in order to find the boundary points corresponding to such angles. Nevertheless, the portion of a slice lying on a constant- $\theta$  plane is again formed by isolated points, and a discretization on angle  $\phi$  would almost never encounter such points. Slices of the boundary curves in Fig. 8, in fact, can only be obtained by analytical methods specific for such platforms [20], since other methods based on discretization exhibit similar drawbacks [13–15]. The presented technique, on the contrary, is robust to such situations. If the same equations considered for the standard platform are now used for determining the constant-position workspace of the special platform, we obtain the results shown in Fig. 8. This workspace occupies the whole range of the roll-pitch-yaw angles in this case, but the curves represent true motion impediments to be avoided when planning trajectories for the platform.

## CONCLUSIONS

This paper has introduced a new approach for computing three-dimensional slices of the workspace on Stewart platforms of arbitrary geometry. A distinguishing feature of the approach is that it unifies, under a single method, the obtention of any possible slice of the workspace, while previous approaches mostly concentrate on particular slices, like the constant-position or the constant-orientation slice. In fact, a total of twenty slices can

**TABLE 2.** Performance data at  $\sigma = 0.1$ .

| Platform | Workspace          | Joint limits       | $n_s$ | $t_s$ (sec) |
|----------|--------------------|--------------------|-------|-------------|
| Standard | Const. position    | Active             | 6526  | 66          |
|          |                    | Active and passive | 3196  | 69          |
|          | Const. orientation | Active             | 1306  | 4           |
|          |                    | Active and passive | 1621  | 21          |
| Special  | Planar mode        | Active             | 1849  | 6           |
|          |                    | Active and passive | 1391  | 14          |
|          | Const. position    | Active             | 8448  | 252         |

be obtained by fixing three pose parameters, and computing any of them might be necessary depending on the specific task to be performed with the platform. Additional advantages of the method have been discussed and illustrated with examples, like the ability to compute all connected components of a slice, to detect motion barriers present in its interior, and the possibility to take passive joint limits into account.

Clearly, the primary application of the method is in the context of robot design, as it allows studying the motion capability of a particular platform relatively fast, before its actual construction. However, the method might also be helpful in the context of collision-free trajectory planning, where a main issue is how to sample the workspace efficiently and with good coverage, in order to compute proper roadmaps of the workspace in short times [8]. While fast in favorable situations, current plan-

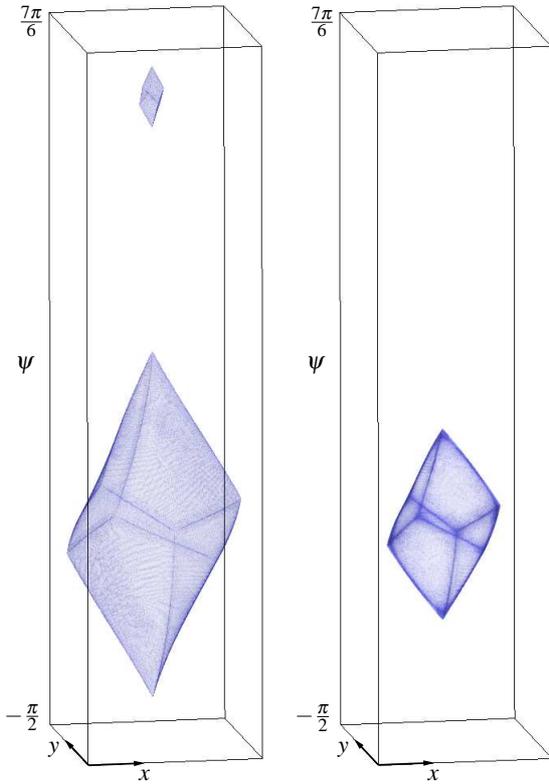


FIGURE 6. Boundaries of the planar workspace.

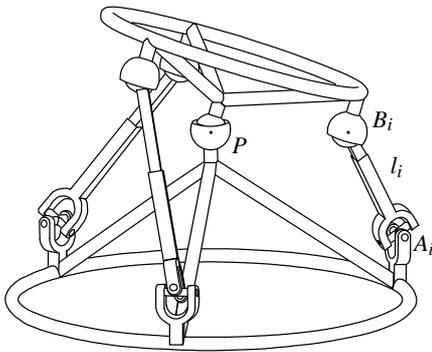


FIGURE 7. A 3-UPS/S platform.

ners draw samples from conservative estimates of the workspace, which makes them less efficient and possibly incomplete on highly-constrained situations. The performance of such planners would notably increase in such cases, however, if samples were drawn from accurate representations like those provided by the proposed method.

Finally, it is worth noting that the method shows potential of being able to cope with additional constraints, like leg-leg colli-

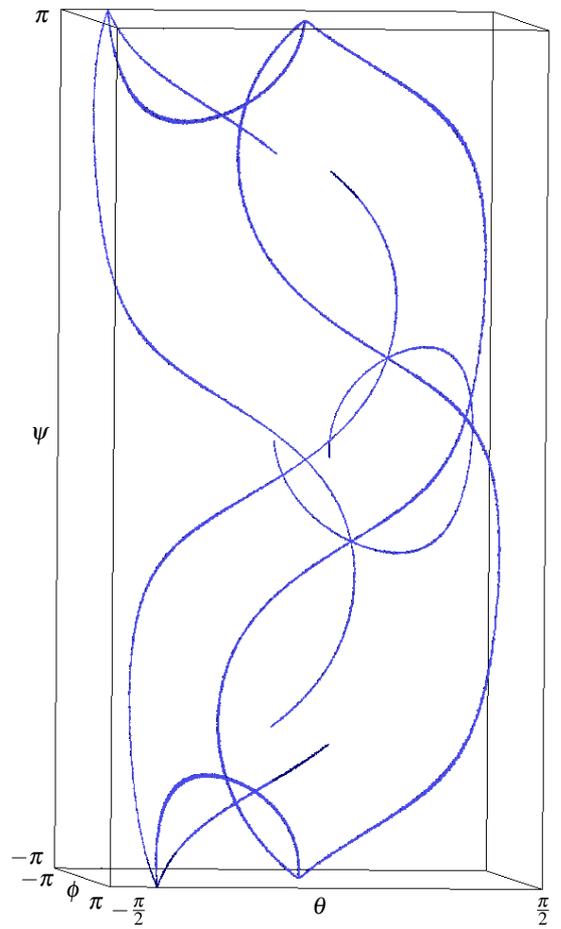


FIGURE 8. Degenerate boundaries of the constant-position workspace of the special platform.

sions or singularity-avoidance constraints. These constraints can in principle be formulated in the form assumed by the proposed approach, but further work needs to be done in order to achieve a mild formulation, leading to an acceptable computational burden.

## ACKNOWLEDGMENT

The authors wish to thank Dr. Josep M. Porta for fruitful discussions around the topic of the paper, and for his help during the implementation of the method on the CUIK platform. This work has been partially supported by the Spanish Ministry of Science and Innovation under contract DPI2010-18449, and by a Juan de la Cierva contract supporting the third author.

## REFERENCES

- [1] Stewart, D., 1965. "A platform with six degrees of freedom". *Proceedings of the Institution of Mechanical Engineers*, **180**, pp. 371–386.
- [2] Dasgupta, B., and Mruthyunjaya, T. S., 2000. "The Stewart platform manipulator: a review". *Mechanism and Machine Theory*, **35**(1), pp. 15–40.
- [3] McInroy, J., and O'Brien, G., 1999. "Precise, fault-tolerant precision pointing using a Stewart platform". *IEEE/ASME Transactions on Mechatronics*, **4**, pp. 91–95.
- [4] Preumont, A., Horodincu, M., Romanescu, I., de Marneffe, B., Avraam, M., Deraemaeker, A., Bossens, F., and Hanieh, A., 2007. "A six-axis single-stage active vibration isolator based on Stewart platform". *Journal of Sound and Vibration*, **300**, pp. 644–661.
- [5] Shaw, D., and Chen, Y., 2001. "Cutting path generation of the Stewart platform-based milling machine using an end-mill". *International Journal of Production Research*, **39**, pp. 1367–1383.
- [6] van Silfhout, R. G., 1999. "High-precision hydraulic Stewart platform". *Review of Scientific Instruments*, **70**, pp. 3488–3494.
- [7] Shoham, M., Burman, M., Zehavi, E., Joskowicz, L., Batkalin, E., and Kunicher, Y., 2003. "Bone-mounted miniature robot for surgical procedures: Concept and clinical applications". *IEEE Transactions on Robotics and Automation*, **19**(5), october.
- [8] Cortés, J., and Siméon, T., 2003. "Probabilistic motion planning for parallel mechanisms". In Proc. of the Int. Conf. on Robotics and Automation, Vol. 3, pp. 4354–4359.
- [9] Gosselin, C., 1990. "Determination of the workspace of 6-DOF parallel manipulators". *ASME J. Mech. Des.*, **112**, pp. 331–336.
- [10] Merlet, J. P., 1992. "Geometrical determination of the workspace of a constrained parallel manipulator". In Advances in Robot Kinematics, Ferrare, Italy, September 7–9, pp. 326–329.
- [11] Merlet, J.-P., et al., 1999. "Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries". *International Journal of Robotics Research*, **18**(9), pp. 902–916.
- [12] Romdhane, L., 1994. "Orientation workspace of fully parallel mechanisms". *European Journal of Mechanics, A/Solids*, **13**(4), pp. 541–553.
- [13] Merlet, J.-P., 1995. "Determination of the orientation workspace of parallel manipulators". *Journal of Intelligent and Robotic Systems*, **13**(2), pp. 143–160.
- [14] Bonev, I., and Ryu, J., 2001. "A new approach to orientation workspace analysis of 6-DOF parallel manipulators". *Mechanism and Machine Theory*, **36**(1), pp. 15–28.
- [15] Pernkopf, F., and Husty, M., 2006. "Workspace analysis of Stewart-Gough-type parallel manipulators". *Proc. of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, **220**(7), pp. 1019–1032.
- [16] Jiang, Q., and Gosselin, C., 2009. "Evaluation and representation of the theoretical orientation workspace of the gough-stewart platform". *Journal of Mechanisms and Robotics*, **1**(2).
- [17] Haug, E. J., Luh, C.-M., Adkins, F. A., and Wang, J.-Y., 1996. "Numerical algorithms for mapping boundaries of manipulator workspaces". *ASME Journal of Mechanical Design*, **118**, pp. 228–234.
- [18] Tsai, K., and Lin, J., 2006. "Determining the compatible orientation workspace of Stewart-Gough parallel manipulators". *Mechanism and Machine Theory*, **41**(10), pp. 1168–1184.
- [19] Abdel-Malek, K., Adkins, F., Yeh, H. J., and Haug, E., 1997. "On the determination of boundaries of manipulator workspaces". *Robotics and Computer Integrated Manufacturing*, **13**(1), pp. 63–72.
- [20] Bonev, I. A., and Gosselin, C. M., 2006. "Analytical determination of the workspace of symmetrical spherical parallel mechanisms". *IEEE Transactions on Robotics*, **22**(5), pp. 1011–1017.
- [21] Merlet, J., 2006. *Parallel robots*. Springer-Verlag New York Inc.
- [22] Porta, J. M., Ros, L., and Thomas, F., 2009. "A linear relaxation technique for the position analysis of multi-loop linkages". *IEEE Transactions on Robotics*, **25**(2), pp. 225–239. See also <http://www-iri.upc.es/groups/gmr/cuikweb>.
- [23] Bohigas, O., Ros, L., and Manubens, M., 2011. A numerical method for complete determination of workspace boundaries on general manipulators. Submitted.