# Singularities of Non-redundant Manipulators: A Short Account and a Method for their Computation in the Planar Case

Oriol Bohigas[a], Montserrat Manubens[a], and Lluís Ros[a]

[a] *Institut de Robòtica i Informàtica Industrial (UPC-CSIC)*
*Llorens Artigas 4-6, 08028 Barcelona, Catalonia*
*E-mails:* {*obohigas,mmanuben,lros@iri.upc.edu*}

**Abstract**

The study of the singularity set is of utmost utility in understanding the local and global behavior of a manipulator. After reviewing the mathematical conditions that characterize this set, and their kinematic and geometric interpretation, this paper shows how these conditions can be formulated in an amenable manner in planar manipulators, allowing to define a conceptually-simple method for isolating the set exhaustively, even in higher-dimensional cases. As a result, the method delivers a collection of boxes bounding the location of all points of the set, whose accuracy can be adjusted through a threshold parameter. Such boxes can then be projected to the input or output coordinate spaces, obtaining informative diagrams, or portraits, on the global motion capabilities of the manipulator. Examples are included that show the application of the method to simple manipulators, and to a complex mechanism that would be difficult to analyze using common-practice procedures.

*Keywords:* Singularity set, planar manipulator, forward singularity, inverse singularity, box approximation, branch-and-prune method.

## 1. Introduction

Singularity Analysis is a central topic of Robot Kinematics. It has as a goal to study certain configurations, termed *singular* or *critical*, where important changes take place in the kinetostatic performance of a manipulator. Motion control or dexterity losses can arise, and there may appear unresolvable or uncontrollable end-effector forces, among other effects. The study of singularities is therefore motivated by a desire to avoid these configurations, but it may be helpful to operate close to them sometimes, such as when handling heavy objects, drilling, or fine-positioning, where extreme force or motion transformation ratios are often required. Independently of the context of application, however, it is clear that reliable tools to compute and visualize the whole singularity set are needed to properly assist the robot design and programming processes.

Numerous mathematical conditions aimed at characterizing singularity have been given in the literature [1, 2, 3, 4], even for manipulators of general architecture [5, 6, 7, 8, 9]. The earliest attempt to provide a general framework to determine and classify all singular configurations can be attributed to Gosselin and Angeles [5], who proposed the use of input/output velocity equations to define the well-known "Type I" and "Type II" singularities, where the velocity of the end-effector does not determine the velocities of the actuators, and *vice versa*. The approach was sound, but neglected the role played by passive joint velocities, and it was later found that further singularity types existed that could not be framed into their formalism [6, 7, 8]. This observation led Zlatanov to define singular configurations in a more general way, as those where the forward or the inverse instantaneous kinematic problems[1] become undetermined [8], and to identify three fundamental types of singularities—*input*, *output*, and *C-space* singularities, also characterized by Park independently [9]—, which can be further classified into six lower-level types according to the kinematic degeneracy occurring in them. Zlatanov's characterization of singularity is probably the most systematic and general one proposed so far in the literature, and accommodates, as special cases, the earlier Type I/II singularities, and subtle singularities, such as constraint [10, 11] or architecture singularities [12, 13].

[1] Understood as the computation of the *overall* configuration velocity, given the input or output velocities.

These advances in mathematical characterization, however, have not been paralleled by corresponding advances in the development of general algorithms for computing the entire singularity set. Previous methods for studying the set are effective, but restrict their attention to narrowly-defined classes of manipulators [14, 15, 16, 17, 18, 19, 20, 21], or to particular singularity types [22], and a general approach able to isolate all possible singularities on a large class of manipulators is still lacking. To help covering such gap, this paper proposes a numerical method for computing the singularity set of planar manipulators of *general* architecture, i.e., encompassing open or closed kinematic chains interconnected in any possible way, by means of revolute or prismatic pairs.

The method builds upon earlier work on position analysis [23, 24]. It is based on describing the singularity set as the solution of a system of quadratic equations, and on exploiting the particular form of these equations to define a branch-and-prune strategy that can approximate the set in a multi-resolutive fashion. As a result, a collection of boxes forming an outer envelope of the set is delivered, which can be computed at the desired precision. The method can also be used to derive useful representations, or *portraits*, of the singularity set, defined as projections of the C-space of the manipulator to the input and output spaces, with all singularity points indicated. These diagrams provide valuable information on the reachable areas, possible motion impediments, and safe navigation regions of the manipulator in each of such spaces.

A main assumption of the paper is that the studied manipulators are non-redundant, to allow a more simple and symmetric presentation of results. However, the analysis of mechanisms with redundant actuation should also be tackleable with machinery similar to the one presented. Also, the emphasis is on illustrating the method on closed-chain mechanisms, because they are those exhibiting the whole range of singular phenomena, but the results remain applicable to arbitrary multibody systems.

The rest of the paper is structured as follows. Section 2 reviews the mathematical conditions that characterize the singularity set of any non-redundant manipulator, the main singularity types, and the kinematic consequences of traversing the configurations of each type. The presentation is terse in comparison to systematic treatments like [8], but it provides geometric arguments that are difficult to find elsewhere, and summarizes necessary background for the rest of the paper. Section 3 focuses on the planar case, and uses the previous conditions to develop systems of quadratic equations describing the singularity set. Sections 4 and 5 describe the method proposed to solve these systems numerically, and how the computed solutions can be processed to obtain the aforementioned portraits. Section 6 illustrates the application of the method to manipulators with a well-known singularity set, and to a highly-complex mechanism that would be difficult to analyze using common-practice approaches. Section 7, finally, provides the main conclusions of the paper and outlines points deserving further attention.

## 2. Singular configurations

### *2.1. Mathematical conditions*

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

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

which expresses the assembly constraints imposed by the joints [25, 23, 24]. Here, $\boldsymbol{\Phi}(\boldsymbol{q}) : \mathcal{Q} \to \mathcal{E}$ is a differentiable map, where $\mathcal{Q}$ and $\mathcal{E}$ are $n_q$- and $n_e$-dimensional manifolds respectively, and Eq. (1) is meant to include all possible assembly constraints, including those due to mechanical limits on the joints, which can also be modelled as equality constraints (Appendix A).

Let $C$ denote the C-space of the manipulator. That is,

$$C = \{\boldsymbol{q} \in \mathcal{Q} : \boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0}\}. \tag{2}$$

In the usual setting, the differential $\boldsymbol{\Phi}_{\boldsymbol{q}} = \left[\partial\Phi_i/\partial q_j\right]$ is full rank at all points $\boldsymbol{q} \in C$, except on a subset $\mathcal{G} \subset C$ where $C$ may lose the manifold structure. Thus, $C \setminus \mathcal{G}$ is a smooth manifold of dimension $d = n_q - n_e$, whose tangent space at a point $\boldsymbol{q}$ is the $d$-dimensional set

$$T_{\boldsymbol{q}} C = \{\dot{\boldsymbol{q}} : \dot{\boldsymbol{q}} \in \mathrm{Ker}(\boldsymbol{\Phi}_{\boldsymbol{q}})\}.$$

The vector $\boldsymbol{q}$ will be assumed to contain a vector $\boldsymbol{v}$ of $n_v$ input coordinates, corresponding to the actuated degrees of freedom of the manipulator, and a vector $\boldsymbol{u}$ of $n_u$ output coordinates, corresponding to the end-effector variables defining its functionality. This allows to consider the partitions $\boldsymbol{q} = [\boldsymbol{y}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}]^\mathsf{T}$ and $\boldsymbol{q} = [\boldsymbol{z}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}]^\mathsf{T}$ where $\boldsymbol{y}$ and $\boldsymbol{z}$ encompass the $n_y$ and $n_z$ coordinates remaining in $\boldsymbol{q}$ after the removal of $\boldsymbol{v}$ and $\boldsymbol{u}$, respectively, and to write Eq. (1) in either of the following forms:

$$\boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{0}, \tag{3}$$

$$\boldsymbol{\Phi}(\boldsymbol{z}, \boldsymbol{u}) = \boldsymbol{0}. \tag{4}$$

Hereafter, the $\boldsymbol{v}-$, and $\boldsymbol{u}-$ spaces will be denoted by $\mathcal{V}$ and $\mathcal{U}$ respectively, and it will be further assumed that the manipulator is non-redundant, i.e., that $n_v = n_u = d$, which means that the number of inputs, and also the outputs, is the lowest necessary to determine the overall configuration $\boldsymbol{q}$. This implies that $n_y = n_z = n_e$ in particular, so that Eqs. (3) and (4) are well-determined systems of equations in general, for fixed values of $\boldsymbol{v}$ and $\boldsymbol{u}$.

To see the role played by singular configurations, consider the time derivatives of Eqs. (3) and (4):

$$\boldsymbol{\Phi_y}\,\dot{\boldsymbol{y}} + \boldsymbol{\Phi_v}\,\dot{\boldsymbol{v}} = \boldsymbol{0}, \tag{5}$$

$$\boldsymbol{\Phi_z}\,\dot{\boldsymbol{z}} + \boldsymbol{\Phi_u}\,\dot{\boldsymbol{u}} = \boldsymbol{0}. \tag{6}$$

Note that for configurations $\boldsymbol{q}$ on which $\boldsymbol{\Phi_y}$ and $\boldsymbol{\Phi_z}$ are non-singular, we can write Eqs. (5) and (6) in the equivalent form

$$\dot{\boldsymbol{y}} = -\boldsymbol{\Phi}_{\boldsymbol{y}}^{-1}\,\boldsymbol{\Phi_v}\,\dot{\boldsymbol{v}}, \tag{7}$$

$$\dot{\boldsymbol{z}} = -\boldsymbol{\Phi}_{\boldsymbol{z}}^{-1}\,\boldsymbol{\Phi_u}\,\dot{\boldsymbol{u}}, \tag{8}$$

which provide the solution to the forward and inverse instantaneous kinematic problems of the manipulator. However, Eqs. (7) and (8) only hold whenever $\boldsymbol{\Phi_y}$ and $\boldsymbol{\Phi_z}$ are full rank, and only in this case the input and output rates $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{u}}$ will determine unique values for the remaining rates $\dot{\boldsymbol{y}}$ and $\dot{\boldsymbol{z}}$. This must be so because, when $\boldsymbol{\Phi_y}$ is rank-deficient at $\boldsymbol{q}$, Eq. (5) yields, for a given value of $\dot{\boldsymbol{v}}$, either no solution or infinitely-many solutions for $\dot{\boldsymbol{y}}$, in which case it is not possible to determine the velocity $\dot{\boldsymbol{q}}$ of the manipulator by specifying the velocities $\dot{\boldsymbol{v}}$ of the actuators. When $\boldsymbol{\Phi_z}$ is rank-deficient at $\boldsymbol{q}$, Eq. (6) reveals an analogous relation between $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{q}}$. Following these observations, a configuration $\boldsymbol{q} \in C$ is said to be *singular* if either $\boldsymbol{\Phi_y}$ or $\boldsymbol{\Phi_z}$ is rank deficient at $\boldsymbol{q}$, and the set $\mathcal{S}$ of all of such configurations is called the *singularity set* of the manipulator [8].

Note now that $\mathcal{S}$ can be obtained as the union of the solution sets of the following systems of equations

$$\left.\begin{array}{c}\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0}\\ \boldsymbol{\Phi_y}\,\boldsymbol{\xi} = \boldsymbol{0}\\ \|\boldsymbol{\xi}\|^2 = 1\end{array}\right\} \tag{9}$$

$$\left.\begin{array}{c}\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0}\\ \boldsymbol{\Phi_z}\,\boldsymbol{\xi} = \boldsymbol{0}\\ \|\boldsymbol{\xi}\|^2 = 1\end{array}\right\} \tag{10}$$

where the first equation in each system constrains $\boldsymbol{q}$ to be a feasible configuration of the manipulator, and the second and third equations enforce the existence of a non-zero vector $\boldsymbol{\xi}$ in the kernel of the corresponding matrix. The points $\boldsymbol{q}$ satisfying the left (resp. right) system will be called *forward* (resp. *inverse*) singularities.

### *2.2. Kinematic and geometric interpretation*

A rapid inspection of Eqs. (5) and (6) reveals that forward and inverse singularities correspond to configurations in which the locking of the input or output coordinates yields an infinitesimally flexible, or shaky, mechanism [26]. Such a degenerate behaviour has further physical consequences, which can be better appreciated by classifying the points in $\mathcal{S}$ according to the geometric cause of rank deficiency in $\boldsymbol{\Phi_y}$ or $\boldsymbol{\Phi_z}$. Three types of singular configurations can be distinguished according to this criterion:

1. *C-space singularities*, defined as the points $\boldsymbol{q} \in \mathcal{G} \subset C$ where the whole differential $\boldsymbol{\Phi_q}$ is rank deficient, so that both the forward and inverse kinematic problems become unsolvable in the form of Eqs. (7) and (8), independently of the choice of input and output coordinates.
2. *Input singularities*, or the points $\boldsymbol{q} \in C \setminus \mathcal{G}$ where $\boldsymbol{\Phi_y}$ is rank deficient, so that the forward kinematic problem becomes unsolvable.
3. *Output singularities*, or the points $\boldsymbol{q} \in C \setminus \mathcal{G}$ where $\boldsymbol{\Phi_z}$ is rank deficient, so that the inverse instantaneous kinematic problem is unsolvable.

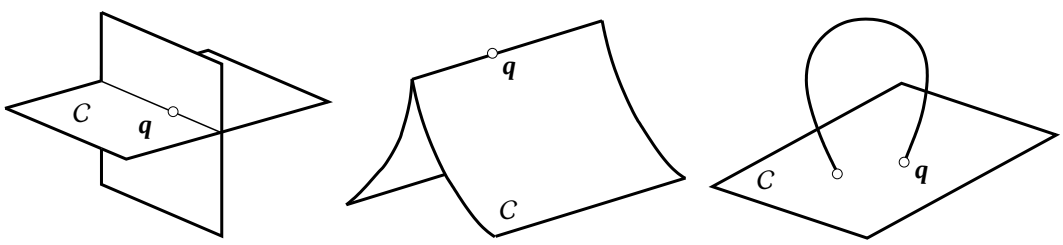


Figure 1: Examples of C-space singularities.

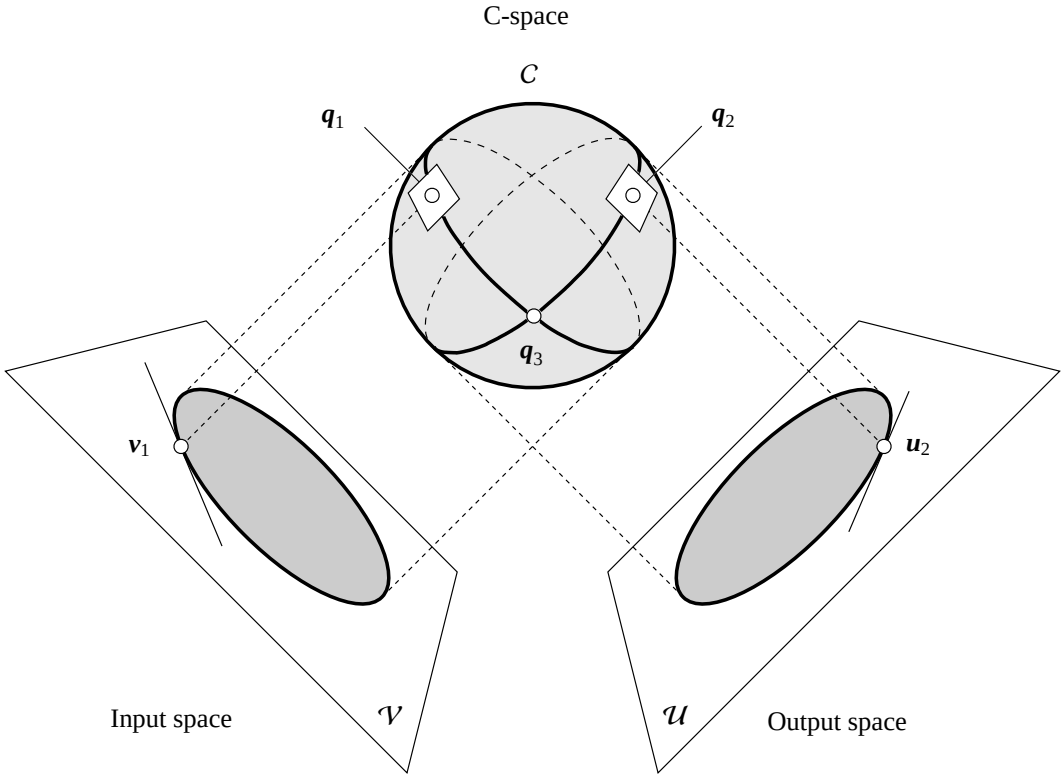


Figure 2: Interpretation of input and output singularities when $\mathcal{Q} = \mathbb{R}^3$, $\mathcal{C}$ is a sphere, and $\mathcal{V}$ and $\mathcal{U}$ are two coordinate planes of $\mathbb{R}^3$. In the figure, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ correspond to an input and an output singularity, respectively, and $\boldsymbol{q}_3$ is both an input and an output singularity. In this example, a smooth trajectory in $\mathcal{V}$ (resp. $\mathcal{U}$) through $\boldsymbol{v}_1$ (resp. $\boldsymbol{u}_2$) does not locally determine a unique smooth trajectory in $\mathcal{C}$.

C-space singularities correspond to points $\boldsymbol{q}$ in which $\mathcal{C}$ may lose the manifold structure, such as bifurcations, ridges, or dimension changes (Fig. 1). Since $\boldsymbol{\Phi_q}$ is rank deficient at such points, the tangent space to $\mathcal{C}$ becomes ill-defined in them, and there is an increase in the instantaneous mobility of the manipulator. The increased mobility cannot be controlled even if we change the location of the actuators, because it is intrinsic to the design of the mechanism. On input and output singularities, contrarily, $\boldsymbol{\Phi_q}$ is full rank and $\mathcal{C}$ has a $d$-dimensional tangent space, but this space has a special position [27]. This is easy to see when $\mathcal{Q} = \mathbb{R}^{n_q}$, in which case $\mathcal{C}$ can be regarded as a subset of $\mathbb{R}^{n_q}$. In such a situation, input singularities correspond to points $\boldsymbol{q}$ where the tangent space to $\mathcal{C}$ projects down to $\mathcal{V} = \mathbb{R}^{n_v}$ as a linear space of dimension lower than $n_v$, and output singularities are the points where the tangent space to $\mathcal{C}$ projects to $\mathcal{U} = \mathbb{R}^{n_u}$ as a subspace of dimension lower than $n_u$ (Fig. 2). Whereas input singularities yield controllability issues (a feasible vector $\dot{\boldsymbol{v}}$ does not determine a unique vector $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{C}$), output singularities correspond to mobility losses of the end-effector (independently of the value of $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\,\mathcal{C}$, $\dot{\boldsymbol{u}}$ is always restricted to a linear subspace of smaller dimension).

The implicit function theorem [28] provides further insight as to the advantages of avoiding each singularity type. As a consequence of the theorem, if $\boldsymbol{\Phi_y}$ is full rank at a point $\boldsymbol{q}_0 = [\boldsymbol{y}_0^\mathsf{T}, \boldsymbol{v}_0^\mathsf{T}]^\mathsf{T}$, a smooth trajectory $\boldsymbol{v}(t) \subset \mathcal{V}$ through $\boldsymbol{v}_0$ will locally correspond to a unique smooth trajectory $\boldsymbol{q}(t)$ on $\mathcal{C}$ through $\boldsymbol{q}_0$, or, in other words, the overall movement of the manipulator will be controllable through the inputs. In a similar way, whenever $\boldsymbol{\Phi_z}$ is full rank at $\boldsymbol{q}_0 = [\boldsymbol{z}_0^\mathsf{T}, \boldsymbol{u}_0^\mathsf{T}]^\mathsf{T}$, a smooth trajectory $\boldsymbol{u}(t)$ through $\boldsymbol{u}_0$ will locally determine a unique smooth trajectory $\boldsymbol{q}(t)$ on $\mathcal{C}$, so that a tracking of the output will be sufficient to predict the overall motion of the manipulator. This one-to-one correspondence between the input or output trajectories, on the one hand, and the manipulator trajectory, on the other hand, is not guaranteed at a singular configuration. This can even be inferred from the simple situation of Fig. 2, which provides, as we see, a powerful image to intuitively understand the critical phenomena that occur at a singularity.

Note finally that, since the rank deficiency of $\boldsymbol{\Phi_q}$ implies the rank deficiency of $\boldsymbol{\Phi_y}$ and $\boldsymbol{\Phi_z}$, forward singularities are the union of C-space and input singularities, whereas inverse singularities are the union of C-space and output singularities. As it turns out, a point $\boldsymbol{q} \in \mathcal{C} \setminus \mathcal{G}$ can be both an input and an output singularity, so that both the forward and the inverse instantaneous kinematic problems may become unsolvable on $\mathcal{C} \setminus \mathcal{G}$. C-space singularities can be singled-out if desired, by defining a system similar to those in Eqs. (9) and (10), but imposing the rank deficiency of $\boldsymbol{\Phi_q}$ instead of that of $\boldsymbol{\Phi_y}$ or $\boldsymbol{\Phi_z}$.

## 3. Formulating the equations of the singularity set

We next show that a particular choice of configuration coordinates allows formulating Eqs. (9) and (10) in an amenable manner on planar manipulators, suitable to adopt a simple branch-and-prune strategy to solve these systems numerically. The formulation closely follows that of reference point coordinates in Multibody Dynamics, which leads to polynomial equations of a simple quadratic form with little manipulation, in comparison to other formulations departing from loop constraints on relative joint displacements [29], or to distance-based formulations [30].

### *3.1. Configuration coordinates and assembly constraints*

Let us assume that our manipulator has $n_b$ links and $n_j$ joints, labelled $L_1, \ldots, L_{n_b}$, and $J_1, \ldots, J_{n_j}$, respectively, where $L_1$ is supposed to be the ground link. We furnish every link $L_l$ with a local reference frame, $\mathcal{F}_l$, letting $\mathcal{F}_1$ act as the absolute frame. We will write $\mathbf{v}^{\mathcal{F}_l}$ to indicate that the components of a vector $\mathbf{v} \in \mathbb{R}^2$ are provided in the basis of $\mathcal{F}_l$, and we will assume that vectors with no superscript are expressed in the basis of $\mathcal{F}_1$. Then, the pose of each link in the manipulator can be specified by the pair $(\mathbf{r}_l, \mathbf{R}_l)$, where $\mathbf{r}_l = (x_l, y_l)$ is the position of the origin of $\mathcal{F}_l$ in frame $\mathcal{F}_1$, and

$$\boldsymbol{R}_l = \begin{bmatrix} \cos\theta_l & -\sin\theta_l \\ \sin\theta_l & \cos\theta_l \end{bmatrix}$$

is the rotation matrix expressing the orientation of $\mathcal{F}_l$ relative to $\mathcal{F}_1$. Note that the link poses cannot be arbitrary though, as they must fulfill the assembly constraints imposed by the joints.

If $J_i$ is a revolute joint connecting links $L_j$ and $L_k$, the assembly constraint of this joint is equivalent to imposing the coincidence of two points on the joint, $P_i$ and $Q_i$, respectively fixed to $L_j$ and $L_k$ [Fig. 3 (a)]. This condition can be formulated as follows

$$\boldsymbol{r}_j + \boldsymbol{R}_j\, \boldsymbol{p}_i^{\mathcal{F}_j} = \boldsymbol{r}_k + \boldsymbol{R}_k\, \boldsymbol{q}_i^{\mathcal{F}_k}, \tag{11}$$

where $\boldsymbol{p}_i^{\mathcal{F}_j}$ and $\boldsymbol{q}_i^{\mathcal{F}_k}$ are the constant position vectors of $P_i$ and $Q_i$ in $\mathcal{F}_j$ and $\mathcal{F}_k$ respectively. The joint angle at $J_i$ is not explicit in Eq. (11), but it can easily be obtained as

$$\alpha_i = \theta_j - \theta_k. \tag{12}$$

If $J_i$ is a prismatic joint, we consider two points $P_i$ and $Q_i$ on the axis of the joint as before, but also a unit vector $\boldsymbol{d}_i$ aligned with the joint [Fig. 3 (b)]. The assembly constraint is then equivalent to forcing $P_i$ to lie on the axis of the joint on $L_k$, defined by $Q_i$ and $\boldsymbol{d}_i$, while keeping the relative angle between $L_j$ and $L_k$ fixed to a constant offset $\delta_i$. These conditions are equivalent to

$$\boldsymbol{r}_j + \boldsymbol{R}_j\, \boldsymbol{p}_i^{\mathcal{F}_j} = \boldsymbol{r}_k + \boldsymbol{R}_k\, \boldsymbol{q}_i^{\mathcal{F}_k} + d_i\, \boldsymbol{R}_k\, \boldsymbol{d}_i^{\mathcal{F}_k}, \tag{13}$$

and

$$\delta_i = \theta_j - \theta_k, \tag{14}$$

where $\boldsymbol{d}_i^{\mathcal{F}_k}$ is the direction vector $\boldsymbol{d}_i$ expressed in $\mathcal{F}_k$, and $d_i$ is the linear displacement of the joint.

In our case, thus, Eq. (1) is the system formed by Eqs. (11)-(14) established for all joints of the manipulator, and $\boldsymbol{q}$ is the vector encompassing the variables $x_l$, $y_l$, and $\theta_l$ of all links, and $\alpha_l$ and $d_l$ for all joints. Note only that, since $L_1$ is the ground link, $\boldsymbol{r}_1 = \boldsymbol{0}$, $\theta_1 = 0$, and $\boldsymbol{R}_1$ is the identity matrix. Thus, for a system of $n_b$ links and $n_j$ joints, the number of variables in $\boldsymbol{q}$ will be $n_q = 3(n_b - 1) + n_j$, and the system in (1) will have $n_e = 3n_j$ equations. Accordingly, the dimension of $C$ will be $d = n_q - n_e = 3(n_b - 1) - 2n_j$ in general, meaning that $d$ of the variables $\alpha_l$ and $d_l$ will be actuated, forming the $\boldsymbol{v}$ vector, and $d$ of the variables $x_i$, $y_i$, and $\theta_i$ will describe the output of the manipulator, forming the $\boldsymbol{u}$ vector.

It is worth noting that, in fact, Eq. (12) is only necessary for each *actuated* revolute joint, and that many of the variables $\boldsymbol{r}_l = (x_l, y_l)$ can be eliminated if closed kinematic chains are present in the manipulator, through a process explained in detail in [24]. The elimination of the $\boldsymbol{r}_l$ variables is based on the observation that Eqs. (11) and (13) arising along a closed chain can be substituted by an equivalent "loop-closure" equation that does not contain any of the $\mathbf{r}_l$ variables. This process simplifies the system, and can always be invoked if desired, but the explanations that follow are equally applicable to both the original and the simplified systems.

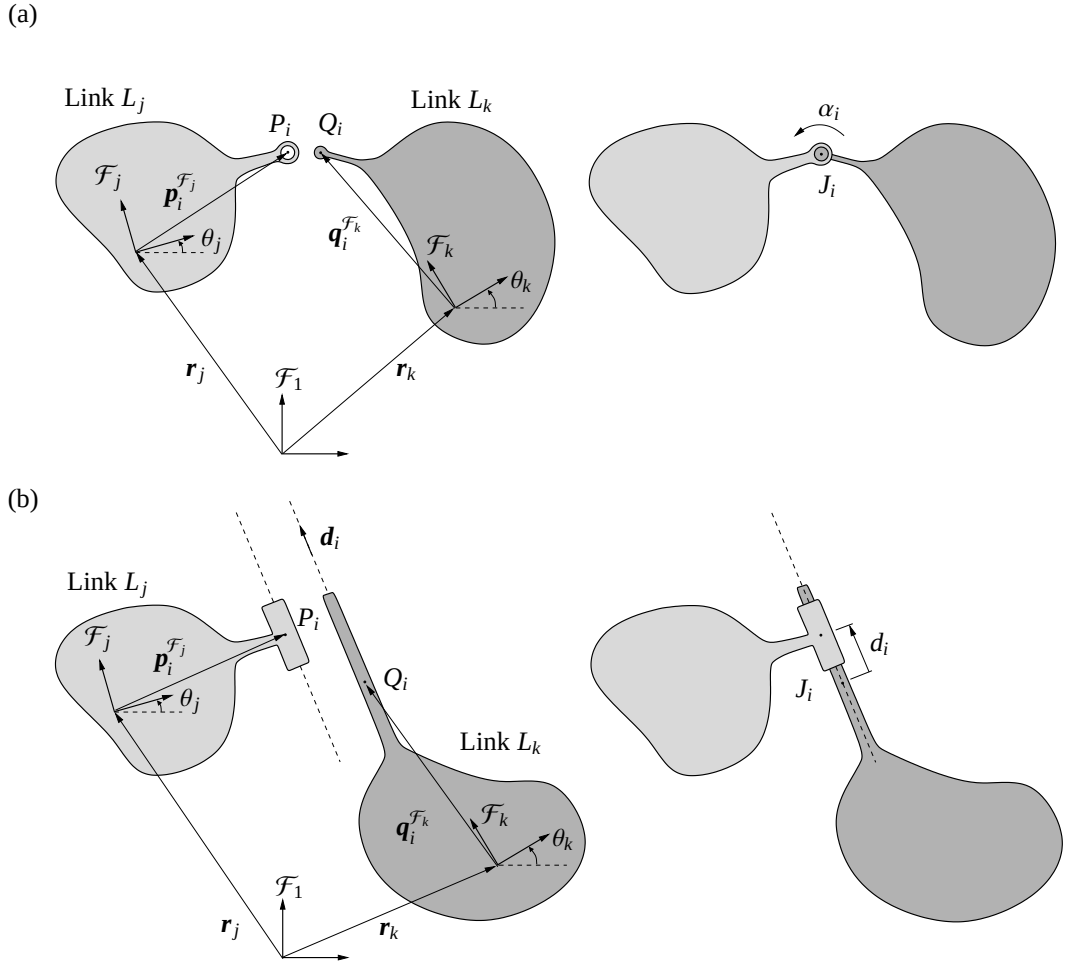


Figure 3: Geometric elements intervening in the assembly of revolute and prismatic pairs.

### *3.2. Reduction to a simple quadratic form*

From the previous formulation, we note that all terms intervening in $\mathbf{\Phi}(\boldsymbol{q})$ are either linear in the $\boldsymbol{q}$ variables, or multilinear in the sines and cosines of the $\theta_i$ variables, which implies that all terms of Eqs. (9) and (10) will also have the same form. The following three steps can be applied now, in order to convert any of these systems into the polynomial form required in Section 4. First, replace each occurrence of Eq. (12) by the equivalent equations

$$\begin{aligned} s_{\alpha_i} &= \sin\theta_j \cos\theta_k - \cos\theta_j \sin\theta_k, \\ c_{\alpha_i} &= \cos\theta_j \cos\theta_k + \sin\theta_j \sin\theta_k, \end{aligned}$$

where the coordinates $s_{\alpha_i}$ and $c_{\alpha_i}$ refer to the sine and cosine of $\alpha_i$, respectively. If prismatic pairs are present, perform a similar replacement on Eqs. (14). Second, to obtain a polynomial system of equations, introduce the changes of variables $c_{\theta_l} = \cos\theta_l$ and $s_{\theta_l} = \sin\theta_l$ for each angle $\theta_l$, together with the equation $c_{\theta_l}^2 + s_{\theta_l}^2 = 1$. Third, use the changes of variables

$$p_k = r_i^2, \tag{15}$$

$$b_k = r_i r_j, \tag{16}$$

on terms of the form $r_i^2$ and $r_i r_j$ of this system, in order to convert it into the expanded form

$$\left.\begin{aligned} \mathbf{\Lambda}(\boldsymbol{x}) &= 0 \\ \mathbf{\Omega}(\boldsymbol{x}) &= 0 \end{aligned}\right\}, \tag{17}$$

where $\boldsymbol{x}$ is an $n_x$-dimensional vector encompassing all of the variables, $\mathbf{\Lambda}(\boldsymbol{x}) = 0$ is a subsystem of linear equations in $\boldsymbol{x}$, and $\mathbf{\Omega}(\boldsymbol{x}) = 0$ is a subsystem gathering all equations of the form of (15) and (16) introduced.

Eq. (17) involves more equations and variables than the original system, but the simpler structure of its equations is beneficial to the branch-and-prune strategy defined next. Another advantage is that it is straightforward to define conservative bounds for all solutions of Eq. (17), since: (1) the variables in $\boldsymbol{x}$ that refer to sines and cosines can only take values in the $[-1, 1]$ interval; (2) simple feasibility intervals for the $x_l$ and $y_l$ variables can be derived from the link dimensions; and (3) intervals for the $p_k$ and $b_k$ variables can be obtained by simple interval operations using Eqs. (15) and (16). From the Cartesian product of such intervals, thus, it is possible to define an initial rectangular box $\mathcal{B} \subset \mathbb{R}^{n_x}$ bounding all solutions of Eq. (17).

## 4. Computing the singularity set

The algorithm for solving Eq. (17) 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 the box is reduced to an empty set, in which case it contains no solution, or the box is "sufficiently" small, in which case it is considered a *solution* box, or the box cannot be "significantly" reduced, in which case it is bisected into two sub-boxes via box splitting (which simply bisects its largest interval). To converge to all solutions, the whole process is recursively applied to the new sub-boxes, until one obtains a collection of solution boxes whose side lengths are below a given threshold $\sigma$.

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

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

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 $[\underline{x_i}, \overline{x_i}]$ denotes the interval of $\mathcal{B}_c$ along dimension $x_i$, then:

1. The portion of the parabola $p_k = x_i^2$ 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 $x_i = \underline{x_i}$ and $x_i = \overline{x_i}$, 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 $[\underline{x_i}, \overline{x_i}] \times [\underline{x_j}, \overline{x_j}]$ vertically to the paraboloid (Fig. 4b).

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

As it turns out, the previous algorithm explores a binary tree whose internal nodes correspond to boxes that have been split at some time, and whose leaves are either solution or empty boxes. The collection of all solution boxes is returned as

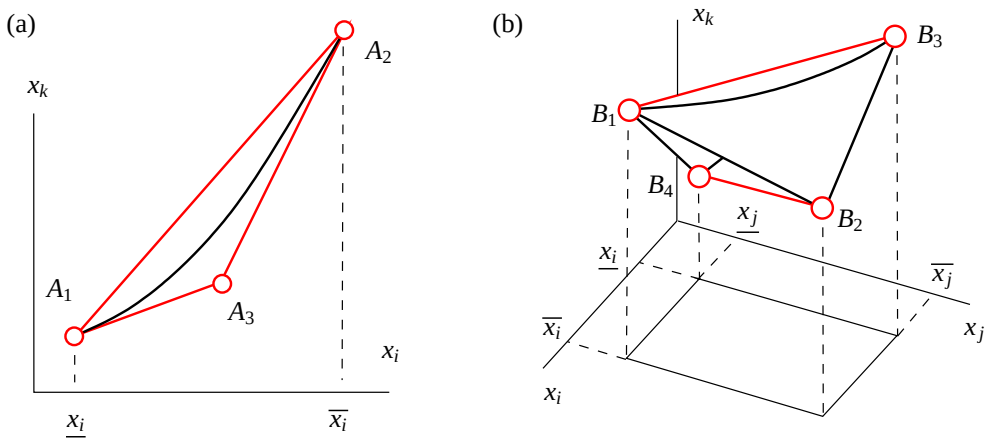


Figure 4: Polytope bounds within box $\mathcal{B}_c$.

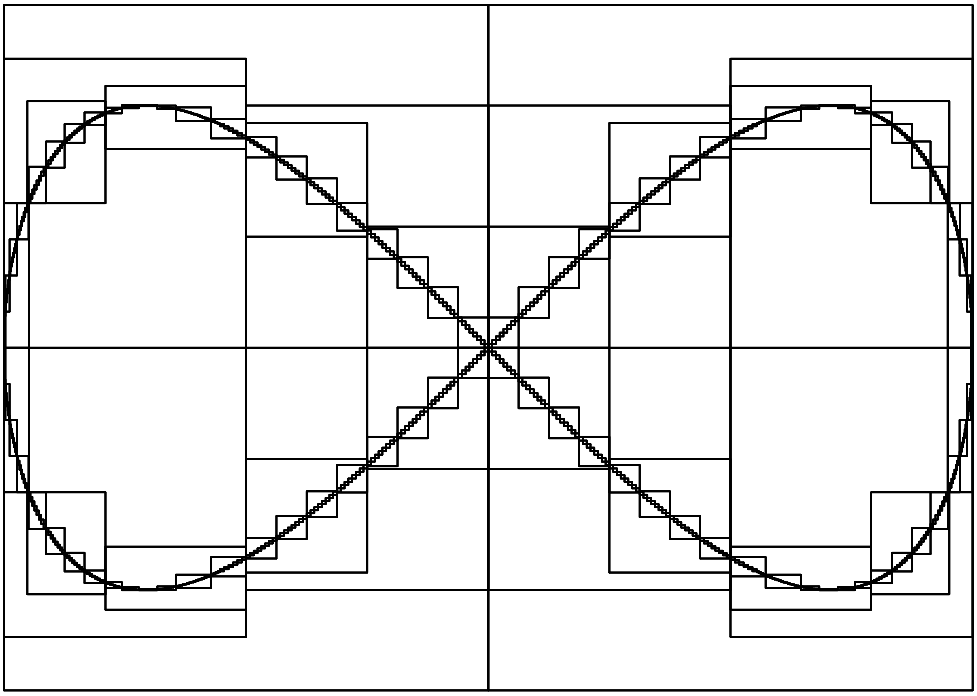

Figure 5: Progression of the algorithm on computing the lemniscate curve of Gerono, defined by the equation $x^4 = (x^2 - y^2)$. The figure shows the initial box, together with intermediate and final box approximations generated by the algorithm.

output upon termination, and it is said to form a *box approximation* of the solution set of Eq. (17), because it forms a discrete envelope of such set, whose accuracy can be adjusted through the $\sigma$ parameter.

Fig. 5 illustrates such approximations on a simple example. Notice that the algorithm is complete, in the sense that it will succeed in isolating all solution points of the solved system accurately, provided that a small-enough value for $\sigma$ is used. Detailed properties of the algorithm, including an analysis of its completeness, correctness, and convergence order, are given in [24].

It is worth noting that the previous algorithm can be naturally parallelized to be run on multi-processor computers. To this end, we can just implement the book-keeping of the search tree on a selected "master" processor which keeps track of the tree leaves at all times. Every leaf that is neither an empty nor a solution box needs to be further reduced. Since box reduction is the most time-consuming task, and several boxes await for it simultaneously, it makes sense to perform the reductions in parallel, by assigning each of them to any of the remaining "slave" processors. A slave processor's task is thus to receive a box from the master processor, to reduce it as much as possible by solving the aforementioned linear programs, and to return the reduced box back to the master, which will queue it for further splitting and reduction, if needed, or mark it as a solution or an empty box.

## 5. Visualizing the singularity set

Even though we have a means to compute $\mathcal{S}$, a non-trivial issue is how to represent this set in a meaningful way, suitable to the needs of a robot designer. Because of the high number of configuration variables typically involved in $\boldsymbol{q}$, $\mathcal{S}$ is often defined in a highly-dimensional space, so that the use of 2- or 3-D projections becomes inevitable to understand its structure. An enlightening choice, as done e.g. in [22, 16, 18, 19], is to project $\mathcal{S}$ to the output space $\mathcal{U}$, since this space encodes the end-effector motion and is easier to interpret. On such a projection, points corresponding to inverse singularities indicate a loss of instantaneous degrees of freedom relative to the $\boldsymbol{u}$ variables, and thus include the boundaries and interior barriers of the workspace relative to such variables [22, 31]. Similarly, $\mathcal{S}$ can be projected to the input space $\mathcal{V}$, as done e.g. in [20, 32, 21], where the forward singularities delimit the motion range that should be reachable by the actuators. Both the $\mathcal{V}$ and $\mathcal{U}$ spaces get partitioned into several regions after such projections, and it is possible to decide which regions correspond to feasible configurations of the manipulator by selecting a point in each region, and solving Eqs. (3) or (4) with $\boldsymbol{v}$ or $\boldsymbol{u}$ fixed to the selected point, using the same numerical method described in Section 4.

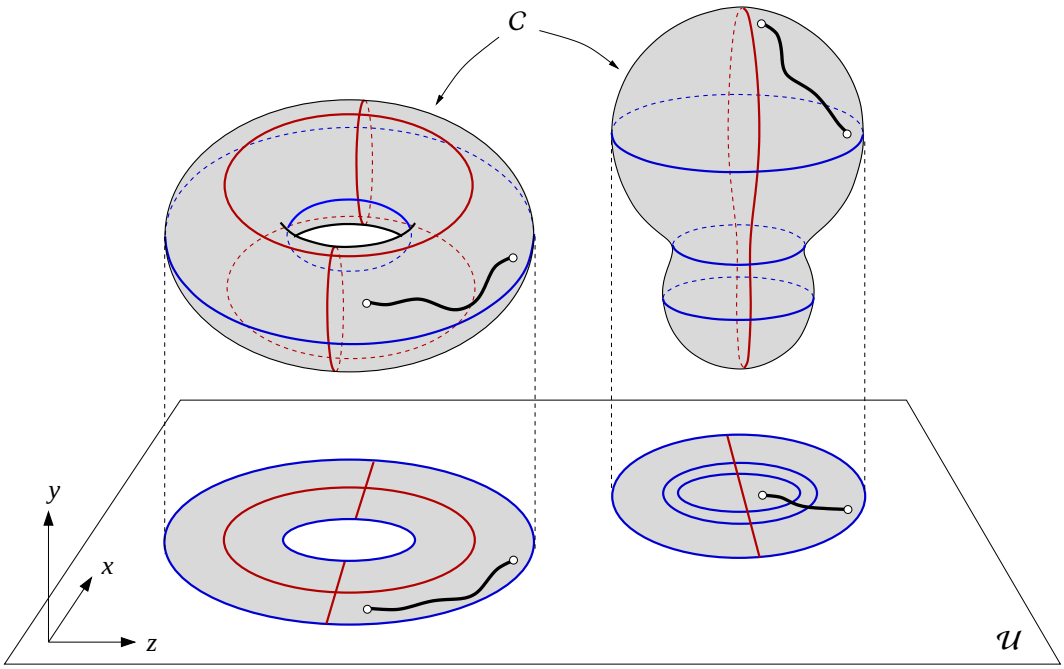


Figure 6: A portrait of a synthetic C-space with two connected components. The $\mathcal{V}$ and $\mathcal{U}$ spaces are assumed to be the *xy*- and *xz*-planes in this case, so that the forward and inverse singularity loci are the red and blue curves, respectively. Only the portrait on the $\mathcal{U}$ space is shown for simplicity. The portrait, as in this case, may reveal the existence of several connected components in $C$. Also, it can be used as a safe navigation map, because paths in the portrait not crossing a projected singularity correspond to singularity-free paths on $C$ (left path). However, the converse is not necessarily true (right path).

The resulting diagrams, which we refer to as singularity *portraits*, convey much global information on the motion capabilities of the manipulator because (Fig. 6):

- The existence of several connected components in $C$ may be revealed by the portrait, and such knowledge may be useful to determine the most appropriate component into which the manipulator should be "assembled" by design, depending on the task to be performed with it.
- A feasible path in $\mathcal{V}$ or $\mathcal{U}$ not crossing a projected singularity corresponds to a singularity-free path in $C$
- Only when approaching a projected singularity some kind of motion degeneracy is to be expected, so that a portrait can be used as a safe navigation map of $C$.

It must be added that the connectivity of the singularity-free regions of $C$ is only partially reflected in the portraits. It is easy to see on the right component of Fig. 6, for example, that distinct points of $C$ may seem to be separated by singularities when looking at the portrait, while they are actually connected by singularity-free paths on $C$. However, robust numerical tools have been given to determine the existence of such paths, and to provide the whole singularity-free region of $C$ that is reachable from a given configuration [33, 34].

## 6. Illustrative examples

We next demonstrate the performance of the method on computing and visualizing the singularities of 3-RRR manipulators, and on a mechanism of a complex structure. Whereas the former serve to verify the correctness of the method on well-studied cases, the latter shows the method capabilities on mechanisms that would be difficult to analyze using common-practice techniques. All computations have been carried out using the parallelized version of the method outlined in Section 4, implemented in C using the libraries of the CUIK Suite [24], and executed on a grid computer with 20 dual quad-core Xeon processors. A table is given at the end of the section, summarizing the size of the solved systems and the main performance data on the reported problems. In all plots that follow, the same color code adopted in Fig. 6 has been used to distinguish the forward and inverse singularity loci, and to identify the regions of $\mathcal{U}$ and $\mathcal{V}$ attainable by the manipulator.

*6.1. Parallel 3-RRR manipulators*

The 3-RRR manipulator consists of a moving platform linked to the ground by means of three legs (Fig. 7), where each leg is a three-revolute chain. The three intermediate joints at points $C_i$ are actuated, allowing to control the three degrees of freedom of the platform, and the remaining joints are passive. The inputs of the manipulator are thus given by the joint angles $\alpha_i$ at the $C_i$ joints, so that $\boldsymbol{v} = [\alpha_1, \alpha_2, \alpha_3]^\mathsf{T}$ in this case. Since the moving platform acts as the end-effector, the output of interest is given by the pose vector $\boldsymbol{u} = [x, y, \theta]^\mathsf{T}$, where $(x, y)$ and $\theta$ provide the position and orientation of the platform respectively (Fig. 7).

Several tools have been proposed to study the singularity set $\mathcal{S}$ of this manipulator [35, 36, 37], which is known to be two-dimensional in general. A good reference summarizing them is [16], where it is shown that the forward singularities can be derived from those of the 3-RPR manipulator [35], whereas the inverse singularities can be generated geometrically, from the so-called vertex-spaces of the legs. These methods are useful, but concentrate on deriving the constant-orientation slices of $\mathcal{S}$ only, so that a reconstruction of the whole singularity surface involves a discretization on the angle $\theta$, which necessarily leaves points of $\mathcal{S}$ out of the representation. Moreover, only projections of the slices on the $(x, y)$-plane are derived, so that the visualization of the singularity surface on the input space, for example, is not straightforward. The method we present in this paper, in contrast, allows to compute the whole singularity surface directly on $C$, and to project it easily to any required space, including $\mathcal{V}$ or $\mathcal{U}$, without incurring in any loss of information.

To compute $\mathcal{S}$, the proposed method requires formulating Eq. (1) as explained in Section 3, by gathering Eqs. (11) and (12) for all joints of the manipulator. This system can be simplified slightly in this case to obtain two loop-closure equations, for instance those relative to the loops starting at $A_1$ and returning back through $A_2$ and $A_3$, plus additional relations providing all input and output coordinates of the manipulator. The resulting system implicitly defines the three-dimensional C-space $C$ of the manipulator, and can be used to formulate Eqs. (9) and (10) through differentiation, using the definitions for the $\boldsymbol{v}$ and $\boldsymbol{u}$ vectors assumed above. Both of these systems can be expanded to the form of Eq. (17), giving rise to a polynomial system with 29 equations and 31 variables in the two cases. The same geometric parameters adopted in [16] have been used in such systems, to ease the comparison of results. They are indicated in Table 1, where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ provide the positions of $A_i$ and $B_i$ in the absolute and relative frames, respectively, and $l_{i,1}$ and $l_{i,2}$ indicate the length of the proximal and distal links of the $i$-th leg.

The singularity surfaces obtained by the method are shown in Fig. 8, projected to the output space. The blue surface corresponds to the inverse singularity locus, which provides the boundaries of the workspace. The red surface corresponds

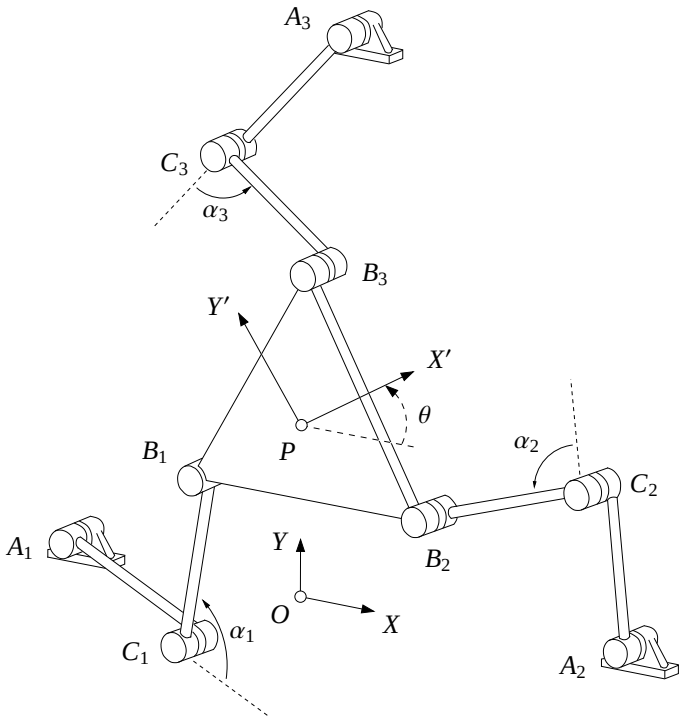


Figure 7: A planar 3-RRR manipulator. Points $A_1$, $A_2$, and $A_3$ are fixed to the ground. Absolute ($OXY$) and relative ($PX'Y'$) reference frames are defined, fixed to the ground and to the moving platform respectively. The platform pose is given by the absolute coordinates $(x, y)$ of a point $P$, and by the angle $\theta$ of $PX'Y'$ relative to $OXY$.

| manipulator | $i$ | $\boldsymbol{a}_i$ | $\boldsymbol{b}_i$ | $l_{i,1}$ | $l_{i,2}$ |
|---|---|---|---|---|---|
| 3-R$\underline{\text{R}}$R | 1 | $(0, 0)$ | $(0, 0)$ | 4 | 3 |
| | 2 | $(-2.386, 0)$ | $(-0.276, 0.276)$ | 4 | 3 |
| | 3 | $(-1.193, -2.067)$ | $(-0.919, 0.184)$ | 4 | 3 |
| 3-$\underline{\text{R}}$RR | 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 |

Table 1: Parameters of the considered 3-RRR manipulators.

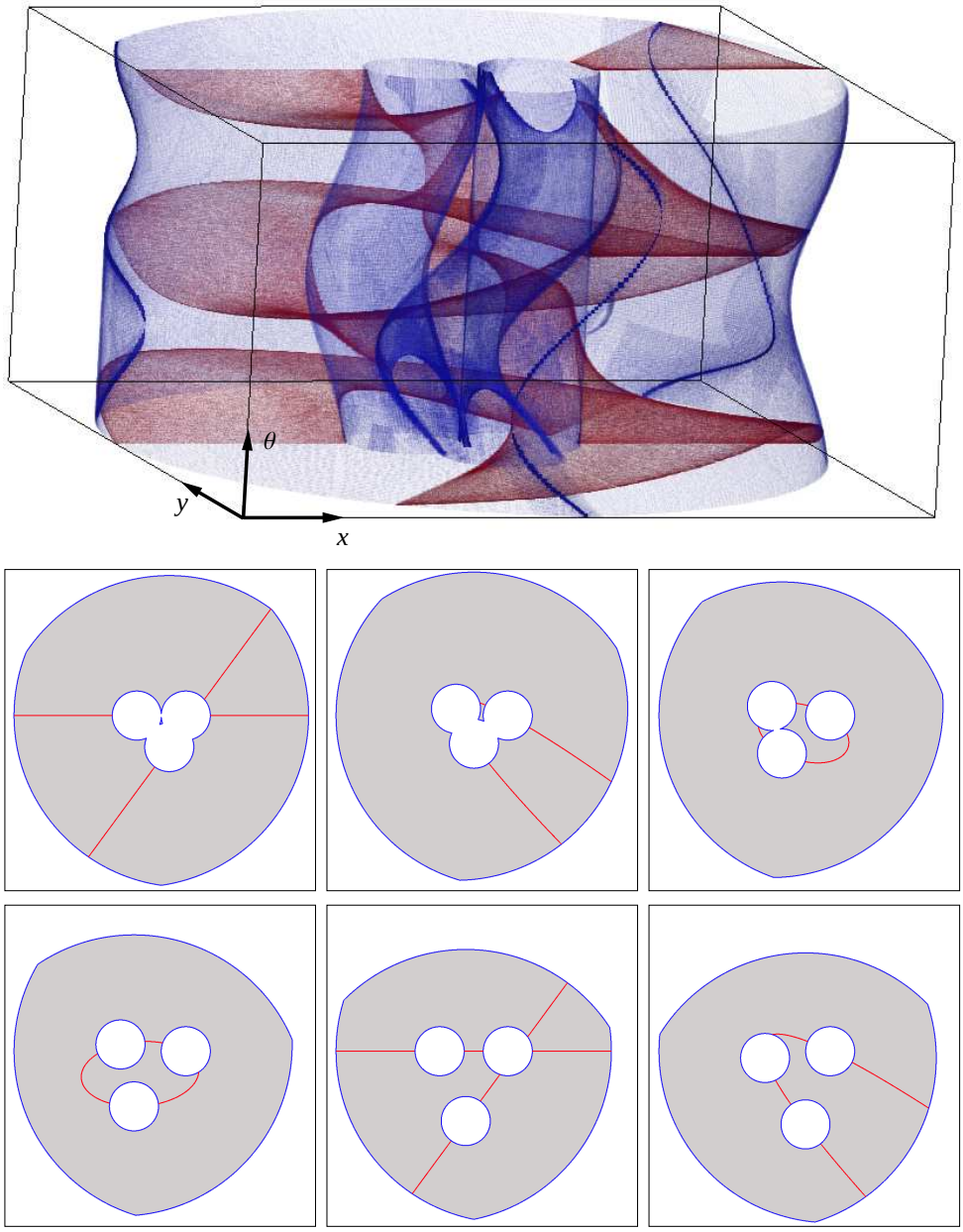


Figure 8: Output portrait obtained for the 3-R$\underline{\text{R}}$R manipulator. Top: Forward (red) and inverse (blue) singularity surfaces in the space defined by $x$, $y$, and $\theta$. The boxes computed are drawn with translucent faces to better appreciate the shape of the surfaces. Bottom: Slices of the output portrait at a constant value of $\theta$. From top to bottom, and from left to right, the values assumed are $\theta = -\pi$, $-\frac{3\pi}{4}$, $-\frac{\pi}{2}$, $-\frac{\pi}{4}$, 0, and $\frac{\pi}{4}$.

to the forward singularity locus, i.e., to configurations where the motion control is compromised, due to the specific choice of actuated degrees-of-freedom. Even though these singularity surfaces appear to be quite complex, it can be shown that the constant-orientation slices of the forward singularity locus can be described by conic sections in the $(x, y)$-plane [16, 35]. Any of these slices can be readily obtained by the proposed method by simply fixing the value of $\theta$ in the equations, obtaining the red curves shown in Fig. 8, bottom, where only parabolas, ellipses or pairs of lines appear as expected. The inverse singularity curves in such plots do also coincide with those obtained through the intersection of vertex spaces [36, 16].

By simply changing the projection coordinates we can easily represent $\mathcal{S}$ in the input space as well, obtaining the results shown in Fig. 9. Here, the forward singularities delimit the motion range of the actuators, and it can be seen how the inverse singularities only appear in planes where one of the $\alpha_i$ angles is either 0 or $\pi$, in agreement with the fact that the platform only loses instantaneous mobility when at least one of the legs is fully extended or folded back [16]. To better understand the

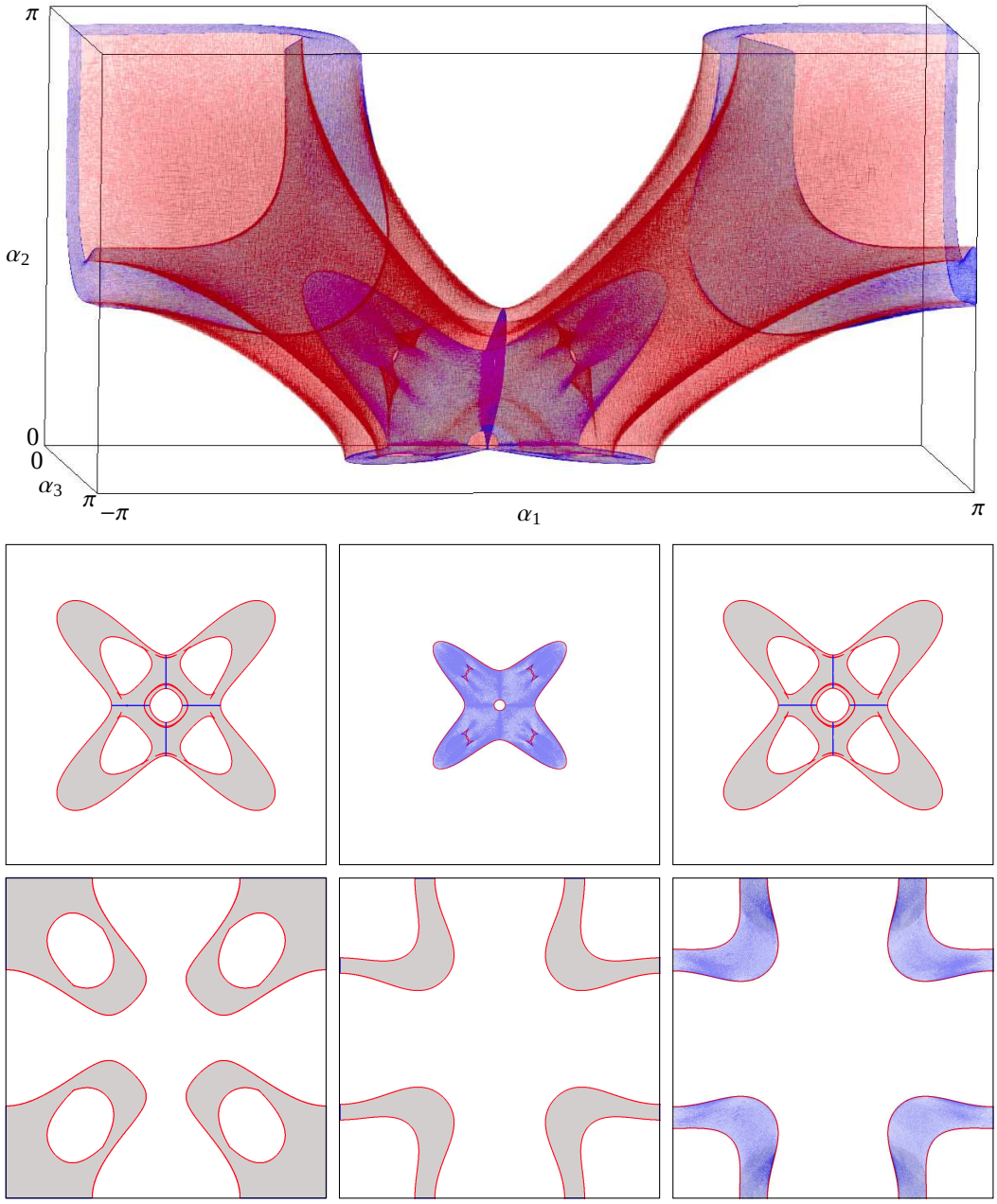


Figure 9: Input portrait of the 3-RRR manipulator. Top: Forward (red) and inverse (blue) singularity surfaces in the space defined by $\alpha_1$, $\alpha_2$, and $\alpha_3$. Only two octants of the space are shown for simplicity, the other octants being obtained by symmetry. Bottom: Slices of the input portrait at different values of $\alpha_3$. From left to right, and from top to bottom, the values assumed are $\alpha_3 = -\frac{\pi}{4}$, 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$, and $\pi$.

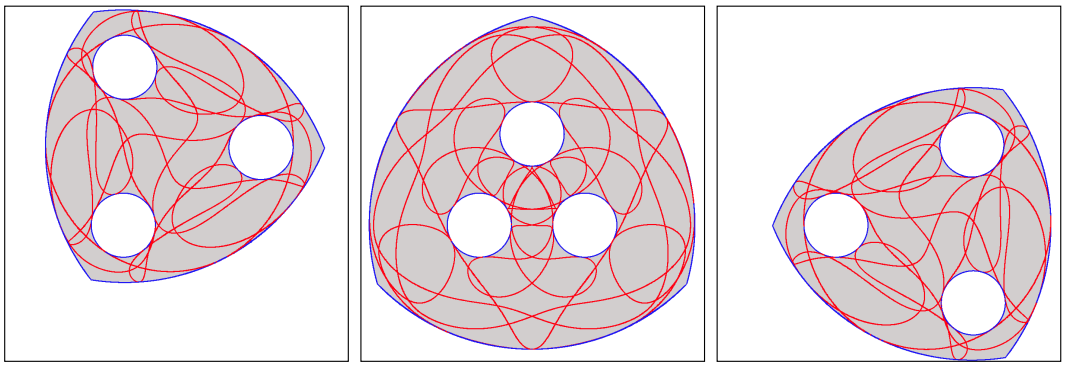

Figure 10: Slices of the output portrait of the 3-RRR manipulator computed by the method at fixed orientations of the platform, assuming the geometric parameters in Table 1, bottom. From left to right, the values $\theta = -\frac{\pi}{4}$, 0 and $\frac{\pi}{4}$ are assumed. The plot of the $\theta = 0$ slice agrees with the one published in [37, 16].

structure of the singularity surface on the input space, some slices are also shown for constant values of $\alpha_3$. Observe how the whole region attainable by the inputs is singular for $\alpha_3 = 0$ or $\alpha_3 = \pi$. On these slices, the inverse singularities are no longer one-dimensional, as one would expect. Whereas this circumstance poses no problem to the proposed method, it may indeed hinder the application of other methods relying on discretization of the $\alpha_3$ angle.

It must be noted that the structure of the singularity set can become quite complex even on simple manipulators. For example, if on the 3-RRR mechanism we mount the actuators in the $A_i$ joints instead of in the $C_i$ ones, the constant-orientation slices of the forward singularity locus are then described by polynomials in $x$ and $y$ of minimal degree 42 [16]. Polynomials of such kind constitute valuable tools for the analysis of the singularity set, but their derivation often requires quite involved manipulations guided by intuition [14, 18, 19, 21], which makes it difficult to apply such a strategy to every new manipulator that has to be analyzed. The proposed method can compute the mentioned slices just as easily as in the case of the 3-RRR manipulator (Fig. 10), but its full potential is more apparent on mechanisms of much higher complexity, where the analytic approach based on descriptive polynomials would be rather difficult to apply.

### *6.2. A complex mechanism*

To illustrate the method on a highly complex situation, we next apply it to compute the singularity set of the 15-link mechanism in Fig. 11 (a). The mechanism consists of five quadrilateral links interconnected through bar links and revolute joints, forming a decagonal ring. If we fix one of the quadrilaterals to the ground, the mechanism has mobility two, so that $C$ will have dimension $d = 2$ in general, and the singularity set will be formed by one or several curves in such space. Assuming that the mechanism is controlled by actuating the $\theta_1$ and $\theta_2$ angles indicated, and that the output is given by the $(x, y)$ coordinates of a point $P$ on link $L$, given in the absolute frame $OXY$, we have $\boldsymbol{v} = [\theta_1, \theta_2]^\mathsf{T}$ and $\boldsymbol{u} = [x, y]^\mathsf{T}$ in this case.

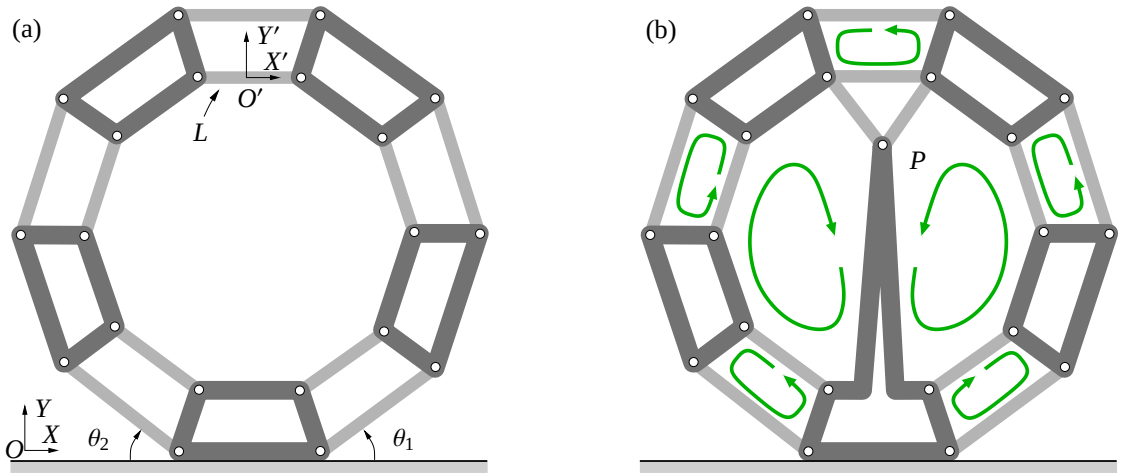


Figure 11: (a) A 15-link mechanism. (b) Its inverse kinematics problem is equivalent to solving the position analysis of a seven-loop truss.

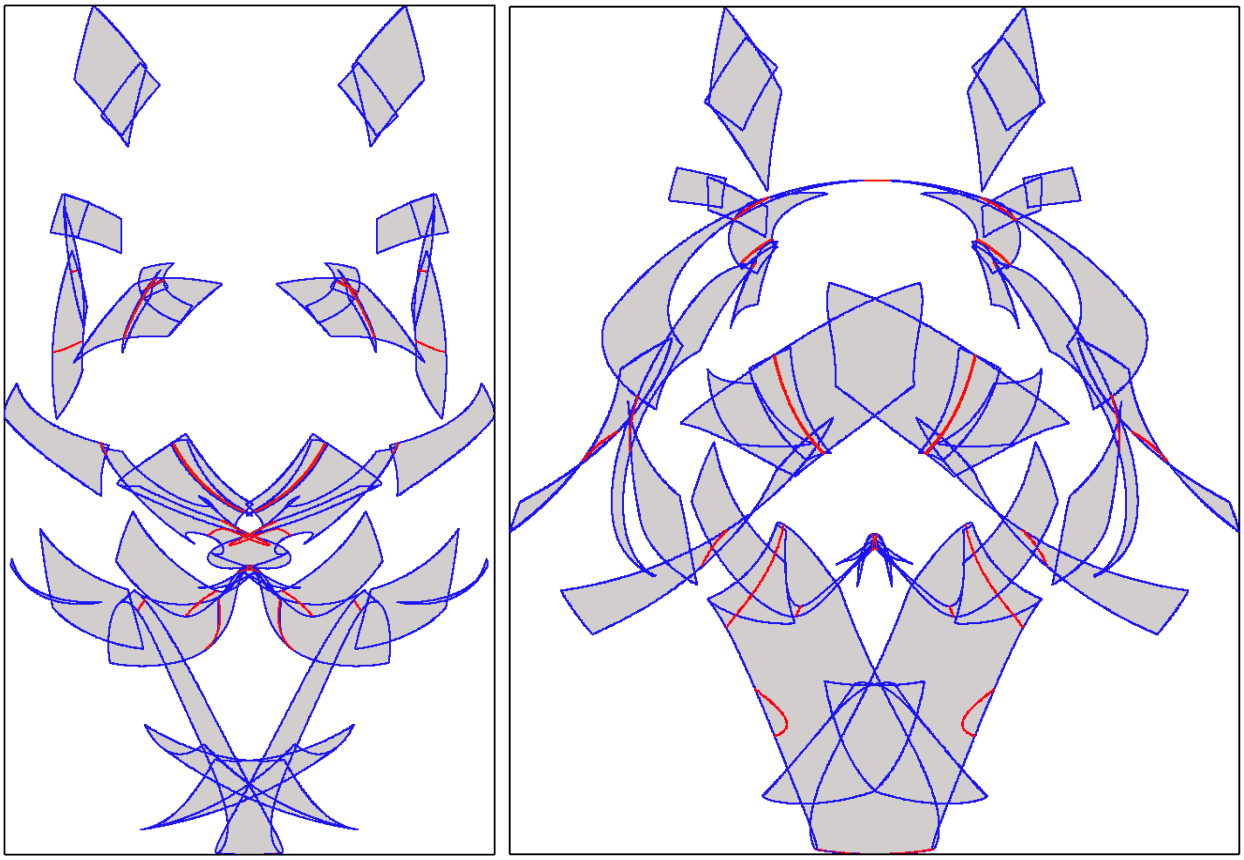

Figure 12: Output portraits of the manipulator in Fig. 11 assuming the geometric parameters mentioned in the text. The angles $\theta_1$ and $\theta_2$ are limited by keeping their cosines to the ranges $[0.5, 0.7]$ (left plot) and $[0.6, 0.8]$ (right plot), with positive sines in both cases. Red and blue curves correspond to forward and inverse singularities, respectively.

The complexity of this mechanism comes from the fact that it involves many links, and all of them move in a highly-coupled manner. This behaviour is apparent from the topology of the mechanism already, but it can be proved through the application of recent Assur Graph Theory tools [38, 39]. On the basis of these observations, we conjecture that the derivation of minimal-degree polynomials describing the singularity set of this manipulator is an extremely difficult task. The computation of such set is even too hard through discretization techniques [40, 41], which define a grid of points in the $\mathcal{U}$ space, solve the inverse kinematics problem for each point, and finally analyze the resulting configurations one-by-one, identifying those that are close to the singularity set. Note that this process boils down to discretizing the $(x, y)$ plane on this mechanism, and that solving the inverse kinematics problem for each position $(x, y)$ is equivalent to finding all configurations of a seven-loop truss [Fig. 11, (b)], which is beyond the capabilities of even the most advanced techniques for position analysis based on characteristic polynomials [42, 43, 44].

Assuming that $P$ is located in position $(0, -1)$ of the frame $O'X'Y'$ of Fig. 11 (a), that all quadrilateral links are squares of side 1, and that all bars are of length 2, except $L$, which is of length $\sqrt{2}$, the method determines the singularity sets shown in Fig. 12. The two plots correspond to two variants of the mechanism that differ on the limits imposed on $\theta_1$ and $\theta_2$ only, which can be modelled by adding a few equations to the system (Appendix A). Note that, in doing so, the configurations where some actuator reaches its limit are considered to be singular, because a loss of mobility occurs in the output link as a consequence.

### *6.3. Performance data*

Table 2 summarizes the main performance data of the method on computing the singularity sets depicted in Figs. 8, 9, 10, and 12. For each figure we provide data relative to each singularity subset considered (using "F" and "I" as a shortcut for the forward and inverse singularity loci), the dimension of the subset (*Dim*), the number of equations ($N_{eq}$) and variables ($N_{var}$) involved in Eq. (17), the $\sigma$ threshold considered, the computation time in seconds, and the number of solution boxes returned by the method ($N_{boxes}$). The two variants of the 15-link mechanism corresponding to the left and right plots of Fig. 12 are indicated as "15-link-a" and "15-link-b", respectively.

| Fig. | Manipulator | Locus/slice | Subset | *Dim* | $N_{eq}$-$N_{var}$ | $\sigma$ | Time (s) | $N_{boxes}$ |
|---|---|---|---|---|---|---|---|---|
| 8 | 3-R<u>R</u>R | Full locus | F | 2 | 29-31 | 0.1 | 2168 | 150538 |
| | | | I | 2 | 29-31 | 0.1 | 1182 | 242185 |
| | | $\theta = -\pi$ | F | 1 | 28-29 | 0.01 | 18 | 2692 |
| | | | I | 1 | 28-29 | 0.01 | 65 | 9652 |
| | | $\theta = -\frac{3\pi}{4}$ | F | 1 | 28-29 | 0.01 | 14 | 1372 |
| | | | I | 1 | 28-29 | 0.01 | 61 | 8828 |
| | | $\theta = -\frac{\pi}{2}$ | F | 1 | 28-29 | 0.01 | 12 | 894 |
| | | | I | 1 | 28-29 | 0.01 | 63 | 8725 |
| | | $\theta = -\frac{\pi}{4}$ | F | 1 | 28-29 | 0.01 | 13 | 1113 |
| | | | I | 1 | 28-29 | 0.01 | 51 | 7748 |
| | | $\theta = 0$ | F | 1 | 28-29 | 0.01 | 17 | 2612 |
| | | | I | 1 | 28-29 | 0.01 | 49 | 7419 |
| | | $\theta = \frac{\pi}{4}$ | F | 1 | 28-29 | 0.01 | 14 | 1658 |
| | | | I | 1 | 28-29 | 0.01 | 46 | 7579 |
| 9 | 3-R<u>R</u>R | Full locus | F | 2 | 29-31 | 0.1 | 2168 | 150538 |
| | | | I | 2 | 29-31 | 0.1 | 1182 | 242185 |
| | | $\alpha_3 = -\frac{\pi}{4}$ | F | 1 | 28-29 | 0.01 | 186 | 22195 |
| | | | I | 1 | 28-29 | 0.01 | 15 | 6655 |
| | | $\alpha_3 = 0$ | F | 1 | 28-29 | 0.01 | 216 | 10158 |
| | | | I | 2 | 28-29 | 0.1 | 489 | 106792 |
| | | $\alpha_3 = \frac{\pi}{4}$ | F | 1 | 28-29 | 0.01 | 198 | 22151 |
| | | | I | 1 | 28-29 | 0.01 | 15 | 6653 |
| | | $\alpha_3 = \frac{\pi}{2}$ | F | 1 | 28-29 | 0.01 | 118 | 23654 |
| | | | I | 1 | 28-29 | 0.01 | 18 | 9851 |
| | | $\alpha_3 = \frac{3\pi}{4}$ | F | 1 | 28-29 | 0.01 | 55 | 13578 |
| | | | I | 1 | 28-29 | 0.01 | 12 | 5885 |
| | | $\alpha_3 = \pi$ | F | 1 | 28-29 | 0.01 | 53 | 11950 |
| | | | I | 2 | 28-29 | 0.1 | 447 | 170170 |
| 10 | 3-<u>R</u>RR | $\theta = -\frac{\pi}{4}$ | F | 1 | 22-23 | 0.01 | 9 | 9276 |
| | | | I | 1 | 22-23 | 0.01 | 59 | 19906 |
| | | $\theta = 0$ | F | 1 | 22-23 | 0.01 | 15 | 14548 |
| | | | I | 1 | 22-23 | 0.01 | 66 | 18917 |
| | | $\theta = \frac{\pi}{4}$ | F | 1 | 22-23 | 0.01 | 10 | 9335 |
| | | | I | 1 | 22-23 | 0.01 | 51 | 19998 |
| 12 | 15-link-a | Full locus | F | 1 | 47-48 | 0.01 | 202 | 5734 |
| | | | I | 1 | 47-48 | 0.01 | 2126 | 117007 |
| | 15-link-b | Full locus | F | 1 | 47-48 | 0.01 | 413 | 3918 |
| | | | I | 1 | 47-48 | 0.01 | 6520 | 117196 |

Table 2: Performance data on the reported examples.

## 7. Conclusions

Despite the maturity of Singularity Analysis, scarce attention has been devoted to the development of numerical algorithms for computing the singularity set of an arbitrary manipulator. Such a gap, which was highlighted in [8] and remained open since then, is partially covered in this paper by providing a method to compute the singularity set of any planar non-redundant manipulator. The method relies on a branch-and-prune strategy whereby an initial box bounding the singularity set is recursively reduced and bisected, producing finer and finer approximations of the set successively, until the accuracy of the result is below a given threshold. The method can isolate the whole singularity set independently of its dimension, with the sole limitations imposed by the curse of dimensionality. Its performance has been illustrated on several examples involving 2- or 3-dimensional C-spaces, both on well-studied manipulators, and on a complex one that would be difficult to analyze through common-practice techniques. The latter is in fact believed to lie among the most difficult mechanisms analyzed so far in the Computational Kinematics literature.

An effort has also been made to provide guidelines on how to represent the singularity set once computed, in order to produce suitable diagrams for the robot designer. On this regard, it has been shown that the set can be easily projected to the input and output spaces to provide global information on the motion capabilities of the manipulator, including the reachable input/output areas, the locations where control or dexterity losses can arise, and a delimitation of regions where manipulator motions can safely be planned. Such diagrams, called *portraits* in the paper, can be further enriched by studying their connectivity if desired, either through the use of well-established tools of local barrier analysis [22], or through recent continuation methods able to trace the singularity-free component of the C-space that is reachable from a given configuration [33, 34].

The natural extension of this research is to deal with the more complex spatial case. Work in this direction is underway already [45, 46], relying on the systematic tools of Screw Theory, and on the singularity classification framework proposed in [8]. Such an extension is under consolidation at the moment, and will be the subject of forthcoming publications [47, 27].

**Acknowledgements**

We thank Josep M. Porta for fruitful discussions around the topic of this paper, and for his help on the implementation of the method. This work has been partially supported by the Spanish Ministry of Education and Science through the I+D project DPI2010-18449, and through a Juan de la Cierva contract supporting the second author. We also want to acknowledge the support received by the CSIC project 2012-50-E-026 letting us build some prototypes of the analyzed mechanisms.

## Appendix A. Modelling joint limits

Mechanical limits on the joints can easily be modelled as equality constraints. Two types of limits need to be treated: those imposed on the linear displacement of a slider joint, and those on the angle rotated by a revolute joint. On the one hand, if $q_i$ is a linear displacement that must satisfy

$$q_i^{min} \leq q_i \leq q_i^{max}, \tag{A.1}$$

note that we can enforce this constraint by setting

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

where $m_i = \frac{1}{2}(q_i^{max} + q_i^{min})$, $h_i = \frac{1}{2}(q_i^{max} - q_i^{min})$, and $d_i$ is a newly-defined auxiliary variable. The values $m_i$ and $h_i$ are the mid-point and half-range of the interval $[q_i^{min}, q_i^{max}]$, and Eq. (A.2) 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 Eq. (A.1) if, and only if, it satisfies Eq. (A.2) for some value of $d_i$. On the other hand, if $q_i$ is a joint angle that must satisfy

$$-\alpha_i \leq q_i \leq \alpha_i, \tag{A.3}$$

then this angle will be represented by its cosine $c_{q_i}$ and its sine $s_{q_i}$ under the proposed formulation. The constraint in Eq. (A.3) is equivalent to $c_{q_i} \geq \cos\alpha_i$, which can be written as

$$c_{q_i} = t_i^2 + \cos\alpha_i, \tag{A.4}$$

where $t_i$ is a new variable that can take any value. Again $q_i$ satisfies Eq. (A.3) if, and only if, it satisfies Eq. (A.4) for some $t_i$.

## References

[1] J. Merlet, Singular configurations of parallel manipulators and Grassmann geometry, International Journal of Robotics Research 8 (5) (1989) 45–56.
[2] P. Ben-Horin, M. Shoham, Singularity condition of six-degree-of-freedom three-legged parallel robots based on Grassmann-Cayley algebra, IEEE Transactions on Robotics 22 (4) (2006) 577–590.
[3] P. S. Donelan, Singularity-theoretic methods in robot kinematics, Robotica 25 (6) (2007) 641–659.
[4] J. Borràs, Singularity-invariant leg rearrangements in Stewart-Gough platforms, Ph.D. thesis, Universitat Politècnica de Catalunya (2011).
[5] C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on Robotics and Automation 6 (3) (1990) 281–290.
[6] D. Zlatanov, R. Fenton, B. Benhabib, Singularity analysis of mechanisms and robots via a motion-space model of the instantaneous kinematics, in: Proc. of the IEEE International Conference on Robotics and Automation, 1994, pp. 980–985.
[7] D. Zlatanov, R. Fenton, B. Benhabib, Singularity analysis of mechanisms and robots via a velocity-equation model of the instantaneous kinematics, in: Proc. of the IEEE International Conference on Robotics and Automation, 1994, pp. 986–991.
[8] D. Zlatanov, Generalized singularity analysis of mechanisms, Ph.D. thesis, University of Toronto (1998).
[9] F. Park, J. Kim, Singularity analysis of closed kinematic chains, ASME Journal of Mechanical Design 121 (1) (1999) 32–38.
[10] D. Zlatanov, I. Bonev, C. Gosselin, Advances in Robot Kinematics: Theory and Applications, Kluwer Academic Publishers, 2002, Ch. Constraint Singularities as C-Space Singularities, pp. 183–192.

[11] D. Zlatanov, I. Bonev, C. Gosselin, Constraint singularities of parallel mechanisms, in: Proc. of the IEEE International Conference on Robotics and Automation, Vol. 1, 2002, pp. 496–502.
[12] O. Ma, J. Angeles, Architecture singularities of parallel manipulators., International Journal of Robotics & Automation 7 (1) (1992) 23–29.
[13] J. Borràs, F. Thomas, C. Torras, Architecture singularities in flagged parallel manipulators, in: Proc. of the IEEE International Conference on Robotics and Automation, 2008, pp. 3844–3850.
[14] B. St-Onge, C. Gosselin, Singularity analysis and representation of the general Gough-Stewart platform, The International Journal of Robotics Research 19 (3) (2000) 271–288.
[15] C. Gosselin, J. Wang, Singularity loci of a special class of spherical three-degree-of-freedom parallel mechanisms with revolute actuators, The International Journal of Robotics Research 21 (7) (2002) 649–659.
[16] I. A. Bonev, Geometric analysis of parallel mechanisms, Ph.D. thesis, Faculté des Sciences et de Génie, Université de Laval (2002).
[17] J. Wang, C. Gosselin, Singularity loci of a special class of spherical 3-DOF parallel mechanisms with prismatic actuators, ASME Journal of Mechanical Design 126 (2) (2004) 319–326.
[18] H. Li, C. Gosselin, M. Richard, B. St-Onge, Analytic form of the six-dimensional singularity locus of the general Gough-Stewart platform, ASME Journal of Mechanical Design 128 (1) (2006) 279–287.
[19] I. A. Bonev, C. M. Gosselin, Analytical determination of the workspace of symmetrical spherical parallel mechanisms, IEEE Transactions on Robotics 22 (5) (2006) 1011–1017.
[20] M. Zein, P. Wenger, D. Chablat, Singular curves in the joint space and cusp points of 3-RPR parallel manipulators, Robotica 25 (6) (2007) 717–724.
[21] P. Wenger, D. Chablat, Kinematic analysis of a class of analytic planar 3-RPR parallel manipulators, in: Proc. of the 5th International Workshop on Computational Kinematics, 2009, pp. 43–50.
[22] E. J. Haug, C.-M. Luh, F. A. Adkins, J.-Y. Wang, Numerical algorithms for mapping boundaries of manipulator workspaces, ASME Journal of Mechanical Design 118 (2) (1996) 228–234.
[23] J. M. Porta, L. Ros, T. Creemers, F. Thomas, Box approximations of planar linkage configuration spaces, ASME Journal of Mechanical Design 129 (4) (2007) 397–405.
[24] J. M. Porta, L. Ros, F. Thomas, A linear relaxation technique for the position analysis of multi-loop linkages, IEEE Transactions on Robotics 25 (2) (2009) 225–239.
[25] J. G. De Jalón, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems, Springer Verlag, 1993.
[26] W. Whiteley, Handbook of discrete and computational geometry, Chapman & Hall/CRC, 2004, Ch. Rigidity and scene analysis, pp. 1327–1354.
[27] O. Bohigas, Numerical computation and avoidance of manipulator singularities, Ph.D. thesis, Universitat Politècnica de Catalunya, to be defended in May 2013.
[28] S. G. Krantz, H. R. Parks, The Implicit Function Theorem: History, Theory and Applications, Birkhäuser, Boston, 2002.
[29] A. Castellet, Solving inverse kinematics problems using an interval method, Ph.D. thesis, Universitat Politècnica de Catalunya (1998).
[30] J. M. Porta, L. Ros, F. Thomas, F. Corcho, J. Cantó, J. J. Pérez, Complete maps of molecular-loop conformational spaces, Journal of Computational Chemistry 28 (13) (2007) 2170–2189.
[31] O. Bohigas, L. Ros, M. Manubens, A complete method for workspace boundary determination on general structure manipulators, IEEE Transactions on Robotics 28 (5) (2012) 993–1006.
[32] E. Macho, O. Altuzarra, C. Pinto, A. Hernandez, Advances in Robot Kinematics: Analysis and Design, Springer, 2008, Ch. Transitions between multiple solutions of the Direct Kinematic Problem, pp. 301–310.
[33] O. Bohigas, M. Henderson, L. Ros, J. M. Porta, A singularity-free path planner for closed-chain manipulators, in: Proc. of the IEEE International Conference on Robotics and Automation, 2012, pp. 2128–2134.
[34] O. Bohigas, M. Henderson, L. Ros, M. Manubens, J. M. Porta, Planning singularity-free paths on closed-chain manipulators, submitted.
[35] J. Sefrioui, C. M. Gosselin, On the quadratic nature of the singularity curves of planar three-degree-of-freedom parallel manipulators, Mechanism and Machine Theory 30 (4) (1995) 533–551.
[36] J. P. Merlet, C. M. Gosselin, N. Mouly, Workspaces of planar parallel manipulators, Mechanism and Machine Theory 33 (1-2) (1998) 7–20.
[37] I. Bonev, C. Gosselin, Singularity loci of planar parallel manipulators with revolute joints, in: Proc. of the 2nd Workshop on Computational Kinematics, 2001, pp. 291–299.
[38] B. Servatius, O. Shai, W. Whiteley, Combinatorial characterization of the Assur graphs from engineering, European Journal of Combinatorics 31 (4) (2010) 1091–1104.
[39] A. Sljoka, O. Shai, W. Whiteley, Checking mobility and decomposition of linkages via pebble game algorithm, in: Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE, 2011.
[40] O. Altuzarra, C. Pinto, R. Avilés, A. Hernández, A practical procedure to analyze singular configurations in closed kinematic chains, IEEE Transactions on Robotics 20 (6) (2004) 929–940.
[41] E. Macho, O. Altuzarra, E. Amezua, A. Hernandez, Obtaining configuration space and singularity maps for parallel manipulators, Mechanism and Machine Theory 44 (11) (2009) 2110–2125.
[42] N. Rojas, F. Thomas, Distance-based position analysis of the three seven-link Assur kinematic chains, Mechanism and Machine Theory 46 (2) (2010) 112–126.
[43] N. Rojas, F. Thomas, On closed-form solutions to the position analysis of Baranov trusses, Mechanism and Machine Theory 50 (2011) 179–196.
[44] N. Rojas, Distance-based formulations for the position analysis of kinematic chains, Ph.D. thesis, Universitat Politècnica de Catalunya (2012).
[45] O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, J. M. Porta, Numerical computation of manipulator singularities, in: Proc. of the IEEE International Conference on Robotics and Automation, 2012, pp. 1351–1358.
[46] O. Bohigas, D. Zlatanov, M. Manubens, L. Ros, On the numerical classification of the singularities of robot manipulators, in: Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE, 2012.
[47] O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, J. M. Porta, A general method for the numerical computation of manipulator singularity sets, submitted.