

# A Complete Method for Workspace Boundary Determination on General Structure Manipulators

Oriol Bohigas, Montserrat Manubens, and Lluís Ros

**Abstract**—This paper introduces a new method for workspace boundary determination on general structure manipulators. The method uses a branch-and-prune technique to isolate a set of output singularities and then classifies the points on such a set according to whether they correspond to motion impediments in the workspace. A detailed map of the workspace is obtained as a result, where all interior and exterior regions, together with the singularity and barrier sets that separate them, get clearly identified. The method can deal with open- or closed-chain manipulators, whether planar or spatial, and is able to take joint limits into account. Advantages over previous general methods based on continuation include the ability to converge to all boundary points, even in higher dimensional cases, and the fact that manual guidance with *a priori* knowledge of the workspace is not required. Examples are included that show the performance of the method on benchmark problems documented in the literature, as well as on new ones unsolved so far.

**Index Terms**—Branch-and-prune method, closed-chain, kinematics, linear relaxation, mechanism design, multibody system, parallel robot, workspace determination.

## I. INTRODUCTION

**A** MAIN problem of robot kinematics is to determine, for a given manipulator, the complete set of poses that its end-effector can adopt, as the manipulator runs through all possible configurations. The determination of such set, usually known as the *workspace* or *accessible output set*, has received substantial attention in the literature, as it finds applications to mechanism design, path planning, and task execution in robotics [2]–[9], as well as to related problems in other domains [10]–[12].

Since the early studies on the problem [13]–[17], efficient workspace determination methods have been given, but most of them are tailored to a particular robot architecture, or class of architectures. An important group of such methods adopts a constructive geometric approach. Representative of them is [18], which computes the constant-orientation workspace of a spatial parallel manipulator, [19], which extends the approach to deal with other physical constraints, or [20], which provides meth-

ods for various planar parallel platforms. Other significant approaches include interval analysis or discretization techniques for Gough-type manipulators [21], [22], optimization-based algorithms for fully serial or parallel robots [4], analytic methods for symmetrical spherical mechanisms [23], and analytic, topologic, or algebraic-geometric procedures for serial manipulators [24]–[28].

The literature on the topic is rich, and elaborate surveys can be found in [2]–[4] and [11], but it should be noted that most previous methods, including [13]–[28], are difficult to apply to manipulators outside the class they consider, because they exploit specificities of the class in some way or another, like the possibility to parametrize the end-effector pose [24], [29], simplifications introduced by dimension symmetries [23], or the feasibility of an algebraic [24]–[28] or geometric [18]–[20] treatment. Even discretization methods, which resort to configuration sampling only [17], [22], [30], rely on the assumption that either the forward or the inverse kinematics problems have a simple solution, which need not be the case in general.

While *ad hoc* methods are desirable because they tend to yield faster or simpler algorithms in particular cases, methods for general structure manipulators are required too, to be able to analyze robots for which no specific solution exists. This paper presents one such method, applicable to planar or spatial multibody systems formed by rigid links and lower pair joints, possibly involving closed kinematics chains, on which particular dimensions, simple forward/inverse kinematics solutions, or the existence of end-effector pose parameterizations can, in principle, not be exploited.

To the authors' best knowledge, only one approach of a similar generality is available in the literature, due to Haug and collaborators [31, and references therein]. In their work, Haug *et al.* apply the idea that the workspace boundaries can be extracted from a set of output singularities, and focus their effort on tracing such singularities numerically using a continuation method. The procedure is elegant and works well in favorable situations, but important weaknesses of the method were identified in [32], including 1) the need to manually guide the method with *a priori* knowledge of the workspace shape, 2) the fact that only cross-sectional curves of the boundary can be traced in higher dimensional cases, or 3) the possibility to miss some boundary segments in the presence of voids within the workspace. Unfortunately, as it will be shown in this paper, the performance of the procedure can further degrade, since one can encounter workspaces with several connected components, hidden regions, or degenerate barriers where the method will produce incomplete or misleading maps of the workspace.

Manuscript received December 15, 2011; accepted April 17, 2012. Date of publication June 25, 2012; date of current version September 28, 2012. This paper was recommended for publication by Associate Editor Y. Choi and Editor W. K. Chung upon evaluation of the reviewers' comments. This work is a revised and extended version of [1], and it was funded in part by the Spanish Government under Contract DPI2010-18449 and by a Juan de la Cierva contract supporting the second author.

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC 08028 Barcelona, Spain (e-mail: obohigas@iri.upc.edu; mmanuben@iri.upc.edu; ros@iri.upc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2012.2196311

In contrast, the method we propose in this paper is complete, in the sense that it is able to isolate all boundary and barrier points of the workspace without guidance, even when voids or degenerate barriers are present. The method, moreover, does not require *a priori* knowledge of the workspace in any form, and is not limited to compute cross-sectional curves of the boundary only. By design, it can isolate the boundary of workspaces of any dimension, with the sole limitations imposed by the curse of dimensionality. The method is based on characterizing a set of singularities relative to the output coordinates with a system of equations of an adequate form, and then exploiting this form to compute all singular points at a required precision, using a numerical technique based on linear relaxations [33]. The singular points are then classified according to whether they correspond to actual motion impediments for the manipulator, obtaining a detailed map of the workspace where interior and exterior regions, and the singularity sets that separate them, get clearly identified.

The rest of the paper is structured as follows. Section II provides working definitions, assumptions, and necessary background for the paper. Section III reviews the continuation method in [31] and identifies a number of situations in which it fails to properly determine the boundary of the workspace. Section IV describes the alternative method we propose, which is robust to such situations. Section V illustrates the performance of the method on several cases, including problematic ones like those described in Section III. Section VI makes some remarks regarding the application of the method to non-smooth manifolds. Section VII, finally, concludes the paper and discusses points that require further attention.

## II. PRELIMINARIES

### A. Basic Definitions and Assumptions

The allowable positions and orientations of all links in a manipulator are usually encoded in a vector  $\tilde{\mathbf{q}}$  of generalized coordinates, subject to a system of equations

$$\tilde{\Phi}(\tilde{\mathbf{q}}) = \mathbf{0} \quad (1)$$

expressing the assembly constraints imposed by the joints [34]. Here,  $\tilde{\Phi}(\tilde{\mathbf{q}})$  is a smooth vector function, and (1) is meant to include all possible assembly constraints, including those due to mechanical limits on the joints, which can also be modeled as equality constraints (see Appendix A).

To analyze the functionality of the manipulator, a vector  $\mathbf{u}$  of  $n_u$  output coordinates is usually defined, encompassing Cartesian coordinates and/or orientation angles of the end-effector. Depending on the specific formulation adopted for (1), such coordinates may or may not be explicit in  $\tilde{\mathbf{q}}$  but, in the latter case, it is always possible to transform (1) into an equivalent system of  $n_e$  equations

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (2)$$

using appropriate manipulations, where  $\mathbf{q}$  is an alternative vector of  $n_q \geq n_e$  generalized coordinates now including  $\mathbf{u}$  explicitly. The vectors  $\mathbf{q}$ ,  $\mathbf{u}$ , and  $\Phi(\mathbf{q})$  take values within smooth manifolds  $\mathcal{Q}$ ,  $\mathcal{U}$ , and  $\mathcal{E}$ , of dimensions  $n_q$ ,  $n_u$ , and  $n_e$ , respectively.

By adopting the partition  $\mathbf{q} = [\mathbf{z}^T, \mathbf{u}^T]^T$ , where  $\mathbf{z}$  accumulates all coordinates in  $\mathbf{q}$ , except those in  $\mathbf{u}$ , (2) can now be written as

$$\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}, \quad (3)$$

and the workspace of the system relative to the  $\mathbf{u}$  coordinates can be defined as the set  $\mathcal{A}$  of points  $\mathbf{u} \in \mathcal{U}$  that satisfy (3) for some  $\mathbf{z}$ .

Let  $\mathcal{C}$  denote the configuration space of the manipulator, i.e.,  $\mathcal{C} = \{\mathbf{q} : \Phi(\mathbf{q}) = \mathbf{0}\}$ . Throughout the paper, we shall assume that  $\Phi(\mathbf{q}) : \mathcal{Q} \rightarrow \mathcal{E}$  is a smooth function whose Jacobian

$$\Phi_{\mathbf{q}} = [\partial\Phi_i / \partial q_j]$$

is full rank at all points  $\mathbf{q} \in \mathcal{C}$  so that  $\mathcal{C}$  will be a smooth manifold of dimension  $d = n_q - n_e$ , but Section VI provides hints on how to deal with the general case. We shall also assume that  $n_u \leq d$ , and that  $\mathcal{A}$  is a subset of  $\mathcal{U}$  of dimension  $n_u$ , which is the common situation in general. In particular,  $n_u \leq d$  implies that  $n_z \geq n_e$  so that the system in (3) is, in principle, not overconstrained for a fixed value of  $\mathbf{u}$ .

Although a direct computation of  $\mathcal{A}$  could be attempted, it is often more effective to delimit  $\mathcal{A}$  by computing its boundary, which is a set of lower dimension. A point  $\mathbf{u}$  lies on the boundary of  $\mathcal{A}$ , denoted  $\partial\mathcal{A}$ , if every open set of  $\mathcal{U}$  containing  $\mathbf{u}$  intersects the interior and the exterior of  $\mathcal{A}$ . The following known conditions will be used to isolate such points [31].

### B. Singularity Conditions and Barrier Determination

Let  $\pi_{\mathbf{u}} : \mathcal{Q} \rightarrow \mathcal{U}$  denote the projection map from  $\mathbf{q}$  onto the  $\mathbf{u}$  variables; i.e.,  $\pi_{\mathbf{u}}(\mathbf{z}, \mathbf{u}) = \mathbf{u}$ . Observe that  $\mathcal{A}$  is exactly the image of  $\mathcal{C}$  through  $\pi_{\mathbf{u}}$ . As shown in Appendix B, moreover, the Jacobian matrix  $\Phi_{\mathbf{z}} = [\partial\Phi_i / \partial z_j]$  must be rank deficient at the points  $\mathbf{q} \in \mathcal{C}$  that project onto some  $\mathbf{u} \in \partial\mathcal{A}$ . Geometrically, these points correspond to critical points of the projection of  $\mathcal{C}$  onto  $\mathcal{U}$ , i.e., to points  $\mathbf{q} \in \mathcal{C}$  where the projection of the tangent space of  $\mathcal{C}$  does not span the tangent space of  $\mathcal{U}$  at  $\mathbf{u} = \pi_{\mathbf{u}}(\mathbf{q})$ , which thus constitute configurations in which the manipulator loses instantaneous mobility relative to the  $\mathbf{u}$  variables. The set  $\mathcal{S}$  of all such critical points will be called the *singularity set* hereafter, and the notation  $\pi_{\mathbf{u}}(\mathcal{S})$  will be used to refer to the projection of  $\mathcal{S}$  onto  $\mathcal{U}$ . Fig. 1(a) illustrates the situation.

A preliminary idea of how the workspace boundary would look like, thus, can be gained by computing all points  $\mathbf{q} \in \mathcal{C}$  for which  $\Phi_{\mathbf{z}}$  is rank deficient, and projecting them to the  $\mathbf{u}$  variables in order to obtain  $\pi_{\mathbf{u}}(\mathcal{S})$ . However, note from the examples in Fig. 1(b) and 1(c) that the rank deficiency of  $\Phi_{\mathbf{z}}$  is a necessary but not sufficient condition for  $\pi_{\mathbf{u}}(\mathbf{q})$  to lie in  $\partial\mathcal{A}$ , as there can be critical points projecting on the interior of  $\mathcal{A}$  too. In fact, as illustrated in the Fig. 1, points  $\mathbf{q}$  where  $\Phi_{\mathbf{z}}$  is rank deficient can be classified into two broad categories. They can be *traversable* or *barrier* singularities depending on whether there exists a trajectory on  $\mathcal{C}$  through  $\mathbf{q}$  whose projection on  $\mathcal{U}$  traverses  $\pi_{\mathbf{u}}(\mathcal{S})$  for each neighborhood of  $\mathbf{q}$  on  $\mathcal{C}$  [3], [35]. Points corresponding to barrier singularities can, in turn, be

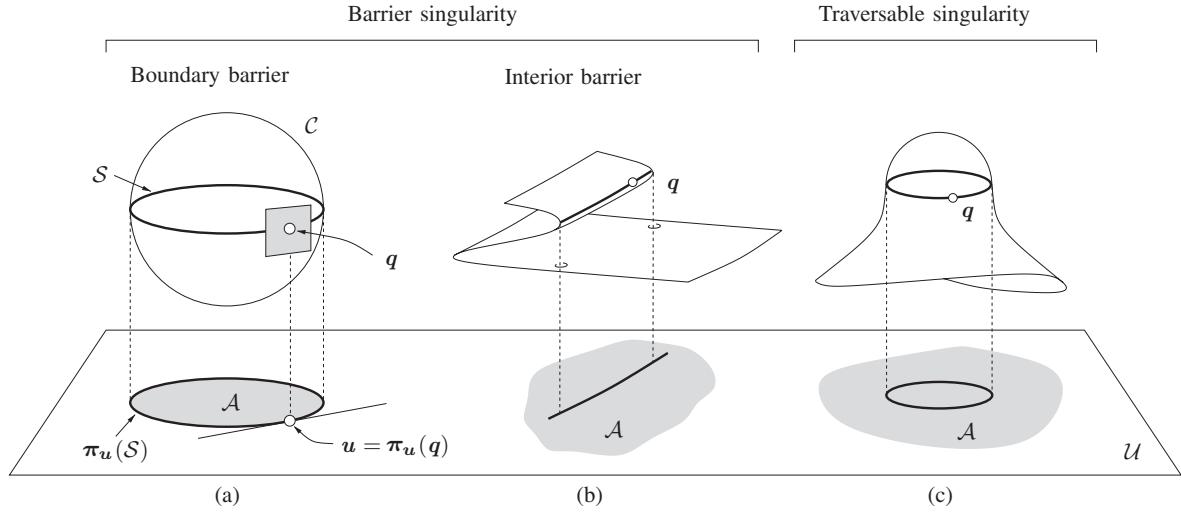


Fig. 1. (a) Sets  $\mathcal{S}$  and  $\pi_{\mathbf{u}}(\mathcal{S})$  when  $\mathcal{Q} = \mathbb{R}^3$ ,  $\mathcal{U} = \mathbb{R}^2$ ,  $\mathcal{C}$  is the sphere  $x^2 + y^2 + z^2 = 1$ , and  $\pi_{\mathbf{u}}$  is the projection map  $f(x, y, z) = (x, y)$ . The workspace relative to the  $(x, y)$  coordinates is the projection of the sphere onto the  $(x, y)$  plane, and the boundaries of such projection necessarily correspond to points on the sphere where the tangent plane projects onto a line of  $\mathbb{R}^2$ . (b) and (c)  $\pi_{\mathbf{u}}(\mathcal{S})$  can also lie in the interior of  $\mathcal{A}$ .

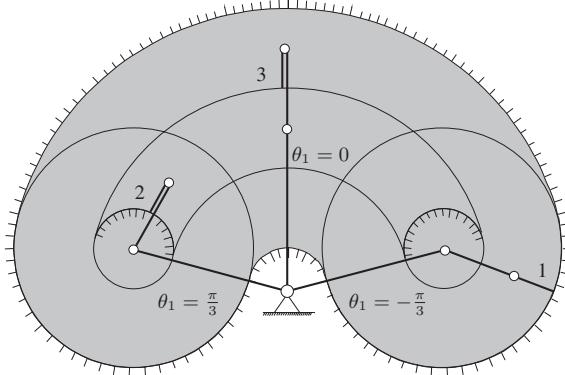


Fig. 2. Workspace of a planar 3R manipulator relative to the  $(x, y)$  coordinates of the tip point of the last link, assuming that the angle  $\theta_1$  of the first joint is restricted to the  $[-\pi/3, \pi/3]$  range. Points corresponding to singularities are indicated in solid lines, and those on boundary and interior barriers are indicated with normal vectors on the forbidden side. Configurations 1–3 correspond to a boundary barrier, an interior barrier, and a traversable singularity, respectively.

classified as *boundary* or *interior* barriers, according to whether they occur over  $\partial\mathcal{A}$  or over the interior of  $\mathcal{A}$ , respectively. An example of each one of these singularities is depicted in Fig. 2 for a planar 3R manipulator.

As noted in [31], [32], and [35], a workspace determination method should ideally detect *all* barriers in the workspace, either interior or on the boundary, as such barriers constitute true motion impediments for the end-effector. A criterion to determine whether a point  $\mathbf{q}_0 = [\mathbf{z}_0^\top, \mathbf{u}_0^\top]^\top \in \mathcal{S}$  corresponds to a barrier or a traversable singularity was given in [31], based on the following idea. Let  $\mathbf{q} = \mathbf{q}(\mathbf{v})$  be a smooth parametrization of  $\mathcal{C}$  in a neighborhood of  $\mathbf{q}_0$ , where  $\mathbf{v}$  is a vector of  $d$  parameters and  $\mathbf{q}_0 = \mathbf{q}(\mathbf{v}_0)$  for some  $\mathbf{v}_0 \in \mathbb{R}^d$ . Since  $\Phi_q$  is full rank for all  $\mathbf{q} \in \mathcal{C}$ , the implicit function theorem guarantees that

such a parametrization exists [36]. If  $\mathbf{n}_0$  is the normal vector to  $\pi_{\mathbf{u}}(\mathcal{S})$  at  $\mathbf{u}_0$ , we can check whether  $\mathbf{q}_0$  corresponds to a barrier singularity by examining the sign of

$$\psi(\mathbf{v}) = \mathbf{n}_0^\top (\mathbf{u}(\mathbf{v}) - \mathbf{u}_0) \quad (4)$$

for all local trajectories  $\mathbf{v} = \mathbf{v}(t)$  crossing  $\mathbf{v}_0$  for some time  $t = t_0$  whose corresponding path  $\mathbf{u} = \mathbf{u}(t)$  is orthogonal to  $\pi_{\mathbf{u}}(\mathcal{S})$  at  $\mathbf{u}_0$ . This sign is mostly determined by a quadratic form corresponding to the second-order term of the Taylor expansion of (4). If the form is positive- or negative-definite, then all paths orthogonal to  $\pi_{\mathbf{u}}(\mathcal{S})$  lie on one side of  $\pi_{\mathbf{u}}(\mathcal{S})$  and  $\mathbf{q}_0$  is a barrier singularity. If the form is indefinite, there are paths in  $\mathcal{A}$  that cross  $\pi_{\mathbf{u}}(\mathcal{S})$  and  $\mathbf{q}_0$  is a traversable singularity. Finally, if this form is semidefinite, we cannot deduce the singularity type unless we examine higher order terms of the Taylor expansion. However, the latter case only occurs on zero-measure subsets of  $\mathcal{S}$  generally.

### III. ISSUES OF THE CONTINUATION METHOD

In order to see the advantages of our approach in comparison with the continuation method in [31], this method is next reviewed briefly, identifying a number of cases where it fails to produce complete maps of the workspace.

We note first that the method in [31] relies on 1-D path tracking procedures, and hence it can only trace  $\partial\mathcal{A}$  explicitly on 1-D boundaries, i.e., when  $n_u = 2$ . Fig. 3(a) explains the method on a simple setting in which  $\mathcal{Q} = \mathbb{R}^3$ ,  $\mathcal{U} = \mathbb{R}^2$ , and  $\mathcal{C}$  contains two connected components,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , which project onto a workspace  $\mathcal{A}$  with two regions. The method begins with a manually provided configuration of the manipulator  $\mathbf{q}_i = [\mathbf{z}_i^\top, \mathbf{u}_i^\top]^\top$ , with  $\mathbf{u}_i \in \mathcal{A}$ , shoots a ray through  $\mathbf{u}_i$  in  $\mathcal{U}$  on an arbitrary direction, and traces this ray until a point  $\mathbf{u}_b \in \partial\mathcal{A}$  is found. The movement along this ray is achieved by continuation of the corresponding trajectory on  $\mathcal{C}$ , i.e., by iteratively

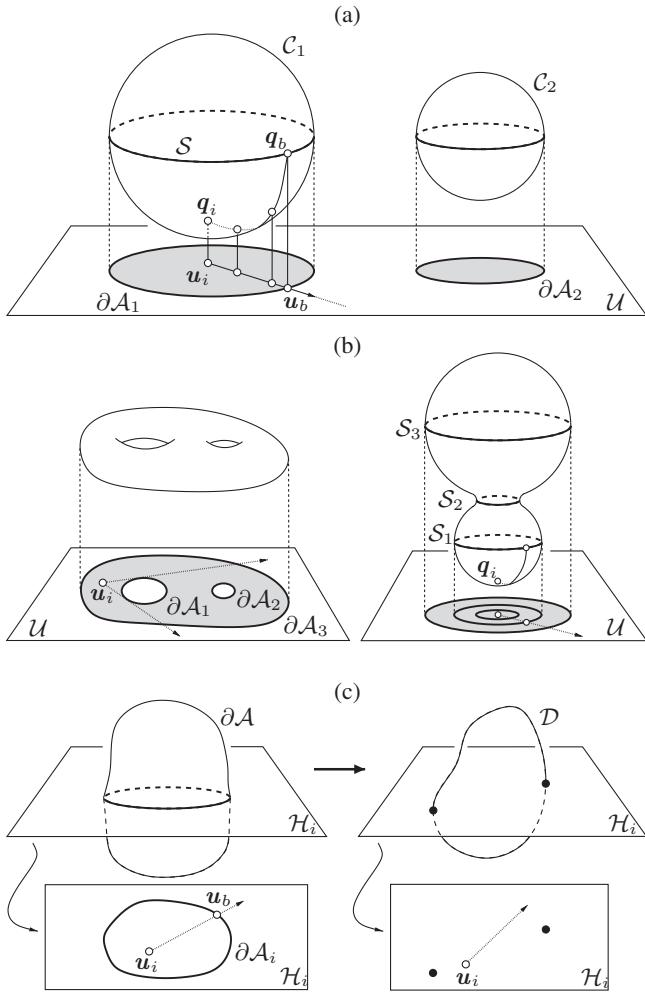


Fig. 3. Performance of the continuation method in multi-component workspaces (a), hidden regions (b), and degenerate barriers (c).

solving  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$  using a Newton method for  $\mathbf{u}$  fixed to discrete values along the ray. This process is repeated until a value of  $\mathbf{u}$  outside  $\mathcal{A}$  is found, which is detected because the Newton method fails to converge, and a dichotomy-search process is then performed locally to find a point  $q_b \in \mathcal{C}$  lying on  $\mathcal{S}$ . A second continuation process is then launched from  $q_b$  to find the connected component of  $\mathcal{S}$  that is reachable from such point, by solving a system of equations that express the rank deficiency of  $\Phi_z$ . Once  $\mathcal{S}$  has been found, the points of  $\pi_{\mathbf{u}}(\mathcal{S})$  are computed by projection, and those corresponding to barrier or traversable singularities are finally detected through second-order analysis.

Because path tracking methods based on continuation are fast and robust to bifurcations [37], this approach will rapidly determine  $\mathcal{S}$  in favorable cases. However, one encounters the following situations in which the method will fail to identify  $\mathcal{A}$  completely.

- 1) *Multicomponent workspaces:* Difficulties arise, for example, when computing  $\partial\mathcal{A}$  on workspaces with several connected components, like the one in Fig. 3(a). Independently of the chosen direction for the ray, note that the previous process will certainly hit  $\partial\mathcal{A}_1$ , but not

$\partial\mathcal{A}_2$ , because the tracking of the ray cannot continue beyond  $\partial\mathcal{A}_1$  using continuation. To converge to all boundary curves, the previous strategy should at least be fed with one point on each connected component of  $\mathcal{C}$ , but no satisfactory method has been given yet to compute such points in general, to the best of our knowledge.

- 2) *Hidden regions:* The continuation method may seem to be able to compute, at least, the boundary of the workspace component to which  $q_i$  belongs, but this is not the case in general. Note that  $\mathcal{S}$  may itself have several connected components, and some of such components could be missed depending on the position of  $u_i$ , even if rays on all possible directions were shot. In Fig. 3(b), left, for example, the continuation method may be able to find  $\partial\mathcal{A}_1$  and  $\partial\mathcal{A}_3$  from  $u_i$ , but not  $\partial\mathcal{A}_2$ , because  $\partial\mathcal{A}_2$  is hidden behind  $\partial\mathcal{A}_1$ . The problem also arises on workspaces with interior barriers, as seen on the right side of Fig. 3(b). Starting the continuation from  $q_i$  allows hitting the interior barrier corresponding to  $\mathcal{S}_1$ , but not the boundary barrier corresponding to  $\mathcal{S}_3$ , thus ignoring a full annular region that actually lies inside the workspace.
- 3) *Degenerate barriers:* When  $\partial\mathcal{A}$  has dimension two or higher, the continuation method slices  $\partial\mathcal{A}$  through hyperplanes  $\mathcal{H}_i$  in order to obtain 1-D curves  $\partial\mathcal{A}_i$  trackable by the method (see the left side of Fig. 3(c)). On manipulators of a special geometry, however, portions of  $\partial\mathcal{A}$  can degenerate into lower dimensional barriers  $\mathcal{D}$ , thus making the slices  $\mathcal{H}_i$  contain isolated points only (see the right side of Fig. 3(c)). The method will clearly miss the lower dimensional barriers in such a situation because the ray-shooting technique will fail to meet them with probability one, independently of the location of  $u_i$ .

As it will be shown in Section V, examples of multicomponent workspaces, hidden regions, and degenerate barriers occur easily on real manipulators and, thus, there is a clear need to develop alternative methods which are robust to such situations. The next section provides one such method.

#### IV. LINEAR RELAXATION METHOD

The proposed method consists in first formulating a system of quadratic equations that characterize  $\mathcal{S}$  (see Section IV-A), then applying a numerical technique that exploits the structure of this system to isolate  $\mathcal{S}$  (see Section IV-B), and finally using a local method to classify the points of  $\mathcal{S}$  as boundary barriers, interior barriers, or traversable singularities (see Section IV-C).

##### A. Equation Formulation

For a manipulator involving lower pairs of any kind, except the helical pair, the formulation proposed in [33] makes (1) adopt the form of a polynomial system of *quadratic* equations (i.e., equations where only monomials of the form  $a$ ,  $a^2$ , or  $ab$  intervene, where  $a$  and  $b$  refer to any two of the variables). We next show that, by adopting a formulation of such kind, it is possible to extend (1) with additional variables and equations,

to obtain a new system

$$\chi(\mathbf{y}) = \mathbf{0} \quad (5)$$

that characterizes the points of  $\mathcal{S}$ . This system will be shown to be quadratic too, which will allow us to define a conceptually simple technique to compute  $\mathcal{S}$  numerically.

To explain how (5) can be derived, we shall distinguish two situations, depending on whether the  $\mathbf{u}$  variables appear explicitly in  $\tilde{\mathbf{q}}$ , or are only determined implicitly by some of the variables in  $\tilde{\mathbf{q}}$ , both illustrated in Section V on particular examples. In all cases, it will be assumed that (1) has been formulated following [33], with  $\tilde{\mathbf{q}}$  containing the pose of the end-effector encoded as a vector  $[\mathbf{p}^T, \mathbf{r}^T]^T$ , where  $\mathbf{p}$  and  $\mathbf{r}$ , respectively, encompass the Cartesian coordinates of a point on the effector, and the components of a rotation matrix giving the orientation of the effector.

*1) Dealing With Explicit Output:* Assume, initially, that  $\tilde{\mathbf{q}}$  explicitly contains  $\mathbf{u}$ . This occurs in positional workspaces, i.e., when  $\mathbf{u}$  includes part or all of the variables in  $\mathbf{p}$ . Then, by adopting the partition  $\tilde{\mathbf{q}} = [\mathbf{z}^T, \mathbf{u}^T]^T$ , (1) can be directly cast into the form of (3) so that  $\mathcal{S}$  will be the set of points  $\mathbf{q} = [\mathbf{z}^T, \mathbf{u}^T]^T$  satisfying  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$  for which  $\Phi_z$  is rank deficient, i.e., the set of points  $\mathbf{q}$  that satisfy

$$\left. \begin{array}{l} \Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0} \\ \Phi_z^T \xi = \mathbf{0} \\ \xi^T \xi = 1 \end{array} \right\} \quad (6)$$

for some  $\xi$ , where  $\xi$  is an  $n_e$ -dimensional vector of unknowns. Clearly, the first equation in (6) constrains  $\mathbf{q}$  to be a valid configuration, and the second and third equations impose the rank deficiency of  $\Phi_z$ . This matrix is of size  $n_e \times n_z$ , with  $n_z \geq n_e$ , and hence it will be rank deficient whenever there exists a linear combination of its rows with non-null coefficients. Since (1) is quadratic, (6) will be quadratic too, because all entries in  $\Phi_z^T$  will be linear terms, and  $\xi^T \xi$  is directly a quadratic expression. Thus,  $\chi(\mathbf{y}) = \mathbf{0}$  adopts the form of (6) in this case, with  $\mathbf{y} = [\mathbf{z}^T, \mathbf{u}^T, \xi^T]^T$ .

*2) Dealing With Implicit Output:* There are situations in which the  $\mathbf{u}$  variables do not all intervene in  $\tilde{\mathbf{q}}$  but, instead, they can be related to a subset  $\tilde{\mathbf{u}}$  of  $n_{\tilde{u}}$  variables in  $\tilde{\mathbf{q}}$ , through a smooth function of the form

$$\tilde{\mathbf{u}} = \mu(\mathbf{u}). \quad (7)$$

This occurs whenever  $\mathbf{u}$  contains orientation angles of the end-effector. Since [33] represents end-effector orientations by rotation matrices, orientation angles are only related implicitly to the components of  $\mathbf{r}$  through a parametrization of the special orthogonal group under consideration [ $SO(2)$  or  $SO(3)$ ], depending on whether the manipulator is planar, or spatial]. In order to transform (1) into the form of (3) it will be possible, in such situations, to consider the partition  $\tilde{\mathbf{q}} = [\tilde{\mathbf{z}}^T, \tilde{\mathbf{u}}^T]^T$ , and subdivide (1) into two subsystems as follows:

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \eta(\tilde{\mathbf{u}}) = \mathbf{0} \end{array} \right\}, \quad (8)$$

where  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  is a subsystem of equations whose solution set can be globally parametrized by (7), and  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0}$  collects

the rest of equations. Since  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  parametrizes the solution set of  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$ ,  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  can be replaced by  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  in (8), obtaining the equivalent system

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \tilde{\mathbf{u}} = \mu(\mathbf{u}) \end{array} \right\} \quad (9)$$

which now contains  $\mathbf{u}$  explicitly. Therefore, (3) adopts the form of (9) in this case, with  $\mathbf{z} = [\tilde{\mathbf{z}}^T, \tilde{\mathbf{u}}^T]^T$ , and

$$\Phi(\mathbf{z}, \mathbf{u}) = \left[ \begin{array}{c} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) \\ \tilde{\mathbf{u}} - \mu(\mathbf{u}) \end{array} \right], \quad (10)$$

so that  $\mathcal{S}$  will be the set of points  $\mathbf{q} = [\mathbf{z}^T, \mathbf{u}^T]^T$  satisfying (9) for which  $\Phi_z$  is rank deficient. Note however that, because of the form of (10),  $\Phi_z$  has the block structure

$$\Phi_z = \left[ \begin{array}{ccc} \Psi_{\tilde{\mathbf{z}}} & \cdots & \Psi_{\tilde{\mathbf{u}}} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \vdots & \mathbf{I}_{n_{\tilde{u}}} \end{array} \right]$$

in this case, where  $\mathbf{I}_{n_{\tilde{u}}}$  is the  $n_{\tilde{u}} \times n_{\tilde{u}}$  identity matrix, so that  $\Phi_z$  will be rank deficient if, and only if, its upper-left block  $\Psi_{\tilde{\mathbf{z}}}$  is rank deficient. Hence,  $\mathcal{S}$  can be characterized as the set of points  $\mathbf{q} = [\tilde{\mathbf{z}}^T, \tilde{\mathbf{u}}^T, \mathbf{u}^T]^T$  that satisfy

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \tilde{\mathbf{u}} = \mu(\mathbf{u}) \\ \Psi_{\tilde{\mathbf{z}}}^T \xi = \mathbf{0} \\ \xi^T \xi = 1 \end{array} \right\}, \quad (11)$$

for some  $\xi$ , where  $\xi$  is a new vector with the appropriate size.

While we could now proceed to isolate  $\mathcal{S}$  by solving (11),  $\mu(\mathbf{u})$  usually introduces trigonometric terms that complicate the solution. Fortunately, since  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  parametrizes the solution set of  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$ , and the  $\mathbf{u}$  variables only intervene in the second equation of (11), we can substitute  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  for  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  in (11), arriving at the equivalent system

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \eta(\tilde{\mathbf{u}}) = \mathbf{0} \\ \Psi_{\tilde{\mathbf{z}}}^T \xi = \mathbf{0} \\ \xi^T \xi = 1 \end{array} \right\}, \quad (12)$$

which is quadratic, because  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}})$  and  $\eta(\tilde{\mathbf{u}})$  are quadratic under the adopted formulation. Thus,  $\chi(\mathbf{y}) = \mathbf{0}$  adopts the form of (12) in this case, with  $\mathbf{y} = [\tilde{\mathbf{z}}^T, \tilde{\mathbf{u}}^T, \xi^T]^T$ .

### B. Equation Solution

A numerical method that is able to solve (5) is next described, based on expanding the equations into a canonical form, and then using a linear relaxation method exploiting this form to isolate the solutions [33].

Let  $y_i$  and  $y_j$  refer to any two variables appearing in (5). The method starts by introducing the changes of variables

$$p_i = y_i^2, \quad (13)$$

$$b_k = y_i y_j, \quad (14)$$

for all  $y_i y_j$  and  $y_i^2$  monomials intervening in (5). This allows the transformation of the system into the expanded form

$$\begin{aligned} \Lambda(\mathbf{x}) &= \mathbf{0} \\ \Omega(\mathbf{x}) &= \mathbf{0} \end{aligned} \quad (15)$$

where  $\mathbf{x}$  is an  $n_x$ -dimensional vector including the original  $\mathbf{y}$  variables, and the newly introduced  $p_i$  and  $b_k$  ones,  $\Lambda(\mathbf{x}) = \mathbf{0}$  is a collection of linear equations in  $\mathbf{x}$ , and  $\Omega(\mathbf{x}) = \mathbf{0}$  is a collection of equations adopting one of the two forms  $x_k = x_i^2$ ,  $x_k = x_i x_j$ , which correspond to the variable changes in (13) and (14).

It can be seen that each variable  $x_i$  of  $\mathbf{x}$  can only take values within a prescribed interval [33] so that from the Cartesian product of all such intervals one can define an initial  $n_x$ -dimensional box  $\mathcal{B}$  that bounds all solutions of (15). The algorithm to isolate such solutions recursively applies two operations on  $\mathcal{B}$ : *box shrinking* and *box splitting*.

Using box shrinking, portions of  $\mathcal{B}$  that contain no solution are eliminated by narrowing some of its defining intervals. This process is repeated until either 1) the box is reduced to an empty set, in which case it contains no solution, or 2) the box is “sufficiently” small, in which case it is considered a *solution* box, or 3) the box cannot be “significantly” reduced, in which case, it is bisected into two subboxes via box splitting (which simply bisects its largest interval). To converge to all solutions, the whole process is recursively applied to the new subboxes until one obtains a collection of solution boxes whose side lengths are below a given threshold  $s^{\max}$ .

The crucial operation in this scheme is box shrinking, which is implemented as follows. Notice first that the solutions falling in some subbox  $\mathcal{B}_c \subseteq \mathcal{B}$  must lie in the linear variety defined by  $\Lambda(\mathbf{x}) = \mathbf{0}$ . 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 two linear programs

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

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

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  $\Omega(\mathbf{x}) = \mathbf{0}$ . These equations can be taken into account by noting that, if  $\mathcal{B}_c$  is adjusted to the graph of the considered equation, and  $[l_i, u_i]$  denotes the interval of  $\mathcal{B}_c$  along dimension  $x_i$ , then we have the following.

- 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  $A_3$  are the points where the parabola intercepts the lines  $x_i = l_i$  and  $x_i = u_i$ , and  $A_2$  is the point where the tangent lines at  $A_1$  and  $A_3$  meet [see Fig. 4(a)].
- 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

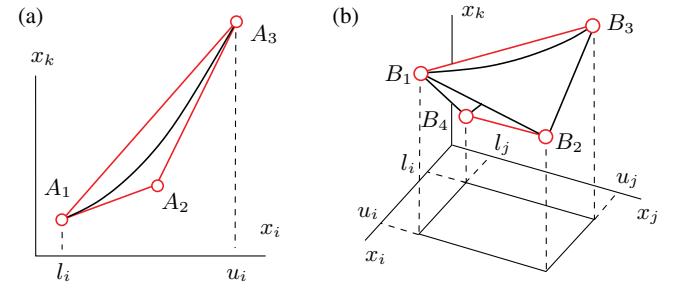


Fig. 4. Polytope bounds within box  $\mathcal{B}_c$ . (a) Points on the parabola  $x_k = x_i^2$  are bound by the triangle  $A_1 A_2 A_3$ . (b) Points on the hyperbolic paraboloid  $x_k = x_i x_j$  are bound by the tetrahedron  $B_1 B_2 B_3 B_4$ .

the rectangle  $[l_i, u_i] \times [l_j, u_j]$  vertically to the paraboloid [see Fig. 4(b)].

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 even 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 is returned as output upon termination, and it is said to form a *box approximation* of the solution set of (15), because the boxes form a discrete envelope of such set, whose accuracy can be adjusted through the  $s^{\max}$  parameter. The algorithm is complete, in the sense that the boxes in  $B$  include all solution points of (15), and hence it will succeed in isolating all points of  $\mathcal{S}$  accurately, provided that a small-enough value for  $s^{\max}$  is used. Detailed properties of the algorithm, including an analysis of its completeness, correctness, and convergence order, are given in [33].

Having obtained  $B$ , it is finally straightforward to obtain a box approximation  $B^S$  of the singular set  $\mathcal{S}$ . If (5) adopts the form of (6), then  $z$  and  $u$  explicitly intervene in (15) as part of  $\mathbf{x}$ , and each box in  $B$  already has ranges along the  $\mathbf{q} = [\mathbf{z}^\top, \mathbf{u}^\top]^\top$  dimensions. Such ranges define a box in  $\mathbf{q}$ -space enclosing points of  $\mathcal{S}$ , and the collection of all of such boxes provides  $B^S$ . If (5) adopts the form of (12),  $B^S$  can be obtained in a similar way. The only difference is that, because the  $u$  variables do not intervene in (12), the boxes in  $B$  do not provide explicit ranges for them. However, if for each box in  $B$  we consider the ranges along the  $\tilde{u}$  variables, we can derive corresponding ranges for the  $u$  variables by solving  $\tilde{u} = \mu(u)$  using interval techniques [33], [38].

### C. Boundary Identification

Once  $B^S$  has been obtained, it remains to check whether the points of  $\mathcal{S}$  enclosed in  $B^S$  correspond to boundary barriers, interior barriers, or traversable singularities. This classification is performed in two stages. This classification, which is illustrated in Fig. 5, is performed in two stages.

In the first stage, we classify the boxes of  $B^S$  according to whether they enclose barrier or traversable singularities. For

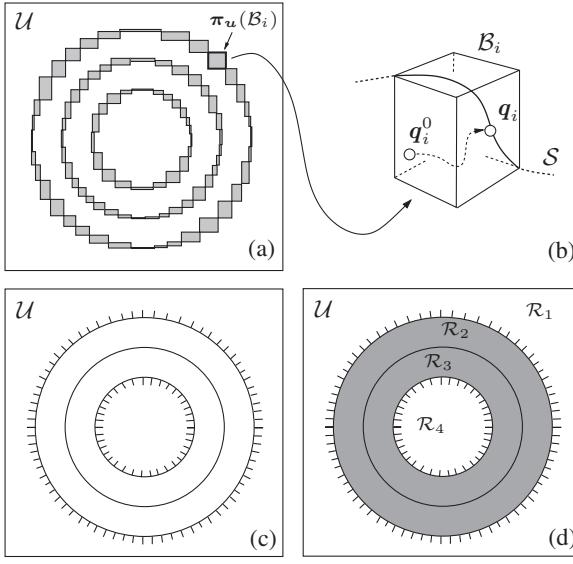


Fig. 5. Boundary identification process. (a) Box approximation of  $\mathcal{S}$  projected onto the output space  $\mathcal{U}$ . (b) Computation of  $q_i \in \mathcal{S}$  for each box  $\mathcal{B}_i$ . (c) Classification of the points of  $\pi_{\mathbf{u}}(\mathcal{S})$  into barrier or traversable singularities. (d) Regions into which  $\pi_{\mathbf{u}}(\mathcal{S})$  subdivides  $\mathcal{U}$  and their classification into interior (grey) or exterior (white) regions.

each box  $\mathcal{B}_i \in \mathcal{B}^{\mathcal{S}}$  [Fig. 5(a)], a point  $q_i \in \mathcal{S}$  is computed [Fig. 5(b)], and the barrier determination method mentioned in Section II-B is applied to this point. The computation of  $q_i$  is done by solving (6) or (11), depending on the situation, using the Newton–Raphson method starting from an arbitrary point  $q_i^0$  inside  $\mathcal{B}_i$ . This procedure will quadratically converge to some point  $q_i \in \mathcal{S}$  provided that the points within  $\mathcal{B}_i$  are close enough to  $\mathcal{S}$ , which can be guaranteed by computing  $\mathcal{B}^{\mathcal{S}}$  using a small-enough  $s^{\max}$  threshold. The singularity type obtained for  $q_i$  (either barrier or traversable singularity) is taken as an estimation of the singularity type of all points in  $\mathcal{B}_i \cap \mathcal{S}$  so that after repeating this process for all of the boxes in  $\mathcal{B}^{\mathcal{S}}$ , it is possible to subdivide  $\mathcal{S}$  into subsets of constant singularity type. If  $q_i$  is a barrier singularity, a normal vector  $\mathbf{n}_i$  pointing toward the forbidden side of the barrier is drawn at  $\mathbf{u}_i = \pi_{\mathbf{u}}(q_i)$  [Fig. 5(c)].

In the second stage, we determine which of the barrier points  $q_i$  computed in the previous stage correspond to boundary or interior barriers. To this end, notice that  $\pi_{\mathbf{u}}(\mathcal{S})$  subdivides  $\mathcal{U}$  into several regions  $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$ , where each region fully lies in the interior or in the exterior of  $\mathcal{A}$ , and a barrier point  $\mathbf{u}_i$  will lie on  $\partial\mathcal{A}$  if, and only if, one of its two neighboring regions is exterior to  $\mathcal{A}$ . Thus, determining which of the barrier points  $q_i$  correspond to boundary barriers boils down to checking whether the regions  $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$  are interior or exterior to  $\mathcal{A}$ . The type of a region  $\mathcal{R}_j$  can be determined by selecting a point  $\mathbf{u}_j$  in the region, and solving  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$  for that point, which is here done by resorting to the numerical technique proposed in [33]. If  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$  has at least one solution, then  $\mathcal{R}_j$  is an interior region; otherwise, it is exterior.

While solving  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$  can be costly, note that it is not necessary to apply this test to most regions because the type of a region can often be decided by noting the following:

TABLE I  
PERFORMANCE DATA ASSUMING  $s^{\max} = 0.1$

Manipulator	Workspace	$d$	$n_v$	$n_c$	$n_s$	$t_s$
D. Butterfly	reachable	1	37	36	145369	218
Stewart platf.	const.-orient.	2	27	25	1677906	45
3-UPS/S platf.	orientation	2	21	19	156699	30
Agile Eye	orientation	1	21	19	100636	17

- 1) If  $\mathbf{u}$  only contains position coordinates of the end-effector, then the outer region will be exterior to  $\mathcal{A}$ , because the effector can only reach a bounded set of positions in practice.
- 2) A region  $\mathcal{R}_j$  whose boundary contains a traversable singularity can be marked as interior, because  $\mathcal{R}_j$  contains trajectories that enter  $\mathcal{R}_j$  through that singularity.
- 3) A region  $\mathcal{R}_j$  whose boundary contains a barrier point  $\mathbf{u}_i$  with  $\mathbf{n}_i$  pointing outward from  $\mathcal{R}_j$  can be marked as interior as well, because such barrier indicates that there are feasible trajectories in  $\mathcal{C}$  projecting inside  $\mathcal{R}_j$ .

In Fig. 5(d), for example, these observations allow identifying  $\mathcal{R}_1$  as an exterior region if  $\mathbf{u}$  only contains position coordinates, and  $\mathcal{R}_2$  and  $\mathcal{R}_3$  as interior regions. Only the type of  $\mathcal{R}_4$  needs to be disambiguated by checking a point in the region.

## V. PERFORMANCE ON SEVERAL SITUATIONS

We next illustrate the performance of the method on various situations. Several representative workspaces on planar and spatial manipulators are computed to emphasize the generality of the approach, as well as to encounter cases of multicomponent workspaces, hidden regions, and degenerate barriers like those described in Section III, which hinder the application of the continuation method in [31].

All experiments reported have been carried out using a parallelized version of the method implemented in C using the libraries of the CUIK platform [33], and executed on a grid computer with four DELL Poweredge units equipped with two Intel Quadcore Xeon E5310 processors and 4 GB of RAM each. For each experiment, Table I provides the dimension of  $\partial\mathcal{A}$  ( $d$ ), the number of variables ( $n_v$ ) and equations ( $n_c$ ) intervening in (5), the amount of CPU time required to solve it ( $t_s$ , in minutes), and the number of solution boxes returned ( $n_s$ ), assuming  $s^{\max} = 0.1$  in all cases.

### A. Multicomponent Workspaces

To illustrate the performance of our approach on complex multiloop linkages, and to give one example of a multicomponent workspace, we apply our method to compute the reachable workspace of a planar mobility-three double-butterfly linkage (see Fig. 6). This mechanism has been used to compare the performance of general position analysis methods [39]–[41], but no complete method has been given to compute the boundaries and barriers of its reachable workspace yet, as far as we know.

For this example, it will be assumed that the end-effector is the upper-left body in Fig. 6, whose pose is determined by point  $P$  and angle  $\theta_1$ , and that two slider joints are mounted to let

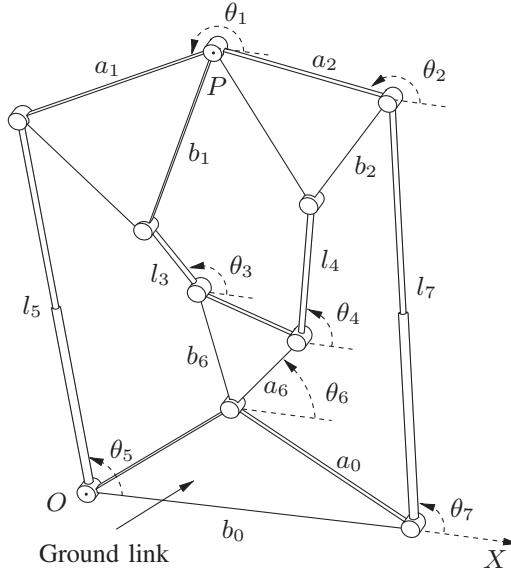


Fig. 6. Planar double-butterfly linkage with variable lengths  $l_5$  and  $l_7$ . The fixed frame is centered at  $O$  with the  $X$ -axis aligned as indicated, and all angles  $\theta_i$  are measured relative to such axis.

lengths  $l_5$  and  $l_7$  vary within the ranges [11, 13] and [10, 12], respectively. As for the parameters in Fig. 6, we adopt the same values considered in [39]–[41]. Namely,  $a_0 = 7$ ,  $a_1 = 7$ ,  $a_2 = 5$ ,  $b_0 = 13$ ,  $b_1 = 6$ ,  $b_2 = 3$ ,  $l_3 = 7$ ,  $l_4 = 9$ ,  $a_6 = 3$ ,  $b_6 = 2$ , and  $\gamma_0 = 36.87^\circ$ ,  $\gamma_1 = 22.62^\circ$ ,  $\gamma_2 = 53.13^\circ$ ,  $\gamma_6 = 36.87^\circ$ , where  $\gamma_i$  is the acute angle between segments  $a_i$  and  $b_i$ . Then, for this manipulator, (1) is formed by the following.

- 1) The loop equations enforcing the closure of the three loops that leave the ground link via  $l_7$ , and return via  $l_4$ ,  $l_3$ , and  $l_5$  [40]

$$\begin{aligned} 0 &= l_7 c_7 + b_2 c_2 c_{\gamma_2} - b_2 s_2 s_{\gamma_2} - l_4 c_4 \\ &\quad - a_6 c_6 + a_0 c_{\gamma_0}, \\ 0 &= l_7 s_7 + b_2 s_2 c_{\gamma_2} + b_2 c_2 s_{\gamma_2} - l_4 s_4 \\ &\quad - a_6 s_6 - a_0 s_{\gamma_0}, \\ 0 &= l_7 c_7 + a_2 c_2 + a_1 c_1 - l_5 c_5 + b_0, \\ 0 &= l_7 s_7 + a_2 s_2 + a_1 s_1 - l_5 s_5, \\ 0 &= l_7 c_7 + a_2 c_2 + b_1 c_1 c_{\gamma_1} - b_1 s_1 s_{\gamma_1} \\ &\quad - l_3 c_3 - b_6 c_6 c_{\gamma_6} + b_6 s_6 s_{\gamma_6} + a_0 c_{\gamma_0}, \\ 0 &= l_7 s_7 + a_2 s_2 + b_1 s_1 c_{\gamma_1} + b_1 c_1 s_{\gamma_1} \\ &\quad - l_3 s_3 - b_6 s_6 c_{\gamma_6} - b_6 c_6 s_{\gamma_6} - a_0 s_{\gamma_0}, \end{aligned}$$

where  $c_{\gamma_i}$  and  $s_{\gamma_i}$  stand for the cosine and sine of  $\gamma_i$ , and  $c_i$  and  $s_i$  for those of  $\theta_i$ .

- 2) The equations providing the  $x$  and  $y$  coordinates of  $P$  relative to the fixed  $OXY$  frame

$$\begin{aligned} x &= b_0 + l_7 c_7 + a_2 c_2, \\ y &= l_7 s_7 + a_2 s_2. \end{aligned}$$



Fig. 7. Box approximation of the set  $\pi_u(\mathcal{S})$  corresponding to the reachable workspace of the linkage in Fig. 6. The boxes can be better appreciated by zooming the electronic version of the paper.

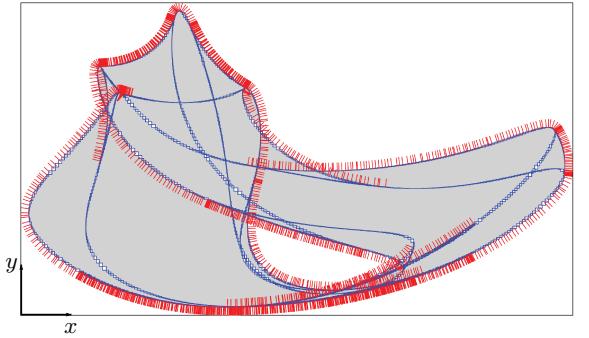


Fig. 8. Results of applying the boundary identification process from Section IV-C to one of the curve components in Fig. 7. The same conventions as in Figs. 2 and 5(d) are used.

- 3) The circle equations constraining  $c_i$  and  $s_i$

$$c_i^2 + s_i^2 = 1.$$

- 4) The joint limit constraints for  $l_5$  and  $l_7$

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

where  $m_i$  and  $h_i$  are the midpoint and half range of the intervals for  $l_5$  and  $l_7$  (see Appendix A).

The reachable workspace is defined as the set of attainable locations for a point on the end-effector, e.g.,  $P$  in our case. Thus,  $\mathbf{u} = [x, y]^T$  for this workspace, and since  $x$  and  $y$  are explicit in the previous equations, we are in the situation

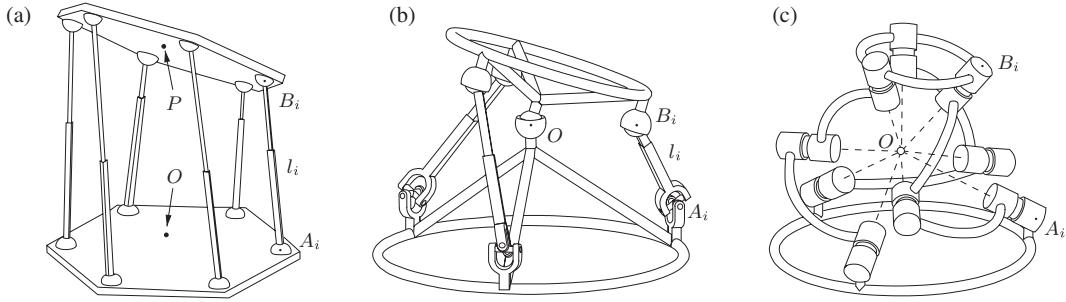


Fig. 9. (a) A 6-6 Stewart platform. (b) and (c) 3-UPS/S and 3-RRR spherical platforms. (b) and (c) are adapted from [23].

of Section IV-A1. Moreover, since  $n_u = 2$ , the boundary of the reachable workspace will be 1-D in general. The proposed method computes the box approximation of  $\pi_u(\mathcal{S})$  shown in Fig. 7 in this case, which delimits three workspace areas corresponding to different assembly modes of the mechanism. The result of the boundary identification process on one of such areas is shown in Fig. 8. Note that, having several connected components, this workspace would be difficult to map out entirely using the method in [31].

### B. Hidden Regions

To show the performance of the method on spatial mechanisms, and to encounter workspaces with hidden regions, we next apply the method to the Stewart platform [see Fig. 9(a)]. This is a challenging test case for any workspace determination method. Its full workspace is 6-D, and its boundary 5-D, which hinders any attempt of computing it exhaustively, due to the curse of dimensionality. For this reason, and because 6-D spaces are impossible to visualize directly in three dimensions, comprehension on this workspace is being gained by obtaining lower dimensional workspaces like 1) the *constant orientation* workspace, or set of attainable locations by a point  $P$  on the platform, for a fixed platform orientation [18], [42], 2) the *constant position* workspace, or set of platform orientations for a fixed position of  $P$  [22], [43], [44], [45], and 3) the *reachable* workspace, or set of locations that  $P$  can attain, with at least one platform orientation [46], [47]. All of these workspaces can be computed by the proposed technique using a proper choice of the  $u$  variables and fixing others to given values. To provide one example where hidden regions arise, we next compute a constant orientation workspace studied in [18].

In order to formulate (1), let  $a_i$  and  $b_i$  denote the position vectors of the base and platform anchor points of the  $i$ th leg, expressed in fixed ( $OXYZ$ ) and moving ( $PX'Y'Z'$ ) frames, respectively. The length of the  $i$ th leg can then be written as

$$l_i^2 = |\mathbf{p} + \mathbf{R}\mathbf{b}_i - \mathbf{a}_i|^2 \quad (16)$$

for  $i = 1, \dots, 6$ , where  $\mathbf{R}$  is a rotation matrix that provides the orientation of  $PX'Y'Z'$  relative to  $OXYZ$ , and  $\mathbf{p} = [x, y, z]^\top$  is the position vector of  $P$  in the  $OXYZ$  frame. Note that if  $\mathbf{r}_j$  refers to the  $j$ th column of  $\mathbf{R}$ , then it must be

$$|\mathbf{r}_1|^2 = 1 \quad (17)$$

$$|\mathbf{r}_2|^2 = 1 \quad (18)$$

$$\mathbf{r}_1 \cdot \mathbf{r}_2 = 0 \quad (19)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (20)$$

for  $\mathbf{R}$  to be a proper rotation matrix. In addition, since the lengths  $l_i$  can only take values within ranges  $[l_i^{\min}, l_i^{\max}]$ , it must be

$$(l_i - m_i)^2 + d_i^2 = h_i^2 \quad (21)$$

for  $i = 1, \dots, 6$ , where  $m_i$  and  $h_i$  are the midpoint and half range of  $[l_i^{\min}, l_i^{\max}]$ .

Since we are computing the constant orientation workspace,  $\mathbf{R}$  will be a known matrix in the previous equations, and (1) will be the system formed by (16) and (21), with  $\tilde{\mathbf{q}} = [l_1, \dots, l_6, d_1, \dots, d_6, x, y, z]^\top$ . Clearly,  $\mathbf{u} = [x, y, z]^\top$  now, and we are in the situation in which  $\mathbf{u}$  intervenes in  $\tilde{\mathbf{q}}$  explicitly, and (5) adopts the form of (6).

Fig. 10 shows 3-D views of the box approximation obtained for  $\pi_u(\mathcal{S})$ , which describes an umbrella-like surface overall. The computation was done assuming the parameters in Table II and with  $\mathbf{R}$  fixed to the identity matrix. In fact, this workspace has an additional connected component symmetric to the one of Fig. 10, which corresponds to the assembly mode of this manipulator where  $P$  sweeps a similar volume for  $z < 0$ . All results obtained are consistent with those in [18].

To better appreciate the shape of the enclosed volume, Fig. 10 plots constant- $z$  slices of  $\pi_u(\mathcal{S})$  indicating the results of the boundary identification process. Note from the plots that it would be difficult to compute such slices by continuation [31], because many slices present multicomponent boundaries and hidden regions that difficult the application of the ray-shooting technique described in Section III. If, for example, the ray is shot from point  $E$  on the  $z = 5.12$  slice, it will not hit the boundary of voids  $V_2$  and  $V_3$  on that slice. While it is true that in [42], the authors were able to compute  $\pi_u(\mathcal{S})$  using the method in [31], they did so by defining particular slices of this set obtained by cutting the umbrella with planes through line  $L$  shown in Fig. 10. This solution avoids the appearance of internal voids within each slice but obviously relies on using *a priori* knowledge of the result.

### C. Degenerate Barriers

Because of the complexity of their defining equations, orientation workspaces are considered among the most difficult ones to compute and represent [7], [22], [44], [45]. Their derivation could be illustrated on the Stewart platform, but we shall do

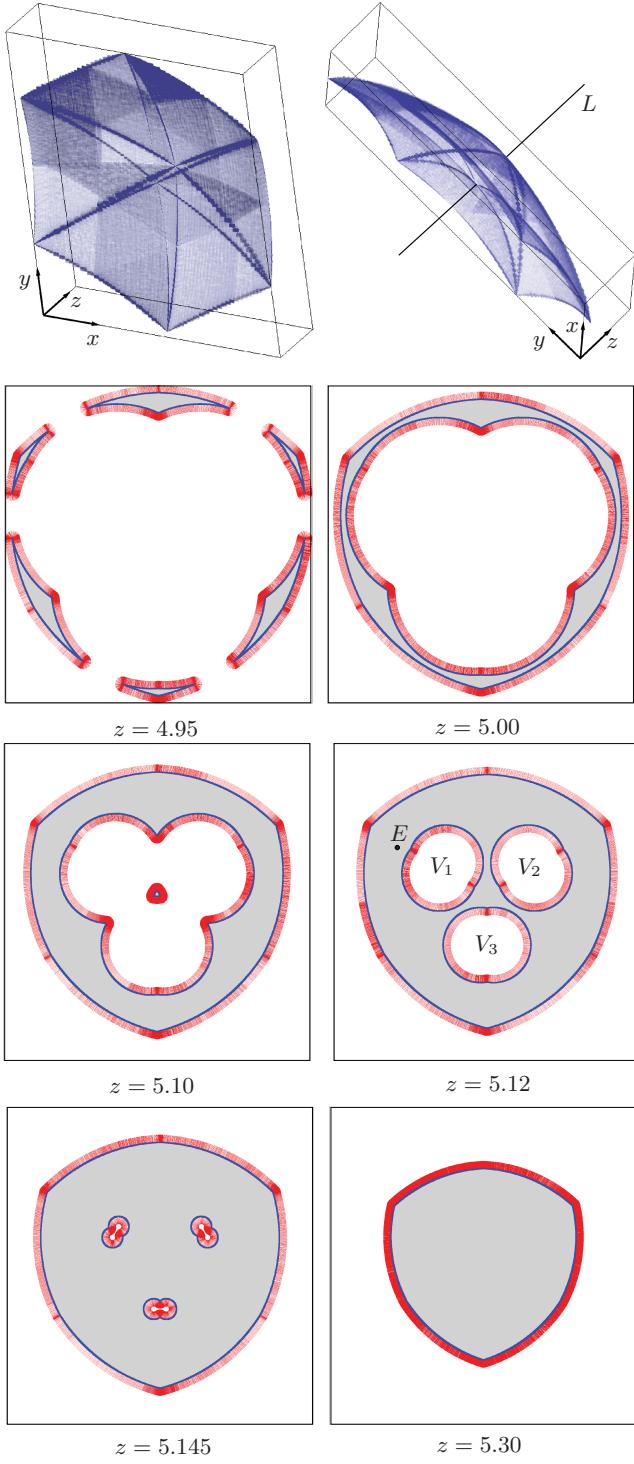


Fig. 10. (Top row) Two views of the boundary of the constant orientation workspace of the Stewart platform. Boxes are semitransparent to better appreciate the shape. (Remaining rows) Slices of the workspace for different values of  $z$ . All points of  $\mathcal{S}$  are classified as barrier singularities in this case.

so on spherical parallel manipulators (SPM) because this will lead to one example of the degenerate barriers mentioned in Section III, which the method in [31] is unable to identify. The examples are taken from [23] and correspond to popular architectures of three degree-of-freedom SPMs: the 3-UPS/S and 3-RRR designs, which are depicted in Fig. 9(b) and 9(c). These

TABLE II  
PARAMETERS OF THE STEWART PLATFORM STUDIED IN [18]

$i$	1	2	3	4	5	6
$a_{i,x}$	92.58	132.58	40.00	-40.00	-132.58	-92.58
$a_{i,y}$	99.64	30.36	-130.00	-130.00	30.36	99.64
$a_{i,z}$	23.10	23.10	23.10	23.10	23.10	23.10
$b_{i,x}$	30.00	78.22	48.22	-48.22	-78.22	-30.00
$b_{i,y}$	73.00	-10.52	-62.48	-62.48	-10.52	73.00
$b_{i,z}$	-37.10	-37.10	-37.10	-37.10	-37.10	-37.10
$l_i^{\min}$	454.5	454.5	454.5	454.5	454.5	454.5
$l_i^{\max}$	504.5	504.5	504.5	504.5	504.5	504.5

are orientational manipulators where the mobile platform can be rotated with respect to the base about a fixed point  $O$  by actuating some of the leg joints. We next compute their orientation workspace and verify the results with those of the analytic method in [23]. Note that whereas the method in [23] is only applicable when certain symmetries hold, the method we propose here remains general.

To derive (1), note that each leg imposes the same constraint on the moving platform, irrespective of the chosen architecture. In a 3-UPS/S platform, for example,  $l_i$  is constrained to take values within some interval  $[l_i^{\min}, l_i^{\max}]$  by design, which limits the angle between  $OA_i$  and  $OB_i$  to some range  $[\alpha_i^{\min}, \alpha_i^{\max}]$ . In a 3-RRA platform, the angle between  $OA_i$  and  $OB_i$  is also limited to some range  $[\alpha_i^{\min}, \alpha_i^{\max}]$  due to mechanical limits on the joints, or to the angles encompassed by the leg links. Both designs are, thus, kinematically equivalent. Moreover, the 3-UPS/S design can be obtained as a special case of the Stewart platform by making three anchor points of such platform coincident, and locking the corresponding legs. Hence, for both the 3-UPS/S and 3-RRA designs, (1) can be formulated as the system formed by (16)–(21) with  $p = [0, 0, 0]^T$ , for  $i = 1, 2, 3$ , assuming that the fixed and moving frames are centered at  $O$ .

In general, the orientation workspace is defined as the set of possible values for three orientation angles of the platform. Although any set of Euler angles could be used for such matter, we shall here adopt the azimuth ( $\phi$ ), tilt ( $\theta$ ), and torsion ( $\sigma$ ) angles assumed in [23] to ease the comparison of results. Using such angles,  $\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\gamma)$ , where  $\gamma = \sigma - \phi$ , and thus, the columns of  $\mathbf{R}$  are

$$\mathbf{r}_1 = \begin{bmatrix} \cos \phi \cos \theta \cos \gamma - \sin \phi \sin \gamma \\ \sin \phi \cos \theta \cos \gamma + \cos \phi \sin \gamma \\ -\sin \theta \cos \gamma \end{bmatrix} \quad (22)$$

$$\mathbf{r}_2 = \begin{bmatrix} -\cos \phi \cos \theta \sin \gamma - \sin \phi \cos \gamma \\ -\sin \phi \cos \theta \sin \gamma + \cos \phi \cos \gamma \\ \sin \theta \sin \gamma \end{bmatrix} \quad (23)$$

$$\mathbf{r}_3 = \begin{bmatrix} \cos \phi \sin \theta \\ \sin \phi \sin \theta \\ \cos \theta \end{bmatrix}. \quad (24)$$

Under the previous convention, Bonev and Gosselin define the orientation workspace as the set of possible values that  $\mathbf{u} = [\phi, \theta, \sigma]^T$  can attain [23], restricting  $\phi \in (-\pi, \pi]$ ,  $\theta \in [0, \pi]$ , and  $\sigma \in (-\pi, \pi]$  to guarantee a one-to-one relationship between the orientations and the corresponding triples  $\{\phi, \theta, \sigma\}$ . We are, thus, in the situation in which  $\mathbf{u}$  does not intervene

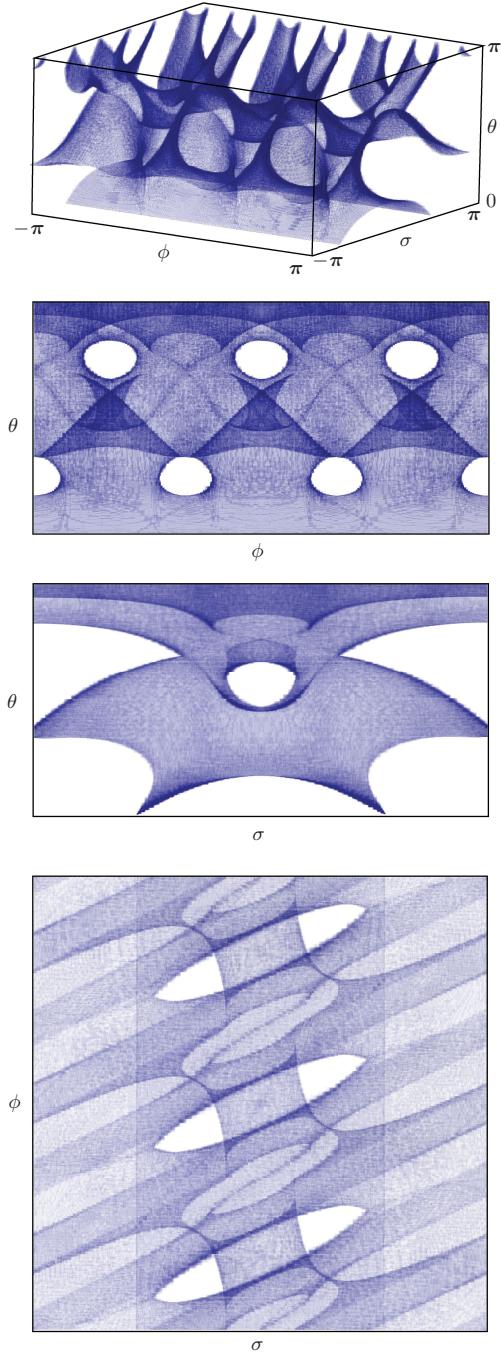


Fig. 11. (Top) 3-D view of the boundary of the orientation workspace of the 3-UPS/S manipulator of Fig. 9(b). (Bottom) Orthogonal projections of that boundary onto the coordinate planes.

explicitly in (1), but it can be related to  $\tilde{\mathbf{u}} = [\mathbf{r}_1^\top, \mathbf{r}_2^\top, \mathbf{r}_3^\top]^\top$  using (22)–(24). Therefore, equation  $\boldsymbol{\eta}(\tilde{\mathbf{u}}) = \mathbf{0}$  is given by (17)–(20) in this case, and  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0}$  consists of (16) and (21) with  $\tilde{\mathbf{z}} = [l_1, l_2, l_3, d_1, d_2, d_3]$ . Overall, (12) contains 19 equations in 21 variables and the boundaries of the orientation workspace are, thus, expected to be 2-D.

For the particular examples shown next, we shall assume the same symmetry conditions as in [23]. Namely,  $A_i$  and  $B_i$  will

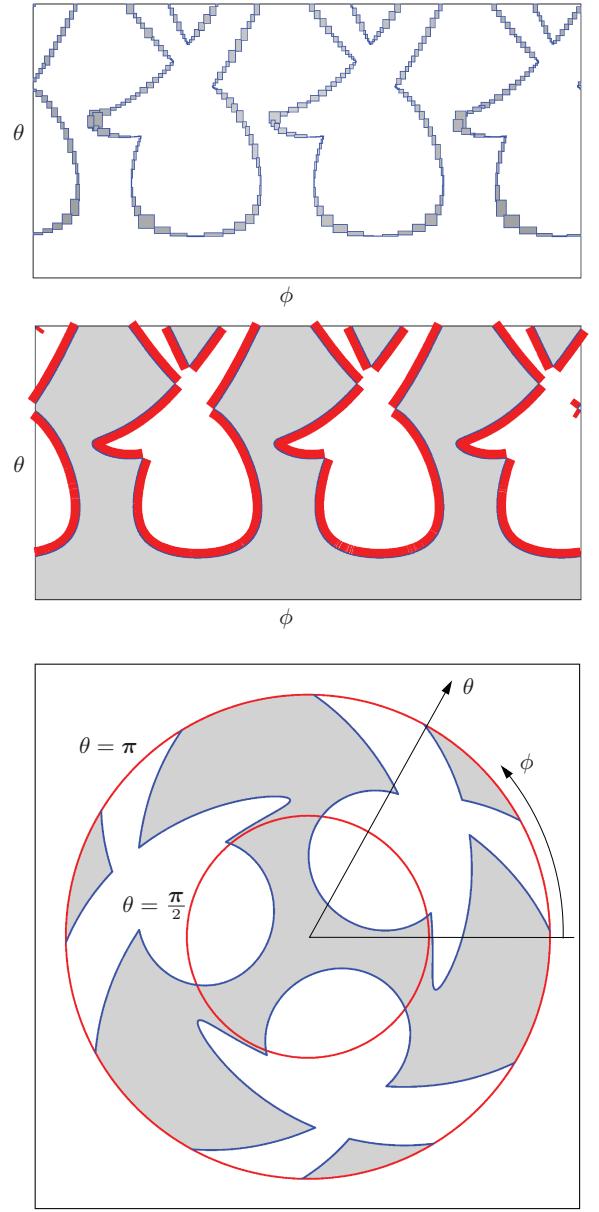


Fig. 12. Constant torsion ( $\sigma$ ) workspace of the spherical parallel manipulator of Fig. 9(b), with parameters  $\beta_1 = 45^\circ$ ,  $\beta_2 = 35^\circ$ ,  $\alpha_{\min} = 20^\circ$ ,  $\alpha_{\max} = 130^\circ$ , and  $\sigma = -30^\circ$ .

lie on a unit sphere centered at  $O$  with position vectors

$$\mathbf{a}_i = \begin{bmatrix} \cos((i-1)\frac{2\pi}{3}) \sin \beta_1 \\ \sin((i-1)\frac{2\pi}{3}) \sin \beta_1 \\ -\cos \beta_1 \end{bmatrix}, \mathbf{b}_i = \begin{bmatrix} \cos((i-1)\frac{2\pi}{3}) \sin \beta_2 \\ \sin((i-1)\frac{2\pi}{3}) \sin \beta_2 \\ -\cos \beta_2 \end{bmatrix}$$

and we will set  $[\alpha_i^{\min}, \alpha_i^{\max}] = [\alpha_{\min}, \alpha_{\max}]$  for all  $i$ .

Fig. 11 shows the resulting box approximation of  $\pi_u(\mathcal{S})$  for  $\beta_1 = 0^\circ$ ,  $\beta_2 = 35^\circ$ ,  $\alpha_{\min} = 20^\circ$ , and  $\alpha_{\max} = 130^\circ$ . These parameters correspond to one of the cases analyzed in [23], where constant-torsion slices of  $\pi_u(\mathcal{S})$  are provided for these manipulators. As expected,  $\pi_u(\mathcal{S})$  is a surface in the  $\{\theta, \phi, \sigma\}$ -space and, by analyzing the neighborhood relationships of the returned boxes, this surface can be shown to contain just one connected component. The top and middle of Fig. 12,

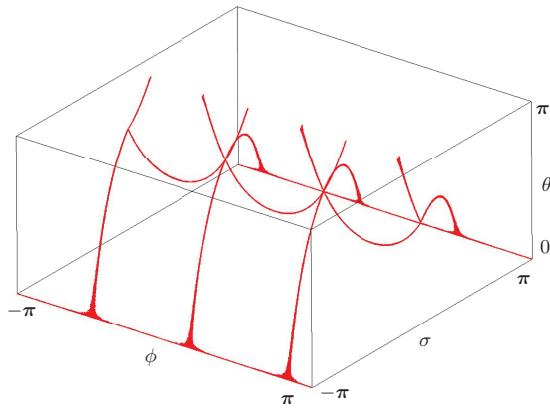


Fig. 13. Workspace of the Agile Eye has degenerate barriers.

respectively, show a  $\sigma = -30^\circ$  slice of the surface shown in Fig. 11 and the barriers identified on such slice. The resulting curve and the interior regions detected match those in [23] when plotted in polar coordinates (see the bottom of Fig. 12).

While the orientation workspace will, generally, have a 2-D boundary, such boundary may degenerate into lower dimensional barriers for particular choices of the geometric parameters, thus posing serious difficulties to the continuation method in [31]. This is what occurs on the *Agile Eye*, for example, a well-known instance of the 3-RRR design in Fig. 9(c), where  $\beta_1 = \beta_2 = \arccos(1/\sqrt{3})$ ,  $\alpha_{\min} = 0$ , and  $\alpha_{\max} = \pi$  [48]. As mentioned in Section III, computing such barriers using [31] is almost impossible because the ray-shooting technique will fail to converge to the barriers almost always. On the contrary, the presented technique is immune to such situations. If the same equations considered to obtain the plot in Fig. 11 are now used for determining the workspace boundaries of the Agile Eye relative to  $\mathbf{u} = [\phi, \theta, \sigma]^T$ , we readily obtain the curves depicted in Fig. 13, which agree with those described in [23]. When analyzed, these curves are seen to be barrier singularities interior to the workspace. In other words, the manipulator will be able to reach any possible orientation, but it will find a motion impediment when trying to traverse across the curves.

## VI. NON-SMOOTH MANIFOLDS

It has been assumed throughout the paper that  $\Phi_q$  is full rank at all points  $\mathbf{q} \in \mathcal{C}$  so that  $\mathcal{C}$  is guaranteed to be a smooth manifold of dimension  $n_q - n_e$  (see Section II). In practice, however, manipulators can be encountered for which  $\Phi_q$  is rank deficient at some points of  $\mathcal{C}$ . Since all maximal minors of  $\Phi_q$  vanish at those points, it is not possible to guarantee that  $\mathcal{C}$  will be locally diffeomorphic to  $\mathbb{R}^{n_q - n_e}$  on them. However, this does not modify the presented algorithm significantly.

Note first that, even if  $\Phi_q$  is not full rank everywhere in  $\mathcal{C}$ ,  $\Phi_z$  must still be rank deficient for a point  $\mathbf{q}$  to belong to  $\partial\mathcal{A}$  (see Appendix A). When computing  $\mathcal{S}$  using the proposed technique, thus, we shall certainly obtain *all* points projecting onto  $\partial\mathcal{A}$ , even those for which  $\Phi_q$  is rank deficient.

Observe also that the points  $\mathbf{q}_0$  of  $\mathcal{S}$  for which  $\Phi_q$  is rank deficient cannot be classified into barrier or traversable singu-

larities, because such classification depends on the existence of a parametrization  $\mathbf{q} = \mathbf{q}(\mathbf{v})$  of  $\mathcal{C}$  in a neighborhood of  $\mathbf{q}_0$  (see Section II-B), which cannot be guaranteed if all maximal minors of  $\Phi_q$  vanish. However, such points correspond to so-called configuration space singularities [49], which can be interpreted as points where the controllability of the manipulator is lost irrespective of the choice of actuated coordinates. Since these configurations are to be avoided, there is no need to classify them, and they can be marked as “uncontrollable” in the returned output.

Finally, we realize that the rank deficiency of  $\Phi_q$  implies the rank deficiency of  $\Phi_z$ . Thus, the set of points for which  $\Phi_q$  is rank deficient will be a subset of dimension lower than that of  $\mathcal{S}$  usually, and the boundary identification process in Section IV-C will not be altered in a significant way.

## VII. CONCLUSION

This paper has presented a complete method for workspace determination on general manipulators. The method delivers a detailed map of the workspace including all motion barriers and output singularities that may be encountered by the manipulator, which is a substantially richer output than that produced by several methods for particular manipulators [20], [22], [43], [44]. In comparison with general methods based on continuation [31], the method is advantageous in that it is *uninformed*, since it does not need to be fed with *a priori* knowledge of the workspace, such as precomputed assembly configurations or suitable slicing directions, and *complete*, as it returns the full workspace map even in the presence of several connected components, hidden regions, or degenerate barriers.

By design, the method is able to compute workspaces of arbitrary dimension but the curse of dimensionality usually restricts its applicability to workspaces of dimension  $n_u \leq 3$  in practice. While workspaces of dimension  $n_u > 3$  arise frequently, e.g., in spatial manipulators, it must be noted that such workspaces are impossible to visualize directly in three dimensions so that in such cases the common practice is to obtain 3-D subsets that are meaningful to the robot designer. In this sense, the paper shows that the reachable, constant-orientation, and constant-position workspaces, which are those typically required in spatial manipulators, can all be computed by the proposed method. Moreover, we emphasize that while, for  $n_u = 3$ , many previous methods are limited to obtain cross-sectional curves of the boundary only [4], [18]–[20], [23], [31], [46], the proposed method will directly isolate the whole surface.

A class of workspaces not treated explicitly in the paper are dexterous workspaces. These are defined as the set of end-effector positions that can be reached with *any* orientation within a given range. While computing such workspaces seems plausible under the proposed approach, several modifications need to be introduced in the method in order to do so. This point, together with the extension of the method to consider further constraints, such as collision constraints, or to compute forward singularities within the workspace, certainly deserve further attention.

## APPENDIX A MODELING JOINT LIMITS

Mechanical limits on the joints can easily be modeled as equality constraints. Two types of limits need to be treated: those referring to distance constraints and those referring to angular constraints. Typically, they arise on slider and revolute joints, respectively, but combinations of both limits may be encountered on other joints. Note on the one hand that, if  $q_i$  is a joint distance that must satisfy

$$q_i^{\min} \leq q_i \leq q_i^{\max} \quad (25)$$

then we can impose this constraint setting

$$(q_i - m_i)^2 + d_i^2 = h_i^2, \quad (26)$$

where  $m_i = \frac{q_i^{\max} + q_i^{\min}}{2}$ ,  $h_i = \frac{q_i^{\max} - q_i^{\min}}{2}$ , and  $d_i$  is a newly defined variable. The values  $m_i$  and  $h_i$  are called the midpoint and half range of the interval  $[q_i^{\min}, q_i^{\max}]$ , and (26) simply constrains the pairs  $(q_i, d_i)$  to take values on a circle of radius  $h_i$  centered at  $(m_i, 0)$  in the  $(q_i, d_i)$  plane. As a consequence,  $q_i$  satisfies (25) if, and only if, it satisfies (26) for some value of  $d_i$ . On the other hand, if  $q_i$  is a joint angle that must satisfy

$$-\alpha_i \leq q_i \leq \alpha_i, \quad (27)$$

then this angle will be represented by its cosine  $c_{q_i}$  and its sine  $s_{q_i}$  under the proposed formulation. The previous constraint is equivalent to  $c_{q_i} \geq \cos \alpha_i$ , which can be written as

$$c_{q_i} = t_i^2 + \cos \alpha_i, \quad (28)$$

where  $t_i$  is a new variable that can take any value. Again  $q_i$  satisfies (27) if, and only if, it satisfies (28) for some  $t_i$ .

## APPENDIX B BOUNDARY CONDITION

We next show that the rank deficiency of  $\Phi_z$  at a point  $q = [z^\top, u^\top]^\top \in \mathcal{C}$  is a necessary condition for  $u$  to belong to  $\partial\mathcal{A}$ , and that the points  $q \in \mathcal{C}$  for which  $\Phi_z$  is rank deficient correspond to critical points of the projection of  $\mathcal{C}$  on  $\mathcal{U}$ .

Observe first that if  $\Phi_z$  is full rank at  $q = [z^\top, u^\top]^\top \in \mathcal{C}$ , then there exists a nonvanishing  $n_e \times n_e$  minor of  $\Phi_z$ , say relative to the variables  $z'$ , and by the implicit function theorem, it is possible to find a function  $z' = F(u')$  relating  $z'$  with the remaining  $u'$  variables, satisfying  $\Phi(F(u'), u') = 0$  [36]. Thus, the  $u'$  variables, which include the  $u$  ones, can be used as a local parameterization of  $\mathcal{C}$  around  $(z', u')$ , implying that arbitrary values in a neighborhood of  $u$  have a corresponding  $z$  satisfying  $\Phi(z, u) = 0$  so that  $u$  must lie in the interior of  $\mathcal{A}$ . Thus,  $\Phi_z$  must be rank deficient for  $u$  to belong to  $\partial\mathcal{A}$ .

Now recall that if  $\Gamma : \mathcal{Q} \rightarrow \mathcal{U}$  is an arbitrary smooth map between two manifolds  $\mathcal{Q}$  and  $\mathcal{U}$  of dimensions  $n_q$  and  $n_u$ , respectively, the differential of  $\Gamma$  at  $q \in \mathcal{Q}$  is the Jacobian matrix  $\Gamma_q = [\partial\Gamma_i / \partial q_j]$ , which relates the vectors in  $T_q \mathcal{Q}$  to the vectors in  $T_u \mathcal{U}$ , the tangent spaces to  $\mathcal{Q}$  at  $q$ , and to  $\mathcal{U}$  at  $u = \Gamma(q)$ , respectively. A point  $q \in \mathcal{Q}$  is said to be a *critical point* of  $\Gamma$  if  $\Gamma_q$  is not surjective at  $q$ , i.e., if  $\Gamma_q(T_q \mathcal{Q})$  is a subspace of  $T_u \mathcal{U}$  of dimension lower than  $n_u$ . One can also speak of the critical

points of  $\Gamma$  with domain restricted to a submanifold  $\mathcal{C} \subseteq \mathcal{Q}$ , to mean the points  $q \in \mathcal{C}$  for which  $\Gamma_q(T_q \mathcal{C})$  does not span  $T_u \mathcal{U}$  entirely.

Suppose now that  $\mathcal{C}$  is defined implicitly by  $\Phi(q) = 0$  as in Section II, where  $\Phi : \mathcal{Q} \rightarrow \mathcal{E}$  is a smooth map whose differential  $\Phi_q$  is full rank at all points of  $\mathcal{C}$ . Then, it can be proved that a point  $q \in \mathcal{C}$  is a critical point of  $\Gamma$  with domain restricted to  $\mathcal{C}$  if, and only if, the matrix  $(\Phi, \Gamma)_q$  has rank less than  $n_e + n_u$  at  $q$  [50, p. 102]. By the previous result, the critical points of the projection map  $\pi_u$  with domain restricted to  $\mathcal{C}$  are the points  $q$  for which

$$(\Phi, \pi_u)_q = \begin{bmatrix} \Phi_z & \dots & \dots & \Phi_u \\ \dots & \dots & \dots & \dots \\ 0 & \vdots & & \mathbf{I}_{n_u} \end{bmatrix}$$

is rank deficient, where  $\mathbf{I}_{n_u}$  denotes the  $n_u \times n_u$  identity matrix. Observe that  $(\Phi, \pi_u)_q$  is rank deficient whenever  $\Phi_z$  is rank deficient, which proves that the mentioned critical points are those for which  $\Phi_z$  is rank deficient.

## ACKNOWLEDGMENT

The authors would like to thank Dr. J. M. Porta for fruitful discussions and for helping to implement the method.

## REFERENCES

- [1] O. Bohigas, L. Ros, and M. Manubens, “A complete method for workspace boundary determination,” in *Proc. 12th Int. Symp. Adv. Robot Kinemat.*, 2010, pp. 329–338.
- [2] D.-Y. Jo, “Numerical analysis of workspaces of multibody mechanical systems,” Ph.D. dissertation, Univ. Iowa, Iowa City, 1988.
- [3] F. A. Adkins, “Numerical continuation and bifurcation methods for mechanism workspace and controllability issues,” Ph.D. dissertation, Univ. Iowa, Iowa City, 1996.
- [4] J. A. Snyman, L. J. du Plessis, and J. Duffy, “An optimization approach to the determination of the boundaries of manipulator workspaces,” *ASME J. Mech. Des.*, vol. 122, no. 4, pp. 447–456, 2000.
- [5] Y. Wang and G. Chirikjian, “Workspace generation of hyper-redundant manipulators as a diffusion process on  $SE(N)$ ,” *IEEE Trans. Robot. Automat.*, vol. 20, no. 3, pp. 399–408, Jun. 2004.
- [6] G. Yang and I. Chen, “Equivolumetric partition of solid spheres with applications to orientation workspace analysis of robot manipulators,” *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 869–879, Oct. 2006.
- [7] J.-P. Merlet and C. Gosselin, “Parallel mechanisms and robots,” in *Handbook of Robotics*. New York: Springer-Verlag, 2008, pp. 269–285.
- [8] Y. Guan, K. Yokoi, and X. Zhang, “Numerical methods for reachable space generation of humanoid robots,” *Int. J. Robot. Res.*, vol. 27, no. 8, pp. 935–950, 2008.
- [9] J. Dai, D. Wang, and L. Cui, “Orientation and workspace analysis of the multifingered metamorphic hand—Metahand,” *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 942–947, Aug. 2009.
- [10] C. Madden, P. Bohnenkamp, K. Kazerounian, and H. Ilies, “Residue level three-dimensional workspace maps for conformational trajectory planning of proteins,” *Int. J. Robot. Res.*, vol. 28, no. 4, pp. 450–463, 2009.
- [11] K. Abdel-Malek, J. Yang, D. Blackmore, and K. Joy, “Swept volumes: fundation, perspectives, and applications,” *Int. J. Shape Model.*, vol. 12, no. 1, pp. 87–127, 2006.
- [12] K. Abdel-Malek and H. Yeh, “Geometric representation of the swept volume using Jacobian rank-deficiency conditions,” *Comput.-Aided Des.*, vol. 29, no. 6, pp. 457–468, 1997.
- [13] B. Roth, “Performance evaluation of manipulators from a kinematic viewpoint,” *Nat. Bureau Stand. Special Pub.*, vol. 459, pp. 39–62, 1976.
- [14] F. Freudenstein and E. Primrose, “On the analysis and synthesis of the workspace of a three-link, turning-pair connected robot arm,” *ASME J. Mech. Trans. Automat. Des.*, vol. 106, no. 3, pp. 365–370, 1984.
- [15] A. Soni and Y. Tsai, “An algorithm for the workspace of a general  $n$ -R robot,” *ASME J. Mech. Trans. Automat. Des.*, vol. 105, pp. 52–57, 1983.

- [16] J. Hansen, K. C. Gupta, and S. M. K. Kazerounian, "Generation and evaluation of the workspace of a manipulator," *Int. J. Robot. Res.*, vol. 2, no. 3, pp. 22–31, 1983.
- [17] J. Rastegar and B. Fardanesh, "Manipulation workspace analysis using the Monte Carlo method," *Mech. Mach. Theory*, vol. 25, no. 2, pp. 233–239, 1990.
- [18] C. Gosselin, "Determination of the workspace of 6-DOF parallel manipulators," *ASME J. Mech. Des.*, vol. 112, pp. 331–336, 1990.
- [19] J. P. Merlet, "Geometrical determination of the workspace of a constrained parallel manipulator," in *Proc. 3rd Int. Symp. Adv. Robot Kinemat.*, 1992, pp. 326–329.
- [20] J. P. Merlet, C. M. Gosselin, and N. Mouly, "Workspaces of planar parallel manipulators," *Mech. Mach. Theory*, vol. 33, nos. 1/2, pp. 7–20, 1998.
- [21] J.-P. Merlet, "Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 902–916, 1999.
- [22] I. Bonev and J. Ryu, "A new approach to orientation workspace analysis of 6-DOF parallel manipulators," *Mech. Mach. Theory*, vol. 36, no. 1, pp. 15–28, 2001.
- [23] I. A. Bonev and C. M. Gosselin, "Analytical determination of the workspace of symmetrical spherical parallel mechanisms," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 1011–1017, Oct. 2006.
- [24] K. Abdel-Malek and H. Yeh, "Analytical boundary of the workspace for general 3-DOF mechanisms," *Int. J. Robot. Res.*, vol. 16, no. 2, 1997.
- [25] M. Ceccarelli, "A formulation for the workspace boundary of general N-revolute manipulators," *Mech. Mach. Theory*, vol. 31, no. 5, pp. 637–646, 1996.
- [26] K. Abdel-Malek, H. Yeh, and N. Khairallah, "Workspace, void, and volume determination of the general 5-DOF manipulator," *J. Struct. Mech.*, vol. 27, no. 1, pp. 89–115, 1999.
- [27] M. Zein, P. Wenger, and D. Chablat, "An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications," *Mech. Mach. Theory*, vol. 41, no. 8, pp. 971–986, 2006.
- [28] E. Ottaviano, M. Husty, and M. Ceccarelli, "Identification of the workspace boundary of a general 3-R manipulator," *ASME J. Mech. Des.*, vol. 128, no. 1, pp. 236–242, 2006.
- [29] Y. Lu, Y. Shi, and B. Hu, "Solving reachable workspace of some parallel manipulators by computer-aided design variation geometry," *Proc. Inst. Mech. Eng. C, Mech. Eng. Sci.*, vol. 222, no. 9, pp. 1773–1781, 2008.
- [30] G. Castelli, E. Ottaviano, and M. Ceccarelli, "A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators," *Mech. Based Des. Struct. Mach.*, vol. 36, no. 1, pp. 14–33, 2008.
- [31] E. J. Haug, C.-M. Luh, F. A. Adkins, and J.-Y. Wang, "Numerical algorithms for mapping boundaries of manipulator workspaces," *ASME J. Mech. Des.*, vol. 118, pp. 228–234, 1996.
- [32] K. Abdel-Malek, F. Adkins, H. J. Yeh, and E. Haug, "On the determination of boundaries to manipulator workspaces," *Robot. Com.-Int. Manuf.*, vol. 13, no. 1, pp. 63–72, 1997.
- [33] J. M. Porta, L. Ros, and F. Thomas, "A linear relaxation technique for the position analysis of multi-loop linkages," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 225–239, Apr. 2009.
- [34] J. G. De Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multi-body Systems*. New York: Springer-Verlag, 1993.
- [35] D. Oblak and D. Kohli, "Boundary surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators," *ASME J. Mech. Trans. Automat. Des.*, vol. 110, no. 4, pp. 389–396, 1988.
- [36] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory and Applications*. Boston, MA: Birkhäuser, 2002.
- [37] E. L. Allgower and K. Georg, *Numerical Continuation Methods*. Berlin, Germany: Springer-Verlag, 1990.
- [38] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, First ed. Philadelphia, PA: SIAM, 2009.
- [39] C. W. Wampler, "Solving the kinematics of planar mechanisms by Dixon's determinant and a complex plane formulation," *ASME J. Mech. Des.*, vol. 123, no. 3, pp. 382–387, Sep. 2001.
- [40] J. M. Porta, L. Ros, T. Creemers, and F. Thomas, "Box approximations of planar linkage configuration spaces," *ASME J. Mech. Des.*, vol. 129, no. 4, pp. 397–405, 2007.
- [41] E. Celaya, T. Creemers, and L. Ros, "Exact interval propagation for the efficient solution of position analysis problems on planar linkages," *Mech. Mach. Theory*, to be published.
- [42] D. Y. Jo and E. J. Haug, "Workspace analysis of closed-loop mechanisms with unilateral constraints," *ASME Adv. Des. Autom.*, vol. 19, no. 3, pp. 53–60, 1989.
- [43] J. P. Merlet, "Determination of the orientation workspace of parallel manipulators," *J. Intell. Rob. Syst.*, vol. 13, no. 2, pp. 143–160, 1995.
- [44] Q. Jiang and C. Gosselin, "Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform," *Mech. Mach. Theory*, vol. 44, no. 6, pp. 1281–1293, 2009.
- [45] Y. Cao, Z. Huang, H. Zhou, and W. Ji, "Orientation workspace analysis of a special class of Stewart-Gough parallel manipulators," *Robotica*, vol. 28, no. 7, pp. 989–1000, 2010.
- [46] C. M. Luh, F. A. Adkins, E. J. Haug, and C. C. Qiu, "Working capability analysis of Stewart platforms," *ASME J. Mech. Des.*, vol. 118, pp. 220–227, 1996.
- [47] F. Pernkopf and M. Husty, "Workspace analysis of Stewart-Gough-type parallel manipulators," *Proc. Inst. Mech. Eng. C, Mech. Eng. Sci.*, vol. 220, no. 7, pp. 1019–1032, 2006.
- [48] I. Bonev, D. Chablat, and P. Wenger, "Working and assembly modes of the Agile Eye," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2317–2322.
- [49] D. Zlatanov, "Generalized singularity analysis of mechanisms," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 1998.
- [50] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1988.



**Oriol Bohigas** received the double M.S. degree in mechanical engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Catalonia, Spain, and in aeronautical engineering from the Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), Toulouse, France, in 2006. After a research internship at Airbus France, he worked for two years as a space engineer with the Centre National d'Etudes Spatiales (CNES). He is currently working toward the Ph.D. degree with the Kinematics and Robot Design Group, Institut de Robòtica i Informàtica Industrial, CSIC-UPC. His current research interests include workspace and singularity analysis of robot mechanisms.



**Montserrat Manubens** received the M.S. degree in mathematics from the Universitat de Barcelona, Barcelona, Spain, in 2001 and the Ph.D. (Hons.) degree in computer algebra from the Universitat Politècnica de Catalunya (UPC), Barcelona, Catalonia, Spain, in 2008. From 2009 to 2010, she worked with the Robotics Group, Institut de Recherche en Communications et Cybrénétique de Nantes, Nantes, France, in the analysis of cuspidal robots. Since 2011, she has been holding a Juan de la Cierva contract with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC. Her current research interests include mathematics and kinematics, with applications to robotics.



**Lluís Ros** received the M.S. degree in mechanical engineering in 1992 and the Ph.D. (Hons.) degree in industrial engineering in 2000, both from the Universitat Politècnica de Catalunya (UPC), Barcelona, Catalonia, Spain. From 1993 to 1996, he worked with the Control of Resources Group of the Institut de Cibernètica, Barcelona. He is currently a Faculty Researcher with the Spanish National Research Council at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC. His current research interests include geometry and kinematics, with applications to robotics.