A Numerical Method for Complete Determination of Workspace Boundaries on General Manipulators*

Oriol Bohigas, Lluís Ros, and Montserrat Manubens
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain

Abstract

This paper introduces a new method for workspace boundary determination on general manipulators. The method uses a branch-and-prune technique to isolate a set of output singularities, and then classifies the points on such set according to whether they correspond to motion impediments in the workspace. A detailed map of the workspace is obtained in the end, where all interior and exterior regions, together with the singularity 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, and a simpler algorithmic structure. Examples are included that show the performance of the method on benchmark problems documented in the Literature, and on new ones unsolved so far.

Keywords: Workspace determination, closed-chain manipulator, multi-body system, branch-and-prune method, linear relaxation, quadratic equation solving.

1 Introduction

A main problem of manipulator 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 (Jo 1988; Adkins 1996; Kyatkin and Chirikjian 1999; Snyman et al. 2000; Badescu and Mavroidis 2004; Merlet and Gosselin 2008; Guan et al. 2008), as well as to related problems in other domains (Madden et al. 2009).

Efficient workspace determination methods exist, but most of them are tailored to a particular robot architecture, or class of architectures. A large group of such methods adopt a constructive geometric approach to the problem. Representative of them is the work by Gosselin (1990), which computes the constant-orientation workspace of a spatial parallel manipulator, the work by Merlet (1995), which extends the approach to deal with other physical constraints, and that by Merlet et al. (1998), which provides methods for various planar parallel platforms. Other significant approaches include discretization (Bonev and Ryu 2001) or interval analysis (Merlet et al. 1999) techniques for Gough-type manipulators, optimization-based algorithms for fully serial/parallel robots (Snyman et al. 2000), analytic methods for symmetrical spherical mechanisms (Bonev and Gosselin 2006), and analytic, topologic, or algebraic-geometric studies for 3R manipulators (Abdel-Malek and Yeh 1997; Zein et al. 2006; Ottaviano et al. 2006). The Literature on the topic is extensive, and we only touch upon it briefly here. Elaborate surveys can be found in (Jo 1988; Adkins 1996; Snyman et al. 2000; Merlet and Gosselin 2008).

While specific solutions are desirable, because they tend to yield faster algorithms, general solutions are required too, to analyze robots for which no specific method exists. In this paper, we ar-

*This work is an extended and updated version of (Bohigas et al. 2010), presented at the 12th International Symposium on Advances in Robot Kinematics, Piran-Portoroz, Slovenia, June 2010.

gue that a solution for lower-pair manipulators of general structure is possible, by extending a recent method for kinematic constraint solving (Porta et al. 2007, 2009) successfully applied to problems of considerable complexity (Rosales et al. 2010).

Up to the authors' knowledge, only one approach of similar generality is available in the Literature, due to Haug and co-workers (Haug et al. 1996, and refs. therein). Similarly to Haug et al., we visualize the workspace by obtaining its boundary, and we extract this boundary from a set of output singularities, but the strategy we adopt to formulate and compute such singularities is substantially different.

In their method, Haug et al. (1996) first slice the singularity set into parallel curves, and then employ a continuation scheme to trace all of such curves in detail. Although elegant and robust to bifurcations, such a procedure requires to be fed with at least one point for each connected component of a traced curve, but no satisfactory method has been given yet to compute such points in general, as far as we know. While Haug et al. (1996) provide a ray-shooting technique to this end, Abdel-Malek et al. (1997) mention several situations that could make the technique miss some boundary segments, specially in the presence of voids internal to the workspace. The performance of ray-shooting strategies can even be worse, however, since degenerate boundaries can be found where *all* points will be missed with probability one, as shown in Secs. 3 and 5.2.2 of this paper.

In contrast, the method we propose in this paper is *complete*, in the sense that it is able to isolate all boundary points of the workspace, even if voids or degenerate boundaries are present. The method is based on formulating the equations of the singularity set in an appropriate form, and then exploiting this form to compute all singular points, using a numerical procedure based on linear relaxations (Porta et al. 2009). Such singularities are then classified according to whether they correspond to actual motion impediments in the workspace, obtaining a detailed map of the workspace in the end, where interior and exterior regions, and the sets that separate them, get clearly identified.

The rest of the paper is structured as follows. Section 2 provides working definitions, assumptions, and necessary background for the paper. Section 3 briefly reviews the continuation method by Haug et al. (1996) and shows several situations in which it fails to entirely determine the boundary of the workspace. The alternative method we propose, which is immune to such situations, is then provided in Section 4, and its performance on various test cases is analyzed in Section 5. Section 6 makes some remarks regarding the application of the method to non-smooth manifolds. Section 7, finally, provides the paper's conclusions and discusses points requiring further attention. To make the presentation as self-contained as possible, proofs of some results used in the paper are included as appendices.

2 Background

2.1 Preliminaries

The allowable positions and orientations of all links in a manipulator are generally encoded in a vector $\tilde{\boldsymbol{q}}$ of $n_{\tilde{q}}$ generalized coordinates, subject to a system of $n_{\tilde{e}} \leq n_{\tilde{q}}$ equations of the form

$$\tilde{\boldsymbol{\Phi}}(\tilde{\boldsymbol{q}}) = 0, \tag{1}$$

which expresses the assembly constraints imposed by the joints (De Jalón and Bayo 1993). Here $\tilde{\boldsymbol{\Phi}}(\tilde{\boldsymbol{q}})$ is a smooth function, and Eq. (1) is meant to include all possible kinematic constraints, including those due to mechanical limits on the active or passive joints, which can also be modelled as equality constraints (Appendix A).

To analyze the functionality of the manipulator, a vector $\boldsymbol{u}$ 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 Eq. (1), such coordinates may or may not be explicit in $\tilde{\boldsymbol{q}}$ but, in the latter case, it is always possible to transform Eq. (1) into an equivalent system of n_e equations

$$\boldsymbol{\Phi}(\boldsymbol{q}) = 0 \tag{2}$$

using appropriate manipulations, where $\boldsymbol{q}$ is an alternative vector of $n_q \geq n_e$ generalized coordinates, now including $\boldsymbol{u}$ explicitly.

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

$$\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0, \tag{3}$$

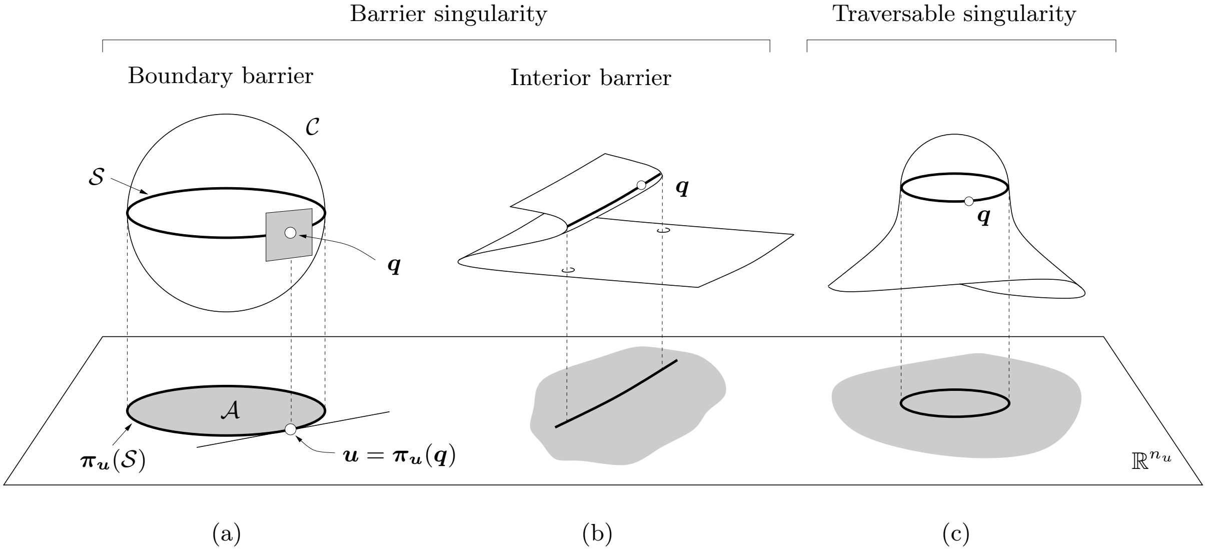


Figure 1: (a) Sets $\mathcal{S}$ and $\boldsymbol{\pi_u}(\mathcal{S})$ when $\mathcal{C}$ is the sphere $x^2 + y^2 + z^2 = 1$ and $\boldsymbol{\pi_u}$ is the projection map $f(x, y, z) = (x, y)$. The workspace relative to the (x, y) coordinates is the projection of the sphere onto the (x, y) plane, and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

and the workspace of the system relative to the $\boldsymbol{u}$ coordinates can be defined as the set $\mathcal{A}$ of points $\boldsymbol{u} \in \mathbb{R}^{n_u}$ that satisfy Eq. (3) for some $\boldsymbol{z}$.

Throughout the paper, we shall assume that $\boldsymbol{\Phi}(\boldsymbol{q})$ is a smooth function from $\mathbb{R}^{n_q}$ to $\mathbb{R}^{n_e}$, whose Jacobian

$$\boldsymbol{\Phi_q} = [\partial\Phi_i/\partial q_j]$$

is full rank at all points $\boldsymbol{q}$ satisfying Eq. (2), so that the set $\mathcal{C}$ of such points will be a smooth manifold of dimension $n_q - n_e$. This assumption is introduced for ease of explanation only, and will be removed in Section 6.

Moreover, we shall concentrate on manipulators for which $\mathcal{A}$ is a subset of $\mathbb{R}^{n_u}$ of dimension n_u, which is the common situation in practice. This implies that $\mathcal{A}$ contains open sets U of $\mathbb{R}^{n_u}$ such that, for any $\boldsymbol{u}_0 \in U$, the system $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}_0) = 0$ has at least one solution. Thus, in particular, the number n_z of coordinates in $\boldsymbol{z}$ must be greater than or equal to the number n_e of equations in Eq. (3).

Although a direct computation of $\mathcal{A}$ could in principle be attempted, it is often more effective to delimit $\mathcal{A}$ by computing its boundary, because such boundary is a set of lower dimension. A point $\boldsymbol{u} \in \mathcal{A}$ lies on the boundary of $\mathcal{A}$, denoted $\partial\mathcal{A}$, if every open set of $\mathbb{R}^{n_u}$ containing $\boldsymbol{u}$ intersects the interior and the exterior of $\mathcal{A}$.

We next provide necessary conditions that the points $\boldsymbol{u} \in \partial\mathcal{A}$ must fulfill. Such conditions, properly transformed into analytic form, will set up the basis of the boundary detection algorithm proposed in Section 4.

2.2 Singularity conditions

Let $\boldsymbol{\pi_u} : \mathbb{R}^{n_q} \longrightarrow \mathbb{R}^{n_u}$ denote the projection map from $\mathbb{R}^{n_q}$ onto the $\boldsymbol{u}$ variables. That is, $\boldsymbol{\pi_u}(\boldsymbol{z}, \boldsymbol{u}) = \boldsymbol{u}$. Observe that $\mathcal{A}$ is exactly the image of $\mathcal{C}$ through $\boldsymbol{\pi_u}$. It is not difficult to see, also, that at the points $\boldsymbol{q} \in \mathcal{C}$ that project onto some $\boldsymbol{u} \in \partial\mathcal{A}$ the matrix $\boldsymbol{\Phi_z} = [\partial\Phi_i/\partial z_j]$ must be rank deficient (see Appendix B). Geometrically, these points correspond to critical points of the projection of $\mathcal{C}$ onto $\mathbb{R}^{n_u}$, i.e., to points $\boldsymbol{q} \in \mathcal{C}$ where the tangent space to $\mathcal{C}$ projects on $\mathbb{R}^{n_u}$ as a linear space of dimension lower than n_u. The set $\mathcal{S}$ of all such critical points will be called the *singularity set* hereafter, and the notation $\boldsymbol{\pi_u}(\mathcal{S})$ will be used to refer to the projection of $\mathcal{S}$ onto $\mathbb{R}^{n_u}$. The situation is illustrated in Fig. 1a with an example.

Kinematically, $\mathcal{S}$ can be interpreted as the set

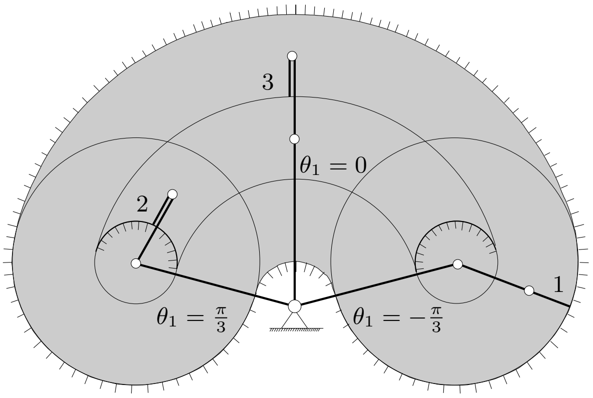


Figure 2: Workspace of a planar 3R manipulator relative to the (x, y) coordinates of the tip point of the last link, assuming that the angle θ_1 of the first revolute joint is restricted to the $[-\pi/3, \pi/3]$ range. Points corresponding to singularities are indicated in solid lines, and those relative to boundary and interior barriers are indicated with normal vectors on the forbidden side. Configurations 1, 2, and 3 correspond to a boundary barrier, an interior barrier, and a traversable singularity, respectively.

of configurations for which the manipulator loses instantaneous mobility relative to the $\boldsymbol{u}$ variables (Appendix C). When $\boldsymbol{u}$ contains all position and orientation coordinates of the end effector, $\mathcal{S}$ is known as the set of type-I singularities (Gosselin and Angeles 1990), end-effector singularities (Park and Kim 1999), or points where the instantaneous inverse kinematic problem is unsolvable (Zlatanov 1998).

A preliminary idea of how the workspace boundary would look like, thus, can be gained by computing all points $\boldsymbol{q} \in \mathcal{C}$ for which $\boldsymbol{\Phi}_{\boldsymbol{z}}$ is rank deficient, and projecting them to the $\boldsymbol{u}$ variables so as to obtain $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$. Note from Figs. 1b and 1c, however, that the rank deficiency of of $\boldsymbol{\Phi}_{\boldsymbol{z}}(\boldsymbol{q})$ is a necessary but not sufficient condition for $\boldsymbol{\pi}_{\boldsymbol{u}}(\boldsymbol{q})$ 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 Fig. 1, points $\boldsymbol{q}$ where $\boldsymbol{\Phi}_{\boldsymbol{z}}$ is rank deficient can be classified into two broad categories. They can be *traversable* or *barrier* singularities depending on whether there exists a trajectory through $\boldsymbol{q}$ whose projection on $\mathbb{R}^{n_u}$ traverses $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ or not, for each neighborhood of $\boldsymbol{q}$ on $\mathcal{C}$ (Oblak and Kohli 1988; Adkins 1996). Points corresponding to barrier singularities can in turn be 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.

We next provide a criterion to determine whether a point $\boldsymbol{q}_0 \in \mathcal{S}$ corresponds to a barrier or a traversable singularity.

2.3 Barrier conditions

Let $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{v})$ be a smooth parameterization of $\mathcal{C}$ in a neighborhood of some $\boldsymbol{q}_0 = [\boldsymbol{z}_0{}^\mathsf{T}, \boldsymbol{u}_0{}^\mathsf{T}]^\mathsf{T} \in \mathcal{S}$, where $\boldsymbol{v}$ is a vector of $n_q - n_e$ paremeters and $\boldsymbol{q}_0 = \boldsymbol{q}(\boldsymbol{v}_0)$ for some $\boldsymbol{v}_0 \in \mathbb{R}^{n_q - n_e}$. Since $\boldsymbol{\Phi}_{\boldsymbol{q}}$ is full rank for all $\boldsymbol{q} \in \mathcal{C}$, the Implicit Function Theorem guarantees that such a parameterization exists (Krantz and Parks 2002). Let $\boldsymbol{n}_0$ be the normal to $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ at $\boldsymbol{u}_0$, which can be computed as indicated in Appendix C. We can determine whether $\boldsymbol{q}_0$ corresponds to a barrier singularity by examining the sign of

$$\psi(\boldsymbol{v}) = \boldsymbol{n}_0{}^\mathsf{T}(\boldsymbol{u}(\boldsymbol{v}) - \boldsymbol{u}_0) \tag{4}$$

for all local trajectories parameterized in time $\boldsymbol{v} = \boldsymbol{v}(t)$ crossing $\boldsymbol{v}_0$ for some $t = t_0$, whose corresponding path $\boldsymbol{u} = \boldsymbol{u}(t)$ is orthogonal to $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ at $\boldsymbol{u}_0$. This can be done by resorting to the second-order Taylor expansion of $\psi(\boldsymbol{v})$ around $\boldsymbol{v}_0$

$$\psi(\boldsymbol{v}) \simeq \psi(\boldsymbol{v}_0) + \delta\boldsymbol{v}^\mathsf{T}\psi_{\boldsymbol{v}}(\boldsymbol{v}_0) + \frac{1}{2}\delta\boldsymbol{v}^\mathsf{T}\psi_{\boldsymbol{v}\boldsymbol{v}}(\boldsymbol{v}_0)\delta\boldsymbol{v},$$

where $\psi_{\boldsymbol{v}}$ and $\psi_{\boldsymbol{v}\boldsymbol{v}}$ are the gradient and Hessian of $\psi(\boldsymbol{v})$, and $\delta\boldsymbol{v} = (\boldsymbol{v} - \boldsymbol{v}_0)$ is a small displacement whose corresponding $\delta\boldsymbol{u} = (\boldsymbol{u} - \boldsymbol{u}_0)$ is orthogonal to $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$.

Note that the first term of the previous expansion vanishes because $\psi(\boldsymbol{v}_0) = \boldsymbol{n}_0{}^\mathsf{T}(\boldsymbol{u}_0 - \boldsymbol{u}_0) = 0$. Moreover, the time derivative of Eq. (4) for $\boldsymbol{v} = \boldsymbol{v}(t)$ is

$$\dot{\psi}(t) = \boldsymbol{n}_0{}^\mathsf{T}\dot{\boldsymbol{u}}(t),$$

which is the component of $\dot{\boldsymbol{u}}(t)$ along $\boldsymbol{n}_0$. As shown in Appendix C, $\dot{\psi}(t_0)$ must vanish irrespectively of the chosen trajectory $\boldsymbol{v} = \boldsymbol{v}(t)$. Thus, since for $t = t_0$ it is $\dot{\psi} = \psi_{\boldsymbol{v}}\dot{\boldsymbol{v}} = 0$ for all $\dot{\boldsymbol{v}}$, we conclude that $\psi_{\boldsymbol{v}}(\boldsymbol{v}_0) = 0$ as well, meaning that the second term of the Taylor expansion also vanishes.

In conclusion, the sign of $\psi(\boldsymbol{v})$ is mostly determined by the definiteness properties of the quadratic form $\delta\boldsymbol{v}^{\mathsf{T}}\psi_{\boldsymbol{vv}}(\boldsymbol{v}_0)\delta\boldsymbol{v}$. If this form is positive- or negative-definite, then all trajectories orthogonal to $\boldsymbol{\pi_u}(\mathcal{S})$ lie on one side of $\boldsymbol{\pi_u}(\mathcal{S})$ and $\boldsymbol{q}_0$ is a barrier singularity. If this form is indefinite, there are trajectories in $\mathcal{A}$ that cross $\boldsymbol{\pi_u}(\mathcal{S})$ and $\boldsymbol{q}_0$ is a traversable singularity. Lastly, if this form is semi-definite, 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.

The definiteness test just outlined can be implemented by checking the eigenvalues of the matrix form of $\delta\boldsymbol{v}^{\mathsf{T}}\psi_{\boldsymbol{vv}}(\boldsymbol{v}_0)\delta\boldsymbol{v}$ (Haug et al. 1996).

3 Related work

As mentioned in the introduction, the workspace determination method we propose can be compared, in terms of generality, to the continuation method by Haug et al. (1996). To properly see the advantages of our method, we next point out the main drawbacks of the continuation method.

We note first that, since the continuation method relies on one-dimensional path tracking procedures, it can only trace $\partial\mathcal{A}$ explicitly on one-dimensional boundaries, i.e., when $n_u = 2$. Fig. 3a explains the method on a simple case in which $\mathcal{C}$ contains two connected components, $\mathcal{C}_1$ and $\mathcal{C}_2$. The method begins with an assembled configuration of the manipulator, $\boldsymbol{q}_i = [\boldsymbol{z}_i{}^{\mathsf{T}}, \boldsymbol{u}_i{}^{\mathsf{T}}]^{\mathsf{T}}$, with $\boldsymbol{u}_i \in \mathcal{A}$, shoots a ray through $\boldsymbol{u}_i$ in $\mathbb{R}^{n_u}$ on an arbitrary direction, and traces this ray until a point $\boldsymbol{u}_b \in \partial\mathcal{A}$ is found. The actual movement along the ray is achieved by continuation of the corresponding trajectory on $\mathcal{C}$, i.e., by iteratively solving $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0$ using a Newton method, for $\boldsymbol{u}$ fixed to discrete values along the ray. This process is repeated until a point outside $\mathcal{A}$ is found, which is detected because the Newton method fails to converge, and a dichotomy-search process is then performed locally, in order to find a point $\boldsymbol{q}_b \in \mathcal{C}$ lying on $\mathcal{S}$. A second continuation process is then launched from $\boldsymbol{q}_b$ to find the connected component of $\mathcal{S}$ that is reachable from $\boldsymbol{q}_b$, by solving a system of equations that expresses the rank deficiency condition of $\boldsymbol{\Phi_z}$. Once $\mathcal{S}$ has been found, the points of $\boldsymbol{\pi_u}(\mathcal{S})$ are computed by projection, and those corresponding to barrier or traversable singularities are finally detected.

Because path tracking methods based on continuation are fast, this approach will rapidly determine $\mathcal{S}$ in favorable situations: the ray shooting technique will succeed in computing $\boldsymbol{u}_b$ in a few iterations, and the tracking of $\mathcal{S}$ can be made robust to path bifurcations, if mature techniques of one-dimensional continuation are employed (Rheinboldt 1986; Allgower and Georg 1990). Despite the elegance and efficiency of the strategy, however, it is not difficult to find situations in which this approach will fail to identify $\partial\mathcal{A}$ completely, even if a configuration $\boldsymbol{q}_i$ is known beforehand.

Difficulties arise, for example, when computing $\partial\mathcal{A}$ on workspaces with several connected components, like the one in Fig. 3a. 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, thus, 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.

The continuation method may seem to be able to compute, at least, the boundary of the workspace component to which $\boldsymbol{q}_i$ belongs, but this is not even the case in general. Note that $\partial\mathcal{A}$ may itself have several connected components, and some of such components could be missed depending on the position of $\boldsymbol{u}_i$, even if rays on all possible directions are shot. In Fig. 3b, for example, the continuation method may be able to find $\partial\mathcal{A}_1$ and $\partial\mathcal{A}_3$ from $\boldsymbol{u}_i$, but not $\partial\mathcal{A}_2$, because $\partial\mathcal{A}_2$ is hidden behind $\partial\mathcal{A}_1$. The problem also arises on workspaces without interior voids, as seen in Fig. 3c. Starting the continuation from $\boldsymbol{q}_i$ allows hitting the interior barrier corresponding to $\mathcal{S}_1$, but not the boundary barrier corresponding to $\mathcal{S}_3$, thus ignoring a full circular area that actually lies inside the workspace.

Problems arise on degenerate boundaries as well, i.e., on those whose dimension is lower than expected. When $\partial\mathcal{A}$ is of dimension larger than one, Haug et al. (1996) slice $\partial\mathcal{A}$ through hyperplanes $\mathcal{H}_i$ in order to obtain one-dimensional curves $\partial\mathcal{A}_i$ trackable by their method (Fig. 3d). On manipulators of a special geometry, however, these higher-dimensional boundaries can degenerate into lower-dimensional sets, thus making the slices $\mathcal{H}_i$ contain isolated points of $\partial\mathcal{A}$ only (Fig. 3e). Note that the method will miss the whole boundary in the latter

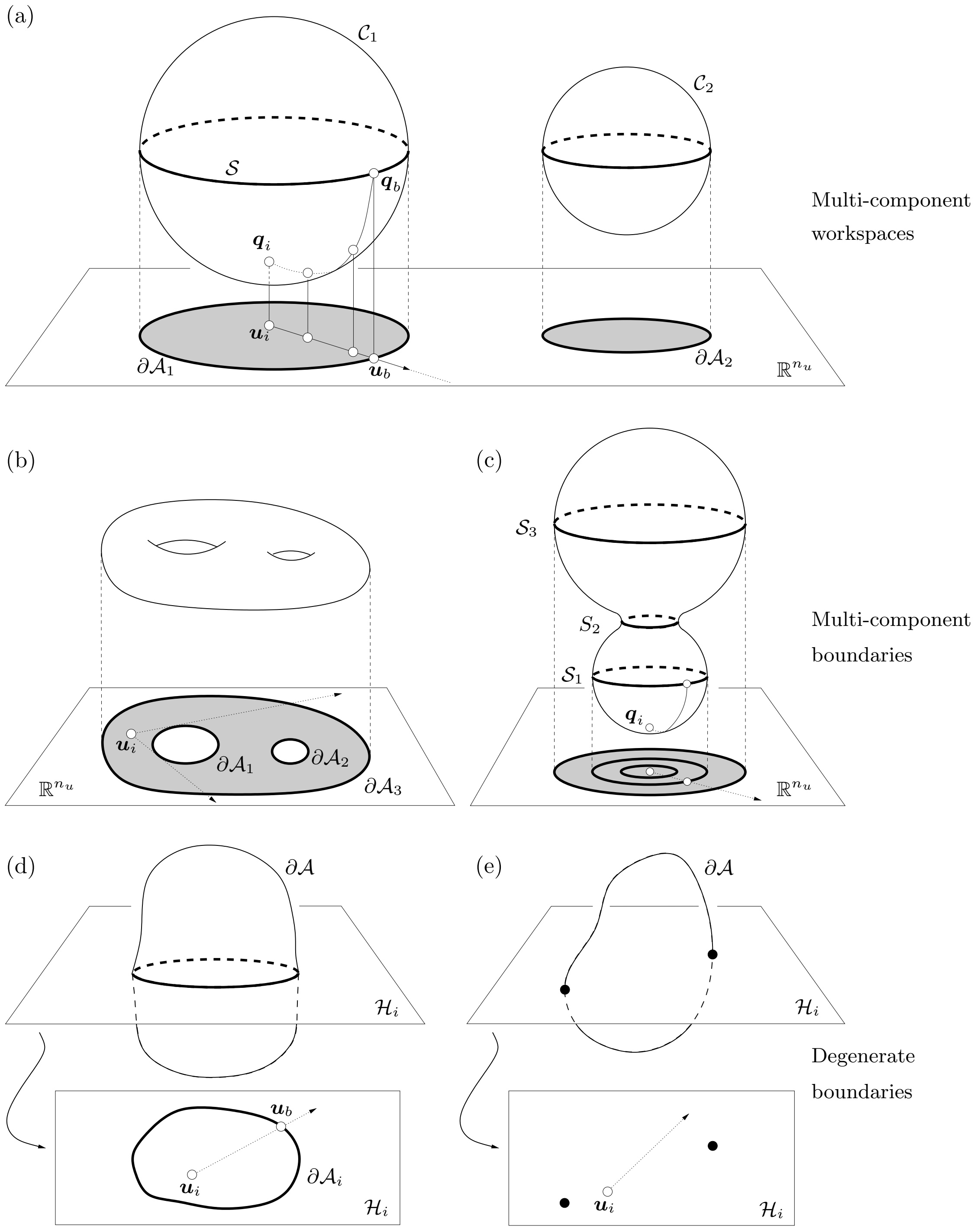


Figure 3: Performance of the continuation method by Haug et al. (1996) in several situations.

situation, because the ray shooting technique will fail to meet $\partial\mathcal{A}$ with probability one on each slice, independently of the location of $\boldsymbol{u}_i$.

As it will be shown in Section 5, examples of multi-component workspaces, multi-component boundaries, and degenerate boundaries occur easily on real manipulators, and, thus, there is a clear need to develop alternative methods which are robust to such situations. Next section presents one, using the results of Section 2.

4 Computational method

The proposed method consists in first formulating a system of equations that characterize $\mathcal{S}$ (Section 4.1), then applying a numerical technique that exploits the structure of this system to isolate $\mathcal{S}$ (Section 4.2), and finally using a local method to classify the points of $\mathcal{S}$ as boundary barriers, interior barriers, or traversable singularities (Section 4.3).

4.1 Equation formulation

For a manipulator involving lower pairs of any kind, except the helical pair, the formulation proposed by Porta et al. (2009) makes Eq. (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 variables). We next show that, by adopting this formulation, it is possible to extend Eq. (1) with additional variables and equations, so as to obtain a new system

$$\boldsymbol{\chi}(\boldsymbol{y}) = 0 \tag{5}$$

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

To explain how Eq. (5) can be derived, we shall distinguish two situations, depending on whether the $\boldsymbol{u}$ variables appear explicitly in $\tilde{\boldsymbol{q}}$, or are only determined implicitly by some of the variables in $\tilde{\boldsymbol{q}}$ (examples of both situations will be worked out in detail in Section 5). In any case, it will be assumed that Eq. (1) has been formulated following Porta et al. (2009), with $\tilde{\boldsymbol{q}}$ containing the pose of the end-effector encoded as a vector $[\boldsymbol{p}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}]^\mathsf{T}$, where $\boldsymbol{p}$ and $\boldsymbol{r}$ respectively encompass the Cartesian coordinates of a point on the effector, and the components of a rotation matrix giving the orientation of the effector.

4.1.1 Dealing with explicit output

Assume initially that $\tilde{\boldsymbol{q}}$ explicitly contains $\boldsymbol{u}$. This occurs in positional workspaces, i.e., when $\boldsymbol{u}$ includes part or all of the variables in $\boldsymbol{p}$. Then, by adopting the partition $\tilde{\boldsymbol{q}} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}]^\mathsf{T}$, Eq. (1) can be directly cast into the form of Eq. (3) so that $\mathcal{S}$ will be, according to Section 2.2, the set of points $\boldsymbol{q} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}]^\mathsf{T}$ satisfying $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0$ for which $\boldsymbol{\Phi}_{\boldsymbol{z}}$ is rank deficient.

Note then that, algebraically, $\mathcal{S}$ can be seen as the set of points $\boldsymbol{q}$ that satisfy

$$\left.\begin{array}{r} \boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0 \\ {\boldsymbol{\Phi}_{\boldsymbol{z}}}^\mathsf{T}\boldsymbol{\xi} = 0 \\ \boldsymbol{\xi}^\mathsf{T}\boldsymbol{\xi} = 1 \end{array}\right\}, \tag{6}$$

for some $\boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is an n_e-dimensional vector of unknowns. Clearly, the first equation in (6) constrains $\boldsymbol{q}$ to be a valid configuration, and the second and third equations impose the rank deficiency of $\boldsymbol{\Phi}_{\boldsymbol{z}}$. 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 Eq. (1) is quadratic, Eq. (6) will be quadratic too, because all entries in ${\boldsymbol{\Phi}_{\boldsymbol{z}}}^\mathsf{T}$ will be linear terms, and $\boldsymbol{\xi}^\mathsf{T}\boldsymbol{\xi}$ is directly a quadratic expression. Thus, $\boldsymbol{\chi}(\boldsymbol{y}) = 0$ adopts the form of Eq. (6) in this case, with $\boldsymbol{y} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}, \boldsymbol{\xi}^\mathsf{T}]^\mathsf{T}$.

4.1.2 Dealing with implicit output

There are situations in which the variables in $\boldsymbol{u}$ do not all intervene in $\tilde{\boldsymbol{q}}$ but, instead, they can be related to a subset $\tilde{\boldsymbol{u}}$ of $n_{\tilde{u}}$ variables in $\tilde{\boldsymbol{q}}$, through a smooth function of the form

$$\tilde{\boldsymbol{u}} = \boldsymbol{\mu}(\boldsymbol{u}). \tag{7}$$

This occurs whenever $\boldsymbol{u}$ contains orientation angles of the end-effector. Since Porta et al. (2009) represent end-effector orientations by rotation matrices, orientation angles are only related implicitly to the components of $\boldsymbol{r}$ through a parameterization 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 Eq. (1) into a system of the form of Eq. (3) it will be possible, in such situations, to consider the partition

$$\tilde{\boldsymbol{q}} = [\tilde{\boldsymbol{z}}^{\mathsf{T}}, \tilde{\boldsymbol{u}}^{\mathsf{T}}]^{\mathsf{T}},$$

and subdivide Eq. (1) into two subsystems, as

$$\left.\begin{array}{c} \boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0 \\ \boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0 \end{array}\right\}, \tag{8}$$

where $\boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0$ is a subsystem of equations whose solution set can be globally parametrized by Eq. (7), and $\boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0$ collects the rest of equations. Since $\tilde{\boldsymbol{u}} = \boldsymbol{\mu}(\boldsymbol{u})$ parametrizes the solution set of $\boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0$, we can substitute $\boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0$ for $\tilde{\boldsymbol{u}} = \boldsymbol{\mu}(\boldsymbol{u})$ in Eq. (8), obtaining the equivalent system of equations

$$\left.\begin{array}{c} \boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0 \\ \tilde{\boldsymbol{u}} = \boldsymbol{\mu}(\boldsymbol{u}) \end{array}\right\}, \tag{9}$$

which now contains $\boldsymbol{u}$ explicitly.

Therefore, Eq. (3) adopts the particular form of Eq. (9) in this case, with $\boldsymbol{z} = [\tilde{\boldsymbol{z}}^{\mathsf{T}}, \tilde{\boldsymbol{u}}^{\mathsf{T}}]^{\mathsf{T}}$, and

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

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

$$\boldsymbol{\Phi}_{\boldsymbol{z}} = \left[\begin{array}{c:c} \boldsymbol{\Psi}_{\tilde{\boldsymbol{z}}} & \boldsymbol{\Psi}_{\tilde{\boldsymbol{u}}} \\ \hdashline \boldsymbol{0} & \boldsymbol{I}_{n_{\tilde{u}}} \end{array}\right]$$

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

$$\left.\begin{array}{c} \boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0 \\ \tilde{\boldsymbol{u}} = \boldsymbol{\mu}(\boldsymbol{u}) \\ \boldsymbol{\Psi}_{\tilde{\boldsymbol{z}}}{}^{\mathsf{T}}\tilde{\boldsymbol{\xi}} = 0 \\ \tilde{\boldsymbol{\xi}}^{\mathsf{T}}\tilde{\boldsymbol{\xi}} = 1 \end{array}\right\} \tag{11}$$

for some $\tilde{\boldsymbol{\xi}}$, where $\tilde{\boldsymbol{\xi}}$ is a vector with as many components as the number of rows in $\boldsymbol{\Psi}_{\tilde{\boldsymbol{z}}}$.

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

$$\left.\begin{array}{c} \boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0 \\ \boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0 \\ \boldsymbol{\Psi}_{\tilde{\boldsymbol{z}}}{}^{\mathsf{T}}\tilde{\boldsymbol{\xi}} = 0 \\ \tilde{\boldsymbol{\xi}}^{\mathsf{T}}\tilde{\boldsymbol{\xi}} = 1 \end{array}\right\}, \tag{12}$$

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

4.2 Equation Solution

A numerical method able to solve $\boldsymbol{\chi}(\boldsymbol{y}) = 0$ is next described, based on the approach proposed by Porta et al. (2009). The approach entails expanding the equations to a canonical form, and then using a branch-and-prune method exploiting this form to isolate the solutions.

Let y_i and y_j refer to any two variables appearing in $\boldsymbol{\chi}(\boldsymbol{y}) = 0$. The method starts by introducing the changes of variables

$$p_i = y_i^2 \tag{13}$$

$$b_k = y_i y_j \tag{14}$$

for all $y_i y_j$ and y_i^2 monomials intervening in $\boldsymbol{\chi}(\boldsymbol{y}) = 0$. This allows transforming this system into the expanded form

$$\left.\begin{array}{c} \boldsymbol{\Lambda}(\boldsymbol{x}) = 0 \\ \boldsymbol{\Omega}(\boldsymbol{x}) = 0 \end{array}\right\}, \tag{15}$$

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

$$x_k = x_i^2,$$

$$x_k = x_i x_j,$$

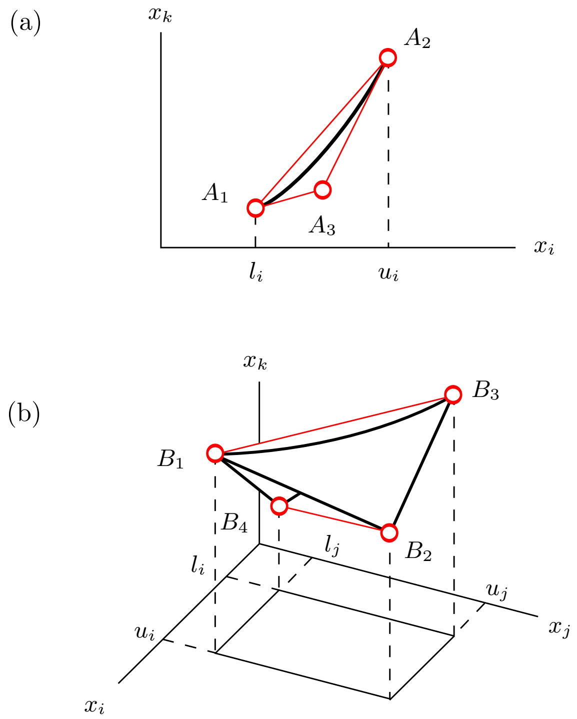


Figure 4: Polytope bounds within box $\mathcal{B}_c$. (a) The points on the parabola $x_k = x_i^2$ are bound by the triangle $A_1A_2A_3$. (b) The points on the hyperbolic paraboloid $x_k = x_i x_j$ are bound by the tetrahedron $B_1B_2B_3B_4$.

corresponding to the changes of variables in Eqs. (13) and (14), respectively.

It can be seen that each variable x_i of $\boldsymbol{x}$ can only take values within a prescribed interval (Porta et al. 2009), so that from the Cartesian product of all such intervals one can define an initial n_x-dimensional box $\mathcal{B}$ which bounds all solutions of Eq. (15). The algorithm to isolate such solutions recursively applies two operations on $\mathcal{B}$: box *shrinking* and box *splitting*.

Using box shrinking, portions of $\mathcal{B}$ containing 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 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 $s^{\max}$.

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

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

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 $\boldsymbol{\Omega}(\boldsymbol{x}) = 0$. These equations can be taken into account by noting that, if $[l_i, u_i]$ denotes the interval of $\mathcal{B}_c$ along dimension x_i, then:

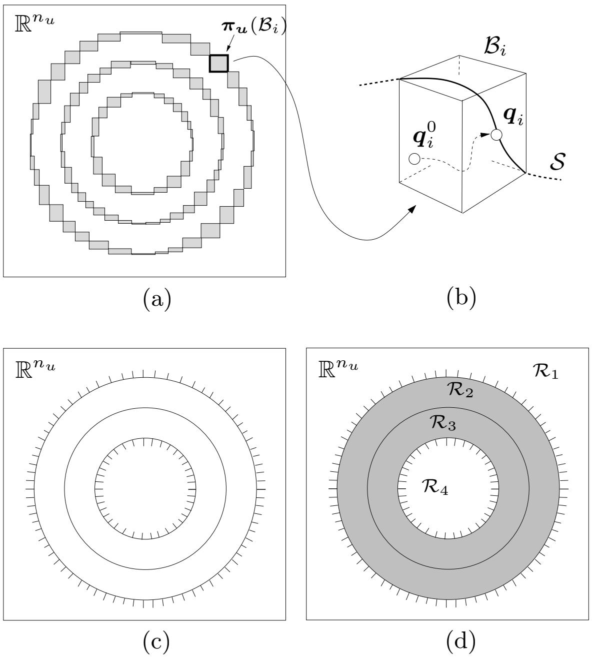
1. The portion of the parabola $x_k = x_i^2$ lying inside $\mathcal{B}_c$ is bound by the triangle $A_1A_2A_3$, where A_1 and A_2 are the points where the parabola intercepts the lines $x_i = l_i$ and $x_i = u_i$, and A_3 is the point where the tangent lines at A_1 and A_2 meet (Fig. 4a).

2. The portion of the hyperbolic paraboloid $x_k = x_i x_j$ lying inside $\mathcal{B}_c$ is bound by the tetrahedron $B_1B_2B_3B_4$, where the points $B_1, \ldots, B_4$ are obtained by lifting the corners of the rectangle $[l_i, u_i] \times [l_j, u_j]$ vertically to the paraboloid (Fig. 4b).

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, which is returned as output upon termination, is said to form a *box approximation* of the solution set of Eq. (15), because the boxes in B form a discrete envelope of such set, whose accuracy can be adjusted through the $s^{\max}$ parameter. Notice that the algorithm is complete, in the sense that the boxes in B include all solution points of Eq. (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 (Porta et al. 2009).

Having obtained B, it is finally straightforward to obtain a box approximation $B^{\mathcal{S}}$ of the singular set $\mathcal{S}$. If Eq. (5) adopts the form of Eq. (6), then $\boldsymbol{z}$ and $\boldsymbol{u}$ explicitly intervene in Eq. (15), as part of $\boldsymbol{x}$, and each box in B already has ranges along the $\boldsymbol{q} = [\boldsymbol{z}^{\mathsf{T}}, \boldsymbol{u}^{\mathsf{T}}]^{\mathsf{T}}$ dimensions. Such ranges define a box in $\boldsymbol{q}$-space enclosing points of $\mathcal{S}$, and the collection of all such boxes provides $B^{\mathcal{S}}$. If Eq. (5) adopts the form of Eq. (12), $B^{\mathcal{S}}$ can be obtained in a similar way. The only difference is that, because the $\boldsymbol{u}$ variables do not intervene in Eq. (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 $\tilde{\boldsymbol{u}}$ variables, we can derive corresponding ranges for the $\boldsymbol{u}$ variables by evaluating $\boldsymbol{u} = \boldsymbol{\mu}^{-1}(\tilde{\boldsymbol{u}})$ using interval arithmetic (Moore et al. 2009).



(a) (b) (c) (d)

Figure 5: Boundary identification process. (a) Box approximation of $\mathcal{S}$ projected onto the output space $\mathbb{R}^{n_u}$. (b) Computation of $\boldsymbol{q}_i \in \mathcal{S}$ for each box $\mathcal{B}_i$. (c) Classification of the points of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ into barrier or traversable singularities. (d) Regions into which $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ subdivides $\mathbb{R}^{n_u}$ and their classification into interior (grey) or exterior (white) regions, using barrier information only.

4.3 Boundary identification

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

In a first stage, we classify the boxes of $B^{\mathcal{S}}$ according to whether they enclose barrier or traversable singularities. For each box $\mathcal{B}_i \in B^{\mathcal{S}}$ (Fig. 5a), a point $\boldsymbol{q}_i \in \mathcal{S}$ is computed (Fig. 5b), and the classification method explained in Section 2.3 is applied to this point. The computation of $\boldsymbol{q}_i$ is done by solving Eq. (6) or Eq. (11), depending on the situation, using a Newton-Raphson method starting from an arbitrary point $\boldsymbol{q}_i^0$ inside $\mathcal{B}_i$. This procedure will quadratically converge to some point $\boldsymbol{q}_i \in \mathcal{S}$ provided that the points within $\mathcal{B}_i$ are close enough to $\mathcal{S}$, which can be guaranteed by computing $B^{\mathcal{S}}$ using a small $s^{\max}$ threshold. The singularity type obtained for $\boldsymbol{q}_i$ (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 $B^{\mathcal{S}}$, it is possible to subdivide $\mathcal{S}$ into subsets of constant singularity type. If $\boldsymbol{q}_i$ is a barrier singularity, a normal vector $\boldsymbol{n}_i$ pointing towards the forbidden side of the barrier is drawn at $\boldsymbol{u}_i = \boldsymbol{\pi}_{\boldsymbol{u}}(\boldsymbol{q}_i)$ (Fig. 5c).

In a second stage, we determine which of the barrier points $\boldsymbol{q}_i$ computed in the previous stage correspond to boundary or interior barriers. To this end, notice that $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ subdivides $\mathbb{R}^{n_u}$ into several regions $\mathcal{R}_1, \ldots, \mathcal{R}_{n_r}$, where each region $\mathcal{R}_j$ fully lies either in the interior or in the exterior of $\mathcal{A}$. Also notice that a barrier point $\boldsymbol{u}_i$ 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 $\boldsymbol{u}_i$ correspond to boundary barriers boils down to checking whether the regions $\mathcal{R}_1, \ldots, \mathcal{R}_{n_r}$ are interior or exterior to $\mathcal{A}$. The type of a region $\mathcal{R}_j$ can be determined by selecting a point $\boldsymbol{u}_j$ in the region, and solving $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}_j) = 0$ for that point, which is here done by resorting to the numerical technique proposed by Porta et al. (2009). If $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}_j) = 0$ has at least one solution, then $\mathcal{R}_j$ is an interior region, otherwise it is exterior.

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

- If $\boldsymbol{u}$ only contains position coordinates of the end-effector, then the outer region will necessarily be exterior to $\mathcal{A}$, because the effector can only reach a compact set of positions in practice.

- A region $\mathcal{R}_j$ whose boundary contains a traversable singularity can be marked as interior, because $\mathcal{R}_j$ contains trajectories that leave $\mathcal{R}_j$ through that singularity.

- A region $\mathcal{R}_j$ whose boundary contains a barrier point $\boldsymbol{u}_i$ with $\boldsymbol{n}_i$ pointing outwards 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 $\mathcal{R}_j$.

For example, these observations allow identifying $\mathcal{R}_1$ as an exterior region in Fig. 5d, and $\mathcal{R}_2$ and $\mathcal{R}_3$ as interior regions. Only the type of $\mathcal{R}_4$ needs to be deambiguated checking a point in the region.

5 Performance tests

We next illustrate the performance of the method by computing the boundaries of several workspaces of planar and spatial manipulators. Manipulators with diverse mechanical structures have been chosen to emphasize the generality of the method, and to illustrate how it performs on the situations described in Section 3, which hinder the application of continuation methods. The method is able to completely determine $\partial\mathcal{A}$ in all situations.

In all plots of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ presented, points corresponding to barrier singularities will be indicated with red vectors on them, drawn on the side of forbidden motion as in Figs. 2 and 5. Also, interior and exterior regions of the workspace will be colored as in such figures.

All results reported have been obtained by using a parallelized version of the method, implemented in C using the libraries of the CUIK platform (Porta et al. 2009), 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 one. For each experiment, we provide the dimension of $\partial\mathcal{A}$ (dim), the number of variables (n_v) and equations (n_c) intervening in Eq. (5), the amount of CPU time required to solve it (t_s), the number of solution boxes returned (n_s), and the accuracy threshold assumed ($s^{\max}$). A summary of such data will be provided in the end.

5.1 Planar manipulators

5.1.1 A 3-RPR manipulator

The derivation of Eq. (5) (Section 4.1), its numerical solution (Section 4.2), and the identification of $\partial\mathcal{A}$ (Section 4.3) are first illustrated on the 3-RPR manipulator of Fig. 6. This example is chosen for its simplicity to describe in detail the application of the method. It is also illustrative of situations

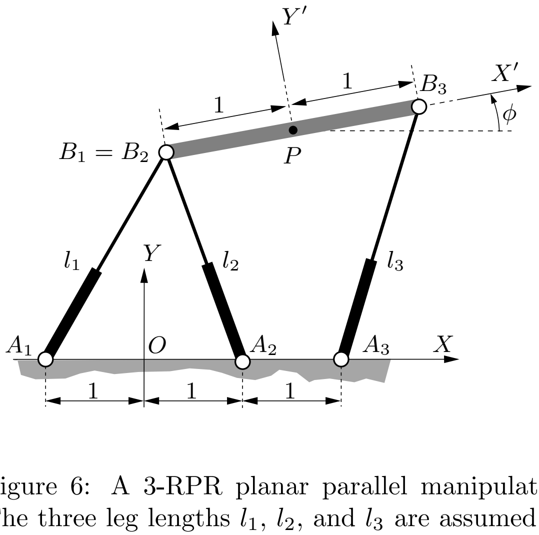


Figure 6: A 3-RPR planar parallel manipulator. The three leg lengths l_1, l_2, and l_3 are assumed to take values within the ranges $[\sqrt{2}, 2]$, $[\sqrt{2}, 2]$, and $[1, 3]$, respectively.

where multi-component workspaces arise (Fig. 3a), which are difficult to map out entirely by continuation. The same test was used by Luh et al. (1996) to exemplify the method by Haug et al. (1996), and by Snyman et al. (2000) as a means of verification. Along the way, we shall correct an error of the results by Luh et al. (1996).

To formulate Eq. (1) in the form assumed in Section 4.1, let $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ denote the position vectors of the anchor points A_i and B_i of the ith leg, relative to the fixed (OXY) and moving $(PX'Y')$ reference frames, respectively. Then, the leg lengths can be written as

$$l_i^2 = |\boldsymbol{p} + \boldsymbol{R}\boldsymbol{b}_i - \boldsymbol{a}_i|^2,$$

for $i = 1, 2, 3$, where $\boldsymbol{p} = [x, y]$ is the position vector of P relative to the fixed frame, and $\boldsymbol{R}$ is the 2×2 rotation matrix of angle ϕ. For the particular dimensions in Fig. 6 these equations become

$$\begin{aligned} l_1^2 &= y^2 - 2ys + x^2 + 2x - 2xc - 2c + 2, \\ l_2^2 &= y^2 - 2ys + x^2 - 2x - 2xc + 2c + 2, \\ l_3^2 &= y^2 + 2ys + x^2 - 4x + 2xc - 4c + 5, \end{aligned} \qquad (16)$$

where c and s refer to the sine and cosine of ϕ and must thus satisfy

$$c^2 + s^2 = 1. \qquad (17)$$

To model the fact that the lengths l_i can only take values within the intervals $[a_i, b_i]$ indicated in Fig. 6, we define $m_i = \frac{b_i + a_i}{2}$ and $h_i = \frac{b_i - a_i}{2}$, and include the constraints

$$(l_i - m_i)^2 + d_i^2 = h_i^2, \qquad (18)$$

for $i = 1, 2, 3$, where the new d_i variables take values within the $[0, h_i]$ range (Appendix A). Thus, in this case, Eq. (1) is the quadratic system formed by Eqs. (16)-(18) with

$$\tilde{\boldsymbol{q}} = [l_1, l_2, l_3, d_1, d_2, d_3, c, s, x, y].$$

To exemplify the situations of Secs. 4.1.1 and 4.1.2, in which the output variables occur in $\tilde{\boldsymbol{q}}$ explicitly, or only implicitly, we next determine the position and full workspaces of this manipulator.

Computing the reachable workspace. The reachable workspace is defined as the set of attainable locations for point P on the moving platform. Thus, $\boldsymbol{u} = [x, y]^{\mathsf{T}}$ for this workspace, and since x and y are already present in $\tilde{\boldsymbol{q}}$, $\mathcal{S}$ is characterized by Eq. (6), which contains $n_c = 16$ equations in $n_v = 17$ variables, where $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0$ is formed by Eqs. (16)-(18).

The progress of the proposed algorithm on solving this system is shown in Fig. 7, top row. Figs. 7a and 7b show intermediate box approximations of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$, after 11 and 25 seconds, containing 677 and 4316 boxes respectively. Fig. 7c displays the final result, which contains $n_s = 50338$ solution boxes, computed in $t_s = 2.83$ minutes at $s^{\max} = 0.01$ (boxes are too small to be appreciated). Fig. 7c also shows the results of the boundary identification procedure explained in Section 4.3, where singularity and region types are indicated using the same conventions as in Figs. 2 and 5. The procedure identifies as barrier singularities those points corresponding to the outer contour, as well as the points of the inner segments $8-8'$ and $8-8''$, which are labelled as in (Luh et al. 1996) for ease of comparison.

As seen in Fig. 7c, this workspace contains four connected components that are symmetric in pairs, corresponding to the symmetric ways of assembling this manipulator. While the components in the middle seem to be connected at point D, there is in fact no trajectory in $\mathcal{C}$ allowing to cross from one component to the other. This fact is consistent with the double barrier arising at D, as seen in the figure, and will become clear upon inspecting the boundaries of the full workspace below. Note that, starting from $\boldsymbol{u}_i$ in Fig. 7c, Haug et al.'s method is able to compute only the upper-right component (Luh et al. 1996).

It is worth mentioning that the segment $8'$-$8''$ was erroneously marked as an interior barrier by Luh et al. (1996), while we detect it as a traversable singularity. This segment corresponds to point P tracing a circle around point B_1, when l_1 is fixed to its lowest value $\sqrt{2}$, while keeping the platform aligned with leg 1. The result by Luh et al. (1996) must be erroneous, because P can really cross this segment from any of its two sides, as shown in Fig. 8. The platform can start from a position where P is to the right of the segment (Fig. 8a), then slide down along line L until it hits the segment (Fig. 8b), and, locking l_1 and l_2 to their values in this configuration, finally perform a rotation about point B_1 by actuating l_3 (Fig. 8c).

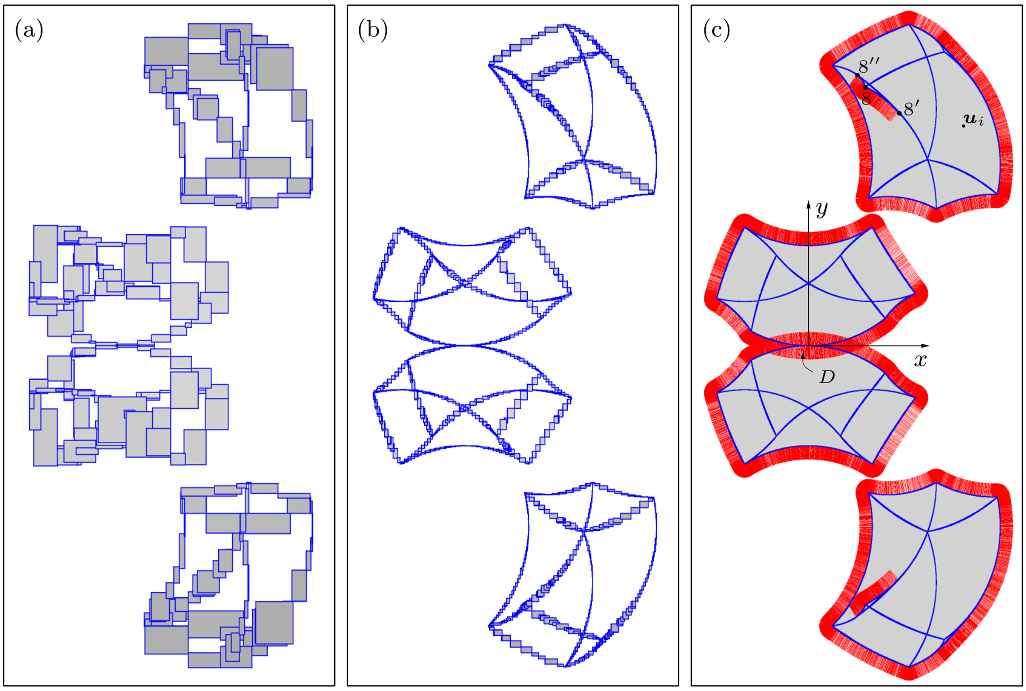


Figure 7: Progress of the proposed algorithm on computing the reachable workspace of the manipulator in Fig. 6. Barrier and traversable singularities are labelled in (c) using the same convention as in Figs. 2 and 5.

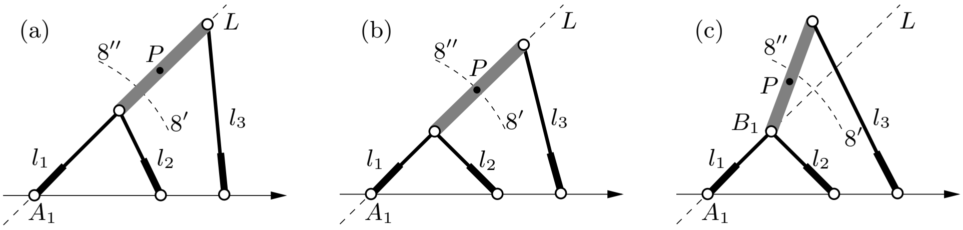


Figure 8: A trajectory in which point P crosses the segment $8'$-$8''$.

Computing the full workspace. The *full* workspace of the 3-RPR manipulator is the set of possible poses for the upper platform. Thus, $\boldsymbol{u} = [x, y, \phi]^{\mathsf{T}}$ for this workspace, but note that ϕ is not present in $\tilde{\boldsymbol{q}}$, so that we are in the situation of Section 4.1.2. However, $\boldsymbol{u}$ is related to $\tilde{\boldsymbol{u}} = [x, y, s, c]$ through the parametric expression

$$\left.\begin{aligned} x &= x \\ y &= y \\ c &= \cos(\phi) \\ s &= \sin(\phi) \end{aligned}\right\}, \tag{19}$$

implying that Eq. (7) is Eq. (19) in this case. Note that $\tilde{\boldsymbol{u}} = [x, y, s, c]^{\mathsf{T}}$ satisfies Eq. (19) for some $\boldsymbol{u} = [x, y, \phi]^{\mathsf{T}}$ if, and only if, it satisfies Eq. (17), which means that, regarding the formulation of Eq. (12), $\boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0$ is simply Eq. (17), and $\boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0$ is the system formed by Eqs. (16) and (18).

Eq. (12) encompasses $n_c = 14$ equations and $n_v = 16$ variables in total, thus yielding a solution set of dimension two in general. A box approximation of this set has been computed by the proposed algorithm in $t_s = 13.6$ minutes, at $s^{\max} = 0.1$, ob-

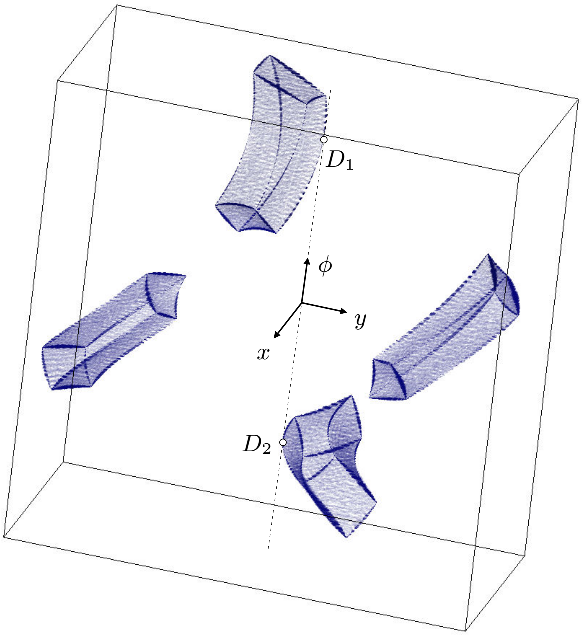


Figure 9: Box approximation of $\partial\mathcal{A}$ of the 3-RPR manipulator in Fig. 6. Boxes are shown with semi-transparent faces, to better appreciate the shape of the enclosed volumes. Points D_1 and D_2 correspond to point D in Fig. 7.

taining $n_s = 218795$ solution boxes. The corresponding box approximation of $\mathcal{S}$ is shown in Fig. 9, projected onto the x, y, and ϕ coordinates.

As shown in the figure, four connected components arise, as anticipated in Fig. 7, each enclosing a cuboid-shaped region of the workspace. All points of $\mathcal{S}$ are classified as boundary barriers in this example; i.e., there are no interior barriers nor traversable singularities relative to this workspace. Note also that, because the boundary is two-dimensional, Haug et al.'s method would only be able to obtain one-dimensional slices of such boundary, whereas the proposed method directly isolates the whole surface.

5.1.2 General multi-loop manipulators

The workspaces in Section 5.1.1 can also be obtained by resorting to specific methods for 3-RPR manipulators (Merlet et al. 1998). However, because they exploit geometric particularities of the manipulator, such methods cannot deal with general multi-loop manipulators. To illustrate the applicability of the proposed technique even to such cases, we show results on the manipulator of Fig. 10, known as the double-butterfly linkage. This manipulator has been used repeatedly in the past to compare the performance of general position analysis methods (Nielsen and Roth 1999; Wampler 2001; Porta et al. 2007; Celaya et al. 2007), but no method has been given yet for computing its workspace, as far as we know.

For this test case, it is assumed that the end-effector is the upper left body in Fig. 10, whose pose is determined by point P and angle θ_1, and that two slider joints are mounted, to let lengths l_5 and l_7 vary within the ranges $[11, 13]$ and $[10, 12]$, respectively.

As for the parameters in Fig. 10, we adopt the same values considered by Nielsen and Roth (1999), Wampler (2001), Porta et al. (2007), and Celaya et al. (2007). 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 $\gamma_0 = 36.87°$, $\gamma_1 = 22.62°$, $\gamma_2 = 53.13°$,

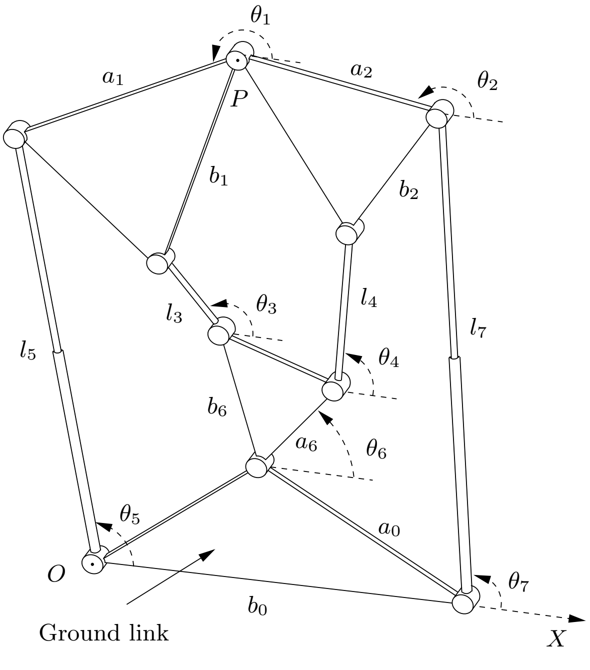


Figure 10: The double-butterfly linkage with variable lengths l_5 and l_7. The fixed reference frame is centered at O with the X-axis aligned as indicated.

$\gamma_6 = 36.87°$, where γ_i is the acute angle between segments a_i and b_i.

By application of the Grübler-Kutzbach criterion (Gogu 2005), it is easy to realise that this manipulator has three degrees of freedom, so that the boundary of its reachable and full workspaces will be one- and two-dimensional, respectively. Such boundaries can be obtained in a way analogous to that for the 3-RPR manipulator (Section 5.1.1), by taking into account that Eq. (1) is formed by:

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:

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

   where c_{γ_i} and s_{γ_i} stand for the cosine and sine of γ_i, and c_i and s_i for those of θ_i.

2. The equations providing the x and y coordinates of P relative to the fixed OXY frame:

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

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 mid-point and half-range of the intervals for l_5 and l_7 (Appendix A).

Details on how these equations can be derived for general multi-loop linkages, and for this one in particular, are given by Porta et al. (2007).

Fig. 11 shows, in blue color, a box approximation of the set $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ corresponding to the reachable workspace of this manipulator, computed by solving Eq. (6) at $s^{\max} = 0.1$ in $t_s = 218$ minutes, where $n_c = 36$ and $n_v = 37$. The resulting approximation contains $n_s = 145369$ boxes and delimits three workspace areas, corresponding to different assembly modes of the mechanism. Note from Fig. 11 that, having several connected components, this workspace would again be difficult to map out entirely using continuation methods (Haug et al. 1996).

The result of the boundary identification process of Section 4.3 is shown in Fig. 12 for one of the curve components in Fig. 11. From the barrier singularities detected, we automatically infer that all grey regions indicated are interior to the workspace. Only the type of region $\mathcal{R}$ is ambiguous, but it is finally detected as an exterior region by checking a point inside it, as explained in Section 4.3.

For completeness, we mention that constant orientation workspaces (the attainable locations for point P for θ_1 fixed to some value) are also easy to compute under the given approach. Their defining equations are simply those corresponding to the reachable workspace, with θ_1 fixed to the desired value. As an example, Fig. 11 shows, overlaid in red color, the set $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ relative to such workspace for $\theta_1 = 30°$, computed for $s^{\max} = 0.1$, which includes $n_s = 5831$ boxes computed in $t_s = 17$ minutes.

5.2 Spatial manipulators

Both the 3-RPR and double-butterfly planar manipulators illustrate situations with multi-component workspaces (Fig. 3a). To encounter situations with multi-component or degenerate boundaries (Figs. 3b and 3c), we next consider Stewart platforms (Section 5.2.1) and spherical manipulators (Section 5.2.2).

5.2.1 Stewart platforms

The Stewart platform (Fig. 13a) is a challenging test case for any workspace determination method. Since its full workspace is six-dimensional, the boundary will be five-dimensional in general, which hinders any attempt of computing it exhaustively, due to the curse of dimensionality. For this reason, and because six-dimensional spaces

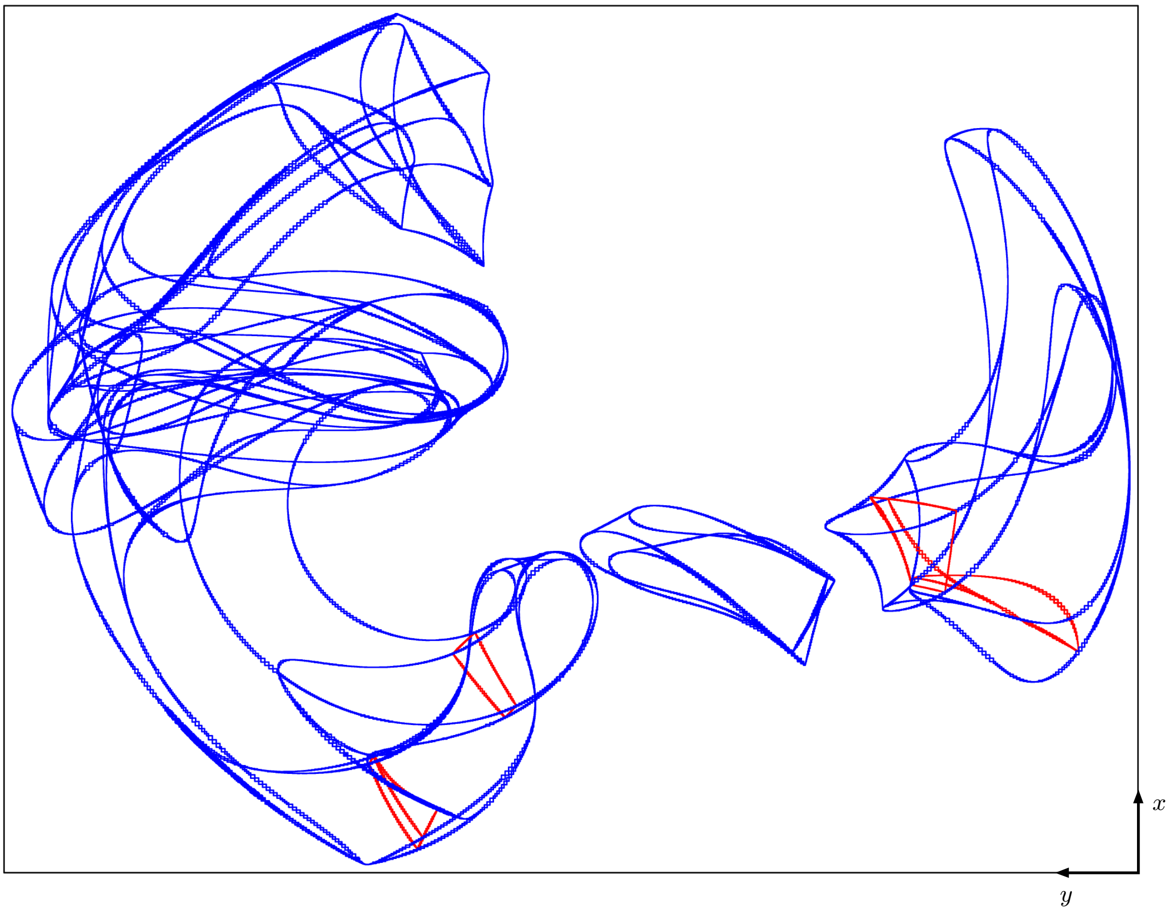


Figure 11: Set $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ corresponding to the reachable workspace (blue) and constant orientation workspace (red) of the double-butterfly linkage of Fig. 10.

are impossible to visualize directly anyway, comprehension on the full workspace of the Stewart platform has been gained by determining lower-dimensional workspacess like (1) the *constant orientation* workspace, or set of attainable locations by a point P on the platform, for a fixed platform orientation (Jo and Haug 1989; Gosselin 1990), (2) the *constant position* workspace, or set of platform orientations for a fixed position of P (Merlet 1995; Bonev and Ryu 2001; Cao et al. 2010), and (3) the *reachable* workspace, or set of locations that P can attain, with at least one platform orientation (Luh et al. 1996; Pernkopf and Husty 2006).

i	1	2	3	4	5	6
$x_{\boldsymbol{a}_i}$	92.58	132.58	40.00	−40.00	−132.58	−92.58
$y_{\boldsymbol{a}_i}$	99.64	30.36	−130.00	−130.00	30.36	99.64
$z_{\boldsymbol{a}_i}$	23.10	23.10	23.10	23.10	23.10	23.10
$x_{\boldsymbol{b}_i}$	30.00	78.22	48.22	−48.22	−78.22	−30.00
$y_{\boldsymbol{b}_i}$	73.00	−10.52	−62.48	−62.48	−10.52	73.00
$z_{\boldsymbol{b}_i}$	−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

Table 1: Geometric paramaters of the Stewart platform studied by Gosselin (1990).

All of these workspaces can be computed by the proposed technique, using a proper choice of the $\boldsymbol{u}$ variables, and fixing others to given values. To illustrate, we next show results on deriving the constant orientation workspace for a particular plat-

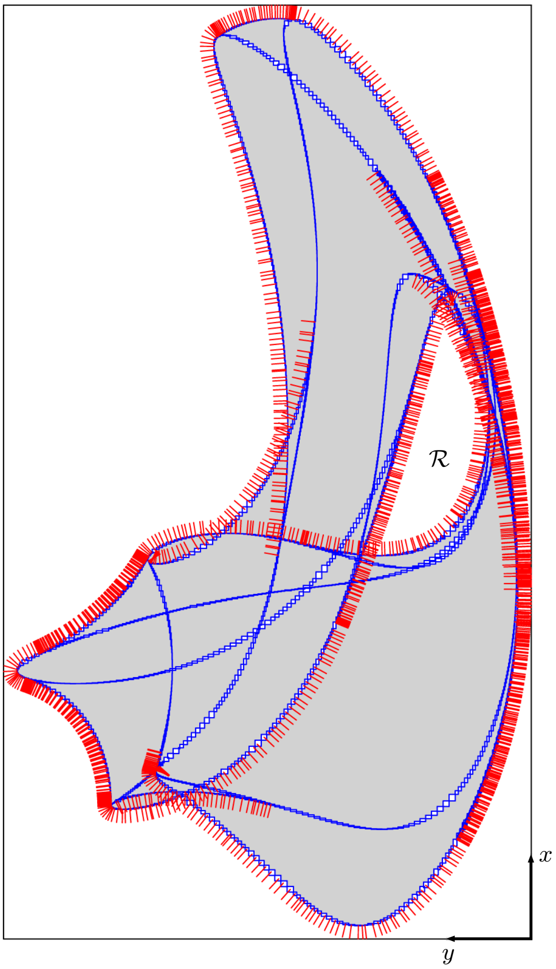


Figure 12: Results of applying the boundary identification process from Section 4.3 to one of the curve components in Fig. 11.

form studied previously by Gosselin (1990).

To formulate the kinematic equations for the Stewart platform, let $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ denote the position vectors of the base and platform anchor points of the ith leg, expressed in fixed ($OXYZ$) and moving ($PX'Y'Z'$) reference frames, respectively. Then, the length of the ith leg can be written as

$$l_i^2 = |\boldsymbol{R}\boldsymbol{b}_i + \boldsymbol{p} - \boldsymbol{a}_i|^2 \qquad (20)$$

for $i = 1, \ldots, 6$, where $\boldsymbol{R}$ is a rotation matrix that provides the orientation of $PX'Y'Z'$ relative to $OXYZ$, and $\boldsymbol{p} = [x, y, z]^\mathsf{T}$ is the position vector of P in the $OXYZ$ frame. Note that if $\boldsymbol{r}_j$ refers to the jth column of $\boldsymbol{R}$, then

$$|\boldsymbol{r}_1|^2 = 1, \qquad (21)$$

$$|\boldsymbol{r}_2|^2 = 1, \qquad (22)$$

$$\boldsymbol{r}_1 \cdot \boldsymbol{r}_2 = 0, \qquad (23)$$

$$\boldsymbol{r_3} = \boldsymbol{r}_1 \times \boldsymbol{r}_2, \qquad (24)$$

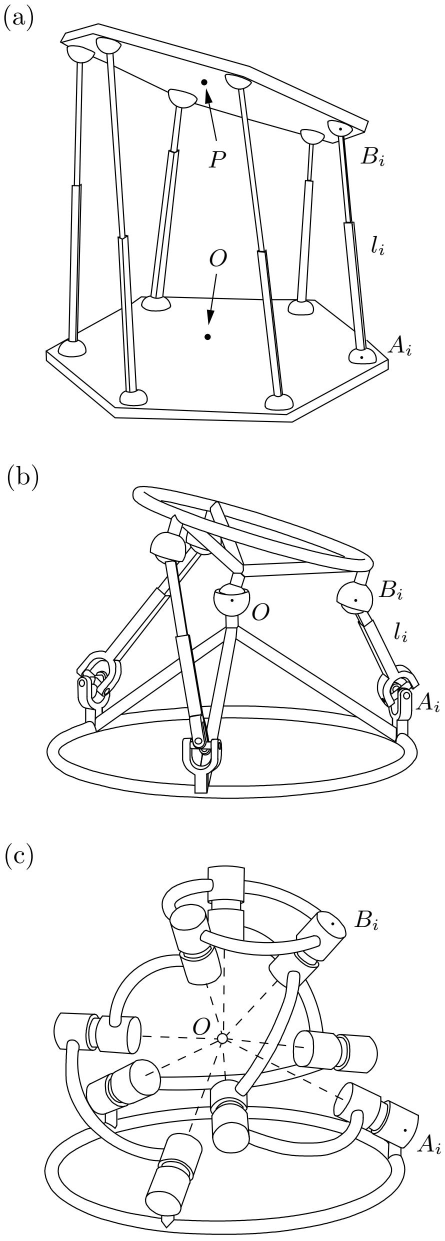


Figure 13: The 6-6 Stewart platform (a), and the 3-UPS/S (b) and 3-RRR (c) spherical platforms.

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

$$(l_i - m_i)^2 + d_i^2 = h_i^2 \tag{25}$$

for $i = 1, \ldots, 6$, where m_i and h_i are the mid-point and half-range of $[l_i^{\min}, l_i^{\max}]$ (see Appendix A).

When computing the constant orientation workspace, $\boldsymbol{R}$ will be a known matrix in the previous equations and, thus, Eq. (1) will be the system formed by Eqs. (20) and (25), for $i = 1, \ldots, 6$, with

$$\tilde{\boldsymbol{q}} = [l_1, \ldots, l_6, d_1, \ldots, d_6, x, y, z]^{\mathsf{T}}.$$

Clearly, $\boldsymbol{u} = [x, y, z]^{\mathsf{T}}$ for this workspace, and we are in the situation of Section 4.1.1, in which $\boldsymbol{u}$ intervenes in $\tilde{\boldsymbol{q}}$ explicitly. Thus, $\mathcal{S}$ is characterized by Eq. (6), for which $n_v = 27$, $n_c = 25$, and $\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = 0$ is the system formed by Eqs. (20) and (25).

Fig. 14 shows 3D views of the box approximation obtained for $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ under the proposed technique, using the parameters indicated in Table 1, assuming that the orientation matrix has been fixed to the identity matrix. The approximation was computed at $s^{\max} = 0.1$ in $t_s = 45$ CPU minutes, and consists of $n_s = 1677906$ boxes describing an umbrella-like surface overall. In fact, this workspace has an additional connected component symmetric to the one of Fig. 14, 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 by Gosselin (1990).

To better appreciate the shape of the enclosed volume, Fig. 14 plots constant-z slices of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ indicating the results of the boundary identification process of Section 4.3. Note from the plots that it would be difficult to compute such slices by continuation (Haug et al. 1996), because many slices present multi-component boundaries and internal voids that difficult the application of the ray-shooting technique described in Section 3. 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, because V_2 and V_3 are hidden behind V_1. This situation clearly occurs on other slices as well. While it is true that Luh et al. (1996) were able to compute $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ using the method by Haug et al. (1996), they did so by defining particular slices of this set, obtained by cutting the umbrella with planes through line L shown in Fig. 14. This solution avoids the appearance of internal voids within each slice but, obviously, it relies on a-priori knowledge of the result.

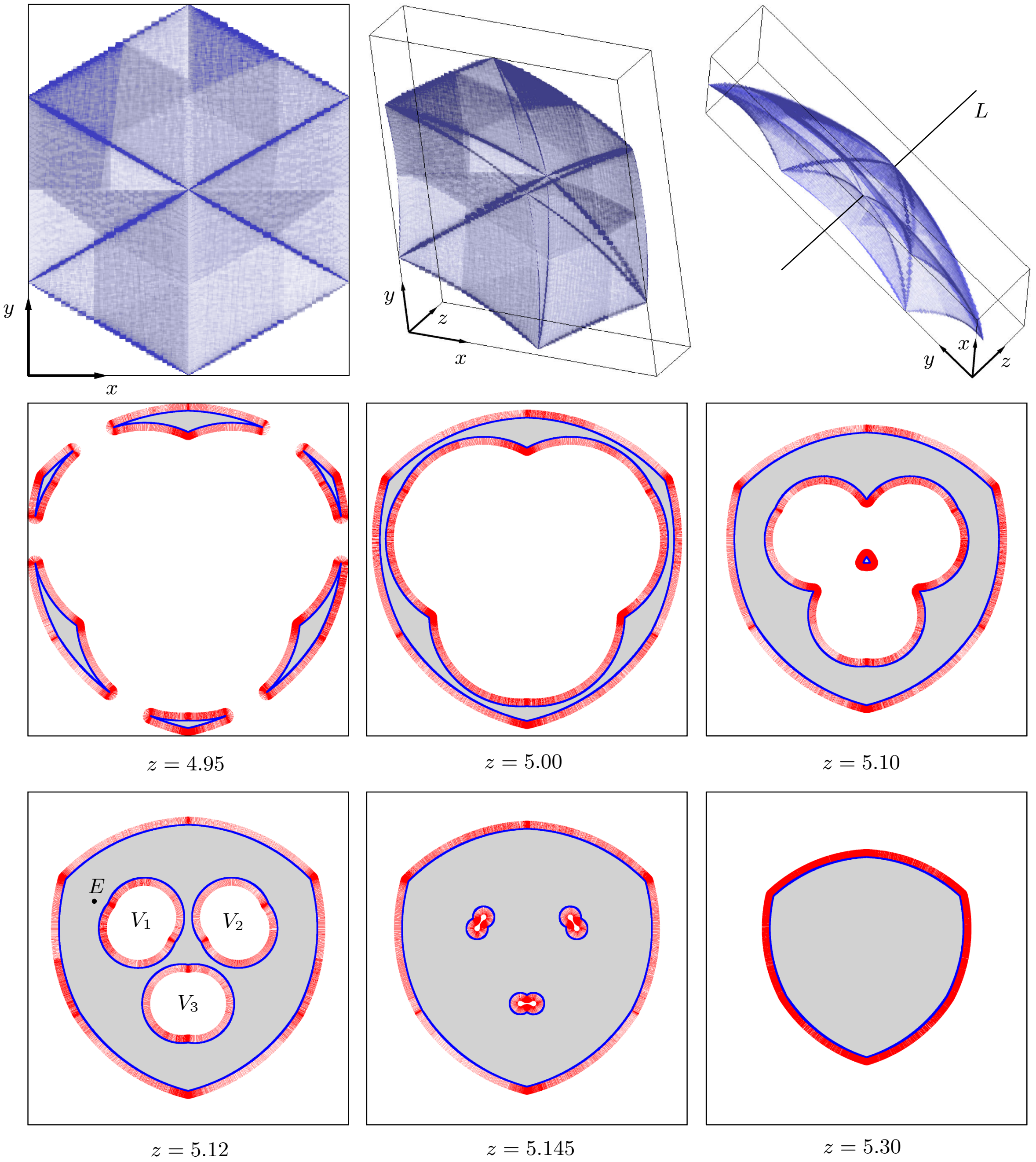
5.2.2 Spherical platforms

Because of the complexity of their defining equations, orientation workspaces are considered among the most difficult ones to compute and represent (Bonev and Ryu 2001; Merlet and Gosselin 2008; Jiang and Gosselin 2009; Cao et al. 2010). Their derivation could be illustrated on Stewart platforms, but we shall do so on spherical parallel manipulators (SPM) because this will lead to one example of the degenerate boundaries mentioned in Section 3, where continuation methods are particularly unsuccessful.

The examples are taken from Bonev and Gosselin (2006), and correspond to popular architectures of three degree-of-freedom SPMs: the 3-UPS/S and 3-RRR designs, depicted in Fig. 13b and 13c. These 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 by Bonev and Gosselin (2006).

To derive Eq. (1) for such manipulators, note that each leg imposes the same constraint on the moving platform, irrespectively 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-RRR 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. Thus, both designs are kinematically equivalent. In turn, 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-RRR designs, Eq. (1) can be formulated as the system formed by Eqs. (20) to (25) with $\boldsymbol{p} = 0$, for $i = 1, 2, 3$, assuming that both the fixed and moving reference frames are centered at O.

In general, the orientation workspace is defined as the set of possible values for three orientation



$z = 4.95$ $z = 5.00$ $z = 5.10$

$z = 5.12$ $z = 5.145$ $z = 5.30$

Figure 14: Top row: Different views of the boundary of the constant orientation workspace of a Stewart platform, computed at $s^{\max} = 0.1$. Boxes are shown with semitransparent walls to better appreciate the resulting shape. Middle and bottom rows: Slices of the previous boundary for different values of z, computed at $s^{\max} = 0.01$. All points of $\mathcal{S}$ are classified as barrier singularities in this case.

angles of the platform. Although any set of Euler angles could be used for such matter, we shall here adopt the tilt-and-torsion convention assumed by Bonev and Gosselin (2006) to facilitate the comparison of results. The moving platform is oriented in two stages under this convention: The *tilt* stage, in which the platform rotates an angle ϕ about the Z' axis, and then an angle θ about the rotated Y' axis, and the *torsion* stage, in which it rotates an angle $\sigma - \phi$ around the rotated Z' axis. The platform orientation is thus defined by the angles ϕ, θ, and σ (called azimuth, tilt, and torsion, respectively), and $\boldsymbol{R}$ can be written as

$$\boldsymbol{R} = \boldsymbol{R}_z(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\gamma),$$

where $\gamma = \sigma - \phi$. Thus, the columns of $\boldsymbol{R}$ are

$$\boldsymbol{r}_1 = \begin{bmatrix} \cos(\phi)\cos(\theta)\cos(\gamma) - \sin(\phi)\sin(\gamma) \\ \sin(\phi)\cos(\theta)\cos(\gamma) + \cos(\phi)\sin(\gamma) \\ -\sin(\theta)\cos(\gamma) \end{bmatrix}, \quad (26)$$

$$\boldsymbol{r}_2 = \begin{bmatrix} -\cos(\phi)\cos(\theta)\sin(\gamma) - \sin(\phi)\cos(\gamma) \\ -\sin(\phi)\cos(\theta)\sin(\gamma) + \cos(\phi)\cos(\gamma) \\ \sin(\theta)\sin(\gamma) \end{bmatrix}, \quad (27)$$

$$\boldsymbol{r}_3 = \begin{bmatrix} \cos(\phi)\sin(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\theta) \end{bmatrix}. \quad (28)$$

Under the previous convention, Bonev and Gosselin (2006) define the orientation workspace as the set of possible values that $\boldsymbol{u} = [\phi, \theta, \sigma]^\mathsf{T}$ can attain, 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 of Section 4.1.2, in which $\boldsymbol{u}$ does not intervene explicitly in the variables of Eq. (1), but it can be related to $\tilde{\boldsymbol{u}} = \left[\boldsymbol{r}_1{}^\mathsf{T}, \boldsymbol{r}_2{}^\mathsf{T}, \boldsymbol{r}_3{}^\mathsf{T}\right]^\mathsf{T}$ using Eqs. (26)-(28).

Therefore, $\boldsymbol{\eta}(\tilde{\boldsymbol{u}}) = 0$ is given by Eqs. (21)-(24), and $\boldsymbol{\Psi}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}}) = 0$ consists of Eqs. (20) and (25) with $\tilde{\boldsymbol{z}} = [l_1, l_2, l_3, d_1, d_2, d_3]$. Overall, Eq. (12) contains $n_c = 19$ equations and $n_v = 21$ variables and, thus, the boundaries of the orientation workspace are expected to be 2-dimensional.

For the particular examples shown next, we shall assume the same symmetry conditions as Bonev and Gosselin (2006). Namely, A_i and B_i will lie on a unit sphere centered at O, with position vectors

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

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

and all ranges for the leg angles will be the same, with $\alpha_{i,min} = \alpha_{min}$ and $\alpha_{i,max} = \alpha_{max}$, $i = 1, 2, 3$. Note however that whereas Bonev and Gosselin's technique is only valid if such symmetries hold, the method we propose remains general.

Fig. 15 shows the resulting box approximation of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$, computed at $s^{\max} = 0.1$ for $\beta_1 = 0°$, $\beta_2 = 35°$, $\alpha_{min} = 20°$, $\alpha_{max} = 130°$. These parameters correspond to one of the cases analyzed explicitly by Bonev and Gosselin (2006), who provide constant-torsion slices of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ for these manipulators. The approximation was obtained in $t_s = 30$ minutes and contains $n_s = 156699$ solution boxes. As expected, $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ is a surface in the $\{\theta, \phi, \sigma\}$-space and, by analyzing the neighborhood relationships of the returned boxes, it can be shown to contain just one connected component. Figs. 16 top and middle show, respectively, a $\sigma = -30°$ slice of the box approximation shown in Fig. 15, and the results of the boundary identification process in Section 4.3 applied to such slice. The resulting curve, and the inner regions detected, match those by Bonev and Gosselin (2006) when plotted in polar coordinates (Fig. 16, bottom).

While the orientation workspace will generally have a two-dimensional boundary, such boundary may degenerate into a lower-dimensional set for particular choices of the geometric parameters, thus posing serious difficulties to the continuation method by Haug et al. (1996). This is what occurs on the *Agile Eye* for example, a well-known instance of the 3-RRR design in Fig. 13c, with parameters $\beta_1 = \beta_2 = \arccos(1/\sqrt{3})$, $\alpha_{min} = 0$, and $\alpha_{max} = \pi$ (Bonev et al. 2006). As mentioned in Section 3, computing the boundary by continuation is almost impossible in such cases, because, having to proceed on a slice-by-slice basis, the ray shooting technique will fail to converge to the boundary almost always. On the contrary, the presented technique is immune to such situations: If the same equations considered for ob-

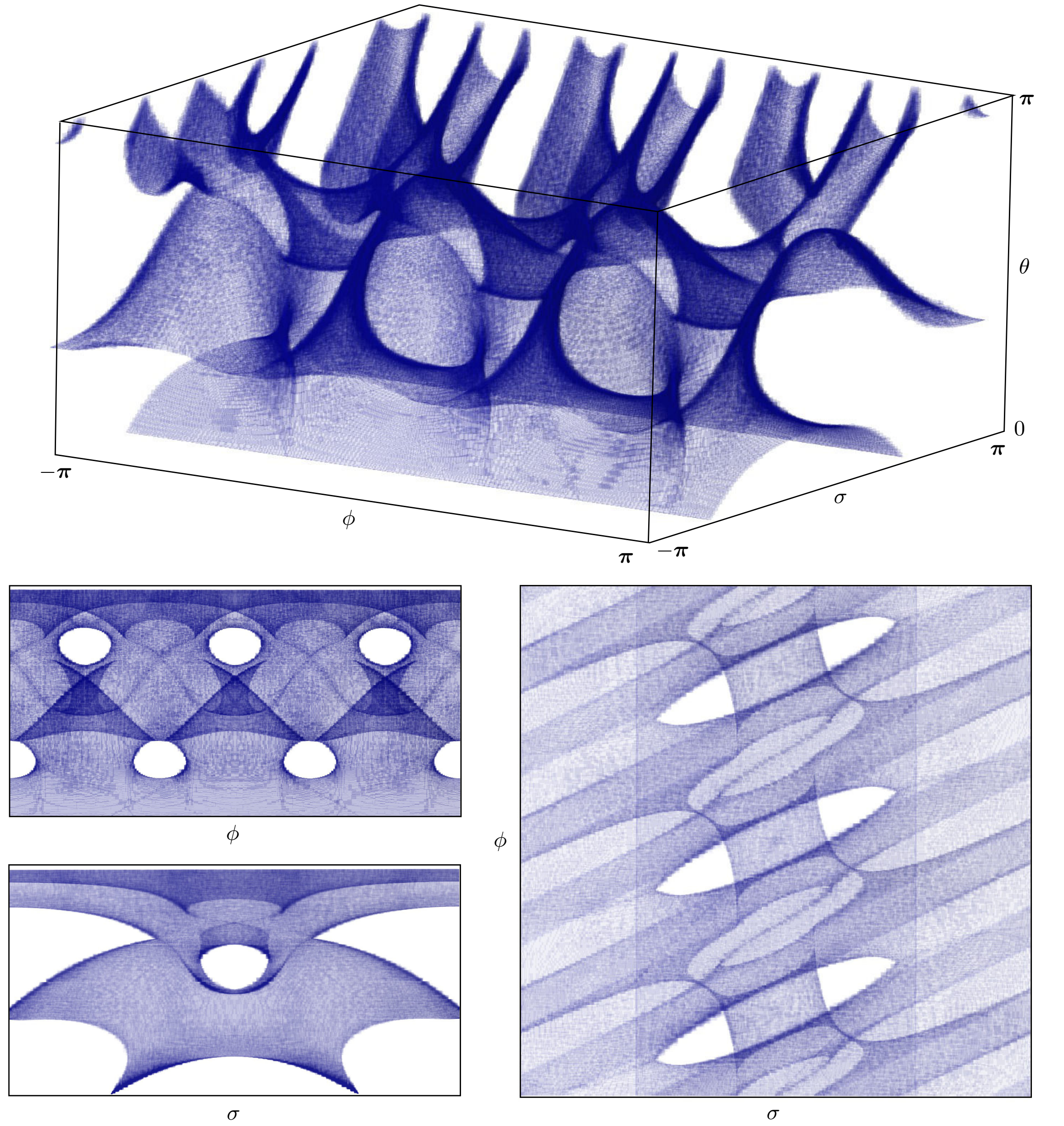


Figure 15: Top: 3D view of a box approximation for the set $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ corresponding to the orientation workspace of the 3-UPS/S manipulator of Fig. 13b. Bottom: orthogonal projections of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ onto the $\phi-\theta$, $\sigma-\theta$, and $\sigma-\phi$ planes. Boxes are drawn with semitransparent walls to better appreciate the shape of the surface.

taining the plot in Fig. 15 are now used for determining the workspace of the Agile Eye relative to $\boldsymbol{u}=[\phi,\theta,\sigma]^{\mathsf{T}}$, we obtain the boundary curves depicted in Fig. 17 ($n_s = 100636$ boxes in $t_s = 17$ minutes, for $s^{\max} = 0.1$), which agree with those described by Bonev and Gosselin (2006).

Manipulator	Workspace	$\boldsymbol{dim}$	$\boldsymbol{n_v}$	$\boldsymbol{n_c}$	$\boldsymbol{s^{\max}}$	$\boldsymbol{n_s}$	$\boldsymbol{t}$ (min)
3-RPR	reachable	1	17	16	0.01	50338	2.83
	full	2	16	14	0.1	218795	13.6
Double-butterfly	reachable	1	37	36	0.1	145369	218
	constant-orientation	1	34	33	0.1	5831	17
Stewart	constant-orientation	2	27	25	0.1	1677906	45
3-UPS/S	orientation (full)	2	21	19	0.1	156699	30
Agile Eye	orientation (full)	1	21	19	0.1	100636	17

Table 2: Performance data of the proposed method, on computing the reported workspaces.

5.3 Performance data

Table 2 provides a summary of the performance data for all experiments reported. While execution times may appear to be large, we note that the emphasis of the paper has been on presenting a method as general as possible, valid for a wide range of manipulators. If manipulator particularities are exploited, however, execution times can be considerably shortened. When computing the constant orientation workspace of the Stewart platform, for example, $\boldsymbol{\Phi_z}$ is diagonal under the proposed formulation, and using the fact that leg lengths will never vanish in practice, the rank deficiency condition of $\boldsymbol{\Phi_z}$ can be written as $d_1 d_2 d_3 d_4 d_5 d_6 = 0$. This equation can be converted into quadratic form using appropriate changes of variables, which allows formulating Eq. (5) with much less variables and equations. Introducing such simplifications, the boundary of the constant orientation workspace can be computed in 0.38 minutes, rather than in the 45 minutes reported in Table 2. In a similar way, the boundaries of the orientation workspace of the 3-UPS/S manipulator can be computed in 3.76 minutes instead of in the 30 minutes indicated in the table, and those of the Agile Eye in 2.55 minutes instead of 17 minutes (Bohigas et al. 2011).

6 Non-smooth manifolds

Throughout the paper, it has been assumed that $\boldsymbol{\Phi_q}$ is full rank at all points $\boldsymbol{q} \in \mathcal{C}$, so that $\mathcal{C}$ is guaranteed to be a smooth manifold of dimension $n_q - n_e$ (Section 2). In practice, however, manipulators can be encountered for which $\boldsymbol{\Phi_q}$ is rank deficient at some points of $\mathcal{C}$. Since all maximal minors of $\boldsymbol{\Phi_q}$ vanish at those points, it is not possible to guarantee on them, using the Implicit Function Theorem, that $\mathcal{C}$ is locally diffeomorphic to $\mathbb{R}^{n_q - n_e}$. This circumstance, however, does not modify the presented algorithm significantly.

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

Moreover, observe that the points of $\mathcal{S}$ for which $\boldsymbol{\Phi_q}$ is rank deficient cannot be classified into barrier or traversable singularities, as such classification depends on the existence of a parameterization $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{v})$ of $\mathcal{C}$ in a neighborhood of $\boldsymbol{q}_0$ (Section 2.3), which cannot be guaranteed if all maximal minors of $\boldsymbol{\Phi_q}$ vanish. However, we note that points where $\boldsymbol{\Phi_q}$ is rank deficient correspond to so-called increased instantaneous mobility singularities (Zlatanov 1998), or configuration space singularities (Park and Kim 1999), which can be interpreted as points where the controllability of the manipulator is lost, irrespectively of the choice of actuated coordinates. These configurations are to be avoided and, hence, they can simply be marked as uncontrollable in the returned output.

Finally, we realise that the rank deficiency of $\boldsymbol{\Phi_q}$ is a stronger condition than the rank deficiency of $\boldsymbol{\Phi_z}$. Thus, the set of points for which $\boldsymbol{\Phi_q}$ is rank deficient will be a subset of dimension lower than that of $\mathcal{S}$ in general, so that the boundary identification process in Section 4.3 will not be altered in a significant way.

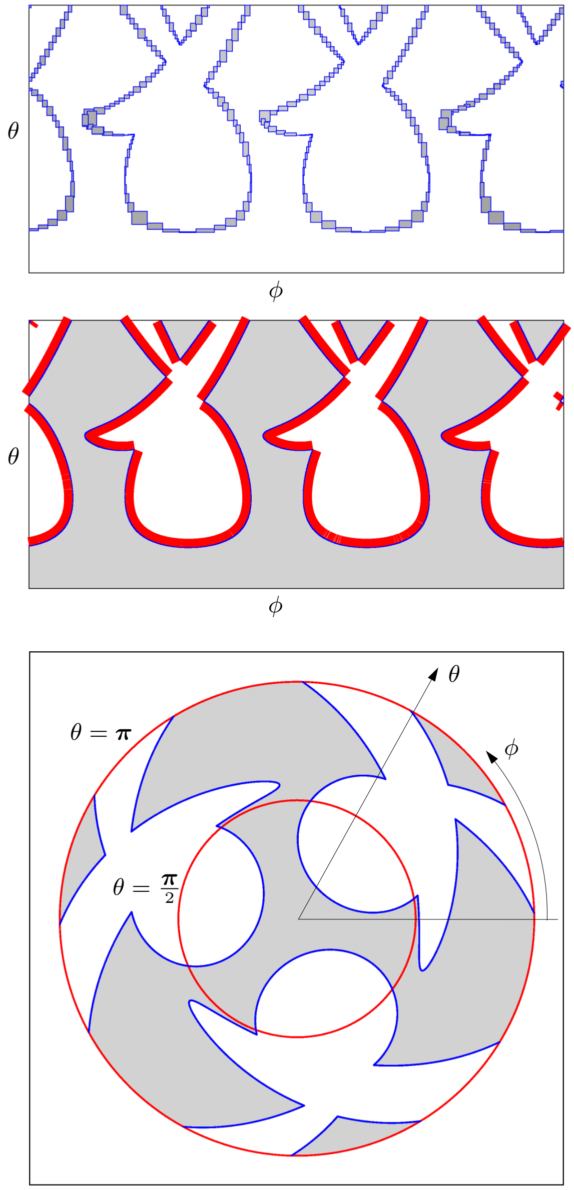


Figure 16: Constant torsion (σ) workspace of the spherical parallel manipulator of Fig. 13b, with parameters $\beta_1 = 45°$, $\beta_2 = 35°$, $\alpha_{min} = 20°$, $\alpha_{max} = 130°$, $\sigma = -30°$.

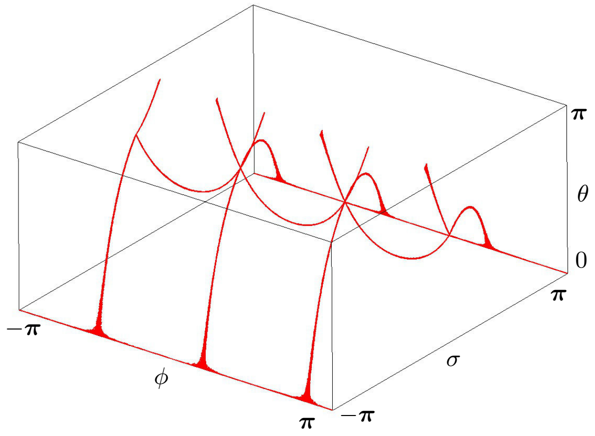


Figure 17: The workspace of the Agile Eye has one-dimensional boundaries. With probability one, continuation methods would fail to compute any point on such boundary.

7 Conclusions

This paper has presented a new method for workspace boundary determination on general manipulators. A principal advantage of the method is its ability to converge to all boundary points, even on multi-component or degenerate boundaries. Previous methods for the same purpose cannot ensure this property, as shown in the paper, because they rely on continuation, which requires the availability of one point for each connected component of the boundary. Since no method has been given yet to compute such points, existing continuation methods presently rely on ray-shooting techniques, which require a-priori knowledge of manipulator configurations, and may miss boundary segments in several situations. Such drawbacks are all circumvented by the alternative method we propose.

Using the method, the computation of the boundary becomes feasible on three- or lower-dimensional workspaces (like those of planar or spherical manipulators). For six-dimensional workspaces (like those of spatial manipulators) such a computation is a difficult task independently of the methodology used, because the curse of dimensionality must inevitably be faced. In order to visualize such workspaces, the common practice is to compute lower-dimensional subsets meaningful to the robot designer. In this respect, the boundaries of the reachable workspace, the constant-orientation workspace, and the constant-position workspace can all be computed by the proposed method, using a proper choice of the $\boldsymbol{u}$ variables, and fixing other variables to known values if required. It is also worth noting that, while most previous methods are only able to obtain one-dimensional slices of such boundaries, the proposed method directly isolates 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 identifying the boundary of such workspaces seems plausible using a similar approach, several modifications need to be introduced into the proposed method in order to be able to do so. This point, together with the extension of the method to consider other constraints, such as collision constraints, certainly deserve further attention.

Acknowledgements

The authors wish to thank Dr. Josep M. Porta for fruitful discussions around the topic of the paper, and for helping to implement the method on the CUIK library. This work has been partially supported by the Spanish Ministry of Science and Innovation under contract DPI2010-18449.

A Modelling joint limits

This appendix shows that joint limit constraints can easily be modelled as equality constraints.

There are two types of limit constraints that need to be treated: those refering to distance constraints and those refering to angular constraints. Typically, they arise on slider and revolute joints, respectively, but combinations of both may be encountered on any other joint.

Note on the one hand that, if q_i is a joint distance that must satisfy

$$q_i^{min} \leq q_i \leq q_i^{max}, \tag{29}$$

then we can equivalently re-write this constraint as

$$(q_i - m_i)^2 + d_i^2 = h_i^2, \tag{30}$$

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 new variable. The values m_i and h_i are called the mid-point and half-range of the interval $[q_i^{min}, q_i^{max}]$, and Eq. (30) 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 (29) if, and only if, it satisfies Eq. (30) for some value of d_i.

On the other hand, if q_i is a joint angle that must satisfy

$$-\alpha_i \leq q_i \leq \alpha_i, \tag{31}$$

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

$$c_{q_i} = t_i^2 + \cos\alpha_i, \tag{32}$$

where t_i is a new variable that can take any value. Again q_i satisfies (31) if, and only if, it satisfies Eq. (32) for some t_i.

B Boundary condition

This appendix shows that the rank deficiency of $\boldsymbol{\Phi_z}$ at a point $\boldsymbol{q} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}]^\mathsf{T} \in \mathcal{C}$ is a necessary condition for $\boldsymbol{u}$ to belong to $\partial\mathcal{A}$. Also, results from the theory of smooth maps are recalled, to show that points $\boldsymbol{q} \in \mathcal{C}$ for which $\boldsymbol{\Phi_z}$ is rank deficient correspond to critical points of the projection of $\mathcal{C}$ on $\mathcal{U}$ (Canny 1988).

Observe first that if $\boldsymbol{\Phi_z}$ is full rank at a point $\boldsymbol{q} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}]^\mathsf{T} \in \mathcal{C}$, then there exists a non-vanishing $n_e \times n_e$ minor of $\boldsymbol{\Phi_z}$, say relative to the variables $\boldsymbol{z'}$, and by the Implicit Function Theorem it is possible to find a function $\boldsymbol{z'} = \boldsymbol{F}(\boldsymbol{u'})$ relating $\boldsymbol{z'}$ with the remaining $\boldsymbol{u'}$ variables, satisfying $\boldsymbol{\Phi}(\boldsymbol{F}(\boldsymbol{u'}), \boldsymbol{u'}) = 0$ (Krantz and Parks 2002). Thus, the $\boldsymbol{u'}$ variables, which include the $\boldsymbol{u}$ ones, can be used as a local parameterization of $\mathcal{C}$ around $(\boldsymbol{z'}, \boldsymbol{u'})$, implying that arbitrary values in a neighborhood of $\boldsymbol{u}$ have a corresponding $\boldsymbol{z}$ satisfying $\Phi(\boldsymbol{z}, \boldsymbol{u}) = 0$, so that $\boldsymbol{u}$ must lie in the interior of $\mathcal{A}$. Thus, $\boldsymbol{\Phi_z}$ must be rank-deficient at $\boldsymbol{q}$, for $\boldsymbol{u}$ to belong to $\partial\mathcal{A}$.

Now let $\boldsymbol{\Gamma} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_u}$ be an arbitrary smooth map. Recall that the differential of $\boldsymbol{\Gamma}$ at a point $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is the Jacobian matrix $\boldsymbol{\Gamma_q} = [\frac{\partial \Gamma_i}{\partial q_j}]$, which relates infinitesimal changes in the domain of $\boldsymbol{\Gamma}$ to the corresponding changes in the value of $\boldsymbol{\Gamma}$. A point $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is said to be a *critical point* of $\boldsymbol{\Gamma}$, if $\boldsymbol{\Gamma}_{\boldsymbol{q}}$ is rank deficient. At a critical point, thus, the images of the infinitesimal changes in $\boldsymbol{q}$ do not span the whole set of possible changes in the value of $\boldsymbol{\Gamma}$, because $\boldsymbol{\Gamma}_{\boldsymbol{q}}$ is not surjective.

One can also speak of the critical points of $\boldsymbol{\Gamma}$ with domain restricted to some submanifold $\mathcal{C} \subseteq \mathbb{R}^{n_q}$, to mean the points $\boldsymbol{q} \in \mathcal{C}$ for which $\boldsymbol{\Gamma}_{\boldsymbol{q}}$ is rank deficient. Since $\boldsymbol{\Gamma}_{\boldsymbol{q}}$ maps the tangent space to $\mathcal{C}$ at $\boldsymbol{q}$, to the tangent space to $\mathbb{R}^{n_u}$ at $\boldsymbol{\Gamma}(\boldsymbol{q})$, at such points it happens, geometrically, that the former space gets mapped to a subspace of $\mathbb{R}^{n_u}$ of dimension lower than n_u. The images through $\boldsymbol{\Gamma}$ of these points are called the *critical values* of the restriction of $\boldsymbol{\Gamma}$ to $\mathcal{C}$.

Suppose now that $\mathcal{C}$ is defined implicitly by $\boldsymbol{\Phi}(\boldsymbol{q}) = 0$ as in Section 2, where $\boldsymbol{\Phi} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_e}$ is a smooth map. In order for $\mathcal{C}$ to really have a manifold structure we assume that $\boldsymbol{\Phi}_{\boldsymbol{q}}$ has full rank at all points in $\mathcal{C}$, so that, by the Implicit Function theorem, it will always be possible to define a proper parameterization of a neighborhood of any $\boldsymbol{q} \in \mathcal{C}$. The following holds under this setting (Canny 1988, page 102).

A point $\boldsymbol{q} \in \mathcal{C}$ is a critical point of $\boldsymbol{\Gamma}$ with domain restricted to $\mathcal{C}$ if, and only if, the matrix

$$(\boldsymbol{\Phi}, \boldsymbol{\Gamma})_{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{q}} \\ \cdots \\ \boldsymbol{\Gamma}_{\boldsymbol{q}} \end{bmatrix},$$

has rank less than $n_e + n_u$ at $\boldsymbol{q}$.

By the previous result the critical points of the projection map $\boldsymbol{\pi}_{\boldsymbol{u}}$ with domain restricted to $\mathcal{C}$ are the points $\boldsymbol{q}$ for which the matrix

$$(\boldsymbol{\Phi}, \boldsymbol{\pi}_{\boldsymbol{u}})_{\boldsymbol{q}} = \left[\begin{array}{c:c} \boldsymbol{\Phi}_{\boldsymbol{z}} & \boldsymbol{\Phi}_{\boldsymbol{u}} \\ \hdashline \boldsymbol{0} & \boldsymbol{I}_{n_u} \end{array}\right],$$

is rank deficient, where $\boldsymbol{I}_{n_u}$ denotes the $n_u \times n_u$ identity matrix. Note thus that $(\boldsymbol{\Phi}, \boldsymbol{\pi}_{\boldsymbol{u}})_{\boldsymbol{q}}$ is rank deficient if, and only if, its upper-left block $\boldsymbol{\Phi}_{\boldsymbol{z}}$ is rank deficient.

C Loss of output mobility

Let $\boldsymbol{q}(t) = (\boldsymbol{z}(t), \boldsymbol{u}(t))$ be an arbitrary trajectory in $\mathcal{C}$, parametrized in t, traversing a given point $\boldsymbol{q}_0 = (\boldsymbol{z}_0, \boldsymbol{u}_0) \in \mathcal{S}$ for $t = t_0$. Let $\boldsymbol{n}_0$ denote the normal vector to $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ at $\boldsymbol{u}_0$. This appendix shows that the component of $\dot{\boldsymbol{u}}(t_0)$ along $\boldsymbol{n}_0$ must vanish, independently of the chosen trajectory $\boldsymbol{q}(t)$, so that $\dot{\boldsymbol{u}}(t_0)$ must necessarily lie in the tangent space to $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ at $\boldsymbol{u}_0$.

Making the substitution $\boldsymbol{q} = \boldsymbol{q}(t)$ in Eq. (2), and evaluating the time derivatives of this equation at $\boldsymbol{q} = \boldsymbol{q}_0$ yields

$$\boldsymbol{\Phi}_{\boldsymbol{u}}(\boldsymbol{q}_0)\dot{\boldsymbol{u}}(t_0) + \boldsymbol{\Phi}_{\boldsymbol{z}}(\boldsymbol{q}_0)\dot{\boldsymbol{z}}(t_0) = 0, \qquad (33)$$

Since $\boldsymbol{q}_0 \in \mathcal{S}$, $\boldsymbol{\Phi}_{\boldsymbol{z}}$ must be rank deficient at $\boldsymbol{q}_0$, and there must exist a non-null vector $\boldsymbol{\xi}_0$ such that

$$\boldsymbol{\Phi}_{\boldsymbol{z}}{}^{\mathsf{T}}\boldsymbol{\xi}_0 = 0. \qquad (34)$$

Multiplying both sides of Eq. (33) by $\boldsymbol{\xi}_0{}^{\mathsf{T}}$, and taking into account Eq. (34), we get

$$\boldsymbol{\xi}_0{}^{\mathsf{T}}\boldsymbol{\Phi}_{\boldsymbol{u}}(\boldsymbol{q}_0)\dot{\boldsymbol{u}}(t_0) = 0. \qquad (35)$$

Because this holds for all vectors $\dot{\boldsymbol{u}}(t_0)$ in the tangent space of $\boldsymbol{\pi}_{\boldsymbol{u}}(\mathcal{S})$ at $\boldsymbol{u}_0$, the normal vector $\boldsymbol{n}_0$ must be

$$\boldsymbol{n}_0 = \boldsymbol{\Phi}_{\boldsymbol{u}}(\boldsymbol{q}_0)^{\mathsf{T}}\boldsymbol{\xi}_0. \qquad (36)$$

Now, the component of $\dot{\boldsymbol{u}}(t)$ along $\boldsymbol{n}_0$ is given by

$$\dot{\boldsymbol{u}}_{\boldsymbol{n}_0} = \boldsymbol{n}_0{}^{\mathsf{T}}\dot{\boldsymbol{u}}(t_0),$$

but using Eqs. (35) and (36) we finally realise that

$$\dot{\boldsymbol{u}}_{\boldsymbol{n}_0} = \boldsymbol{\xi}_0{}^{\mathsf{T}}\boldsymbol{\Phi}_{\boldsymbol{u}}(\boldsymbol{q}_0)\dot{\boldsymbol{u}}(t_0) = 0,$$

on all possible trajectories $\boldsymbol{q}(t)$ in $\mathcal{C}$ through $\boldsymbol{q}_0$.

References

Abdel-Malek, K., Adkins, F., Yeh, H. J., and Haug, E. (1997). On the determination of boundaries to manipulator workspaces. *Robotics and Computer Integrated Manufacturing*, 13(1):63–72.

Abdel-Malek, K. and Yeh, H. (1997). Analytical boundary of the workspace for general 3-DOF mechanisms. *The International Journal of Robotics Research*, 16(2):198.

Adkins, F. A. (1996). *Numerical continuation and bifurcation methods for mechanism workspace and controllability issues.* PhD thesis, Ph.D. Thesis, The University of Iowa, Iowa City, Iowa 52242.

Allgower, E. L. and Georg, K. (1990). *Numerical Continuation Methods*. Srpinger-Verlag, Berlin.

Badescu, M. and Mavroidis, C. (2004). New performance indices and workspace analysis of reconfigurable hyper-redundant robotic arms. *The International Journal of Robotics Research*, 23(6):643–659.

Bohigas, O., Ros, L., and Manubens, M. (2010). A complete method for workspace boundary determination. *Advances in Robot Kinematics: Motion in Man and Machine*, pages 329–338.

Bohigas, O., Ros, L., and Manubens, M. (2011). Computing end-effector singularities of Stewart platforms. In preparation.

Bonev, I., Chablat, D., and Wenger, P. (2006). Working and assembly modes of the Agile Eye. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2317–2322.

Bonev, I. and Ryu, J. (2001). A new approach to orientation workspace analysis of 6-DOF parallel manipulators. *Mechanism and Machine Theory*, 36(1):15–28.

Bonev, I. A. and Gosselin, C. M. (2006). Analytical determination of the workspace of symmetrical spherical parallel mechanisms. *IEEE Transactions on Robotics*, 22(5):1011–1017.

Canny, J. (1988). *The Complexity of Robot Motion Planning*. The MIT Press, Cambridge, Massachusetts.

Cao, Y., Huang, Z., Zhou, H., and Ji, W. (2010). Orientation workspace analysis of a special class of Stewart-Gough parallel manipulators. *Robotica*, 28(7):989–1000.

Celaya, E., Creemers, T., and Ros, L. (2007). Exact interval propagation for the efficient solution of planar linkages. In *Proceedings of 12th World Conference in Mechanism and Machine Science*.

De Jalón, J. G. and Bayo, E. (1993). *Kinematic and Dynamic Simulation of Multibody Systems*. Springer Verlag.

Gogu, G. (2005). Mobility of mechanisms: a critical review. *Mechanism and machine theory*, 40(9):1068–1097.

Gosselin, C. (1990). Determination of the workspace of 6-DOF parallel manipulators. *ASME Journal of Mechanical Design*, 112:331–336.

Gosselin, C. and Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6(3):281–290.

Guan, Y., Yokoi, K., and Zhang, X. (2008). Numerical methods for reachable space generation of humanoid robots. *The International Journal of Robotics Research*, 27(8):935–950.

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*, 118:228–234.

Jiang, Q. and Gosselin, C. (2009). Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform. *Mechanism and Machine Theory*, 44(6):1281–1293.

Jo, D.-Y. (1988). *Numerical Analysis of Workspaces of Multibody Mechanical Systems*. PhD thesis, Ph. D. Thesis, The University of Iowa, Iowa City, Iowa 52242.

Jo, D. Y. and Haug, E. J. (1989). Workspace analysis of closed-loop mechanisms with unilateral constraints. *Advances in Design Automation ASME DE*, 19(3):53–60.

Krantz, S. G. and Parks, H. R. (2002). *The Implicit Function Theorem: History, Theory and Applications*. Birkhäuser, Boston.

Kyatkin, A. and Chirikjian, G. (1999). Computation of robot configuration and workspaces via the fourier transform on the discrete-motion group. *The International Journal of Robotics Research*, 18(6):601–615.

Luh, C. M., Adkins, F. A., Haug, E. J., and Qiu, C. C. (1996). Working capability analysis of Stewart platforms. *Journal of Mechanical Design*, 118:220–227.

Madden, C., Bohnenkamp, P., Kazerounian, K., and Ilieş, H. (2009). Residue level three-dimensional workspace maps for conformational trajectory planning of proteins. *The International Journal of Robotics Research*, 28(4):450–463.

Merlet, J.-P. (1995). Determination of the orientation workspace of parallel manipulators. *Journal of Intelligent and Robotic Systems*, 13(2):143–160.

Merlet, J.-P. et al. (1999). Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries. *The International Journal of Robotics Research*, 18(9):902–916.

Merlet, J.-P. and Gosselin, C. (2008). Parallel mechanisms and robots. In *Handbook of Robotics*, pages 269–285. Springer-Verlag.

Merlet, J. P., Gosselin, C. M., and Mouly, N. (1998). Workspaces of planar parallel manipulators. *Mechanism and Machine Theory*, 33(1-2):7–20.

Moore, R. E., Kearfott, R. B., and Cloud, M. J. (2009). *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 1st edition.

Nielsen, J. and Roth, B. (1999). Solving the input/output problem for planar mechanisms. *ASME Journal of Mechanical Design*, 121(2):206–211.

Oblak, D. and Kohli, D. (1988). Boundary surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators. *Journal of mechanisms, transmissions, and automation in design*, 110(4):389–396.

Ottaviano, E., Husty, M., and Ceccarelli, M. (2006). Identification of the workspace boundary of a general 3-R manipulator. *Journal of Mechanical Design*, 128(1):236–242.

Park, F. C. and Kim, J. W. (1999). Singularity analysis of closed kinematic chains. *ASME Journal of Mechanical Design*, 121:32–38.

Pernkopf, F. and Husty, M. (2006). Workspace analysis of stewart-gough-type parallel manipulators. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 220(7):1019–1032.

Porta, J. M., Ros, L., Creemers, T., and Thomas, F. (2007). Box approximations of planar linkage configuration spaces. *ASME Journal of Mechanical Design*, 129(4):397–405.

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):225–239. See also `http://www-iri.upc.es/groups/gmr/cuikweb`.

Rheinboldt, W. (1986). *Numerical Analysis of Parametrized Nonlinear Equations*. John Wiley and Sons, New York.

Rosales, C., Ros, L., Porta, J., and Suárez, R. (2010). Synthesizing grasp configurations with specified contact regions. *The International Journal of Robotics Research*. In press.

Snyman, J. A., du Plessis, L. J., and Duffy, J. (2000). An optimization approach to the determination of the boundaries of manipulator workspaces. *ASME Journal of Mechanical Design*, 122(4):447–456.

Wampler, C. W. (2001). Solving the kinematics of planar mechanisms by Dixon's determinant and a complex plane formulation. *ASME Journal of Mechanical Design*, 123(3):382–387.

Zein, M., Wenger, P., and Chablat, D. (2006). An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications. *Mechanism and Machine Theory*, 41(8):971–986.

Zlatanov, D. (1998). *Generalized Singularity Analysis of Mechanisms*. PhD thesis, University of Toronto.