Planning Singularity-free Paths on Closed-Chain Manipulators

Oriol Bohigas, Michael E. Henderson, Lluís Ros, Montserrat Manubens, and Josep M. Porta

***Abstract*—This paper provides an algorithm for computing singularity-free paths on closed-chain manipulators. Given two non-singular configurations of the manipulator, the method attempts to connect them through a path that maintains a minimum clearance with respect to the singularity locus at all points, which guarantees the controllability of the manipulator everywhere along the path. The method can be applied to non-redundant manipulators of general architecture, and it is resolution-complete. It always returns a path whenever one exists at a given resolution, or determines path non-existence otherwise. The strategy relies on defining a smooth manifold that maintains a one-to-one correspondence with the singularity-free C-space of the manipulator, and on using a higher-dimensional continuation technique to explore this manifold systematically from one configuration, until the second configuration is found. If desired, the method can also be used to compute an exhaustive atlas of the whole singularity-free component reachable from a given configuration, which is useful to rapidly resolve subsequent planning queries within such component, or to visualize the singularity-free workspace of any of the manipulator coordinates. Examples are included that demonstrate the performance of the method on illustrative situations.**

***Index Terms*—Closed-chain motion planning, singularity avoidance, singularity-free path or workspace, higher-dimensional continuation, assembly-mode changing.**

I. Introduction

A Fundamental task in Robotics is the computation of feasible C-space paths between two configurations of a manipulator [1]. The subject has received substantial attention, and efficient algorithms exist that solve hard instances of the problem on open chains, even in cluttered environments and highly-dimensional situations [2, 3]. However, when the manipulator involves closed kinematic chains, —as it occurs in parallel robots, multi-arm manipulation systems, or reconfigurable mechanisms, for instance—, the problem is much harder. A number of kinematic loop-closure constraints relate configuration parameters by non-linear equations, giving rise to C-spaces of a complex topological structure. Such spaces may have several connected components and lower-dimensional singularity sets, and may not even admit a global parameterization [4], which complicates the extension of open-chain path planning methods to also deal with the closed-chain case.

A few strategies have been given to plan feasible C-space paths on general closed-chain manipulators [5]–[10], but none of them accounts for the so-called *singular* configurations, where the kinetostatic performance of the manipulator dramatically degrades. Many types of singularities can be distinguished [11], but those of primary interest in this paper are the *direct* or *forward* singularities, which compromise the velocity control of a manipulator, leading to malfunction or a breakage of the structure. Unless the use of involved control strategies would be affordable [12, 13], any path connecting two configurations should avoid crossing such critical configurations. In many contexts, moreover, a minimum clearance with respect to the singularity locus should be maintained, to avoid the negative effects of the increased shakiness in the vicinity of such locus. While it is true that many commercial manipulators are designed not to include singularities in their workspace, this is not always possible, and the availability of a singularity-free path planner would broaden the range of possibilities of the robot designer.

Several works confront the problem of on-line singularity avoidance [14], but only a few tackle the more general problem of planning paths between distant configurations. They include an algorithm based on deforming a parametrized path between the query configurations [15], a variational approach that reduces the problem to a boundary value problem [16], and a numerical technique based on treating the singularity locus as a collection of obstacles [17]. All of these methods work well in favorable situations, but [15] and [16] mention limitations relative to proving path existence in some cases, and the one in [17] is computationally intensive, as it requires constructing polytope approximations of the entire singularity set before searching for a feasible path. In some way or another, also, the methods in [15]–[17] exploit the fact that the considered C-spaces have closed-form parameterizations, so that it is difficult to extend them to tackle manipulators with a more complex architecture. The method provided in this paper, in contrast, makes no recourse to closed-form parameterizations, and can be applied to any non-redundant closed-chain manipulator, with the sole limitations imposed by the curse of dimensionality. As opposed to [17], it treats singularities implicitly, not explicitly as obstacles, resulting in a computationally less intensive approach.

The method was preliminary introduced in [18] and is now presented and demonstrated with thorough detail. It relies on defining a system of equations whose solution manifold

O. Bohigas, L. Ros, M. Manubens, and J. M. Porta are with the Kinematics and Robot Design Group at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain (e-mails: {obohigas,lros,mmanuben,porta}@iri.upc.edu).

M. E. Henderson is with the Numerical Analysis Group in the Mathematical Sciences Department at IBM's Thomas J. Watson Research Center, Yorktown Heights, NY USA (e-mail: mhender@us.ibm.com).

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under contract DPI2010-18449.

The paper has supplementary downloadable material which includes videos of the experiments in Figs. 11 and 12. Contact O. Bohigas for futher questions about this material.

corresponds to the singularity-free subset of the C-space, so that maneuvering through such manifold guarantees singularity avoidance at all times. Then, an extension of the higher-dimensional continuation strategy given in [19] is defined, which allows exploring this manifold systematically until a path joining the start and goal configurations is found, or path non-existence is determined by exhaustion of the search.

The rest of the paper is organized as follows. Section II prepares the ground for the paper and describes the conditions that characterize the forward singularities of a manipulator, explaining their significance from the physical and geometrical standpoints. Based on such background, then, Section III describes the proposed singularity-free path planning method. Section IV shows the performance of the method on illustrative situations. Section V finally provides the paper's conclusions, and highlights points deserving further attention.

II. Preliminaries

The allowable positions and orientations of all links in a manipulator can always be encoded in a vector $\boldsymbol{q}$ of n_q generalized coordinates, subject to a system of n_e equations

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

expressing the assembly constraints imposed by the joints [20]–[22]. Here, $\boldsymbol{\Phi}(\boldsymbol{q}) : \mathcal{Q} \rightarrow \mathcal{E}$ is a differentiable map, and $\mathcal{Q}$ and $\mathcal{E}$ are n_q- and n_e-dimensional manifolds, respectively. Note that no particular choice of generalized coordinates is assumed, because the method is equally applicable under any such choice, as long as $\boldsymbol{\Phi}(\boldsymbol{q})$ satisfies the previous condition.

Let $\mathcal{C}$ denote the C-space of the manipulator. That is,

$$\mathcal{C} = \{\boldsymbol{q} \in \mathcal{Q} : \boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0}\}.$$

In the usual setting, the differential $\boldsymbol{\Phi_q} = [\partial \Phi_i / \partial q_j]$ is full rank at all points $\boldsymbol{q} \in \mathcal{C}$, except on a subset $\mathcal{G}$ of points where $\mathcal{C}$ may lose the manifold structure. Thus, $\mathcal{C} \setminus \mathcal{G}$ is a smooth manifold of dimension $d = n_q - n_e$, with a well-defined tangent space. The points of $\mathcal{G}$ can arise even if $\boldsymbol{\Phi}(\boldsymbol{q})$ is differentiable and are called C-space singularities [11]. They typically correspond to bifurcations, ridges, or dimension changes of $\mathcal{C}$ (Fig. 1).

The vector $\boldsymbol{q}$ will be assumed to contain a subvector $\boldsymbol{v}$ of n_v coordinates, corresponding to the actuated degrees of freedom, or *inputs*, of the manipulator. This allows considering the partition $\boldsymbol{q} = [\boldsymbol{y}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}]^\mathsf{T}$, where $\boldsymbol{y}$ encompasses the n_y coordinates of $\boldsymbol{q}$ not present in $\boldsymbol{v}$, and to write (1) as

$$\boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{0}. \tag{2}$$

It will be further assumed that the manipulator is non-redundant, meaning that the number of inputs is the lowest required to fix a configuration. Thus, $n_v = d$, and $n_y = n_q - n_v = n_q - d = n_e$.

To see the role played by singular configurations, consider the time derivative of (2),

$$\boldsymbol{\Phi_y} \dot{\boldsymbol{y}} + \boldsymbol{\Phi_v} \dot{\boldsymbol{v}} = \boldsymbol{0}. \tag{3}$$

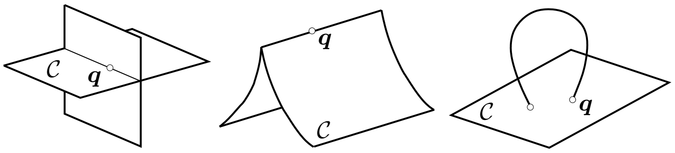

Fig. 1. Examples of C-space singularities.

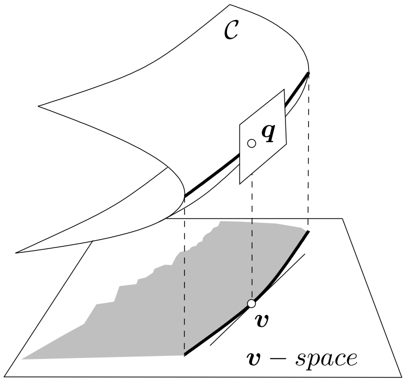

Fig. 2. Interpretation of points $\boldsymbol{q} \in \mathcal{C} \setminus \mathcal{G}$ where $det(\boldsymbol{\Phi_y}) = 0$.

Note that for configurations $\boldsymbol{q}$ for which $\boldsymbol{\Phi_y}$ is full rank we can write (3) in the equivalent form

$$\dot{\boldsymbol{y}} = -\boldsymbol{\Phi}_{\boldsymbol{y}}^{-1} \boldsymbol{\Phi_v} \dot{\boldsymbol{v}}, \tag{4}$$

which provides the time rates of the $\boldsymbol{y}$ coordinates in terms of the time rates of the input coordinates $\boldsymbol{v}$, i.e., the solution to the forward instantaneous kinematic problem. However, (4) only holds whenever the matrix $\boldsymbol{\Phi_y}$ is full rank, and only in this case the input rates $\dot{\boldsymbol{v}}$ will determine a single value for the vector $\dot{\boldsymbol{y}}$. This must be so because, if $\boldsymbol{\Phi_y}$ is rank-deficient in $\boldsymbol{q}$, then, for a given value of $\dot{\boldsymbol{v}}$, (3) yields either no solution or infinitely-many solutions for $\dot{\boldsymbol{y}}$, and it is not possible to uniquely determine the velocity of the manipulator by specifying the velocities of the actuators. On the other hand, if $\boldsymbol{\Phi_y}$ is full rank along a time-parametric path $\boldsymbol{q}(t) = [\boldsymbol{y}(t), \boldsymbol{v}(t)] \in \mathcal{C}$, the Implicit Function Theorem [23] guarantees a one-to-one correspondence between the path $\boldsymbol{v}(t)$ and the path $\boldsymbol{q}(t)$, so that the motion of the manipulator will be controllable through the inputs along the path, even under slight perturbations of $\boldsymbol{v}(t)$. Following these observations, a configuration $\boldsymbol{q} \in \mathcal{C}$ is said to be a *forward singularity* if $det(\boldsymbol{\Phi_y}) = 0$.

Geometrically, the matrix $\boldsymbol{\Phi_y}$ may be rank deficient either because $\boldsymbol{\Phi_q}$ itself is rank deficient, or because only $\boldsymbol{\Phi_y}$ is. In the first case, $\boldsymbol{q}$ is in $\mathcal{G}$, and in the second case, the tangent space to $\mathcal{C}$ at $\boldsymbol{q}$ projects onto the $\boldsymbol{v}$-space as a linear variety of dimension lower than n_v (Fig. 2). Nevertheless, both kinds of situations need to be avoided if a safe and accurate motion path for the manipulator is desired.

Inverse singularities that give rise to dexterity losses of the end-effector might also be of interest in some cases. Such singularities can be characterized in a way analogous to forward singularities, as certain configurations for which another Jacobian matrix is rank-deficient [24, 25].

III. The method

Let $\mathcal{C}_s$ denote the set of forward singularities of $\mathcal{C}$, and define $\mathcal{C}_{sfree} = \mathcal{C} \setminus \mathcal{C}_s$. We call these sets the *singularity locus* and the *singularity-free C-space* of the manipulator, respectively. A singularity-free path joining two configurations of $\mathcal{C}_{sfree}$, $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$, is a continuous map $\boldsymbol{\tau} : [0,1] \rightarrow \mathcal{C}_{sfree}$, such that $\boldsymbol{\tau}(0) = \boldsymbol{q}_s$ and $\boldsymbol{\tau}(1) = \boldsymbol{q}_g$. The planning problem we deal with, thus, boils down to computing such paths between two predefined configurations, $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$. The solution method proposed is based on defining a proper system of equations characterizing $\mathcal{C}_{sfree}$ (Section III-A), and then resorting to a continuation method that uses these equations to trace $\mathcal{C}_{sfree}$ from $\boldsymbol{q}_s$, until $\boldsymbol{q}_g$ is eventually found (Section III-B).

A. A system of equations defining $\mathcal{C}_{sfree}$

In our case, $\mathcal{C}_{sfree}$ is the set of points $\boldsymbol{q} = [\boldsymbol{y}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}]^\mathsf{T} \in \mathcal{C}$ for which $det(\boldsymbol{\Phi_y}) \neq 0$. However, to be able to apply a continuation method, it is necessary to turn the latter condition into equality form. To this end, we introduce an auxiliary variable b and note that $det(\boldsymbol{\Phi_y}) \neq 0$ if, and only if, $det(\boldsymbol{\Phi_y}) \cdot b = 1$ for some value of b. Thus, a system of equations characterizing $\mathcal{C}_{sfree}$ is given by

$$\left.\begin{array}{r} \boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0} \\ det(\boldsymbol{\Phi_y}) \cdot b = 1 \end{array}\right\} \tag{5}$$

If other singularities need to be avoided [24, 25], the determinant of the Jacobian matrix defining them can be added as an additional factor in the second equation of this system. For ease of explanation, however, this extension will not be considered in this paper.

For convenience, the system in (5) will be written as

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0} \tag{6}$$

hereafter, where

$$\boldsymbol{x} = [\boldsymbol{q}^\mathsf{T}, b]^\mathsf{T},$$

and

$$\boldsymbol{F}(\boldsymbol{x}) = \left[\begin{array}{c} \boldsymbol{\Phi}(\boldsymbol{q}) \\ det(\boldsymbol{\Phi_y}) \cdot b - 1 \end{array}\right].$$

Let $\mathcal{M}$ be the set of points $\boldsymbol{x}$ that satisfy (6), and define the function

$$b(\boldsymbol{q}) = \frac{1}{det(\boldsymbol{\Phi_y}(\boldsymbol{q}))}. \tag{7}$$

Note that the points $\boldsymbol{x} \in \mathcal{M}$ are in one-to-one correspondence with the points $\boldsymbol{q} \in \mathcal{C}_{sfree}$, because $\boldsymbol{q} \in \mathcal{C}_{sfree}$ if, and only if, $\boldsymbol{x} = [\boldsymbol{q}, b(\boldsymbol{q})]^\mathsf{T}$ satisfies (5). Accordingly, all paths in $\mathcal{C}_{sfree}$ are uniquely represented in $\mathcal{M}$, and *viceversa*. Thus, the original problem of computing a singularity-free path in $\mathcal{C}$ from $\boldsymbol{q}_s$ to $\boldsymbol{q}_g$ can be reduced to that of connecting

$$\boldsymbol{x}_s = [\boldsymbol{q}_s^\mathsf{T}, b(\boldsymbol{q}_s)]^\mathsf{T}$$

and

$$\boldsymbol{x}_g = [\boldsymbol{q}_g^\mathsf{T}, b(\boldsymbol{q}_g)]^\mathsf{T}$$

through some path in $\mathcal{M}$. This reduction is advantageous because, by letting the path planner operate in $\mathcal{M}$, instead of in $\mathcal{C}$ directly, guarantees that any computed path in $\mathcal{M}$ will have a corresponding path in $\mathcal{C}$ lying entirely in $\mathcal{C}_{sfree}$. This eliminates the need of checking singularity crossings in the planner, which may be rather difficult due to the intricate structure of the singularity locus [15].

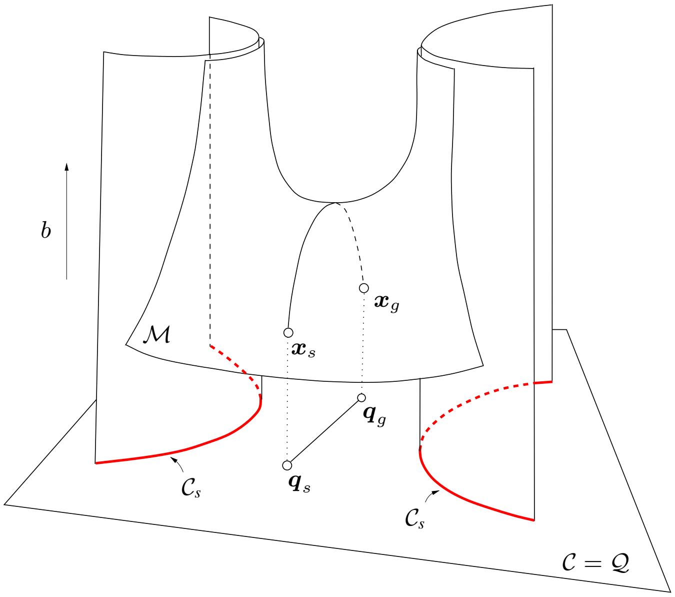

Fig. 3. The original problem of computing a singularity-free path in $\mathcal{C}$ connecting $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ is transformed into one of finding an arbitrary path in $\mathcal{M}$ connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$.

The correspondence of the two problems is schematically illustrated in Fig. 3. The horizontal plane in the bottom represents $\mathcal{C}$, which in this example coincides with the ambient space $\mathcal{Q}$ for simplicity, and the singularity locus $\mathcal{C}_s$ is represented by two red parabolas in this plane. To construct $\mathcal{M}$, we add a new dimension b to $\mathcal{Q}$ (the vertical axis in the figure), and we lift every point $\boldsymbol{q} \in \mathcal{C}$ to the point $\boldsymbol{x} = [\boldsymbol{q}^\mathsf{T}, b(\boldsymbol{q})]^\mathsf{T}$. Then, $\mathcal{M}$ can be thought of as a new manifold extending infinitely in the direction b, as the projection $\boldsymbol{q}$ of a point $\boldsymbol{x} \in \mathcal{M}$ approaches $\mathcal{C}_s$.

Two important observations are in order regarding the search for a path. On the one hand, note that the differential $\boldsymbol{F_x}$ has the block structure

$$\boldsymbol{F_x} = \left[\begin{array}{c:c} \boldsymbol{\Phi_q} & \boldsymbol{0} \\ \hdashline * & det(\boldsymbol{\Phi_y}) \end{array}\right],$$

from which we see that $\boldsymbol{F_x}$ is full rank at all points $\boldsymbol{x} \in \mathcal{M}$, because $\boldsymbol{\Phi_y}$ (and hence $\boldsymbol{\Phi_q}$) is full rank at such points. By the Implicit Function Theorem, this implies that $\mathcal{M}$ has the structure of a smooth manifold everywhere [23], which is beneficial from the point of view of applying a continuation method to explore $\mathcal{M}$ [26], because no bifurcations, ridges, or dimension changes are to be found during such exploration, and no recourse to branch-switching operations will be necessary [27]. On the other hand, observe that, in practice, all the $\boldsymbol{q}$ coordinates have known bounds [22], like those derived from mechanical limits on the joints which, if necessary, can be taken into account by adding extra equations to the system [25]. Moreover, $|b|$ should be maintained below a given threshold b_{max} to guarantee some clearance from $\mathcal{C}_s$,

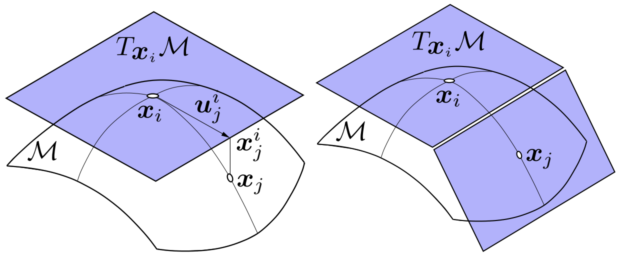

Fig. 4. The higher-dimensional continuation method applied to a two-dimensional manifold in $\mathbb{R}^3$. Left: A point $\boldsymbol{x}_j$ in $\mathcal{M}$ can be obtained by orthogonally projecting a point $\boldsymbol{x}^i_j$ on $T_{\boldsymbol{x}_i}\mathcal{M}$. Right: If a new chart is defined at $\boldsymbol{x}_j$, it must be properly coordinated with the chart at $\boldsymbol{x}_i$ so that their projections smoothly cover the manifold.

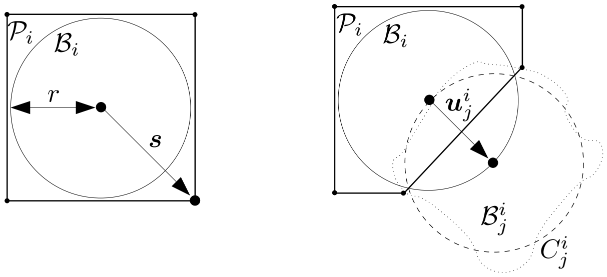

Fig. 5. Polytope-based chart construction. Left: The domain for chart C_i, $\mathcal{P}_i$, is a box including a ball of radius r around x_i, both defined in the tangent space associated with the chart. Right: $\mathcal{P}_i$ is refined using a ball $\mathcal{B}^i_j$ that approximates C^i_j, the projection on C_i of the part of the manifold covered by C_j.

determined by the task to be accomplished, and by the characteristics of the underlying mechanical and control systems. As a result, the search for a path in $\mathcal{M}$ must be restricted to a given domain $\mathcal{D}$ of the $\boldsymbol{x}$-space, usually defined as the Cartesian product of a number of intervals derived from the coordinate bounds.

B. Exploring $\mathcal{M}$ for a path

To determine a singularity-free path connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ we can gradually construct an atlas of $\mathcal{M} \cap \mathcal{D}$. An atlas is a collection of charts where each chart C_i defines a local map between a domain $\mathcal{P}_i \subset \mathbb{R}^d$ and an open set around a given point $\boldsymbol{x}_i \in \mathcal{M}$, initially $\boldsymbol{x}_s$. The atlas will be computed using the higher-dimensional continuation method proposed in [26]. This method defines a local map for chart C_i using the matrix $\boldsymbol{\Psi}_i$, whose columns define an orthonormal basis of $T_{\boldsymbol{x}_i}\mathcal{M}$, the d-dimensional tangent space of $\mathcal{M}$ at $\boldsymbol{x}_i$. The map is determined by first selecting a d-dimensional vector $\boldsymbol{u}^i_j$ of parameters (Fig. 4-left), which is used to generate a point $\boldsymbol{x}^i_j \in \mathbb{R}^{n_q+1}$ in the neighborhood of $\boldsymbol{x}_i$ as follows:

$$\boldsymbol{x}^i_j = \boldsymbol{x}_i + \boldsymbol{\Psi}_i\, \boldsymbol{u}^i_j. \tag{8}$$

Then, a point $\boldsymbol{x}_j$ in $\mathcal{M}$ is computed by orthogonally projecting $\boldsymbol{x}^i_j$. This projection is obtained by solving

$$\left.\begin{array}{r} \boldsymbol{F}(\boldsymbol{x}_j) = \boldsymbol{0} \\ \boldsymbol{\Psi}_i^{\mathsf{T}}\,(\boldsymbol{x}_j - \boldsymbol{x}^i_j) = \boldsymbol{0} \end{array}\right\}$$

using a Newton method initialized at $\boldsymbol{x}^i_j$. At each iteration of the method, $\boldsymbol{x}_j$ is updated with the increment $\Delta\boldsymbol{x}_j$ fulfilling

$$\left[\begin{array}{c} \boldsymbol{F}_{\boldsymbol{x}_i} \\ \boldsymbol{\Psi}_i^{\mathsf{T}} \end{array}\right] \Delta\boldsymbol{x}_j = -\left[\begin{array}{c} \boldsymbol{F}(\boldsymbol{x}_i) \\ \boldsymbol{\Psi}_i^{\mathsf{T}}\,(\boldsymbol{x}_j - \boldsymbol{x}^i_j) \end{array}\right]. \tag{9}$$

The update is applied until the norm of the right-hand side of (9) becomes negligible, or for a maximum number of iterations.

Each point in the manifold is the potential center of a new chart (see Fig. 4-right), and a method due to Henderson can be used to decide where to place the chart centers to ensure a good coverage of the manifold [26]. In his approach, the domain $\mathcal{P}_i$ of chart C_i is initialized as a d-dimensional hypercube enclosing a ball $\mathcal{B}_i$ of radius r, both defined in $T_{\boldsymbol{x}_i}\mathcal{M}$, as illustrated in Fig. 5-left. A vertex of $\mathcal{P}_i$ exterior to $\mathcal{B}_i$, with position vector $\boldsymbol{s}$, is then used to generate a point $\boldsymbol{x}^i_j$, using (8) with

$$\boldsymbol{u}^i_j = \alpha\,\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}, \tag{10}$$

where α is initialized to r. If the projection from $\boldsymbol{x}^i_j$ to $\mathcal{M}$ does not converge, or if the new chart C_j at $\boldsymbol{x}_j$ is too far or too different from C_i, i.e., if

$$\|\boldsymbol{x}_j - \boldsymbol{x}^i_j\| > \epsilon, \tag{11}$$

or

$$det(\boldsymbol{\Psi}_i^{\mathsf{T}}\boldsymbol{\Psi}_j) < 1 - \epsilon, \tag{12}$$

for a given threshold ϵ, the new chart is discarded and a new attempt of chart generation is performed with a smaller α. This procedure adapts the size of the area covered by each chart to the local curvature of the manifold. When C_j is valid, it is used to crop $\mathcal{P}_i$ from the intersection between $\mathcal{B}_i$ and C^i_j, the projection on $T_{\boldsymbol{x}_i}\mathcal{M}$ of the part of the manifold covered by C_j. This projection is approximated by a ball $\mathcal{B}^i_j$ of radius r in $T_{\boldsymbol{x}_i}\mathcal{M}$, centered at the point given by $\boldsymbol{u}^i_j$, as shown in Fig. 5-right. The intersection of $\mathcal{B}_i$ and $\mathcal{B}^i_j$ defines a new face of $\mathcal{P}_i$ that eliminates some of its vertices (in particular the one given by $\boldsymbol{s}$) and generates new ones. Symmetrically, the polytope $\mathcal{P}_j$ associated with C_j is cropped using C_i. When C_i is surrounded by other charts, $\mathcal{P}_i$ becomes a convex polytope included in $\mathcal{B}_i$, and C_i is considered to be *closed*, meaning that no further expansion of the atlas need to be attempted from that chart. Charts whose center is out of the domain $\mathcal{D}$ are also considered closed. When all charts are closed, the connected component of $\mathcal{M}$ containing the initial point $\boldsymbol{x}_s$ gets fully covered. If a path exists from $\boldsymbol{x}_s$ to $\boldsymbol{x}_g$, a chart centered at $\boldsymbol{x}_g$ must be connected to the atlas built from $\boldsymbol{x}_s$ and, thus, a solution path can be determined by searching the graph implicitly defined by the chart centers and their neighborhood relations. If the chart centered at $\boldsymbol{x}_g$ is not reached, path non-existence can be established at the considered value of r.

As an example, Fig. 6 illustrates the progression of the algorithm on tracing the surface of a torus from a given point. Each picture shows the part of the atlas computed after m steps, with open and closed charts colored in red and blue, respectively. Since a torus is a 2-dimensional manifold, the

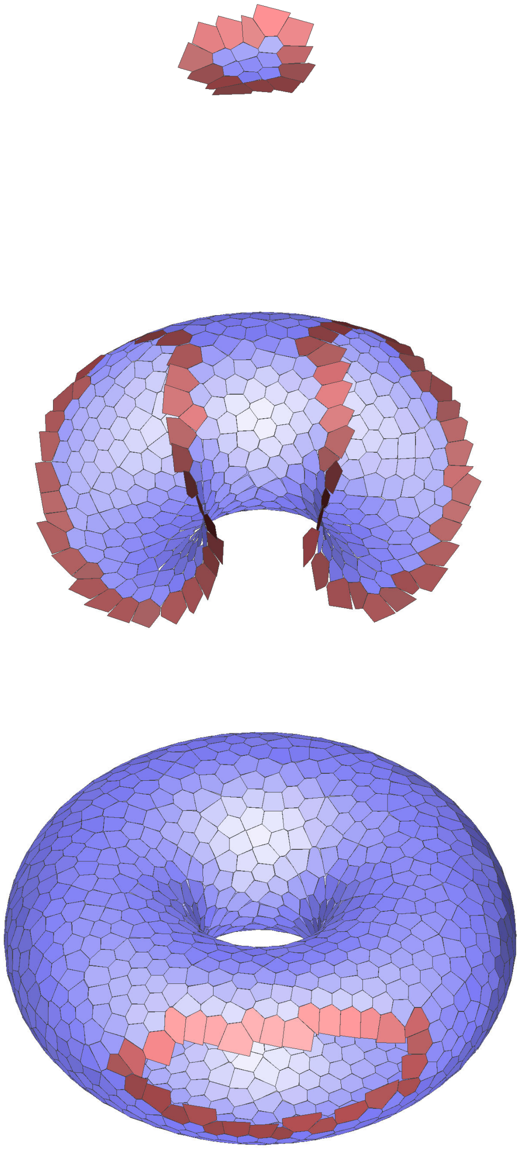

Fig. 6. Progression of the algorithm on the surface of the torus defined by $(\sqrt{x^2+y^2}-R_1)^2+z^2-R_2^2=0$, with $R_1=0.5$, $R_2=0.8$, and using the continuation parameters $r=0.1$ and $\epsilon=0.25$. Red and blue polygons correspond to open and closed charts, respectively.

Algorithm 1: Singularity-free path planner.

```
PathComputation($\boldsymbol{F}, \boldsymbol{x}_s, \boldsymbol{x}_g, \boldsymbol{C}, \boldsymbol{H}, r, \epsilon$)
input  : The functions $\boldsymbol{F}$ defining $\mathcal{M}$, the start and goal
         configurations, $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$, a function $\boldsymbol{C}$ giving the
         cost of the transition between configurations, a
         heuristic function $\boldsymbol{H}$ providing a lower bound of the
         cost of the transition between configurations, and the
         parameters $r$ and $\epsilon$ used to build the atlas.
output: An singularity-free path connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$.
1  $C_s \leftarrow$NewChart($\boldsymbol{F}, \boldsymbol{x}_s, r$)}
2  $C_g \leftarrow$NewChart($\boldsymbol{F}, \boldsymbol{x}_g, r$)}
3  $\mathcal{A} \leftarrow \{C_s, C_g\}$
4  $\mathcal{H} \leftarrow \{C_s\}$
5  $\mathcal{V} \leftarrow \emptyset$
6  $p(s) \leftarrow 0$
7  $c(s) \leftarrow 0$
8  $h(s) \leftarrow \boldsymbol{H}(\boldsymbol{x}_s, \boldsymbol{x}_g)$
9  $C_i \leftarrow C_s$
10 while $\mathcal{H} \neq \emptyset$ and $C_i \neq C_g$ do
11     $C_i \leftarrow$ExtractMin($\mathcal{H}, h$)
12     if $C_i \neq C_g$ and $c(i) < \infty$ then
13         while Open($C_i$) do
14             $\alpha \leftarrow r$
15             $\boldsymbol{s} \leftarrow$Vertex($\mathcal{P}_i$) s.t. $\boldsymbol{s} \notin \mathcal{B}_i$
16             repeat
17                 $C_j \leftarrow$CreateNeighborChart($C_i, \alpha, \boldsymbol{s}$)
18                 $\alpha \leftarrow \alpha \cdot 0.9$
19             until not SimilarCharts($C_i, C_j, \epsilon$)
20             $\mathcal{A} \leftarrow \mathcal{A} \cup \{C_j\}$
21         $\mathcal{V} \leftarrow \mathcal{V} \cup \{C_i\}$
22         $\boldsymbol{x}_i \leftarrow$ Center($C_i$)
23         forall $C_j \in$ Neighbor($C_i$) do
24             if $C_j \notin \mathcal{V}$ then
25                 $\boldsymbol{x}_j \leftarrow$ Center($C_j$)
26                 $t \leftarrow c(i) + \boldsymbol{C}(\boldsymbol{x}_i, \boldsymbol{x}_j)$
27                 if $C_i \notin \mathcal{H}$ or $t < c(j)$ then
28                     $\mathcal{H} \leftarrow \mathcal{H} \cup \{C_j\}$
29                     $p(j) \leftarrow i$
30                     $c(j) \leftarrow t$
31                     $h(j) \leftarrow t + \boldsymbol{H}(\boldsymbol{x}_j, \boldsymbol{x}_g)$
32 if $C_i = C_g$ then
33     Return(ReconstructPath($s, g, p$))
34 else
35     Return($\emptyset$)
```

balls $\mathcal{B}_i$ are circles in this case, and the polytopes $\mathcal{P}_i$ are polygons.

Algorithm 1 gives the pseudo-code of the singularity-free path planner proposed in this paper. The planner implements an A* search strategy [28], taking into account the cost and heuristic functions provided. The former function evaluates the cost of the transition between configurations, and the latter provides a lower bound of this cost. In our implementation we use the Euclidean distance in $\mathbb{R}^{n_q+1}$ for both functions, and thus the planner will compute shortest paths in $\mathcal{M}$, up to the resolution used to define the atlas. More sophisticated cost functions can be used, though, considering the travel time or avoiding collisions. In the latter case, the cost function only has to assign an infinite cost to the transitions between charts that cause collisions [19]. At the beginning, the algorithm defines charts at the start and goal configurations and uses them to initialize the atlas (lines 1 to 3). Additionally, it defines the set of charts from where the search can be expanded $\mathcal{H}$ (line 4) and the set of charts already processed along the search, $\mathcal{V}$ (line 5). It then initializes the pointers to the best parent for each chart (line 6), the cost to reach the initial chart (line 7), and the heuristic estimating the cost from this chart to the goal (line 8). After that, the algorithm iterates while there are charts in $\mathcal{H}$ and the goal chart has not been reached yet (lines 10 to 31). In this iteration, the chart C_i with minimum expected cost to the goal is extracted from $\mathcal{H}$ (line 11). If C_i is not the goal chart (line 12), and while it is not a closed chart (line 13), all its neighbors are generated (lines 14 to 20). Note that charts whose center is out of the domain $\mathcal{D}$ are considered closed and, thus, their neighbors are not generated. If necessary, the neighbors are generated by selecting a vertex of $\mathcal{P}_i$ not in $\mathcal{B}_i$ and using this vertex to define the parameters as in (10). The process of generating a neighboring chart is repeated until the conditions given in (11) and (12) hold (line 19). When this happens, the new chart is added to the atlas, coordinating it with the charts already in it (line 20). When all the neighbors for C_i are eventually generated, the chart is added to $\mathcal{V}$ (line 21) and the search is expanded from it. For each of the non-processed neighbors, the tentative cost to the neighbor via C_i is computed (line 26). Charts not yet in $\mathcal{H}$, or charts where the tentative cost is lower than the best cost computed up to the moment, are added to $\mathcal{H}$ (line 28), changing their parent chart (line 29), setting their new cost (line 30), and, finally, computing the heuristic estimation of the cost to the goal (line 31). At the end of the search, if the goal was found, the path connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ is derived using the pointers to the parent chart stored in p (line 33). Otherwise, an empty path is returned (line 35) indicating that it is not possible to determine a singularity-free path at the used resolution.

The cost of the algorithm at each step is dominated by the cost of two searches among the set of charts: one to find the potential neighbors of a new chart when adding it to the atlas (line 20), and another one to find an open chart from which to continue the search. The performace of the first search can be increased using a k-d tree storing the centers of the charts. If $\mathcal{H}$ is implemented using a heap, both the extraction of the next chart to be expanded (line 11) and the insertion of a new chart in $\mathcal{H}$ (line 28) are logarithmic in the number of expandable charts.

IV. Illustrative examples

The performance of the planner is next illustrated in two situations: first on a fictitious three-dimensional C-space, and then on a 3-$\underline{\text{R}}$RR manipulator. The former case is chosen for its simplicity, to illustrate and visualize the method in three dimensions, and the latter shows the method's performance on a real application of considerable complexity. Note that in all cases we make no use of closed-form parameterizations of $\mathcal{C}$. Also, despite the singularity locus and the workspace are shown for reference in the figures (derived using [24, 29] and [30] in our case), explicit knowledge of such sets is not used by the planner in its computations. All results reported have been obtained from an implementation in C of the method available in [31], executed on a MacBook Pro equipped with a 2.66 GHz Intel Core i7 processor.

A. A three-dimensional example

Consider the fictitious C-space defined implicitly by

$$\boldsymbol{\Phi}(q_1, q_2, q_3) = q_1 - \sigma\ \cos(\omega\ (q_2^2 + q_3^2)) = 0,$$

with $\sigma = 0.5$ and $\omega = 0.25$. It is not difficult to see that this equation defines a sinusoidal surface in the space of $\boldsymbol{q} = [q_1, q_2, q_3]^\mathsf{T}$, shown in Fig. 7 for $\boldsymbol{q} \in [-1, 1] \times [-20, 20] \times [-20, 20]$.

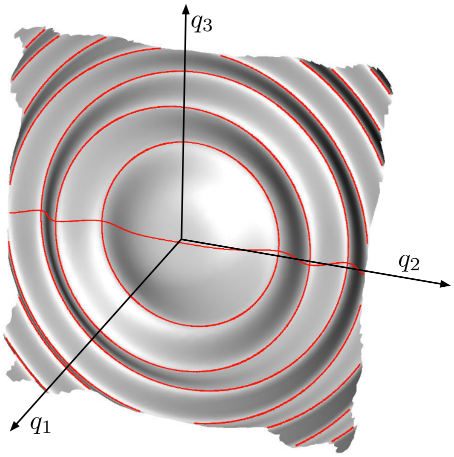

Fig. 7. A fictitious three-dimensional C-space with its singularities highlighted in red, assuming that q_1 and q_2 are the actuated degrees of freedom.

Let us assume for this example that the vector of actuated degrees of freedom is $\boldsymbol{v} = [q_1, q_2]^\mathsf{T}$, so that $\boldsymbol{y} = [q_3]$. Then, the forward singularities are given by the equation

$$det(\boldsymbol{\Phi}_{\boldsymbol{y}}) = \frac{\partial \boldsymbol{\Phi}}{\partial q_3} = 2\,\sigma\,\omega\,q_3\ \sin(\omega\ (q_2^2 + q_3^2)) = 0,$$

which, for non-null parameters w and σ, holds whenever $\omega\ (q_2^2 + q_3^2) = n\,\pi$ with $n \in \mathbb{Z}$, or when $q_3 = 0$. Thus, the singularity locus is formed by the red circles and the sinusoidal line shown in Fig. 7. Note that, in accordance to Fig. 2, the points of such locus are those where the tangent plane to the C-space projects vertically on a line, in the space defined by $\boldsymbol{v} = [q_1, q_2]^\mathsf{T}$.

Fig. 8 shows the results obtained by the planner when trying to connect the configurations $\boldsymbol{q}_s = [0, 4.33, -0.38]^\mathsf{T}$ and $\boldsymbol{q}_g = [0, -4.33, -0.38]^\mathsf{T}$. To compare the results, the figure shows the computed path, in green, when navigating $\mathcal{C}$ (left figure) and $\mathcal{M}$ (right figure), so that the crossing of singularities is permitted and avoided respectively. In both cases the planner returns the shortest path up to the resolution of the generated atlas. The charts of this atlas are shown in blue in both figures, their shape becoming more clear when zooming in the electronic version of the paper.

Note that the path on the left figure crosses the singularity locus twice, while the path on the right figure, although longer, avoids crossing it. The latter path approaches the singularity locus, but we note that a certain clearance is always guaranteed, because the value of $|b|$ is always kept below a given threshold b_{max}. In this case, the value $b_{max} = 12$ was used, but alternative paths with a larger clearance could be obtained by simply reducing b_{max}. Nevertheless, b_{max} should always be chosen larger than the maximum of $|b(\boldsymbol{q}_s)|$ and $|b(\boldsymbol{q}_g)|$, so as to guarantee that the domain $\mathcal{D}$ bounding the search space (Section III-A) includes both $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$. The computation of the singularity-free path took 0.04 seconds in this example, using the continuation parameters $r = 0.25$ and $\epsilon = 0.25$.

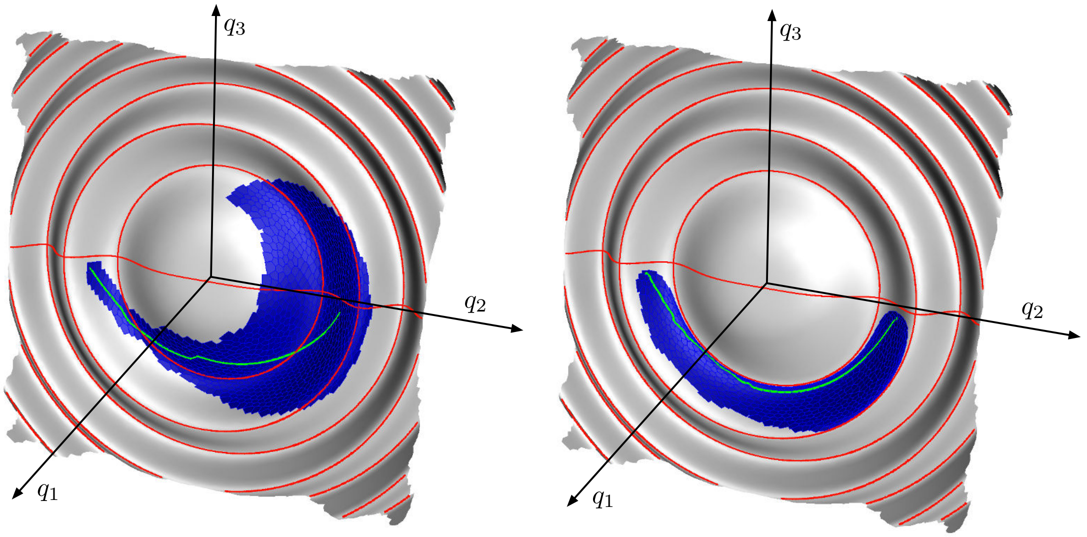

Fig. 8. The path computed by the proposed algorithm when neglecting and considering singularity avoidance (left and right, respectively) on the manifold of Fig. 7. In the plots, the singularity locus is highlighted in red, the charts explored to connect the two query configurations are shown as blue polygons, and the final returned path is shown in green. While the path on the left figure crosses the singularity set twice, the path on the right figure is singularity-free.

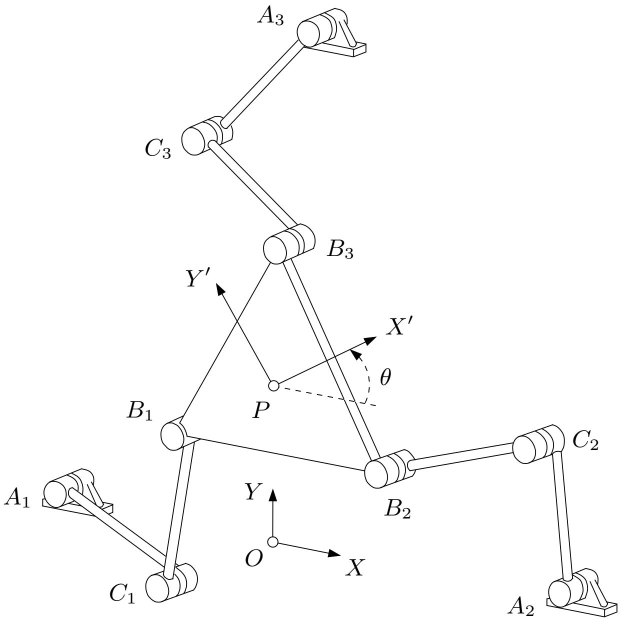

Fig. 9. A 3-$\underline{\text{R}}$RR planar manipulator. Points A_1, A_2, and A_3 are fixed to the ground. The angle θ is measured relative to the OX axis.

B. A 3-$\underline{\text{R}}$RR manipulator

Now, let us consider the planar 3-$\underline{\text{R}}$RR manipulator in Fig. 9, consisting of a moving platform linked to the ground by means of three legs, where each leg is a three-revolute chain. The three joints attached to the ground are actuated, allowing to control the three degrees of freedom of the platform, and the remaining joints are passive. Due to its relatively large workspace, and the fact that the mounting of the actuators on the base reduces weight in the mobile equipment, this is the most common architecture for a 3-DOF planar parallel manipulator [32]. Although the singularity-free path planning problem has been addressed for particular geometries of this manipulator [33], no general path planner has been given yet to avoid its singularities, to the best of our knowledge.

We set the geometry of the manipulator so that the two links of each leg are in a different plane and, thus, they can not collide. In this way, we permit a change of working mode of the leg, which illustrates that such changes can be performed without losing the controllability of the manipulator. Doubts about this fact were initially expressed in the literature. Springs in intermediate leg joints or the exploitation of link inertias were said to be necessary to force the flipping of the legs [32, page 72], but we show that such flippings can be performed without crossing forward singularities, only by controlling the angles of the A_i joints. In fact, the possibility of switching working modes through simple active-joint control has been recently demonstrated using real prototypes [34].

To formulate (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 absolute (OXY) and moving ($PX'Y'$) reference frames (Fig. 9). The assembly constraints imposed by the joints can be formulated as

$$\boldsymbol{a}_i + l_{i1}\begin{bmatrix}\cos\phi_{i1}\\ \sin\phi_{i1}\end{bmatrix} + l_{i2}\begin{bmatrix}\cos\phi_{i2}\\ \sin\phi_{i2}\end{bmatrix} - \boldsymbol{R}\,\boldsymbol{b}_i = \boldsymbol{p}, \tag{13}$$

for $i = 1, 2, 3$, where $\boldsymbol{p} = [x, y]^{\mathsf{T}}$ is the position vector of P relative to the absolute frame, $\boldsymbol{R}$ is the 2×2 rotation matrix of angle θ, and l_{ij} and ϕ_{ij} are the length and absolute orientation angle of the j-th link on the i-th leg. Using the same geometric parameters assumed in [32] and given in Table I, the system in (1) is formed by equations (13) and

$$\boldsymbol{q} = [x, y, \theta, \phi_{11}, \phi_{21}, \phi_{31}, \phi_{12}, \phi_{22}, \phi_{32}]^{\mathsf{T}}. \tag{14}$$

This system implicitly defines a 3-dimensional C-space because $d = n_q - n_e = 9 - 6 = 3$. In this case, since ϕ_{11}, ϕ_{21}, and ϕ_{31} are the actuated degrees of freedom, we have $\boldsymbol{v} = [\phi_{11}, \phi_{21}, \phi_{31}]^{\mathsf{T}}$, and $\boldsymbol{y} = [x, y, \theta, \phi_{12}, \phi_{22}, \phi_{32}]^{\mathsf{T}}$.

TABLE I
PARAMETERS OF THE CONSIDERED 3-$\underline{R}$RR MANIPULATOR.

i	$\boldsymbol{a}_i$	$\boldsymbol{b}_i$	l_{i1}	l_{i2}
1	$(0, 0)$	$(0, 0)$	1	1.35
2	$(2.35, 0)$	$(1.2, 0)$	1	1.35
3	$(1.175, 2.035)$	$(0.6, 0.6\sqrt{3})$	1	1.35

The singularity locus of this manipulator is two-dimensional, and its shape is known to be rather complex. In fact, its slices for a fixed value of θ were studied in [35], and remarkably found to be curves described by a minimal polynomial of degree 42 in x and y. To have an idea, the resulting curve for $\theta = 0$ is shown in red in Fig. 10-top, for the manipulator parameters in Table I. The curve is shown overlaid to the constant orientation workspace of the manipulator (the reachable locations of P when $\theta = 0$), colored in gray, with its boundary curves indicated in blue.

From the figure it may seem that the safe motion range of the manipulator is severely reduced by the presence of singularities—e.g., it appears that P cannot be moved from P_s to P_g while keeping $\theta = 0$—, but note that we are actually observing a projection of the C-space on the xy-plane, and that for each pose $[x, y, \theta]^\mathsf{T}$ of the platform there are up to eight possible inverse kinematic solutions of the manipulator. Each such solution corresponds to a different *working mode* of the mechanism, identified by the sign triple $(\sigma_1, \sigma_2, \sigma_3)$, where σ_i gives the orientation of the triangle $A_iC_iB_i$ [32]. The C-space, thus, has more structure than it looks. It is formed by several "layers" corresponding to the different working modes, and if we project such layers separately, larger singularity-free regions are revealed.

To illustrate, the layers corresponding to the $(+, +, +)$ and $(+, +, -)$ modes for $\theta = 0$ are shown in Fig. 10-bottom, together with the portion of the singularity curve lying on them. A representative configuration is also drawn in each case, with P coinciding on P_s. As we see, P_s and P_g are actually connectable through a singularity-free path lying entirely in the $(+, +, +)$ layer. The proposed planner is able to compute such a path in only 0.05 seconds, using $b_{max} = 3.333$, $r = 0.2$, and $\epsilon = 0.25$, obtaining the results shown in Fig. 11, where the position vectors $\boldsymbol{p}_s = [0.4, -0.6]^\mathsf{T}$ and $\boldsymbol{p}_g = [1.4, 1.5]^\mathsf{T}$ have been assumed for P_s and P_g. Note from the right plot how the manifold $\mathcal{M}$ reaches higher values of b as it approaches the singularities, but the found path avoids these zones and the algorithm returns the shortest possible path in $\mathcal{M}$. The partial atlas generated to resolve the planning query is shadowed in blue in the left plot.

Constant-orientation paths in other layers can be computed if desired, but the full potential of the method comes out when solving complex planning queries in which the platform is allowed to rotate and the manipulator has to change its working mode along the path, to avoid passing through singular configurations (Fig. 12). Due to the difficulty of illustrating a 3-dimensional C-space, the obtained path is shown as a sequence of motion snapshots in this case. The first and last snapshots, (a) and (i), show the start and goal configurations to be connected, which are given by $\boldsymbol{p}_s = [-0.3, -0.9]^\mathsf{T}$ and $\theta_s = 0$, and $\boldsymbol{p}_g = [0.5, 1.9]^\mathsf{T}$ and $\theta_g = -\frac{\pi}{2}$, assuming the working modes $(+, -, -)$ and $(-, +, -)$ respectively. Note that at least two legs should change their working mode along the path, and this can be seen to happen between snapshots (c) and (d) for the first leg, and between (g) and (h) for the second leg. The workspace and the singularities shown in each picture correspond to the orientation of the platform at that moment only, and can be seen to vary with the orientation. Indeed, the path is obtained by exploring a 3-dimensional C-space with a 2-dimensional singularity locus, defined both in a higher-dimensional ambient space, and each picture in Fig. 12 shows a constant-θ slice of such C-space, projected onto the x and y coordinates. The computation of the singularity-free path took 1 second in this example, using $r = 0.3$ and $\epsilon = 0.25$. An animated version of Fig. 12 can be seen in a video attached to this paper, as part of a larger movement planned through three waypoints.

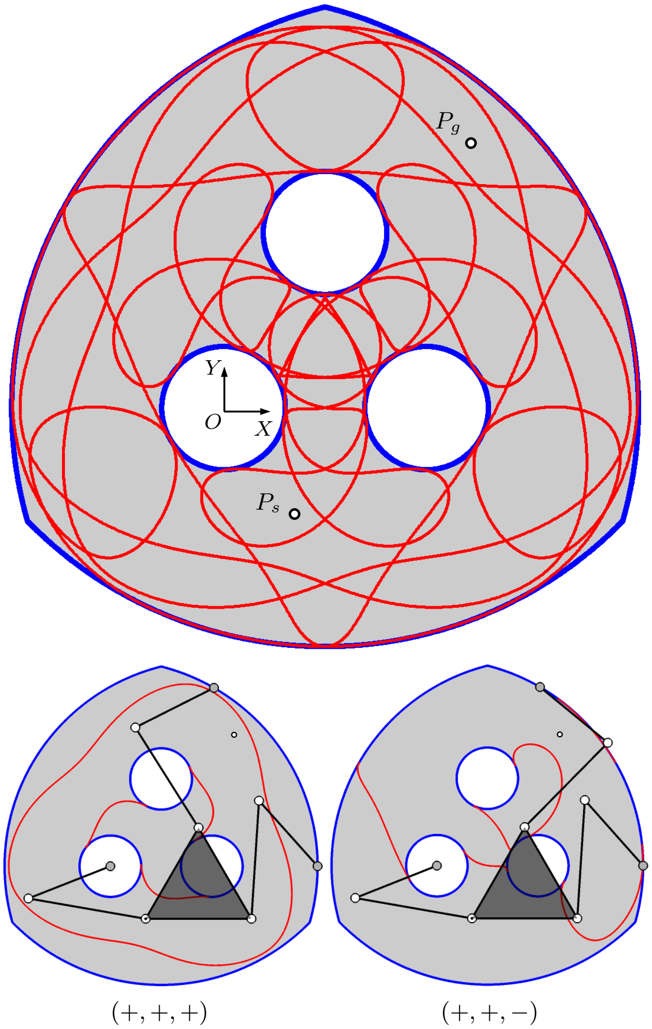

Fig. 10. Top: Boundaries (in blue) of the constant orientation workspace (in gray) and interior singularities (in red) of the 3-$\underline{R}$RR planar manipulator of Table I, assuming $\theta = 0$. Bottom: Layers corresponding to two working modes of the manipulator, differing on the sign of the third leg triangle only, and their corresponding singular curves.

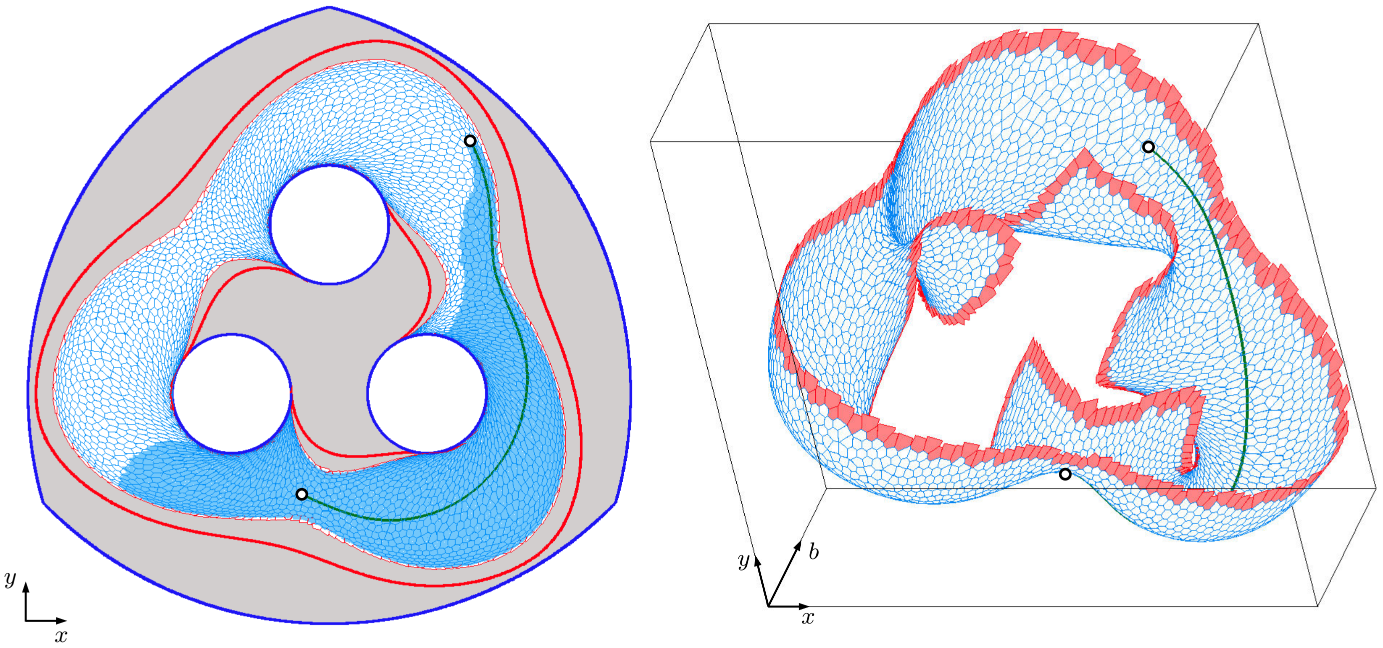

Fig. 11. Results of the method on computing a singularity-free path to connect points P_s and P_g indicated in Fig. 10, assuming that the platform orientation is fixed to $\theta = 0$. Left: The results projected onto the (x, y)-plane. The obtained path is shown in green overlaid onto the atlas of the singularity-free component of $\mathcal{C}$ attainable from the start configuration. The charts are shown colored in white, with blue or red edges depending on whether they lie inside or outside of the domain $\mathcal{D}$. The part actually explored by the algorithm to connect the two configurations is shown shadowed in blue. Right: The same results projected onto the (x, y, b)-space. An animation of this figure is available in the supplementary downloadable material associated with this paper.

Finally, note that inverse singularities could also be avoided by replacing the second equation in (5) by

$$det(\boldsymbol{\Phi_y}) \cdot det(\boldsymbol{\Phi_z}) \cdot b = 1,$$

where in this example $\boldsymbol{z} = [\phi_{11}, \phi_{21}, \phi_{31}, \phi_{12}, \phi_{22}, \phi_{32}]^\mathsf{T}$ [24, 25]. In doing so, we would also ensure a full dexterity of the end effector along the computed path.

V. Conclusions

This paper has introduced a novel approach to compute singularity-free paths on non-redundant closed-chain manipulators of general architecture. Due to the complexity of the involved C-spaces, and of their singularity loci, previous attempts to solve this problem have only considered explicitly-parametrizable C-spaces. In contrast, the approach we present makes no recourse to such parametrizations, and can be applied to any non-redundant closed-chain manipulator with the sole limitations imposed by the computational power available.

The problem has been tackled by defining a system of equations implicitly characterizing the singularity-free C-space of the manipulator, which avoids the need of representing the singularity locus explicitly as an obstacle. The solution manifold of this system can be freely navigated without fear of crossing any forward singularity of the manipulator. Higher-dimensional continuation techniques are then used to progressively construct an atlas of the component of this manifold that contains the start configuration, until the goal configuration is reached, or path non-existence is proved at the resolution of the atlas. Note that, if desired, the method can also be used to detect non-singular transitions between assembly modes [36]–[38], and that it can also generate an exhaustive atlas of the singularity-free C-space component that is reachable from one configuration, which is useful to resolve subsequent planning queries rapidly, or to visualize the singularity-free workspace of the manipulator relative to any set of coordinates. As shown in [39], the latter workspaces are helpful in the context of robot design and analysis.

The resolution completeness of the approach comes at the expense of a computational cost that scales exponentially with the C-space dimension. To deal with higher-dimensional problems, however, we could adapt the approach in [40], which trades off resolution completeness by efficiency and probabilistic completeness, or the approach in [41] which, additionally, guarantees asymptotic optimality. The evaluation of these variants of the planner in the context of singularity-free path planning certainly deserves further attention.

A point left open in the paper is how to establish the threshold b_{max} governing the clearance to the singularity locus. As mentioned, this threshold has to be determined case-wise, depending on the particular context of application. Variants of the method have actually been given for classical or cable-driven hexapods [42, 43], where the clearance is implicitly determined by taking into account the range of forces supported by the actuators.

References

[1] J.-C. Latombe, *Robot motion planning*. Kluwer Academic Pub., 1991.

[2] L. E. Kavraki and S. M. LaValle, *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008, ch. Motion Planning, pp. 109–131.

[3] S. LaValle, *Planning algorithms*. Cambridge Univ Press, 2006.

[4] J. Burdick, “Kinematics and design of redundant robot manipulators,” Ph.D. dissertation, Stanford University, July 1988.

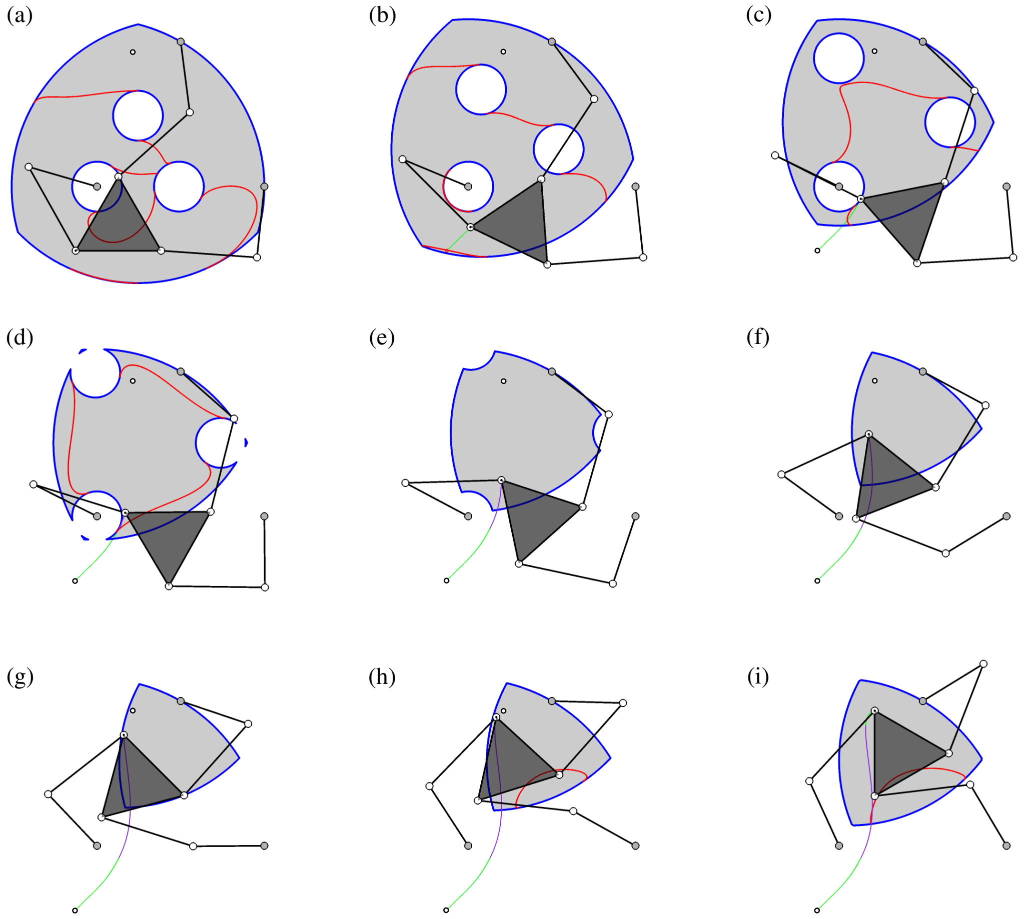

Fig. 12. Several steps of the movement of the manipulator while performing the computed path between configurations (a) and (i). The manipulator is shown overlaid on the constant orientation workspace (in gray with its boundaries in blue) and the singular curves at the corresponding working mode (in red). Changes of working mode occur between steps (c) and (d) for the first leg, and between (g) and (h) for the second leg. An animation of this figure is available in the supplementary downloadable material associated with this paper.

[5] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Algorithmic and Computational Robotics: New Directions. 2000 WAFR*, B. R. Donald, K. Lynch, and D. Rus, Eds. A K Peters Ltd., 2001, pp. 233–246.

[6] J. H. Yakey, S. M. LaValle, and L. E. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Trans. on Robotics*, vol. 17, no. 6, pp. 951–958, 2001.

[7] J. Cortés and T. Siméon, "Sampling-based motion planning under kinematic loop-closure constraints," in *Algorithmic Foundations of Robotics VI*, ser. Springer Tracts in Advanced Robotics, M. Erdmann, D. Hsu, M. Overmars, and A. F. van der Stappen, Eds. Springer, 2005, vol. 17, pp. 75–90.

[8] J. M. Porta, J. Cortés, L. Ros, and F. Thomas, "A space decomposition method for path planning of loop linkages," *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems*, pp. 1882–1888, 2007.

[9] J. M. Porta, L. Jaillet, and O. Bohigas, "Randomized path planning on manifolds based on higher-dimensional continuation," *The Intl. Journal of Robotics Research*, vol. 31, no. 2, pp. 201–215, 2012.

[10] D. Berenson, S. S. Srinivasa, and J. J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The Intl. Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[11] D. Zlatanov, "Generalized singularity analysis of mechanisms," Ph.D. dissertation, University of Toronto, 1998, available through the web address http://goo.gl/rnAUy.

[12] C. Jui and Q. Sun, "Path tracking of parallel manipulators in the presence of force singularity," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 127, pp. 550–563, 2005.

[13] S. Briot and V. Arakelian, "On the dynamic properties and optimum control of parallel manipulators in the presence of singularity," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 2008, pp. 1549–1555.

[14] S. Bhattacharya, H. Hatwal, and A. Ghosh, "Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators," *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 965–974, 1998.

[15] B. Dasgupta and T. Mruthyunjaya, "Singularity-free path planning for the Stewart platform manipulator," *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 711–725, 1998.

[16] S. Sen, B. Dasgupta, and A. K. Mallik, "Variational approach for singularity-free path-planning of parallel manipulators," *Mechanism and Machine Theory*, vol. 38, no. 11, pp. 1165–1183, 2003.

[17] A. K. Dash, I.-M. Chen, S. H. Yeo, and G. Yang, "Workspace generation and planning singularity-free path for parallel manipulators," *Mechanism and Machine Theory*, vol. 40, no. 7, pp. 776–805, 2005.

[18] O. Bohigas, M. Henderson, L. Ros, and J. M. Porta, "A singularity-free path planner for closed-chain manipulators," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, may 2012, pp. 2128–2134.

[19] J. M. Porta and L. Jaillet, "Path planning on manifolds using randomized higher-dimensional continuation," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. C. Lin, Eds. Springer, 2011, vol. 68, pp. 337–353.

[20] J. G. De Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems*. Springer Verlag, 1993.

[21] J. M. Porta, L. Ros, T. Creemers, and F. Thomas, "Box approximations of planar linkage configuration spaces," *ASME Journal of Mechanical Design*, vol. 129, p. 397, 2007.

[22] J. M. Porta, L. Ros, and F. Thomas, "A linear relaxation technique for the position analysis of multi-loop linkages," *IEEE Trans. on Robotics*, vol. 25, no. 2, pp. 225–239, 2009.

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

[24] O. Bohigas, M. Manubens, and L. Ros, "Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case," *Mechanism and Machine Theory*, 2013, in press.

[25] O. Bohigas, "Numerical computation and avoidance of manipulator singularities," Ph.D. dissertation, Universitat Politècnica de Catalunya, 2013, available through the web address http://goo.gl/wlR0i.

[26] M. E. Henderson, "Multiple parameter continuation: Computing implicitly defined k-manifolds," *Intl. Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 451–476, 2002.

[27] ——, "Multiparameter parallel search branch switching," *Intl. Journal of Bifurcation and Chaos in Applied Science and Engineering*, vol. 15, no. 3, pp. 967–974, 2005.

[28] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[29] O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, and J. M. Porta, "Numerical computation of manipulator singularities," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 2012, pp. 1351–1358.

[30] O. Bohigas, M. Manubens, and L. Ros, "A complete method for workspace boundary determination on general structure manipulators," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 993–1006, 2012.

[31] The CUIK project home page, http://www.iri.upc.edu/cuik.

[32] I. A. Bonev, "Geometric analysis of parallel mechanisms," Ph.D. dissertation, Faculté des Sciences et de Génie, Université de Laval, 2002, available through the web address http://goo.gl/lKhw6.

[33] D. Chablat and P. Wenger, "The kinematic analysis of a symmetrical three-degree-of-freedom planar parallel manipulator," in *CISM-IFToMM Symposium on Robot Design, Dynamics, and Control*, 2004.

[34] L. Campos, F. Bourbonnais, I. Bonev, and P. Bigras, "Development of a five-bar parallel robot with large workspace," in *Proc. of the ASME Design Engineering Technical Conferences*, 2010.

[35] I. Bonev and C. Gosselin, "Singularity loci of planar parallel manipulators with revolute joints," in *2nd Workshop on Computational Kinematics (CK2001)*, 2001, pp. 291–299.

[36] C. Innocenti and V. Parenti-Castelli, "Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators," *ASME Journal of Mechanical design*, vol. 120, no. 1, pp. 73–79, 1998.

[37] M. Urizar, V. Petuya, O. Altuzarra, and A. Hernandez, "Assembly mode changing in the cuspidal analytic 3-RPR," *IEEE Trans. on Robotics*, vol. 28, no. 2, pp. 506–513, 2012.

[38] S. Caro, P. Wenger, and D. Chablat, "Non-singular assembly mode changing trajectories of a 6-DOF parallel robot," in *Proc. of the ASME 2012 Intl. Design Engineering Technical Conferences*, 2012, pp. 1–10.

[39] Q. Jiang and C. Gosselin, "Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform," *Mechanism and Machine Theory*, vol. 44, no. 6, pp. 1281–1293, 2009.

[40] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Trans. on Robotics*, vol. 29, no. 1, pp. 105–117.

[41] ——, "Asymptotically-optimal path planning on manifolds," in *Robotics: Science and Systems*, 2012.

[42] O. Bohigas, M. Manubens, and L. Ros, "Planning singularity-free force-feasible paths on the Stewart platform," in *Latest Advances in Robot Kinematics*, J. Lenarcic and M. Husty, Eds. Springer, 2012, pp. 245–252.

[43] ——, "Navigating the wrench-feasible C-space of cable-driven hexapods," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, T. Bruckmann and A. Pott, Eds. Springer, 2013, vol. 12, pp. 53–68.

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

Michael E. Henderson is a member of the Numerical Analysis Group in the Mathematical Sciences Department at IBM's T.J. Watson Research Center. He obtained a B.S. in Engineering Science from The Pennsylvania State University, and a Ph.D. in Applied Mathematics from The California Institute of Technology. His current research interest is nonlinear systems, especially bifurcation theory and continuation methods.

Lluís Ros received the Mechanical Engineering degree in 1992, and the Ph.D. degree (with honors) in Industrial Engineering in 2000, both from the Universitat Politècnica de Catalunya (UPC). From 1993 to 1996 he worked with the Control of Resources Group of the Institut de Cibernètica (Barcelona). He was a visiting scholar at York University (Toronto, 1997), University of Tokyo (Tokyo, 1998), and the Laboratoire d'Analyse et Architecture des Systèmes (Toulouse, 1999). He joined the Institut de Robòtica i Informàtica Industrial (IRI, CSIC-UPC) in 1997, where he is currently an Associate Researcher of the Spanish National Research Council (CSIC) since 2004. His research interests include geometry and kinematics, with applications to robotics, computer graphics, and machine vision.

Montserrat Manubens received the Mathematics degree from the Universitat de Barcelona in 2001, and the Ph.D. degree (with honors) in Computer Algebra from the Universitat Politècnica de Catalunya in 2008. From 2009 to 2010 she worked with the Robotics Group of the Institut de Recherche en Communications et Cybernétique de Nantes (France) in the analysis of cuspidal robots. Since 2011 she holds a Juan de la Cierva contract at the Institut de Robòtica i Informàtica Industrial, (IRI, CSIC-UPC). Her current research interests include mathematics and kinematics, with applications to robotics.

Josep M. Porta received the Engineering degree in Computer Science in 1994, and the Ph.D. degree (with honors) in Artificial Intelligence in 2001, both from the Universitat Politècnica de Catalunya, Spain. From 2001 to 2003 he held a postdoctoral position at the University of Amsterdam, pursuing research in autonomous robot localization using vision. Currently, he is an Associate Researcher of the Spanish National Research Council at the Institut de Robòtica i Informàtica Industrial (IRI, CSIC-UPC), Barcelona, Spain. His current research interests include planning under uncertainty and computational kinematics.