A Complete Method for Workspace Boundary Determination

O. Bohigas, L. Ros, and M. Manubens

Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain.
E-mails: {obohigas,llros,mmanuben}@iri.upc.edu.

Abstract. This paper introduces a new method for workspace boundary determination on general lower-pair multi-body systems. The method uses a branch-and-prune technique to isolate the set of end effector singularities, and then classifies the points in such set according to whether they correspond to actual motion impediments in the workspace. The method can deal with open- or closed-chain systems, and is able to take joint limits into account. Advantages over other methods of similar applicability include its completeness and a simpler algorithmic structure. Examples are included that show its performance on benchmark problems documented in the literature.

Key words: Workspace determination, linear relaxation, closed-chain, multi-body system.

1 Introduction

A main problem of multi-body kinematics is *workspace determination*: for a multi-body system of known geometry, determine the complete set of poses (positions and orientations) that a selected body can adopt, as the system runs through all possible configurations. This special body is also known as the *end effector*, and the set of its poses as the *workspace* of the system. The issue has received substantial attention, as the availability of a good solution finds many applications to mechanism design, path planning, and task execution [1, 2].

Efficient solutions to this problem 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 in [3], which computes the positional workspace of a spatial parallel manipulator, the work in [4], which extends the approach to deal with other physical constraints, and that in [5], which provides methods for various planar parallel manipulators. Other significant approaches include interval analysis techniques for Gough-type manipulators [6], optimization-based algorithms for fully serial/parallel robots [2], analytic methods for symmetrical spherical mechanisms [7], and topologic or algebraic-geometric studies for 3R manipulators [8, 9]. The literature on the

topic is extensive, and we can only touch upon it briefly here. Elaborate surveys can be found in [1, 2].

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 argue that a solution for lower-pair mechanisms of general structure is possible, by extending a recent method for kinematic constraint solving for systems of such generality [10, 11].

Up to the authors' knowledge, only one approach of similar applicability is available in the literature, due to Haug and co-workers [12, and refs. therein]. Similarly to [12], we provide a visualization of the workspace by extracting its boundary from a set of end-effector singularities, but the strategy adopted here to formulate and compute such singularities is substantially different. In [12] the authors slice the singularity set into parallel curves, and a continuation scheme is then employed 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 the boundary, but no satisfactory method has been given to compute such points in general, as far as we know. In fact, the authors mention in [13] several situations that could make the technique miss some boundary segments. On the contrary, the method proposed in this paper is *complete*, in the sense that it is able to isolate all boundary segments of the workspace, including any interior barriers and voids that might be 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 [11].

2 Necessary conditions

The allowable positions and orientations of all links in a multi-body system are usually encoded in an n_q-dimensional vector of generalized coordinates $\mathbf{q}$, subject to a system of $n_e \leq n_q$ equations of the form

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

that expresses the kinematic constraints imposed by the joints. Even if joint limits are present, these can be modelled as equality constraints, as shown in Section 5. Here $\Phi : \mathcal{Q} \to \mathcal{E}$ is a smooth map, and $\mathcal{Q}$ and $\mathcal{E}$ are n_q- and n_e-dimensional manifolds, respectively. In the specific formulation that we will adopt, $\mathcal{Q} = \mathbb{R}^{n_q}$ and $\mathcal{E} = \mathbb{R}^{n_e}$, but they can be arbitrary manifolds in general. We focus on multi-body systems for which the solution set $\mathcal{C}$ of Eq. (1) is a smooth manifold of dimension $n_q - n_e$. This will be the case almost always, because the set of geometric parameters for which $\mathcal{C}$ fails to be a manifold has measure zero in the total space of such parameters. It is useful to consider the partition $\mathbf{q} = [\mathbf{v}^\mathsf{T}, \mathbf{w}^\mathsf{T}, \mathbf{u}^\mathsf{T}]^\mathsf{T}$, where $\mathbf{v} \in \mathcal{V}$ is a vector of n_v input variables corresponding to the actuated degrees of freedom of the multi-body system, $\mathbf{u} \in \mathcal{U}$ is a vector of n_u output variables encoding the pose of the end effector,

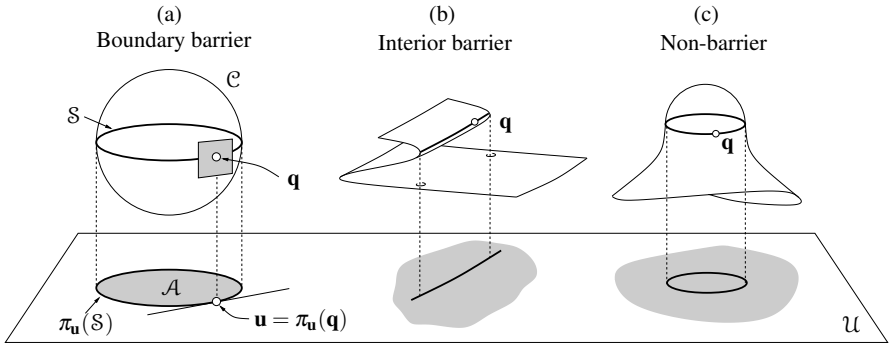

Fig. 1 (a) Sets $\mathcal{S}$ and $\boldsymbol{\pi}_{\mathbf{u}}(\mathcal{S})$ when $\mathcal{C}$ is the sphere $x^2+y^2+z^2=1$ and $\boldsymbol{\pi}_{\mathbf{u}}$ is the projection map $f(x,y,z)=(x,y)$. The "workspace" relative to the (x,y) variables is the projection of the sphere onto the (x,y) plane,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

and $\mathbf{w} \in \mathcal{W}$ is a n_w-dimensional vector encompassing the remaining intermediate variables. By defining $\mathbf{z} = [\mathbf{v}, \mathbf{w}]$, Eq. (1) can be written as $\boldsymbol{\Phi}(\mathbf{z}, \mathbf{u}) = 0$, and the workspace of the system can be defined as the set $\mathcal{A}$ of points $\mathbf{u} \in \mathcal{U}$ for which $\boldsymbol{\Phi}(\mathbf{z}, \mathbf{u}) = 0$ for some $\mathbf{z}$. In general it is easier to obtain a description of the workspace by computing its boundary, because such boundary is an object of lower dimension. A point $\mathbf{u} \in \mathcal{A}$ lies on the boundary of $\mathcal{A}$, denoted $\partial\mathcal{A}$, if every neighborhood of $\mathbf{u}$ intersects $\mathcal{A}$ and the complement of $\mathcal{A}$.

Let $\boldsymbol{\pi}_{\mathbf{u}} : \mathcal{Q} \to \mathcal{U}$ denote the projection map from $\mathcal{Q}$ onto the $\mathbf{u}$ variables. That is, $\boldsymbol{\pi}_{\mathbf{u}}(\mathbf{z}, \mathbf{u}) = \mathbf{u}$. Observe that $\mathcal{A}$ is exactly the image of $\mathcal{C}$ through $\boldsymbol{\pi}_{\mathbf{u}}$. It is not difficult to see, moreover, that the points $\mathbf{q} \in \mathcal{C}$ that project onto some $\mathbf{u} \in \partial\mathcal{A}$ must necessarily be critical points of the projection of $\mathcal{C}$ onto $\mathcal{U}$, i.e., points $\mathbf{q} \in \mathcal{C}$ where the tangent space to $\mathcal{C}$ projects on $\mathcal{U}$ as a linear space of dimension lower than n_u. The set $\mathcal{S}$ of all critical points of the projection of $\mathcal{C}$ on $\mathcal{U}$ will be called the *singularity set* hereafter, and the notation $\boldsymbol{\pi}_{\mathbf{u}}(\mathcal{S})$ will be used to refer to the projection of $\mathcal{S}$ onto $\mathcal{U}$. The situation is illustrated in Fig. 1a with an example.

Kinematically, $\mathcal{S}$ is the set of configurations in which the end effector loses instantaneous mobility [14, 15], which is the set of points $\mathbf{q} \in \mathcal{C}$ for which the matrix $d\boldsymbol{\Phi}_{\mathbf{z}} = [\partial\boldsymbol{\Phi}_i/\partial z_j]$ is rank deficient. This allows a simple algebraic characterization of the points of $\mathcal{S}$. They are the points $\mathbf{q}$ that satisfy

$$\left.\begin{array}{c} \boldsymbol{\Phi}(\mathbf{z}, \mathbf{u}) = 0 \\ d\boldsymbol{\Phi}_{\mathbf{z}}{}^{\mathsf{T}}\xi = 0 \\ \xi^{\mathsf{T}}\xi = 1 \end{array}\right\}, \tag{2}$$

for some ξ, where ξ is an n_e-dimensional vector of unknowns. The first equation in (2) constrains the solutions to points on $\mathcal{C}$. The second and third equations impose the rank deficiency of $d\boldsymbol{\Phi}_{\mathbf{z}}$ (the rows of this matrix are dependent whenever they yield a vanishing linear combination with non-null coefficients). A preliminary idea

of how the workspace boundary would look like, thus, can be gained by computing all points $\mathbf{q}$ that satisfy the previous system, and projecting them to the $\mathbf{u}$ variables, in order to obtain $\pi_{\mathbf{u}}(\mathcal{S})$.

3 Singularity classification

Note that the criticality of $\mathbf{q}$ is a necessary but not sufficient condition for $\pi_{\mathbf{u}}(\mathbf{q})$ to lie in $\partial\mathcal{A}$, as there can be critical points projecting on the interior of $\mathcal{A}$ too. In fact, as illustrated in Fig. 1, points $\mathbf{q}$ satisfying Eqs. (2) can be classified into two broad categories. They can be *non-barrier* or *barrier* singularities, depending on whether there exists a trajectory in the neighborhood of $\mathbf{q}$ on $\mathcal{C}$, passing through $\mathbf{q}$, whose projection on $\mathcal{U}$ traverses $\pi_{\mathbf{u}}(\mathcal{S})$ or not, respectively. Points corresponding to barrier singularities, in turn, can be classified as *boundary* or *interior* singularities, according to whether they occur over $\partial\mathcal{A}$ or over the interior of $\mathcal{A}$, respectively. An example of each one of these singularity types is depicted in Fig. 2, for the particular case of a planar 3R manipulator. We next provide additional criteria to determine which of these singularity types occurs on a given $\mathbf{q}_0 \in \mathcal{S}$.

Let $\mathbf{q} = \mathbf{q}(\mathbf{v})$ be a parameterization of $\mathcal{C}$ in a neighborhood of $\mathbf{q}_0$, with $\mathbf{q}_0 = \mathbf{q}(\mathbf{v}_0)$. Let $\mathbf{n}_0$ be the normal to $\pi_{\mathbf{u}}(\mathcal{S})$ at $\mathbf{u}_0$, which can be computed as indicated in [12]. We can determine whether $\mathbf{q}_0$ corresponds to a boundary barrier by examining the sign of

$$\psi(\mathbf{v}) = {\mathbf{n}_0}^{\mathsf{T}}(\mathbf{u}(\mathbf{v}) - \mathbf{u}_0), \tag{3}$$

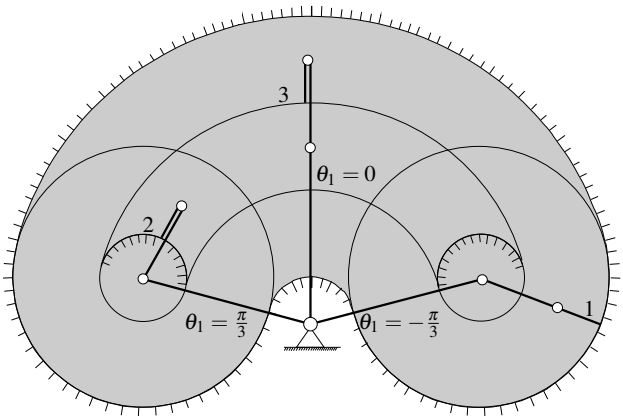

Fig. 2 Position workspace of a planar 3R manipulator, relative to the tip point of the last link, assuming that the angle θ_1 of the first revolute joint is restricted to the $[-\pi/3, \pi/3]$ range. Points corresponding to singularities are indicated in solid lines, and those relative to boundary and interior barriers are indicated with normal vectors on the forbidden side. Configurations 1, 2, and 3, are an example of a boundary barrier, an interior barrier, and a non-barrier singularity, respectively.

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

$$\psi(\mathbf{v}) \simeq \psi(\mathbf{v_0}) + \delta\mathbf{v}^{\mathsf{T}}\psi_{\mathbf{v}}(\mathbf{v_0}) + \frac{1}{2}\delta\mathbf{v}^{\mathsf{T}}\psi_{\mathbf{vv}}(\mathbf{v_0})\delta\mathbf{v}, \tag{4}$$

where $\psi_{\mathbf{v}}$ and $\psi_{\mathbf{vv}}$ are the gradient and Hessian of $\psi(\mathbf{v})$, and $\delta\mathbf{v} = (\mathbf{v} - \mathbf{v}_0)$ is a small displacement whose corresponding $\delta u = (\mathbf{u} - \mathbf{u}_0)$ is orthogonal to $\pi_{\mathbf{u}}(\mathcal{S})$. Note here that the first term of this expansion vanishes because $\psi(\mathbf{v}_0) = \mathbf{n}_0{}^{\mathsf{T}}(\mathbf{u}_0 - \mathbf{u}_0) = 0$. Moreover, the time derivative of Eq. (3) for $\mathbf{v} = \mathbf{v}(t)$ is

$$\dot{\psi}(t) = \mathbf{n}_0{}^{\mathsf{T}}\dot{\mathbf{u}}(t), \tag{5}$$

which is the component of $\dot{\mathbf{u}}(t)$ along $\mathbf{n}_0$. As shown in [12], $\dot{\psi}(t_0)$ must vanish irrespectively of the chosen $\mathbf{v}(t)$. Thus, since for $t = t_0$ it is $\dot{\psi} = \psi_{\mathbf{v}}\dot{\mathbf{v}} = 0$ for all $\dot{\mathbf{v}}$, we conclude that $\psi_{\mathbf{v}}(\mathbf{v_0}) = 0$ too, meaning that the second term of the Taylor expansion also vanishes.

In conclusion, the sign of $\psi(\mathbf{v})$ is mostly determined by the definiteness properties of the quadratic form $\delta\mathbf{v}^{\mathsf{T}}\psi_{\mathbf{vv}}(\mathbf{v_0})\delta\mathbf{v}$. If this form is positive- or negative-definite, then all trajectories orthogonal to $\pi_{\mathbf{u}}(\mathcal{S})$ lie on one side of $\pi_{\mathbf{u}}(\mathcal{S})$ and $\mathbf{q}_0$ is a barrier singularity. If this form is indefinite, there are trajectories in $\mathcal{A}$ that cross $\pi_{\mathbf{u}}(\mathcal{S})$ and $\mathbf{q}_0$ is a non-barrier singularity. Lastly, if this form is semi-definite, we cannot deduce the singularity type unless we examine higher order terms of the Taylor expansion. The latter case may only occur on zero-measure subsets of $\mathcal{S}$, however. The definiteness test just outlined can easily be implemented by checking the eigenvalues of the matrix form of $\delta\mathbf{v}^{\mathsf{T}}\psi_{\mathbf{vv}}(\mathbf{v_0})\delta\mathbf{v}$ [12].

When $\mathbf{q}_0$ is classified as a barrier singularity, finally, it remains to determine whether $\mathbf{u}_0$ lies on $\partial\mathcal{A}$ or in the interior of $\mathcal{A}$. Note that a barrier singularity $\mathbf{q}_0$ will project in the interior of $\mathcal{A}$ if there is some $\mathbf{q} \notin \mathcal{S}$ that projects onto $\mathbf{u}_0$ in $\mathcal{U}$. This test can be implemented by checking whether equation $\Phi(\mathbf{z}, \mathbf{u}) = 0$ for $\mathbf{u}$ fixed to $\mathbf{u}_0$ has a solution $\mathbf{z}$ for which $d\Phi_{\mathbf{z}}$ is full rank.

4 Numerical solution

We next show how Eq. (2) can be solved to determine $\mathcal{S}$, and how the classification scheme just given can be applied to selected points on $\mathcal{S}$, to obtain a detailed picture of the workspace.

As shown in [10, 11] it is always possible to write the first equation in Eq. (2) in a canonical form in which all component functions of $\Phi(\mathbf{z}, \mathbf{u})$ are quadratic polynomials. By quadratic we mean here that if q_i and q_j refer to any two of their variables, they involve monomials of linear, bilinear, or quadratic form only: q_i, $q_i q_j$, or q_i^2.

Note that if $\Phi(\mathbf{z}, \mathbf{u})$ has this special quadratic form, then all equations in System (2) will also have this form, because the entries in $d\Phi_{\mathbf{z}}$ will all be linear in their

variables. It is thus possible to use the numerical method developed in [10, 11] for systems of this kind, in order to isolate the solution set of Eqs. (2) to the desired accuracy. To make the paper self-contained, this method is reviewed briefly next.

Assuming that Eqs. (2) have the aforementioned form, the method starts by introducing the changes of variables $b_k = q_i q_j$ and $p_i = q_i^2$ for all bilinear and quadratic monomials, in order to transform the system into the expanded form

$$\left.\begin{array}{l} \mathbf{L}(\mathbf{x}) = 0 \\ \mathbf{B}(\mathbf{x}) = 0 \\ \mathbf{Q}(\mathbf{x}) = 0 \end{array}\right\}, \tag{6}$$

where $\mathbf{x}$ is a n_x-vector including the original $\mathbf{q}$ variables, and the newly-introduced p_i and b_k ones. Here, $\mathbf{L}(\mathbf{x}) = 0$ is a system of linear equations in $\mathbf{x}$, and $\mathbf{B}(\mathbf{x}) = 0$ and $\mathbf{Q}(\mathbf{x}) = 0$ are systems of bilinear and quadratic equations of the form $b_k - q_i q_j = 0$, $p_i - q_i^2 = 0$, respectively.

It is not difficult to see that, under the used formulation, each variable in $\mathbf{x}$ can only take values within a prescribed interval [11], 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 Eqs. (2). The algorithm then isolates such solutions by recursively applying 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 σ.

The crucial operation in this scheme is box shrinking, which [11] implements as follows. Note first that the solutions falling in some sub-box $\mathcal{B}_c \subseteq \mathcal{B}$ must lie in the linear variety defined by $L(\mathbf{x}) = 0$. Thus, we may shrink $\mathcal{B}_c$ to the smallest possible box bounding this variety inside $\mathcal{B}_c$. The limits of the shrunk box along, say, dimension x_i can be found by solving the two linear programs

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

However, note that the solutions must also lie on the parabolas $p_i = q_i^2$ of $\mathbf{Q}(\mathbf{x}) = 0$, and on the hyperbolic paraboloids $b_k = q_i q_j$ of $\mathbf{B}(\mathbf{x}) = 0$. The two facts can be taken into account by noting that the portion of the parabola $p_i = q_i^2$ lying inside $\mathcal{B}_c$ is bounded by two half planes, and that the points of $\mathcal{B}_c$ verifying $b_k = q_i q_j$ necessarily lie inside a tetrahedron defined by four points, obtained by clipping $\mathcal{B}_c$ with $b_k = q_i q_j$. Thus, the inequalities relative to such bounds can be added to **LP1** and **LP2** above, in order to take these constraints into account, 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.

Upon termination, this algorithm will deliver a collection of n_s boxes containing all points in $\mathcal{S}$, forming a discrete envelope of this set whose accuracy can be adjusted through the σ parameter (next section illustrates such kind of output on a particular example). Finally, a point $\mathbf{q}_0 \in \mathcal{S}$ is computed for each one of the returned boxes, by solving Eq. (2) using a Newton method starting anywhere in the box, to be able to apply the classification method described in Section 3. Note that σ can always be chosen small enough so as to allow a rapid computation of $\mathbf{q}_0$.

5 A comparative example

The proposed technique has been implemented in C, using the libraries of the CUIK platform [11]. We next illustrate the performance of this implementation, on computing the position workspace of the mechanism in Fig. 3, i.e., the set of locations for point P, as the mechanism runs through all of its configurations. This mechanism is particularly useful to compare the results of the proposed approach with those of the continuation technique in [12], published in [16]. It was used in [2] too, as a means of verification.

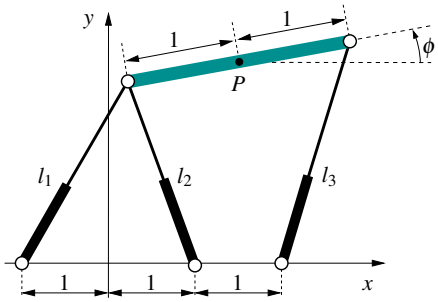

Fig. 3 A 3-RPR planar parallel mechanism.

The method in [12] starts shooting a ray through an initial point $\mathbf{u}_i \in \mathcal{A}$, corresponding to an assembled configuration of the mechanism, and tracks this ray using continuation, until a point $\mathbf{u}_b \in \partial\mathcal{A}$ is found (Fig. 4a). A second continuation process is then launched to track the connected component of $\pi_{\mathbf{u}}(\mathcal{S})$ that is reachable from $\mathbf{u}_b$ (Fig. 4b), whose points are finally classified into barrier and non-barrier singularities using the criteria of Section 3 (Fig. 4c). Note that this scheme is only able to detect the boundaries of the connected component of the workspace to which $\mathbf{u}_i$ belongs, while the algorithm we propose detects all components, as shown in Fig. 4, bottom row. To apply the presented approach, we first write Eq. (1) in the canonical form required in Section 4. For this, being $[x,y]^{\mathsf{T}}$ the coordinates of P, the slider variables l_i can be written as

$$\begin{aligned} l_1^2 &= y^2 - 2ys + s^2 + x^2 + 2x - 2xc - 2c + c^2 + 1, \\ l_2^2 &= y^2 - 2ys + s^2 + x^2 - 2x - 2xc + 2c + c^2 + 1, \\ l_3^2 &= y^2 + 2ys + s^2 + x^2 - 4x + 2xc - 4c + c^2 + 4, \end{aligned} \tag{7}$$

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

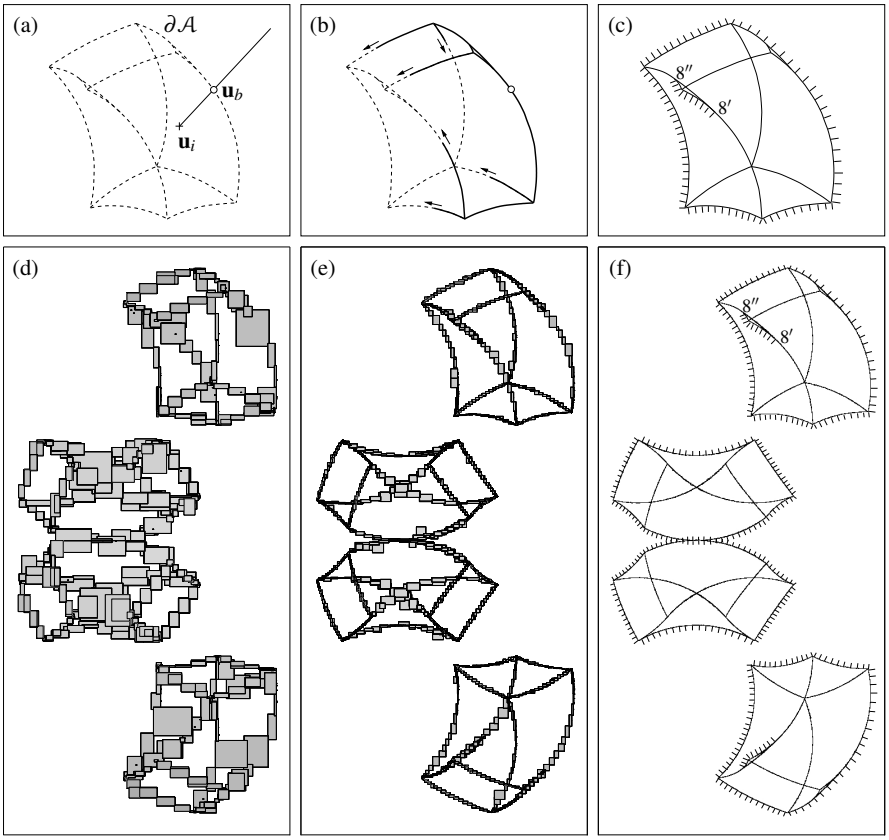

Fig. 4 Progress of the continuation algorithm in [12] (top row) compared to that of the proposed algorithm (bottom row) on computing the position workspace of the mechanism in Fig. 3. Note that the continuation algorithm is only able to isolate one connected component of the workspace, whereas the proposed one isolates them all. Figs. c and f follow the same convention as Fig. 2.

$$c^2 + s^2 = 1. \tag{8}$$

In this particular case, the slider variables l_i are only allowed to take values within prescribed ranges $[a_i, b_i]$, where $a_1 = a_2 = \sqrt{2}$, $a_3 = 1$, $b_1 = b_2 = 2$, and $b_3 = 3$. By defining $m_i = \frac{b_i + a_i}{2}$ and $h_i = \frac{b_i - a_i}{2}$, these constraints can be formulated using the slack variable technique of Optimization as

$$\begin{aligned} l_i &= m_i + h_i s_i, \\ c_i^2 + s_i^2 &= 1, \end{aligned} \tag{9}$$

for $i = 1, 2, 3$, which allows to integrate them readily into Eq. (7) as equalities. Thus, Eq. (1) is the system formed by equations (7), (8) and (9) in this case.

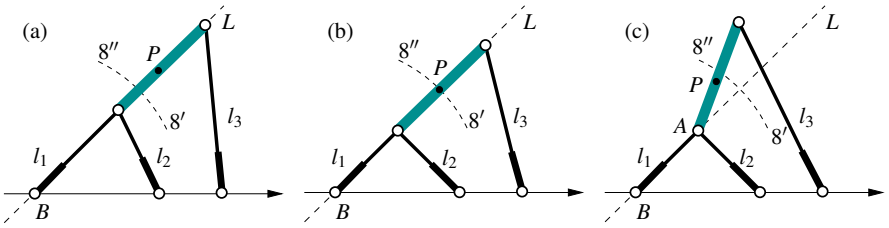

Fig. 5 A trajectory in which point P crosses the segment 8'-8".

The boundaries of the position workspace correspond to critical points of the projection of $\mathcal{C}$ onto the xy plane, i.e., to the solutions of Eq. (2) with $\mathbf{u} = [x,y]^{\mathsf{T}}$, and $\mathbf{z} = [c_1,c_2,c_3,c,s,s_1,s_2,s_3]^{\mathsf{T}}$. Eq. (2), thus, constitutes a polynomial system of 19 equations in 20 variables in this example, with a one-dimensional solution set. The progress of the proposed algorithm on isolating this set is shown in Fig. 4, bottom row. Figs. 4d and 4e show intermediate approximations of $\pi_{\mathbf{u}}(\mathcal{S})$ after 27 and 49 seconds, containing 1282 and 4846 boxes, respectively. Fig. 4f displays the final result, which contains 152082 boxes (the boxes are too small to be appreciated). The overall computation was done using $\sigma = 0.01$, and it took 780 seconds on a parallelized version of the CUIK platform, on a grid of eight DELL Poweredge computers equipped with two Intel Quadcore Xeon E5310 processors and 4Gb of RAM each one. Fig. 4f also shows the results of the classification process given in Section 3 applied to one point for each one of the returned boxes. It is worth mentioning that the segment $8'$-$8''$ was erroneously marked as an interior barrier in [16], while we detect it as a singularity of type non-barrier. This segment corresponds to point P tracing a circle around point B, when l_1 is fixed to its lowest value $\sqrt{2}$, while keeping the platform aligned with leg 1. The result in [16] must be erroneous, because P can really cross this segment from any of its two sides, as shown in Fig. 5. The platform can start from a position where P is to the right of the segment (Fig. 5a), then slide down along line L until it hits the segment (Fig. 5b), and, locking l_1 and l_2 to their values in this configuration, finally perform a rotation about point A by actuating l_3 (Fig. 5c).

6 Conclusions

This paper has introduced a new approach to compute workspace boundaries of general multi-body systems. A principal advantage of the method is its ability to converge to *all* boundary points, as discussed in the paper. Previous methods for the same purpose cannot ensure this property, since they are based on continuation, which requires the availability of one point for each connected component of the sought boundary, and no previous work on workspace analysis has shown how to compute all of such points in general, to the authors' knowledge.

The computation of an exhaustive representation of the workspace boundary becomes feasible when the workspace of the end effector is of dimension three or lower. However, for workspaces of larger dimension it turns out a difficult task, independently of the methodology used, as the curse of dimensionality must inevitably be faced. In order to visualize such workspaces, several authors introduce lower-dimensional representations of the workspace which are easier to compute and meaningful to the robot designer, like the *reachable* workspace, the *constant orientation* workspace, or the *constant position* workspace [6]. It is worth noting that all of these workspaces can be computed by the technique herein proposed, using a proper choice of the **u** variables, and fixing them to given values, if necessary.

Acknowledgements The authors thank Josep M. Porta for fruitful discussions on the topic of the paper. This work was funded by the Spanish Ministry of Science, under contract DPI2007-60858.

References

1. Merlet, J.-P., Gosselin, C.: Parallel mechanisms and robots. In: Handbook of Robotics. Springer-Verlag, 2008, pp. 269–285.
2. Snyman, J. A., du Plessis, L. J., Duffy, J.: An optimization approach to the determination of the boundaries of manipulator workspaces. ASME J. of Mech. Des. **122**(4), 447–456 (2000)
3. Gosselin, C.: Determination of the workspace of 6-DOF parallel manipulators. ASME J. of Mech. Des. **112**, 331-336 (1990)
4. Merlet, J. P.: Determination of the orientation workspace of parallel manipulators. J. of Intelligent and Robotic Systems. **13**(2), 143–160 (1995)
5. Merlet, J. P., Gosselin, C. M. , Mouly, N.: Workspaces of planar parallel manipulators. Mech. Mach. Theory. **33**(1-2), 7–20 (1998)
6. Merlet, J., et al.: Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries. Int. J. of Robotics Research. **18**(9), 902–916 (1999)
7. Bonev, I. A., Gosselin, C. M.: Analytical determination of the workspace of symmetrical spherical parallel mechanisms. IEEE Trans. on Robotics. **22**(5), 1011–1017 (2006)
8. Zein, M., Wenger, P., Chablat, D.: An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications. Mech. Mach. Theory. **41**(8), 971–986 (2006)
9. Ottaviano, E., Husty, M., Ceccarelli, M.: Identification of the workspace boundary of a general 3-R manipulator. ASME J. of Mech. Des. **128**(1), 236–242 (2006)
10. Porta, J. M., Ros, L., Creemers, T., Thomas, F.: Box approximations of planar linkage configuration spaces. ASME J. of Mech. Des. **129**(4), 397–405 (2007)
11. Porta, J. M., Ros, L., Thomas, F.: A linear relaxation technique for the position analysis of multi-loop linkages. IEEE Trans. on Robotics. **25**(2), 225–239 (2009)
12. Haug, E. J., Luh, C.-M., Adkins, F. A., Wang, J.-Y.: Numerical algorithms for mapping boundaries of manipulator workspaces. ASME J. of Mech. Des. **118**, 228–234 (1996)
13. Abdel-Malek, K., Adkins, F., Yeh, H. J., Haug, E.: On the determination of boundaries to manipulator workspaces. Robotics and Comput. Integrated Manuf. **13**(1), 63–72 (1997)
14. Zlatanov, D.: Generalized singularity analysis of mechanisms. Ph.D. dissertation, University of Toronto (1998)
15. Park, F. C., Kim, J. W.: Singularity analysis of closed kinematic chains. ASME J. of Mech. Des. **121**, 32–38 (1999)
16. Luh, C. M., Adkins, F. A., Haug, E. J., Qiu, C. C.: Working capability analysis of Stewart platforms. ASME J. of Mech. Des. **118**(2), 220–227 (1996)