

USB Proceedings

2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)

Paraninfo Building, University of Zaragoza
Zaragoza, Spain
10 - 13 September, 2019

Organized by

University of Zaragoza, Spain

Sponsored by

The Institute of Electrical and Electronics Engineers (IEEE)
IEEE Industrial Electronics Society (IES)
Aragón Institute for Engineering Research (I3A), Spain

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint or republication permission, write to IEEE Copyrights Manager, IEEE Operations Center, 445 Hoes Lane, Piscataway, NJ 08854. All rights reserved. Copyright ©2019 by IEEE.

IEEE Catalog Number: CFP19ETF-USB
ISBN: 978-1-7281-0302-0

Simulation Tool for Analyzing the Kinetostatic Effects of Singularities in Parallel Robots

Adrián Peidró, José M. Marín, Luis Payá, and Óscar Reinoso
Automation, Robotics and Computer Vision Laboratory
Miguel Hernández University
03202 Elche, Spain
{apeidro, jmarin, lpaya, o.reinoso}@umh.es

Abstract—Singularities produce important changes in the kinetostatic properties of parallel robots, such as the ability to resist external forces with zero actuation torques (or the inability to resist them at all), or losses of control or dexterity. These kinetostatic effects of singularities can be graphically visualized by means of four velocity and force ellipsoids that degenerate when the robot crosses a singularity. This paper presents an educational simulation tool to help students to understand these effects by means of the visualization of the aforementioned ellipsoids, at the same time that the PID control of the robot is simulated under external forces, in order to demonstrate how singularities affect the control of the robot.

Index Terms—Education, Parallel robot, PID control, Simulation, Singularity

I. INTRODUCTION

Parallel robots are composed of many open kinematic chains connected in parallel, and they offer some advantages over traditional serial robots used in factories, such as higher stiffness, higher force-to-weight ratio, and higher dynamic capabilities. For this reason, parallel robots find several applications in industry: vehicle simulators, pick-and-place tasks, packaging, medical and rehabilitation robotics, etc. The importance of parallel robots suggests that they should be taught in undergraduate courses on robotics, but these courses are usually focused only on serial robots due to time limitations.

When teaching and learning serial robotics, simulation and modeling tools are frequently used for helping students to understand several kinematic and dynamic concepts of these robots. Many educational toolboxes and simulators have been developed to study such concepts. Many of these tools are toolboxes or packages of commercial programs such as Matlab or Labview, which are frequently used in engineering education: *Robotics Toolbox* [3], *ARTE* [4], *Robolab* [8], and others [1], [9]. Other authors have developed standalone educational tools that do not depend on commercial programs, for example: *RoboAnalyzer* [13] or *EJS+EjsRL* [7].

While educational simulation tools for serial robotics are very common, those for parallel robots or closed-chain mechanisms are less frequent. This is because serial robots are relatively easy to model for general architectures, using general formulations based on Denavit-Hartenberg parameters, whereas parallel robots have a wider variety of architectures

and it is usually much more difficult to develop general-purpose tools able to simulate general-architecture parallel robots or analyze their kinematic and dynamic problems. One of such existing tools is *GIM* [12], which allows the user to simulate the forward kinematics, and visualize the singularities, workspace, and force/velocity ellipsoids of general-architecture closed chain manipulators. Another tool is *Mech-Analyzer* [6], which allows students to analyze the kinematics and dynamics of four-bar-like mechanisms and other linkages. General-purpose commercial tools for multi-body analysis such as Matlab-SimMechanics or ADAMS can also be used to model and simulate parallel robots with educational purposes [15], but using these tools effectively usually requires a long time while students learn how to model robots with them, a precious time that may be used in other learning tasks [13].

Although there exist some educational tools for analyzing some kinematic and dynamic aspects of parallel robots, none of them is completely suitable for studying the important kinetostatic effects of their singularities. This is because these effects can be better understood while simulating the control of the robots, since this is how real robots are usually governed: one specifies the desired configuration and some controller (e.g., a PID in the simplest case) applies the necessary control torques to try to take the robot to the target configuration. Through this control process, it is easier to understand the kinetostatic effects of singularities and how they affect the control in a real setting. Based on this, this paper presents an educational simulation tool that allows students to analyze these effects realistically, while simulating the PID control of a parallel robot and visualizing four ellipsoids that represent the velocity and force capabilities of the robot at all times.

This paper is organized as follows. Section II introduces four ellipsoids that are useful for analyzing the kinetostatic effects of different types of singularities of parallel robots. Next, Section III presents the developed simulation tool for studying these effects while simulating the PID control of the robot, and Section IV illustrates the use of this tool through some examples. Finally, Section V concludes this paper.

II. SINGULARITIES OF PARALLEL ROBOTS

A parallel robot is a manipulator whose end-effector or gripper is controlled through two or more serial kinematic chains connected in parallel. Generally, the kinematic configuration

Work funded by the Spanish Government under project DPI2016-78361-R.

of the robot is uniquely determined by a pair (\mathbf{x}, \mathbf{q}) , where $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is a vector that contains n parameters that parameterize the position and/or orientation of the end-effector, and $\mathbf{q} = [q_1, \dots, q_n]^T \in \mathbb{R}^n$ is a vector that contains n angular and/or linear displacements of the actuated joints of the robot. n is the number of degrees of freedom (DOF) of the robot. \mathbf{x} and \mathbf{q} are related through n scalar equations:

$$\mathbf{C}(\mathbf{x}, \mathbf{q}) = [C_1(\mathbf{x}, \mathbf{q}), \dots, C_n(\mathbf{x}, \mathbf{q})]^T = \mathbf{0}_{n \times 1} \quad (1)$$

where C_i are constraint functions of (\mathbf{x}, \mathbf{q}) . The forward kinematic problem consists in solving \mathbf{x} from (1) for given \mathbf{q} , whereas the inverse kinematics consists in solving \mathbf{q} from (1) for given \mathbf{x} . Time-differentiating (1) yields a velocity equation:

$$\frac{\partial \mathbf{C}}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{A} \dot{\mathbf{x}} + \mathbf{B} \dot{\mathbf{q}} = \mathbf{0}_{n \times 1} \quad (2)$$

where:

- \mathbf{A} is the $n \times n$ Jacobian matrix composed of the derivatives of functions C_i with respect to variables x_i ,
- \mathbf{B} is the $n \times n$ Jacobian matrix composed of the derivatives of functions C_i with respect to variables q_i ,
- $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$ are the velocities of the end-effector and actuated joints, respectively.

In general, both matrices \mathbf{A} and \mathbf{B} depend on the configuration (\mathbf{x}, \mathbf{q}) of the robot. These two matrices define two types of singularities, which have different effects on the kinetostatic properties of parallel robots [5], as explained next.

A. Type-I Singularities

Type-I singularities are also known as *serial singularities*, and these occur when $\det \mathbf{B} = 0$. Assuming that $\det \mathbf{A} \neq 0$, we can solve $\dot{\mathbf{x}}$ from (2):

$$\dot{\mathbf{x}} = -\mathbf{A}^{-1} \mathbf{B} \dot{\mathbf{q}} = \mathbf{J} \dot{\mathbf{q}}, \quad \mathbf{J} := -\mathbf{A}^{-1} \mathbf{B} \quad (3)$$

where \mathbf{J} is the direct Jacobian matrix of the manipulator. Equation (3) is the solution of the “instantaneous” forward kinematic problem, in which one solves the output velocities $\dot{\mathbf{x}}$ in terms of the input velocities $\dot{\mathbf{q}}$. Note that, if $\det \mathbf{B} = 0$, then $\det \mathbf{J} = 0$. According to (3), the serial singularity condition $\det \mathbf{J} = 0$ has the following effects on the input and output velocities of the manipulator:

- (S1): there exists a kernel $\ker(\mathbf{J})$ of nonzero input velocities $\dot{\mathbf{q}} \neq \mathbf{0}$ that map to $\dot{\mathbf{x}} = \mathbf{0}$, i.e., there are some actuator velocities that do not move the end-effector.
- (S2): the range space of \mathbf{J} is a lower dimensional subspace of \mathbb{R}^n , i.e., some output velocities $\dot{\mathbf{x}}$ are unattainable. Only those velocities in the range space of \mathbf{J} will be achievable. In this situation, it is often said that there is a loss of dexterity or “instantaneous degrees of freedom”.

A well-known method to graphically illustrate property (S2) is by means of a velocity ellipsoid that encodes the information of the output velocities attainable by the manipulator in all possible directions ([14], p. 153). If the input velocities $\dot{\mathbf{q}}$ are varied along a unit sphere (i.e., $\|\dot{\mathbf{q}}\|^2 = 1$), then their images $\dot{\mathbf{x}}$ under the linear transformation \mathbf{J} of (3) describe an

ellipsoid $E_{\dot{\mathbf{x}}}$ which represents the output velocities achievable by the manipulator at the considered configuration. The lengths of the semi-axes of this ellipsoid coincide with the singular values of \mathbf{J} . Roughly speaking, the orientation of this ellipsoid and the relative length of its semi-axes reflect the ability of the manipulator to generate output velocities in different directions. For example: an ellipsoid where all semi-axes have similar lengths will mean that the manipulator is able to generate output velocities in all directions with approximately the same ease. On the contrary, an ellipsoid with a semi-axis much smaller than the others will mean that output velocities in the direction of the smallest semi-axis will be difficult to obtain (higher input velocities will be required). In the limit, when the manipulator reaches a serial singularity, at least one of these semi-axes will have zero length: in that case, the ellipsoid degenerates and the robot is unable to generate output velocities in the directions of the semi-axes with zero length.

In the static equilibrium (zero velocity and acceleration), and neglecting gravity (e.g., if the robot moves in a horizontal plane), the external forces/torques \mathbf{f} applied at the end-effector and the actuation torques/forces $\boldsymbol{\tau}$ applied at the actuated joints (which are necessary to balance \mathbf{f}) are related through \mathbf{J} :

$$\boldsymbol{\tau} = -\mathbf{J}^T \mathbf{f} \quad (4)$$

The minus sign “-” in (4) is due to the fact that \mathbf{f} is an *external* force applied on the end-effector; if this sign is omitted, then \mathbf{f} would represent the force transmitted to the end-effector by the actuators (which should have opposite sign to the external force, in order to balance it). If $\det \mathbf{J} = 0$, similarly to what happens for property (S1), there exists a kernel $\ker(\mathbf{J}^T)$ of nonzero forces $\mathbf{f} \neq \mathbf{0}$ such that they map to $\boldsymbol{\tau} = \mathbf{0}$, i.e., in serial singularities, some external forces can be resisted with zero actuation torques. These external forces that can be resisted with zero actuation effort are directed along the directions of the unachievable output velocities, corresponding to the semi-axes of $E_{\dot{\mathbf{x}}}$ with zero length. This is reasonable: if the end-effector cannot move in a given direction, then any force trying to move it along that direction will be passively resisted by kinematic constraints. Also, this can be easily checked with the Singular Value Decomposition (SVD) of \mathbf{J} , which yields:

$$\mathbf{J} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (5)$$

where \mathbf{U} and \mathbf{V} are orthogonal matrices, and \mathbf{S} is a diagonal matrix with the singular values of \mathbf{J} arranged in decreasing order along its main diagonal. In a serial singularity, the last k singular values in \mathbf{S} are zero ($k \geq 1$), and the last k columns of \mathbf{U} are a basis of the subspace of unattainable output velocities. On the other hand, the SVD of \mathbf{J}^T is:

$$\mathbf{J}^T = \mathbf{V} \mathbf{S} \mathbf{U}^T \quad (6)$$

According to (6), in a serial singularity, the last k columns of \mathbf{U} are also a basis of $\ker(\mathbf{J}^T)$, so these are the directions of external forces that can be resisted with zero actuation torques.

Equation (4) can also be used to define a second ellipsoid $E_{\boldsymbol{\tau}}$, by transforming a sphere of unit forces ($\|\mathbf{f}\|^2 = 1$) under

(4). The image of such a sphere of unit forces under \mathbf{J}^T is another ellipsoid $E_{\boldsymbol{\tau}}$ which represents the actuation torques that are necessary to balance these forces. If the robot adopts a configuration which is a serial singularity, at least one of the semi-axes of $E_{\boldsymbol{\tau}}$ will have zero length, which means that external forces cannot transmit some efforts to the actuators (in particular, they cannot transmit those efforts directed along the semi-axes with zero length). Following a reasoning similar to the one followed in Eqs. (5) and (6), one can check that the semi-axes of $E_{\boldsymbol{\tau}}$ with zero length are also the directions of the input velocities $\dot{\mathbf{q}}$ that do not move the end-effector, which graphically illustrates property (S1).

B. Type-II Singularities

Type-II singularities are also known as *parallel singularities*, and they occur when $\det \mathbf{A} = 0$. Assuming that $\det \mathbf{B} \neq 0$, we can solve $\dot{\mathbf{q}}$ from (2), obtaining:

$$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\dot{\mathbf{x}} = \mathbf{J}_{inv}\dot{\mathbf{x}}, \quad \mathbf{J}_{inv} := -\mathbf{B}^{-1}\mathbf{A} \quad (7)$$

where \mathbf{J}_{inv} is the inverse Jacobian matrix of the manipulator. The previous equation is the solution of the instantaneous (or velocity-level) inverse kinematic problem, where one solves the input velocities $\dot{\mathbf{q}}$ that are necessary to achieve a desired output velocity $\dot{\mathbf{x}}$. Note that $\det \mathbf{J}_{inv} = 0$ if $\det \mathbf{A} = 0$. According to the previous equation, if $\det \mathbf{J}_{inv} = 0$, then the velocity capabilities of the manipulator are affected as follows:

- (P1): there exists a kernel $\ker(\mathbf{J}_{inv})$ of nonzero output velocities $\dot{\mathbf{x}} \neq \mathbf{0}$ that map to $\dot{\mathbf{q}} = \mathbf{0}$. This means that, even if we lock the actuators of the manipulator, its end-effector can infinitesimally move in some directions $\dot{\mathbf{x}} \in \ker(\mathbf{J}_{inv})$. In this situation, it is often said that there is a loss of control, or that the manipulator “gains instantaneous degrees of freedom”, since the end-effector is locally movable even with all actuators locked.
- (P2) the range space of \mathbf{J}_{inv} is a lower-dimensional subspace of \mathbb{R}^n : the input velocities $\dot{\mathbf{q}}$ cannot take any value, only those values that belong to the image of \mathbf{J}_{inv} . Thus, the manipulator will only admit input velocities in some directions, while other directions will be infeasible.

As with serial singularities, it is possible to intuitively illustrate properties (P1) and (P2) by means of two ellipsoids, as explained next. In order to visualize property (P2), we compute the image of a sphere of unit output velocities ($\|\dot{\mathbf{x}}\|^2 = 1$) under \mathbf{J}_{inv} : this image will be an ellipsoid $E_{\dot{\mathbf{q}}}$ of input velocities. Analogously to the ellipsoids analyzed previously, in a parallel singularity, some semi-axes of $E_{\dot{\mathbf{q}}}$ will have zero length: input velocities along these axes will be infeasible.

In the static equilibrium (and neglecting gravity), the relationship between the *external* force \mathbf{f} exerted at the end-effector and the actuation torques $\boldsymbol{\tau}$ that resist this force is:

$$\mathbf{f} = -\mathbf{J}_{inv}^T \boldsymbol{\tau} \quad (8)$$

At a parallel singularity, $\det \mathbf{J}_{inv} = 0$ and there exists a kernel $\ker(\mathbf{J}_{inv}^T)$ of nonzero actuation torques that transmit no force at all to the end-effector. Following similar reasonings

as before (5)-(6), it can be shown that these “unsuccessful” torques are parallel to the semi-axes of $E_{\dot{\mathbf{q}}}$ with zero length.

Equation (8) can also be used to visualize the force capabilities of the manipulator at each configuration, by defining a force ellipsoid $E_{\mathbf{f}}$ which is the image of a sphere of unit actuation torques ($\|\boldsymbol{\tau}\|^2 = 1$) under \mathbf{J}_{inv}^T . This ellipsoid reflects the output forces that the end-effector can exert to its environment at the considered configuration, as a result of the actuation torques. In a parallel singularity, the range space of \mathbf{J}_{inv}^T is a lower-dimensional subspace of \mathbb{R}^n , such that the manipulator is unable to generate (or resist) output forces parallel to the semi-axes of $E_{\mathbf{f}}$ with zero length.

Finally, property (P1) can also be illustrated with ellipsoid $E_{\mathbf{f}}$: its semi-axes with zero length define the directions $\dot{\mathbf{x}}$ along which the end-effector is locally movable even with all actuators locked. This can be understood by following similar reasonings as before (5)-(6), but also by following intuitive reasonings: if the manipulator cannot exert forces to constrain the motion of the end-effector along a given direction, then this end-effector will be able to freely move along that direction, without encountering any resistance.

C. Configuration Space and Singularity Loci

The solution set \mathcal{S} of (1), i.e., the set of all feasible pairs (\mathbf{x}, \mathbf{q}) that satisfy the kinematic constraints (1):

$$\mathcal{S} := \{(\mathbf{x}, \mathbf{q}) : \mathbf{C}(\mathbf{x}, \mathbf{q}) = \mathbf{0}\} \quad (9)$$

can be defined as the *configuration space* of the manipulator. As noted above, the determinants of \mathbf{A} and \mathbf{B} depend on the configuration (\mathbf{x}, \mathbf{q}) . All configurations $(\mathbf{x}, \mathbf{q}) \in \mathcal{S}$ that yield $\det \mathbf{B} = 0$ belong to the *serial singularity locus*, whereas all configurations $(\mathbf{x}, \mathbf{q}) \in \mathcal{S}$ satisfying $\det \mathbf{A} = 0$ belong to the *parallel singularity locus*. The graphical representation of these loci is very useful for planning the movements of the robot (e.g., for defining trajectories that do not cross singularities) or designing the manipulator (e.g., for designing a manipulator with such dimensions that its configuration space has large singularity-free regions). Typically, these singularity loci are projected to the input space \mathbf{q} or the output space \mathbf{x} , since the complete configuration $(\mathbf{x}, \mathbf{q}) \in \mathcal{S}$ of the manipulator usually has too many coordinates to be represented in two- or three-dimensional graphs. For example: a manipulator with $n = 2$ degrees of freedom would have four configuration coordinates (x_1, x_2, q_1, q_2) which would require a four-dimensional space to be represented, whereas the input (q_1, q_2) and output (x_1, x_2) coordinates can be projected and represented in 2D plots. This is the representation used in the simulation tool described later in Section III.

Finally, although the analysis presented so far accounts for most of the singularities of parallel robots, and in practice it usually suffices to safely operate them, it is well known that there may occur other types of singularities that the input-output formulation followed in this paper misses to detect. This is because the analysis used here, based on the formulation by Gosselin and Angeles [5], only considers the input(\mathbf{q})-output(\mathbf{x}) kinematic relationships, omitting other variables

neither classified as inputs nor outputs. One type of such missed singularities are the *Redundant Passive Motion (RPM) Singularities*, which occur when some part of the robot can locally or globally move even after locking all inputs and outputs [16]. An RPM singularity clearly occurs in the five-bar manipulator of Fig. 1 in the following situation: if joint B_1 becomes actuated and A_1 passive, and if $l_1 = l_2$, then placing the end-effector P at A_1 will enable the free rotation of bars l_1 and l_2 (which will remain overlapped) about A_1 , without varying the inputs or outputs of the manipulator. RPM singularities will not be considered in this paper, but the interested reader is referred to [2] for more details on these and other singularities, and numerical methods to compute them.

III. DEVELOPED SIMULATION TOOL

This section describes a simulation tool developed for facilitating the study of the kinetostatic effects of singularities in parallel robots, which were explained in the previous section. The developed tool is a new module of the virtual laboratory PaRoLa (Parallel Robotics Laboratory), which allows the user to study several kinematic and dynamic aspects of parallel mechanisms. Some of the analyses allowed by PaRoLa are: analysis of the different solutions of the forward and inverse kinematic problems, visualization of the serial and parallel singularity loci, and simulation of the control of the robot in order to study singularity crossings between different solutions of the forward kinematics [10], [11]. This tool can be accessed through <http://arvc.umh.es/parola>, which links to a webpage where the user can select the robot to simulate. Currently, the kinetostatic analysis of singularities presented in this paper is available for the five-bar manipulator depicted in Fig. 1.

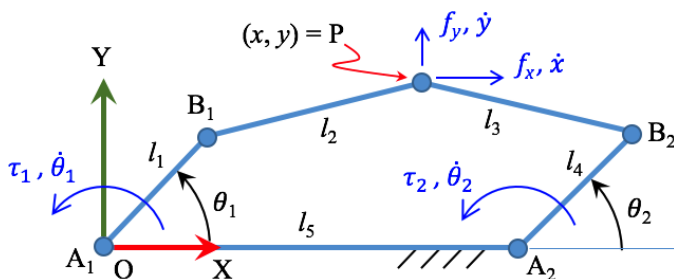


Fig. 1. Five-bar parallel manipulator.

The five-bar mechanism is an $n = 2$ DOF parallel manipulator composed of two RRR serial kinematic chains (“R” denotes “revolute joint”) A_1B_1P and A_2B_2P , whose endpoints are joined at P by means of another R-joint. This manipulator has four articulated mobile bars with lengths $\{l_1, l_2, l_3, l_4\}$, and a fixed bar with length l_5 , connected to the first joint of both chains. The *kinematic inputs* of this robot are the angles $\mathbf{q} = [q_1, q_2]^T = [\theta_1, \theta_2]^T$ between bars $\{l_1, l_4\}$ and the X axis, respectively. Its outputs are the position coordinates $\mathbf{x} = [x_1, x_2]^T = [x, y]^T$ of P with respect to frame OXY (point P is regarded as the end-effector of this manipulator). Angles θ_1 and θ_2 are controlled by means of two actuators connected to joints A_1 and A_2 , which exert torques τ_1 and

τ_2 on bars l_1 and l_4 , respectively (τ_1 and τ_2 are the *dynamic inputs*). Finally, an external force $\mathbf{f} = [f_x, f_y]^T$ is exerted on P. Gravity is neglected, i.e., it is assumed that the robot moves in a horizontal plane. All these input and output variables are illustrated in Fig. 1, which also indicates the input and output velocities, as well as the positive sign of all these variables. For this robot, the constraint functions C_i of (1) are:

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} (x - l_1 c_{\theta_1})^2 + (y - l_1 s_{\theta_1})^2 - l_2^2 \\ (x - l_5 - l_4 c_{\theta_2})^2 + (y - l_4 s_{\theta_2})^2 - l_3^2 \end{bmatrix} \quad (10)$$

where $s_\psi = \sin(\psi)$ and $c_\psi = \cos(\psi)$.

The tool for simulating this robot can be downloaded from <http://arvc.umh.es/parola/5R.html>. This tool is programmed in Java, which will have to be installed and updated in the computer in order to run it. When downloading the .jar simulator and running it, a window like Fig. 2a will pop-up.

This main window has three panels: left panel, central panel, and right panel. The left panel represents the five-bar mechanism, which can be moved by dragging its end-effector P in order to modify its (x, y) output coordinates (this is possible while simulating the inverse kinematics) or by rotating its links l_1 and l_4 in order to modify its input angles θ_1 and θ_2 (this is possible while simulating the forward kinematics). The central panel shows the input space of the manipulator, i.e., a space whose axes represent the input angles θ_1 and θ_2 . In this central panel, the current input angles of the robot are represented as a circle labeled “ROBOT”, which can be dragged while simulating the forward kinematics, in order to simultaneously vary both input coordinates. Both left and central panels also show some red and blue curves which represent the serial and parallel singularity loci, respectively.

Finally, the right panel is a control panel where the user can select which kind of analysis should be performed: forward or inverse kinematic analysis, design analysis of the robot (i.e., varying the dimensions of the robot in order to study how the singularity loci deform as a consequence), or parallel singularity crossing. Some of these functionalities were presented in previous works [10], [11], while the present paper contributes a new functionality which allows the user to study the kinetostatic effects of singularities by means of the four ellipses defined in the previous section. It should be noted that previous tools, such as *GIM* [12], allow the user to visualize the output velocity ($E_{\dot{\mathbf{x}}}$) and force ($E_{\mathbf{f}}$) ellipsoids, while the tool presented in this paper also represents ellipsoids $E_{\dot{\mathbf{q}}}$ and $E_{\boldsymbol{\tau}}$, providing a complete portrait of the input-output velocity and force capabilities of the manipulator at all times.

As next sections will explain, the tool allows for the analysis of kinetostatic effects of singularities from two points of view: from a purely kinematic simulation, or a more realistic control perspective that considers the dynamics of the manipulator.

A. Visualizing the Ellipses While Simulating the Kinematics

The first option to study the kinetostatic effects of singularities using the proposed tool is by simulating the forward or inverse kinematics. To that end, the user must select either the forward or inverse kinematic simulation mode under

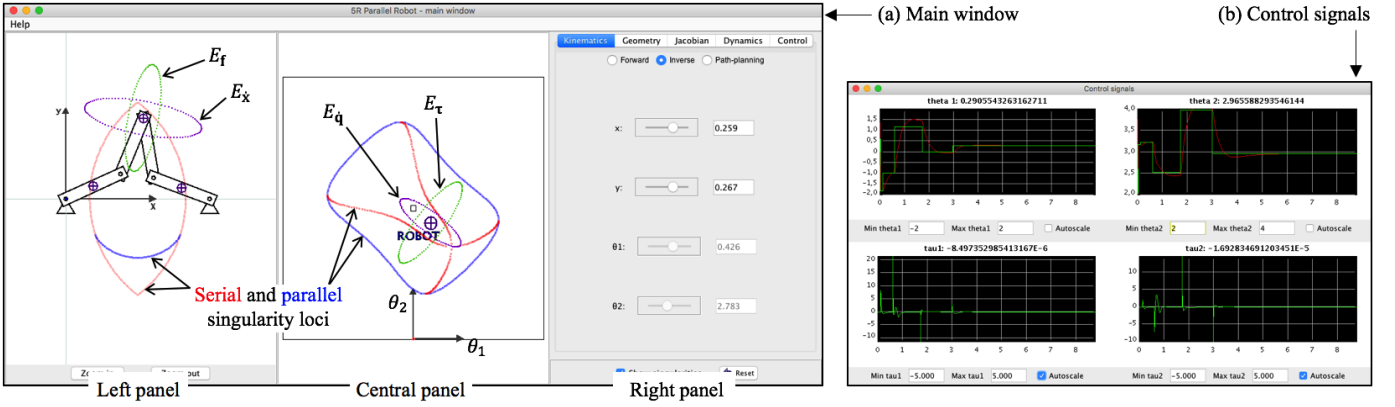


Fig. 2. Simulator for studying the kinetostatic effects of singularities in five-bar parallel manipulators.

the “Kinematics” tab of the control panel. When simulating the forward kinematics, the user can drag the point labeled “ROBOT” in the central panel, modifying in this way the input angles θ_1 and θ_2 . When selecting the inverse kinematics, the user can drag the end-effector P, in order to modify the output variables x and y . In both cases, both left and central panels represent the four ellipsoids defined in Section II. In a robot with $n = 2$ DOF like this, these ellipsoids actually are ellipses.

Ellipses $E_{\dot{x}}$ and E_f are represented in the left panel, and they are centered at point P. Similarly, ellipses $E_{\dot{q}}$ and E_τ are represented in the central panel, and they are centered at the point “ROBOT”, which represents the current values of the input angles, as explained earlier. The velocity ellipses $E_{\dot{x}}$ and $E_{\dot{q}}$ are plotted in purple, whereas the force/torque ellipses E_f and E_τ are plotted in green. Fig. 2a shows an example of non-singular configuration, at which all semiaxes of all four ellipses have nonzero length. Thus, at this configuration, the robot admits input and output velocities in all directions, and it can resist any force with finite and nonzero actuation torques.

By moving the robot under the inverse or forward kinematic modes, as explained above, it is possible to study how these four ellipses transform as the configuration of the robot changes. In particular, it is possible to study how these ellipses degenerate into “segments” when approaching the singularity loci, since some semiaxes have zero length. Note that some configurations may result in the robot apparently “breaking”: this is because some configurations are unreachable.

If the end-effector is placed on a parallel singularity while simulating the inverse kinematics, ellipses E_τ and $E_{\dot{x}}$ will become infinitely large since \mathbf{J}_{inv} becomes singular, whereas ellipses E_f and $E_{\dot{q}}$ degenerate into segments since one of their semiaxes has zero length (see Fig. 3). E_f degenerates into a segment parallel to bars l_2 and l_3 (which remain aligned for parallel singularities), whereas $E_{\dot{q}}$ degenerates into a segment which is tangent to the parallel singularity locus in the (θ_1, θ_2) plane (central panel). At this configuration:

- (P1v) The end-effector admits velocities perpendicular to segment E_f even if we immobilize bars l_1 and l_4 .
- (P1f) The robot cannot resist external forces perpendicular to segment E_f with *finite actuation torques* (but forces

parallel to it can be resisted).

- (P2v) The robot can only perform input movements that are tangent to the parallel singularity locus in the input plane (input velocities in directions perpendicular to segment $E_{\dot{q}}$ are not admitted).
- (P2f) Actuation torques perpendicular to $E_{\dot{q}}$ do not transmit forces to the end-effector.

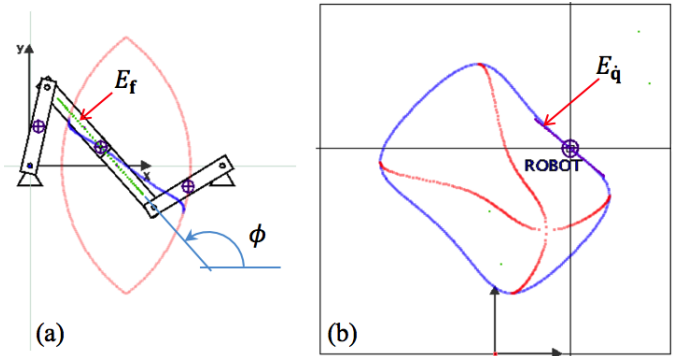


Fig. 3. A parallel singularity.

Similarly, when simulating the forward kinematics, one can place the robot on a serial singularity, observing how ellipses $E_{\dot{q}}$ and E_f become infinitely large, whereas ellipses E_τ and $E_{\dot{x}}$ degenerate into segments since one of their semiaxes has zero length (see Fig. 4). In a serial singularity, one of the two RRR serial kinematic chains is completely extended: in that case, segment $E_{\dot{x}}$ becomes perpendicular to the extended chain, as shown in Fig. 4. In a configuration like this:

- (S1v) Applying input velocities perpendicular to E_τ does not move the end-effector (in this example: these “unsuccessful” input velocities have the form $\dot{\mathbf{q}} = [\dot{\theta}_1, 0]^T$).
- (S1f) No external force \mathbf{f} can transmit to the actuators efforts that are perpendicular to E_τ . In this example, since E_τ is a vertical segment, this means that no external force can transmit horizontal efforts to the actuators, i.e., all external forces will be compensated by actuation torques that satisfy $\boldsymbol{\tau} = [\tau_1, \tau_2]^T = [0, \tau_2]^T$.

- (S2v) The end-effector cannot be moved along the direction perpendicular to segment $E_{\dot{x}}$.
- (S2f) The robot can resist external forces perpendicular to $E_{\dot{x}}$ with zero actuation torques.

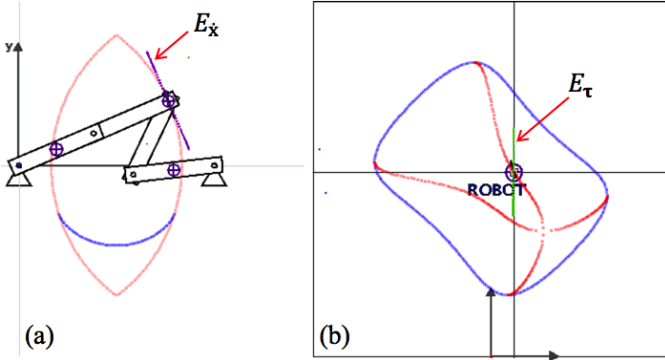


Fig. 4. A serial singularity.

B. Visualizing the Ellipses While Simulating the Control

Although the kinematic simulation presented in the previous subsection allows the user to observe how the four ellipses deform when modifying the configuration of the robot (with the ellipses degenerating into segments at singularities), actually this simulation is not completely realistic since, in reality, one cannot directly modify at will the inputs $\mathbf{q} = [q_1, q_2]^T$ or the outputs $\mathbf{x} = [x, y]^T$, as it is done while simulating the forward or inverse kinematics in the previous subsection, respectively. In real robots, the true inputs by which we can govern the movements of the robot are the actuation torques $\boldsymbol{\tau}$ (or some control voltage, if we also consider the dynamics of the DC motors that drive the actuated joints). Through these torques, we can try to move the robot to the desired configurations, using some control system (e.g., a PID controller). Thus, instead of directly introducing the kinematic inputs or outputs of the robot, as done in Section III-A, we will obtain a more realistic behavior by simulating its control as shown in Fig. 5.

In the control scheme of Fig. 5, we consider an independent PID controller for each actuated joint of the robot. Note that real robots can be controlled more accurately with more sophisticated control techniques, such as Computed Torque Control, but simple decoupled PID controllers will be sufficient for the purposes of the present paper.

The control scheme shown in Fig. 5 is a better approximation of how real robots are governed. Instead of moving the robot by directly introducing its kinematic inputs or outputs (as done in §III-A), we now specify the desired values for the inputs (θ_1^d and θ_2^d) as the setpoints for the controllers. These are compared to the actual inputs of the robot, which can be measured e.g. with encoders. The errors resulting from these comparisons are introduced to PID controllers, which generate actuation torques τ_1 and τ_2 that are applied to the manipulator in order to try to move it to the desired configuration. This setting will offer more realistic simulations that will allow the user to better observe the kinetostatic effects of singularities.

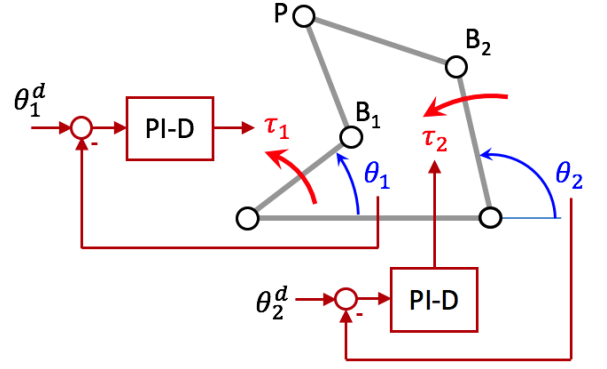


Fig. 5. Decoupled PI-D control scheme.

As for the dynamics of the robot, the simulator considers that all bars are segments with uniform mass density, and it is assumed that joints $\{B_1, B_2, P\}$ have viscous friction proportional to the relative angular velocity between the connecting bars. These friction coefficients, as well as the mass of each bar, can be modified in the “Dynamics” tab of the simulator.

In order to simulate the control, the user accesses the “Control” tab, where the control simulation starts when pressing the button “Start control simulation”. When doing so, the control system will try to drive the inputs of the robot (θ_1 and θ_2) toward their desired setpoints (θ_1^d and θ_2^d), which can be specified using numeric boxes or by clicking directly at the desired values in the central panel of the simulator, which represents plane (θ_1, θ_2) . In the control panel, the user can also modify the Proportional, Integral and Derivative gains for each PID controller of Fig. 5. For each PID, the control law is:

$$\tau_i = P_i(\theta_i^d - \theta_i) + I_i \int (\theta_i^d - \theta_i) dt + D_i(-\dot{\theta}_i) \quad (11)$$

According to (11), the control law actually is a “PI-D”, where the derivative acts only on the controlled variable, omitting the setpoint. In this way, smoother control responses are obtained, since the controller avoids computing the derivative of the setpoints θ_i^d , which are introduced by the user as step changes (which would generate impulses when time-differentiated). The integral term is necessary for eliminating steady-state errors when exerting external forces on the end-effector P.

In the “Control” tab of the control panel, the user can also introduce the external force $\mathbf{f} = [f_x, f_y]^T$ exerted on the end-effector. Finally, in addition to the main window shown in Fig. 2a, there is another window that shows graphs with the time evolution of θ_i^d (green), θ_i (red), and τ_i (green), for $i = 1, 2$ (see Fig. 2b). In these graphs, the quality of the control can be visually assessed (steady-state error, overshoot, etc.).

IV. EXAMPLES

This section demonstrates the usefulness of the proposed simulation tool to analyze some of the kinetostatic effects of singularities introduced in the previous sections, from a control perspective. The values specified next are used for all examples illustrated in this section (these are the default values used

in the simulator). For the geometric design of the robot, the following values are used: $l_1 = l_2 = l_3 = l_4 = 0.2$ m, $l_5 = 0.48$ m. For the dynamic parameters, the used values are: all moving bars have a mass of 0.5 kg, and all passive joints have a viscous friction coefficient of 0.0005 N·m·s/rad. For the PI-D controllers, the gains are: $P_i = 10$, $I_i = 1$, $D_i = 2$.

A. Illustrating Property (S2f)

For this example, no external force is applied on the end-effector at the beginning of the simulation. First, one must select a serial singularity as the setpoint, by clicking on any point of the serial singularity locus in plane (θ_1, θ_2) , in the central panel of the simulator. In the concrete example shown here, the chosen point has the following coordinates: $\theta_1 = 0.4$ rad, and $\theta_2 \approx 3.2636$ rad. This serial singularity is represented in Fig. 4b, whereas the corresponding configuration of the robot is shown in Fig. 4a. In this serial singularity, the A_1B_1P chain of the robot is completely extended.

After waiting some time to allow the PI-D controllers to stabilize the robot at the desired singular configuration, an external force f of 10 N is applied on the end-effector, directed along the semiaxis with zero length of ellipse $E_{\dot{x}}$, which has degenerated into a segment, as shown in Fig. 4a. As discussed in the previous sections, this force should be resisted with zero actuation torques. The force will be introduced so that it tries to move the end-effector towards the interior of the workspace, which is the lens-shaped region enclosed by red serial singularities in Fig. 4a. Thus, the components of the external force to be introduced in the “Control” tab of the simulator are: $f_x = -10 \cos(0.4)$ N and $f_y = -10 \sin(0.4)$ N.

When introducing this force to the simulator, the configuration of the robot may be slightly perturbed initially, but the PI-D controllers will reject this perturbation after some time, stabilizing again the robot at the desired singular configuration while resisting the external force. In steady-state after restoring the desired singular configuration of Fig. 4a, it can be checked in the time plots of the simulator that the actuation torques tend to zero (see Fig. 6), i.e., the external force perpendicular to segment $E_{\dot{x}}$ is resisted with zero actuation torques, which illustrates property (S2f) discussed in previous sections.

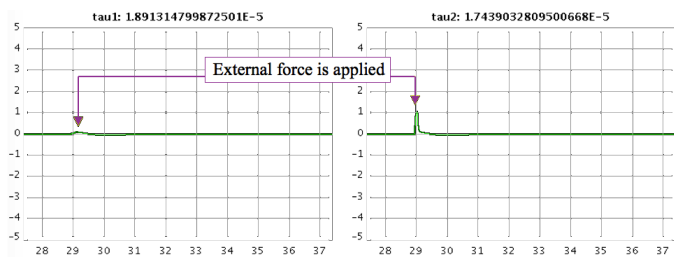


Fig. 6. Perturbation of serial singularity by external force orthogonal to $E_{\dot{x}}$.

B. Illustrating Property (P1f)

For this example, again no external force is applied on the end-effector at the beginning. A parallel singularity is introduced as the setpoint, by clicking on any point of the

parallel singularity locus in plane (θ_1, θ_2) , in the central panel of the simulator. For the concrete example shown in this subsection, the chosen parallel singularity is: $\theta_1 \approx 1.3508$ rad and $\theta_2 \approx 3.6883$ rad, shown in Fig. 3b.

After introducing this setpoint and allowing the PI-D controllers to stabilize the robot at this configuration (Fig. 3a), we can try to apply different external forces to the end-effector, which will perturb the configuration of the robot and will move it away from the desired parallel singularity. The PI-D controllers will try to reject these perturbations and restore the desired singular configuration, while compensating the external forces. However, as discussed in previous sections, the controllers will not be able to compensate any external force: only those parallel to segment E_f shown in Fig. 3a.

For example: starting with the robot stabilized at the parallel singularity of Fig. 3a, an external force of 10N parallel to segment E_f can be applied. In this case, the external force must have an angle ϕ with the X axis, where ϕ is the angle between bars l_2 - l_3 and the X axis (see Fig. 3a), which can be easily obtained from θ_1 and θ_2 as follows:

$$\phi = \text{atan2}(l_1 s_{\theta_1} - l_4 s_{\theta_2}, l_1 c_{\theta_1} - l_5 - l_4 c_{\theta_2}) \approx 2.2967 \text{ rad} \quad (12)$$

Therefore, the external force to be applied to the end-effector will be: $f_x = 10 \cos(\phi) \approx -6.6381$ N and $f_y = 10 \sin(\phi) \approx 7.4790$ N. When applying this force to the robot, starting from the stabilized parallel singularity of Fig. 3a, the robot will be perturbed from this configuration. However, after some transient, the PI-D controllers will be able to reject this perturbation and will restore the configuration of Fig. 3a, reaching again the steady state. By analyzing the graphs of the actuation torques τ_1 and τ_2 in the new steady state after rejecting the previous perturbation, we will obtain a graph similar to Fig. 7, which illustrates the actuation torques necessary for resisting the external force at the desired singular configuration.

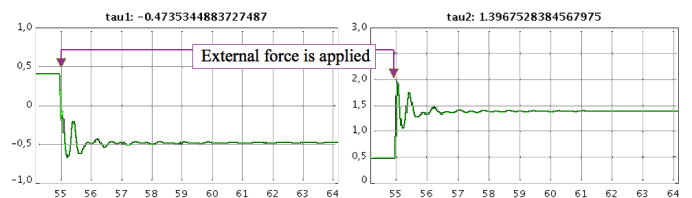


Fig. 7. Perturbation of parallel singularity by force parallel to E_f .

According to Fig. 7, in steady state the external force parallel to bars l_2 - l_3 can be resisted with torques $\tau_1 \approx -0.4735$ N·m and $\tau_2 \approx 1.3967$ N·m. It is possible to validate this control-based simulation using (8). According to (7), J_{inv} is defined as $J_{inv} = -B^{-1}A$, where matrices A and B can be read from the “Jacobian” tab of the simulator for the current configuration of the robot at all times. For the singular configuration of Fig. 3a, the simulator gives the next matrices:

$$A = \begin{bmatrix} 0.265 & -0.299 \\ -0.265 & 0.299 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0649 & 0 \\ 0 & 0.0787 \end{bmatrix} \quad (13)$$

which yield:

$$\mathbf{J}_{inv} = -\mathbf{B}^{-1}\mathbf{A} = \begin{bmatrix} -4.0832 & 4.6071 \\ 3.3672 & -3.7992 \end{bmatrix} \quad (14)$$

In order to validate the actuation torques reported in Fig. 7, which resist the external force at the desired singular configuration, we can solve (8) for this configuration:

$$\begin{bmatrix} -6.6381 \\ 7.4790 \end{bmatrix} = \begin{bmatrix} 4.0832 & -3.3672 \\ -4.6071 & 3.7992 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (15)$$

Since this is a parallel singularity, $\det(\mathbf{J}_{inv}) = 0$ and the previous system has only one independent equation (the second row is about -1.1267 times the first row). By solving τ_2 from any of the two rows of the previous equation, we obtain:

$$\tau_2 = 1.2126 \cdot \tau_1 + 1.97 \quad (16)$$

Thus, any actuation torque $\boldsymbol{\tau} = [\tau_1, \tau_2]^T$ satisfying the previous equation resists the external force. The steady-state torques reported in Fig. 7 satisfy (16), neglecting small roundoff errors.

Next, let us check what happens with the control simulation if the external force is modified, so that it is no longer perfectly parallel to segment E_f of Fig. 3a, but has some component orthogonal to it. For example, if the previous value of $f_x = -6.6381$ N increases to e.g. $f_x = -3.6381$ N, then the robot will be perturbed from the singular configuration of Fig. 3a and the PI-D controllers will try to compensate this force to slowly restore the previous desired configuration. However, in this case, the controllers will be unable to reject this perturbation in finite time: although they try to slowly move the robot toward the singular configuration (while opposing the new external force), the simulator shows that this desired configuration is not completely restored in finite time, and the graphs of the actuation torques show that these torques grow larger as the robot approaches the desired singularity, never stabilizing at constant values (Fig. 8). In this case, the robot gets closer and closer to the singularity but never reaches it, and the actuation torques necessary to continue “pushing” the robot toward the singularity grow continuously, becoming infinite in the limit.

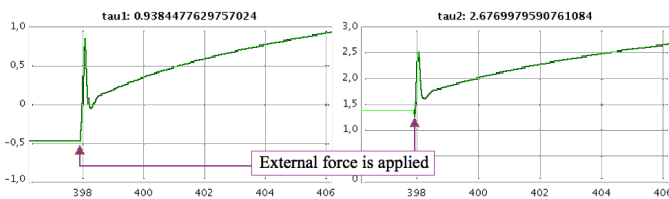


Fig. 8. Perturbation of parallel singularity by force non-parallel to E_f .

This example can also be analyzed with (8), as it was done in (15). However, while (15) was an underdetermined consistent linear system with an infinitude of solutions satisfying (16), this new example yields an inconsistent system since $\text{rank}(-\mathbf{J}_{inv}^T \mathbf{f}) = 1 \neq 2 = \text{rank}(-\mathbf{J}_{inv}^T \mathbf{f})$. Thus, static equilibrium is not possible at the desired parallel singularity, for a force $\mathbf{f} = [-3.6381, 7.4790]^T$ that is not perfectly parallel to the nonzero semiaxis of ellipse E_f . In the control

simulation presented, this is reflected by the fact that the control torques τ_1 and τ_2 grow continuously when trying to approach the desired singular configuration while opposing \mathbf{f} .

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an educational simulation tool for studying the kinetostatic effects of singularities in parallel robots. To that end, the developed tool represents four ellipses that encode the velocity and force capabilities of the robot, both in the input and output spaces. At a singular configuration, some semiaxes of these ellipses have zero length, which denotes the loss of the ability to generate velocities or forces in some directions. These ellipses are combined with the simulation of the PID control of the robot, which allows students to graphically and intuitively analyze these kinetostatic effects, understanding how singularities would affect the control of a real robot. In the future, we plan to extend this simulation to robots with more degrees of freedom, and whose end-effector can both translate and rotate, which poses the problem of mixing translational and angular units.

REFERENCES

- [1] M. Berenguel, F. Rodríguez, J. C. Moreno, J. L. Guzmán, and R. González. Tools and methodologies for teaching robotics in computer science & engineering studies. *Computer Appl. in Engineering Education*, 24(2):202–214, 2016.
- [2] O. Bohigas, M. Manubens, and L. Ros. *Singularities of Robot Mechanisms: Numerical Computation and Avoidance Path Planning*. Springer, 2017.
- [3] P. I. Corke. *Robotics, Vision & Control*. Springer, 2017.
- [4] A. Gil, O. Reinoso, J. M. Marín, L. Paya, and J. Ruiz. Development and deployment of a new robotics toolbox for education. *Computer Appl. in Engineering Education*, 23(3):443–454, 2015.
- [5] C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Trans. on Robotics and Automation*, 6(3):281–290, 1990.
- [6] S. Hampali, C. G. Rajeevlochana, and S. K. Saha. Mechanalyzer: 3d model based mechanism learning software. In *Proceedings of the 14th World Congress in Mechanism and Machine Science*, 2015.
- [7] C. A. Jara, F. A. Candelas, P. Gil, F. Torres, F. Esquembre, and S. Dormido. Ejs+ej srl: An interactive tool for industrial robots simulation, computer vision and remote operation. *Robotics and Autonomous Systems*, 59(6):389 – 401, 2011.
- [8] S. Kucuk and Z. Bingul. An off-line robot simulation toolbox. *Computer Appl. in Engineering Education*, 18(1):41–52, 2010.
- [9] T. J. Mateo and J. M. Andújar. Simulation tool for teaching and learning 3d kinematics workspaces of serial robotic arms with up to 5-dof. *Computer Appl. in Engineering Education*, 20(4):750–761, 2012.
- [10] A. Peidró, A. Gil, J. M. Marín, and Ó. Reinoso. A web-based tool to analyze the kinematics and singularities of parallel robots. *Journal of Intelligent & Robotic Systems*, 81(1):145–163, 2016.
- [11] A. Peidró, O. Reinoso, A. Gil, J. M. Marín, and L. Payá. A virtual laboratory to simulate the control of parallel robots. *IFAC-PapersOnLine*, 48(29):19 – 24, 2015.
- [12] V. Petuya, E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez. Educational software tools for the kinematic analysis of mechanisms. *Computer Appl. in Engineering Education*, 22(1):72–86, 2014.
- [13] C. G. Rajeevlochana and S. K. Saha. Roboanalyzer: 3d model based robotic learning software. In *International Conference on Multi Body Dynamics*, pages 3–13, 2011.
- [14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [15] N. S. Tlale and P. Zhang. Teaching the design of parallel manipulators and their controllers implementing matlab, simulink, simmechanics and cad. *International Journal of Engineering Education*, 21(5):838, 2005.
- [16] D. S. Zlatanov. *Generalized singularity analysis of mechanisms*. University of Toronto, 1999.