A Simulation Tool to Study the Kinematics and Control of 2R$\underline{\text{P}}$R-PR Parallel Robots

Adrián Peidró, Óscar Reinoso, Arturo Gil
José M. Marín, Luis Payá

Systems Engineering and Automation Department
Miguel Hernández University
Avda. de la Universidad s/n, 03202, Elche (Alicante), Spain
e-mail: {apeidro, o.reinoso, arturo.gil, jmarin, lpaya}@umh.es

Abstract: This paper presents an educational simulation tool to analyze the kinematics and dynamic control of 2R$\underline{\text{P}}$R-PR planar parallel robots. The tool, which is very intuitive, allows the robotics students to simulate the forward and inverse kinematic problems of these robots, visualizing the evolution of the solutions to the forward kinematics in the complex space. The tool also allows the students to study the relation between the singularities and the design of the robot, and analyze different methods to change between different assembly modes.

Keywords: Control, Dynamics, Kinematics, Parallel robots, Singularities, Virtual laboratory

1. INTRODUCTION

Parallel robots are robot manipulators in which the end-effector or mobile platform (the part of the robot that usually interacts with the environment) is controlled by two or more legs working in parallel. Due to their closed-chain structure, parallel robots are usually faster and more precise than classical serial robotic arms and have a higher payload-to-weight ratio. However, parallel robots also have smaller workspaces than serial robots, they can adopt singular configurations in which control issues occur, and are generally more difficult to analyze, which makes simulation tools essential to study parallel robots.

Currently, there are many simulation tools to study parallel robots. Petuya et al. (2014) present *GIM*, an educational tool to simulate the forward kinematics, statics and velocity problems of general architecture planar and spatial parallel robot manipulators, as well as to visualize their singularities and workspaces. Ben-Horin et al. (2008) present *SinguLab*, a graphical user interface to study the singularities of Gough-Stewart platforms using Grassmann-Cayley algebra. Other simulators like *DexTAR Sim*[1] and *RoboDK*[2] allow the user to simulate the kinematics and offline programming of 5R and Delta-like parallel robots, respectively. *The CUIK Suite* (Porta et al., 2014) allows the user to solve the kinematic problems of general structure manipulators and visualize their configuration spaces, workspaces and singularity sets. Similarly, the package *Bertini* (Bates et al., 2013) can also be used to solve the kinematic problems of parallel robots, obtaining both the real and imaginary solutions, which can be useful to analyze the kinematics of the robot, as shown in the present paper. Finally, general-purpose commercial programs such as ADAMS and MATLAB/Simulink can also be used to simulate the dynamics and control of parallel robots (Hajimirzaalian et al., 2010; Li and Xu, 2009).

This paper presents a new Java simulation tool to study the kinematics, the singularities, and the dynamic control of a 2-degrees-of-freedom (DOF) planar parallel robot called 2R$\underline{\text{P}}$R-PR. The presented simulator is part of the web-based virtual and remote laboratory PaRoLa (Parallel Robotics Laboratory[3]).

PaRoLa (Peidró et al., 2016, 2015c) is an educational virtual and remote laboratory developed with Easy Java Simulations[4], designed to support and facilitate the study and analysis of parallel robots in robotics courses at undergraduate and master's levels. PaRoLa is composed of a set of Java applets that allow the students to simulate and study the kinematics, singularities, workspace, and dynamics of some parallel manipulators, such as the 5R, the 3$\underline{\text{R}}$RR, and the Delta robots. Also, it allows the students to remotely control a real 5R parallel robot.

The simulator presented in this paper aims at expanding the collection of robots supported by PaRoLa, by allowing the students to simulate the 2R$\underline{\text{P}}$R-PR robot, a 2-DOF parallel robot that can be used as a versatile joint. The presented tool can be used to simulate the forward and inverse kinematic problems of this robot, allowing the user to visualize the evolution of the (complex) solutions to the forward kinematics. Also, this tool can be used to study the relationship between the design of the robot, its singularities, and the ability of the robot to change between different assembly modes without crossing singularities. Finally, the simulator can also be used to simulate the dynamic control of the robot, to study the possibility of changing between different assembly modes by crossing singularities. The presented tool can be freely downloaded from `http://`

[1] http://www.mecademic.com/
[2] http://www.robodk.com/
[3] http://arvc.umh.es/parola
[4] http://fem.um.es/Ejs/

`arvc.umh.es/parola/2rpr_pr.html` (the latest version of Java may be required).

The remainder of this paper is organized as follows. First, Section 2 describes in detail the robot that can be simulated with the proposed tool. Next, Sections 3, 4, and 5 describe how the kinematics, singularities, and dynamic control of the robot can be studied with the presented simulator, respectively. Finally, Section 6 presents the conclusions and the future work.

2. THE 2R$\underline{\text{P}}$R-PR ROBOT

Figure 1 shows the 2R$\underline{\text{P}}$R-PR parallel robot. This robot is a 2-DOF closed-chain planar mechanism composed of a platform connected to a fixed base through one passive slider and two linear actuators (l_1 and l_2) placed in parallel. Due to the passive slider, the platform can only translate along the Y axis (translation y) and rotate in the XY plane (rotation φ). The translation y and rotation φ of the platform can be controlled by means of the linear actuators l_1 and l_2. The geometric design parameters of this robot are $\{a_1, a_2, b_1, b_2\}$. The parameters $\{a_1, a_2\}$ (with $a_1 \neq a_2$) are the positions of the attachment points of the linear actuators to the base along the X axis, whereas the parameters $\{b_1, b_2\}$ (with $b_1, b_2 > 0$) define the size of the platform as shown in Fig. 1.

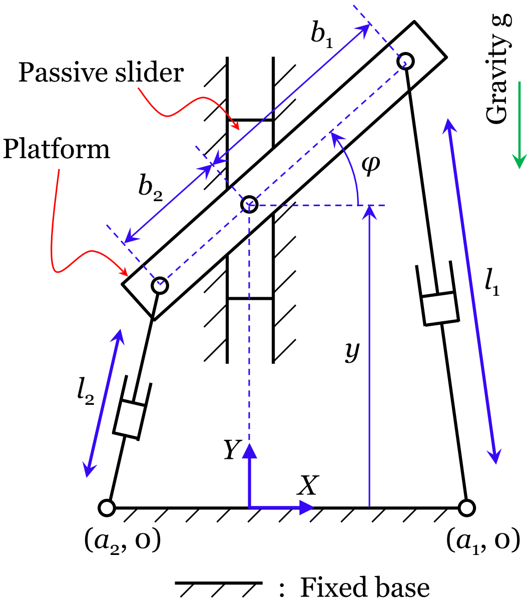


Fig. 1. The 2R$\underline{\text{P}}$R-PR parallel robot.

This parallel mechanism has been studied by some authors. First, this mechanism was proposed by Ridgeway et al. (1996) as a high-stiffness joint for an articulated snake-like robot to explore nuclear facilities. Later, Kong and Gosselin (2002) analyzed the number of different solutions to its forward kinematic problem. Recently, it has been shown that this mechanism can change between different solutions of the forward kinematic problem without crossing singularities (Peidró et al., 2015b). Finally, this mechanism has also been used in a serial-parallel robot for climbing 3D truss structures (Peidro et al., 2015a).

One of the most interesting features of this mechanism is the possibility of using it as a high-stiffness versatile joint, as an alternative to less rigid classical serial joints. Indeed, depending on the pattern of actuation of the actuators l_1 and l_2, the motion of the platform can be a pure translation y, a pure rotation φ, or a combination of them. Thus, used as a joint, this parallel mechanism can behave as a purely prismatic (translational) joint, as a purely revolute (rotational) joint, or as something in between.

In this paper, we present a simulation tool to simulate and study some kinematic and dynamic aspects of this parallel mechanism. The simulation tool, shown in Fig. 2, has six panels p_i ($i = 1, \ldots, 6$):

- p_1: represents the robot in its current configuration, similar to Fig. 1.
- p_2: represents the joint space, i.e. the space of the joint coordinates. The joint coordinates of this robot are the lengths of the linear actuators: l_1 and l_2 (which are represented in the horizontal and vertical axes of the panel p_2, respectively).
- p_3: a control panel in which the user can modify some parameters of the simulation.
- p_4: represents the space of translations y and rotations φ of the platform. The angle φ is represented in the horizontal axis of this panel, whereas the translation y is represented in the vertical axis.
- p_5: represents the complex solutions of the rotation φ, in cylindrical coordinates.
- p_6: represents the complex solutions of the translation y, in rectangular coordinates.

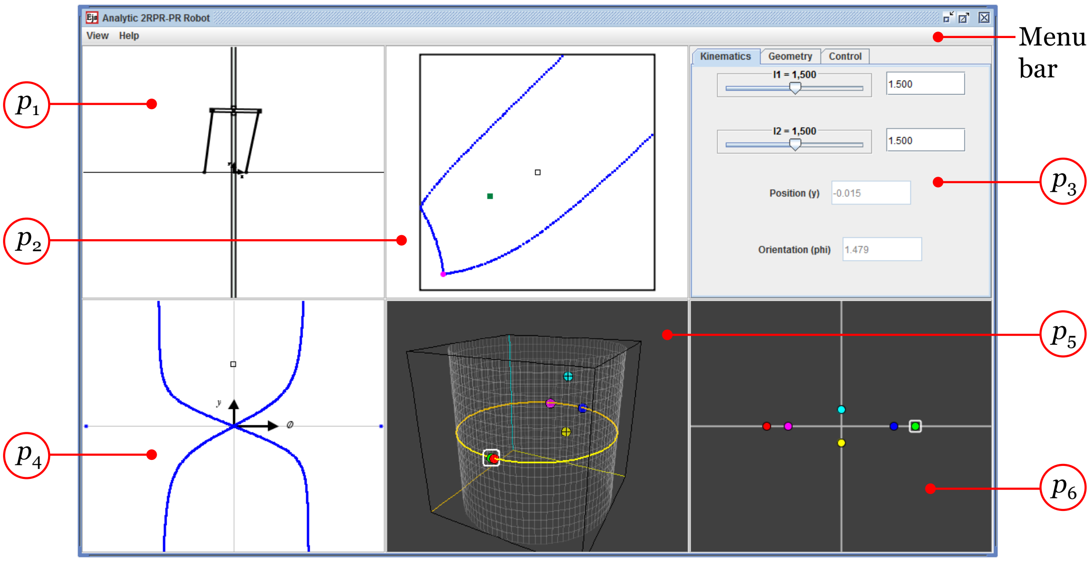
These panels will be described in detail in the next sections, together with the capabilities of the presented tool.

3. SIMULATION OF THE KINEMATICS

This section describes how the forward and inverse kinematic problems of the 2R$\underline{\text{P}}$R-PR robot can be studied with the presented tool.

The forward kinematic problem consists of calculating the position y and orientation φ of the platform in terms of the lengths (l_1, l_2) of the linear actuators. It can be shown that this problem consists of solving a cubic equation and a quadratic equation in sequence, which yields $3 \times 2 = 6$ solutions. However, it has been shown that at most four of these solutions can be real (Kong and Gosselin, 2002), i.e. there can be at most four different real pairs (y, φ) for given lengths (l_1, l_2) (the remaining two solutions are always complex).

To simulate the forward kinematic problem, the user must specify the joint coordinates l_1 and l_2. This can be done using the corresponding boxes and sliders in the "Kinematics" tab of the control panel p_3. Alternatively, the user can click and drag the joint coordinates in the joint space in panel p_2, in which case both joint coordinates can be varied simultaneously (see Fig. 3a). When modifying the value of the joint coordinates, the simulator solves the forward kinematic problem, obtaining the six solutions (y_i, φ_i) ($i = 1, \ldots, 6$), and represents the robot in the configuration associated to one of these six solutions in the panel p_1. The user can select which one



l1 = 1,500 1.500

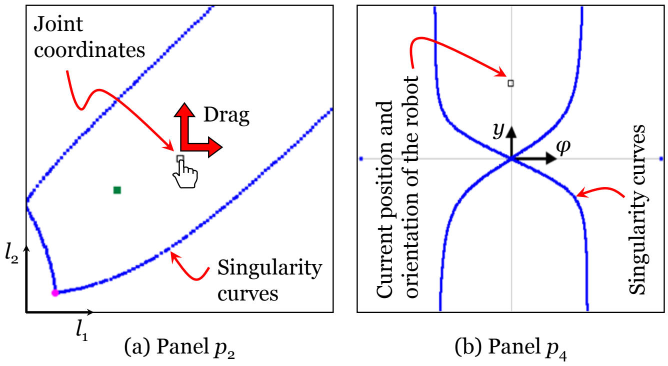
l2 = 1,500 1.500

Position (y) -0.015

Orientation (phi) 1.479

Fig. 2. Main window of the presented tool to simulate the 2R$\underline{\text{P}}$R-PR robot.

of the six solutions is displayed in the panel p_1 using the panels p_5 and p_6, which are described next.



(a) Panel p_2

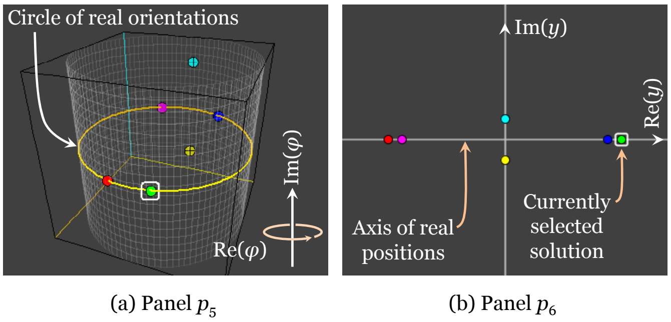
(b) Panel p_4

Fig. 3. (a) To simulate the forward kinematic problem, the user can modify both joint coordinates l_1 and l_2 simultaneously by dragging the small white square in the panel p_2. (b) The current position and orientation of the platform are represented by the tiny square shown in the panel p_4.

When solving (y, φ) in terms of (l_1, l_2), the angle φ, which is in general a complex number, can be calculated as follows (Peidró et al., 2015b):

$$\varphi = \arg\left(\psi + j\sqrt{1-\psi^2}\right) - j\ln\left|\psi + j\sqrt{1-\psi^2}\right| \qquad (1)$$

where $\psi \in \mathbb{C}$ is a root of the aforementioned cubic equation and j is the imaginary unit ($j^2 = -1$). According to the previous equation, the real part of φ is an angle in $[-\pi, \pi]$, whereas the imaginary part of φ takes values in $(-\infty, \infty)$. Thus, the best representation for the complex angle φ is in cylindrical coordinates, in which the angular coordinate is $\mathrm{Re}(\varphi)$, the axial coordinate is $\mathrm{Im}(\varphi)$, and the radial coordinate is an arbitrary positive constant. Regarding the variable y (which in general is also complex), since it is not an angle, it is more appropriate to represent its real and imaginary parts in rectangular axes. Note that the real values of y lie on the horizontal real axis, whereas the real angles φ lie on the circle at zero height of the cylinder, in which $\mathrm{Im}(\varphi) = 0$ (see Fig. 4).



(a) Panel p_5

(b) Panel p_6

Fig. 4. (a) Cylindrical representation of the complex solutions for φ. (b) Rectangular representation of the complex solutions for y.

The panel p_5 represents, in cylindrical coordinates, the six solutions for the angle φ (Fig. 4a). Similarly, the panel p_6 represents, in rectangular coordinates, the six solutions for the coordinate y (Fig. 4b). Recall that, in general, both y and φ can be complex. In these panels, each one of the six different pairs (y_i, φ_i) $(i = 1, \ldots, 6)$ is represented as a circle filled with a different color. The user can click one of these solutions to select which solution should be displayed in the panel p_1 (the currently selected solution is enclosed by a white rounded square, as indicated in Fig. 4b). The selected position and orientation are also shown in the φ-y space in panel p_4, as indicated in Fig. 3b.

Besides permitting the user to choose one among the six solutions, the main purpose of the complex representation of the solutions in the panels p_5 and p_6 is to help the user understand the evolution of the solutions of the forward kinematic problem as the joint coordinates l_1 and l_2 (which are the variables directly controlled) move in the joint space, shown in panel p_2. For example, it can be checked that if the joint coordinates are inside the region R_4 enclosed by the curve of singularities in the joint space, four of the six solutions are real (see Fig. 5a): four angles

φ lie on the circle of real solutions, and the corresponding values of y lie on the horizontal real axis. As the curve of singularities is approached from inside the region R_4, two pairs of real solutions become closer both in the φ and y coordinates (see Fig. 5b), until the singularities are crossed, in which case these solutions "collide" and become complex (see Fig. 5c). Finally, if the joint coordinates are moved outside the region R_4, all the six solutions are non-real (see Fig. 5d). In any case, it can be checked that two of the six solutions always remain complex, as stated earlier [except at a very special singular point of the joint space, at which these two solutions become instantaneously real (Peidró et al., 2015b)].

Finally, the inverse kinematic problem, which consists of calculating the necessary lengths (l_1, l_2) to attain a desired position and orientation (y, ϕ), can also be simulated. In this case, the user should click at the desired position and orientation (y, φ) in the panel p_4. In contrast to the forward kinematic problem, which has four real solutions, the inverse kinematic problem of this robot has a single solution: given a position y and an orientation φ, the linear actuators can only have the following lenths, which can be easily derived from Fig. 1:

$$l_1 = \sqrt{(b_1 \cos\varphi - a_1)^2 + (y + b_1 \sin\varphi)^2} \qquad (2)$$

$$l_2 = \sqrt{(-b_2 \cos\varphi - a_2)^2 + (y - b_2 \sin\varphi)^2} \qquad (3)$$

4. GEOMETRY AND SINGULARITIES

As described in Section 2, the geometry of the robot depends on four design parameters: a_1, a_2, b_1, and b_2. These design parameters determine the shape of the singularity curves in the joint space (panel p_2) and in the φ-y space (panel p_4) of the robot, and the user can modify the value of these four parameters to visualize how the curves of singularities are consequently transformed. The four parameters can be modified in the "Geometry" tab of the control panel (a_1 and a_2 can also be modified dragging the attachment points of the linear actuators to the base in the panel p_1).

The curves shown in the panels p_2 and p_4 are the singularity curves (see Fig. 3), which are dangerous configurations for the robot since the motion of the platform cannot be easily controlled when the robot crosses them. As discussed in the previous section, in the joint space (panel p_2) the singularity curves enclose the region R_4 of pairs (l_1, l_2) for which there are four different real solutions to the forward kinematic problem. Outside the region R_4, all the six solutions to the forward kinematics are complex, which means that the robot cannot be assembled for these combinations of (l_1, l_2) (these combinations are not possible in practice).

Traditionally, the different real solutions to the forward kinematic problem of a parallel robot have been called *assembly modes*. The singularities of parallel robots play a crucial role for changing between different assembly modes of the robot. As discussed in the previous section, for any pair (l_1^*, l_2^*) inside the region R_4, this robot has four different real solutions or assembly modes which we denote by $\{AM_1, AM_2, AM_3, AM_4\}$. We will consider that the robot undergoes a change of assembly mode if, beginning at (l_1^*, l_2^*) with the assembly modes placed in some positions of the complex spaces of panels p_5 and p_6, the robot performs a closed trajectory in the joint space such that, when it returns again to (l_1^*, l_2^*), some of the assembly modes have swapped their positions in the aforementioned complex spaces.

Like in many parallel robots (Peidró et al., 2015c), the assembly mode of the 2R$\underline{\text{P}}$R-PR robot can be changed in two different ways: either crossing singularities (singular transitions) or enclosing special singularities (nonsingular transitions). The first method cannot be properly simulated if we only simulate the forward kinematic problem; the inertial properties of the robot are necessary. Thus, the first method requires including the dynamics in the simulation and will be studied in the next section. The second method, which will be analyzed next, can be simulated by simulating only the forward kinematic problem.

In (Peidró et al., 2015b), it was shown that the assembly mode of the 2R$\underline{\text{P}}$R-PR robot can be changed by enclosing any of two singular points (which we denote by λ_0 and λ_π) at which four assembly modes coalesce. The singular points λ_0 and λ_π are represented in the panel p_2 as a magenta circle and a green square, respectively (see Fig. 6). The user can drag the joint coordinates to perform a closed trajectory that encircles any of these singular points, and observe that this effectively produces a change of assembly mode (see Fig. 6).

The positions of the singularities in the joint space depend on the geometric design parameters of the robot: $\{a_1, a_2, b_1, b_2\}$. When the user modifies the geometry of the robot, both the shape of the singularity curve and the position of the two singular points $\{\lambda_0, \lambda_\pi\}$ change. This is very interesting to study how modifying the design of the robot affects its ability to change its assembly mode by encircling singularities, as explained next. For example, by experimenting with the geometric parameters of the robot, the user can check that, if $a_1 > a_2$, the point λ_π always lies inside the region R_4 enclosed by the singular curve, whereas the point λ_0 lies exactly on the singular curve (see Fig. 7a). This situation is inverted if $a_2 > a_1$ (see Fig. 7b). This means that both points will never be inside the region R_4 simultaneously. Thus, both singular points can never be enclosed simultaneously, because only the points inside the region R_4 can be encircled by real trajectories (see Fig. 7c). Moreover, when $a_1 \approx a_2$, both λ_0 and λ_π lie practically on the singularity curve, which means that neither of these singular points can be enclosed (see Fig. 7d). In that case, the robot is unable to change its assembly mode by enclosing singularities.

5. SIMULATION OF THE DYNAMIC CONTROL

Finally, the dynamic control of the 2R$\underline{\text{P}}$R-PR robot can also be simulated with the presented tool. The simulator assumes that the linear actuators are massless, that the platform has uniformly distributed mass M, and that there exists viscous friction between the fixed base and the passive slider. Also, it is assumed that the gravity acts in the negative direction of the Y axis (see Fig. 1). The user can modify these parameters (or disable gravity) in the "Control" tab of the panel p_3.

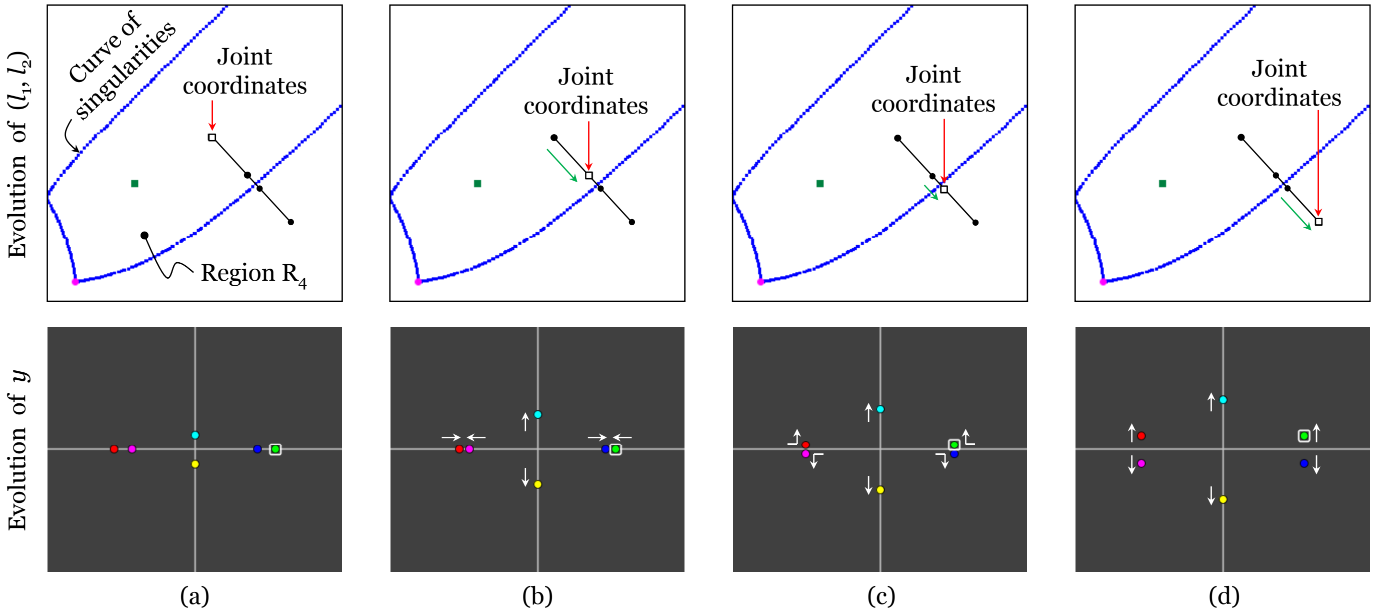


Fig. 5. An example of the evolution of the solutions to the forward kinematic problem as the joint coordinates move in the joint space. (a) Initially, the joint coordinates lie inside the region R_4 enclosed by the singularities, in which there are four real solutions to the forward kinematic problem. (b) As the joint coordinates approach the singularities, two pairs of real solutions get closer. (c) When the singularities are crossed, these two pairs of solutions become complex (they leave the real axis). (c) Out of the region R_4, all the six solutions remain complex.

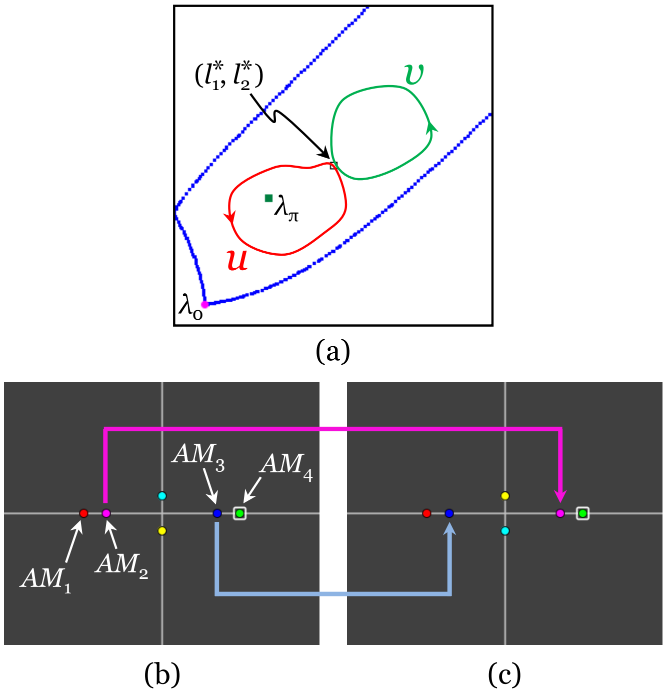


Fig. 6. (a) Two closed trajectories u and v that begin and end at (l_1^*, l_2^*). (b) The solutions to the forward kinematic problem at the beginning of both trajectories. (c) The solutions to the forward kinematic problem at the end of the trajectory u. Since u encloses λ_π, a change of assembly mode occurs because AM_2 and AM_3 swap their positions. On the contrary, the solutions at the end of v are the same as at the beginning.

An independent PD controller, introduced by the user in the "Control" tab, is used to control each linear actuator. The user can specify the desired length for each linear actuator in the control panel, or by clicking directly at the desired lengths in the joint space (panel p_2). The user can visualize the time evolution of the lengths of the linear actuators during the simulation, together with the evolution of the desired or reference lengths, by activating the "View" button in the top menu bar (see Fig. 2).

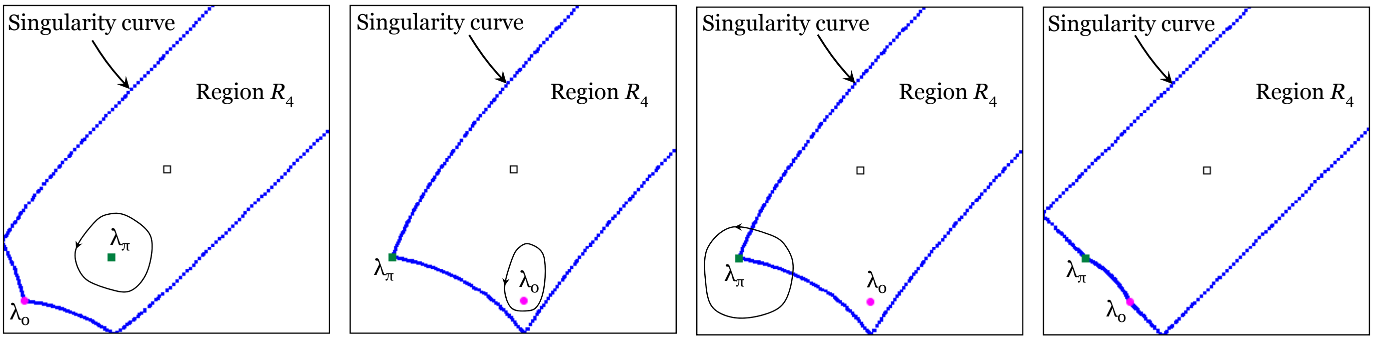
The simulation of the dynamic control of the robot is especially useful to visualize singular transitions, i.e., changes between different assembly modes which occur when crossing singularities. To simulate such a transition, it is necessary to command a trajectory in the joint space which is reflected at the singularity curves (Peidró et al., 2015c). For example, Fig. 8 shows a closed trajectory that begins and ends at point s with different assembly modes at the beginning and at the end. This change of assembly mode is produced by the reflection of the trajectory at the singularity curve. If the trajectory was slightly different and did not undergo a reflection at the singularities, the initial and final assembly modes would be the same.

6. CONCLUSION

This paper has presented an intuitive educational virtual laboratory to simulate 2R$\underline{\text{P}}$R-PR parallel robots and study their kinematics, singularities and dynamic control. In the future, we will add more capabilities to the tool, such as the possibility of testing different path planning algorithms or different advanced control techniques.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministry of Education through grant FPU13/00413, by the Spanish



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

Fig. 7. Variation of the singularities with the geometry of the robot. In all the examples of this figure, the design parameters are: $a_2 = -0.2$, $b_1 = 0.6$, and $b_2 = 0.5$. (a) Example with $a_1 = 0.4$ ($a_1 > a_2$), in which λ_π can be encircled. (b) Example with $a_1 = -1$ ($a_1 < a_2$), in which λ_0 can be encircled. (c) The singular points lying on the curve of singularities cannot be encircled in reality because a real robot cannot execute trajectories that leave the region R_4. (d) An example with $a_1 \approx a_2$: both λ_0 and λ_π lie on the curve of singularities.

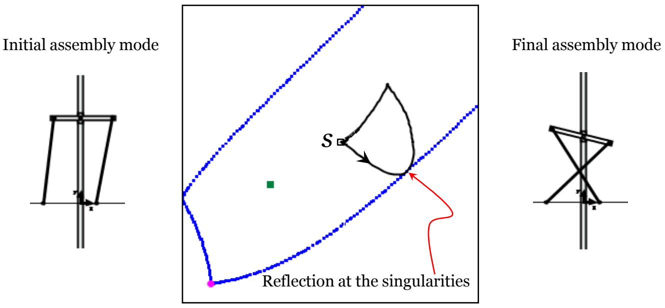


Fig. 8. A trajectory that produces a change of assembly mode by crossing a singularity.

Ministry of Economy through project DPI2013-41557-P, and by the Generalitat Valenciana through project AICO/2015/021.

REFERENCES

Bates, D.J., Hauenstein, J.D., Sommese, A.J., and Wampler, C.W. (2013). *Numerically solving polynomial systems with Bertini*. Software, Environments, and Tools 25. SIAM.

Ben-Horin, P., Shoham, M., Caro, S., Chablat, D., and Wenger, P. (2008). SinguLab - A graphical user interface for the singularity analysis of parallel robots based on Grassmann-Cayley algebra. In: Lenarčič, J., and Wenger, P. (eds.), *Advances in Robot Kinematics: Analysis and Design*, pp. 49–58. Springer Science+Business Media B.V. 2008.

Hajimirzaalian, H., Moosavi, H., and Massah, M. (2010). Dynamics analysis and simulation of parallel robot Stewart platform. In: *Proceedings of the 2nd International Conference on Computer and Automation Engineering*, pp. 472–477. IEEE.

Kong, X. and Gosselin, C.M. (2002). Generation and forward displacement analysis of R$\underline{\text{P}}$R-PR-R$\underline{\text{P}}$R analytic planar parallel manipulators. *ASME Journal of Mechanical Design*, 124(2), pp. 294–300.

Li, Y. and Xu, Q. (2009). Dynamic modeling and robust control of a 3-PRC translational parallel kinematic machine. *Robotics and Computer-Integrated Manufacturing*, 25(3), pp. 630–640.

Peidró, A., Gil, A., Marín, J.M., and Reinoso, Ó. (2016). A web-based tool to analyze the kinematics and singularities of parallel robots. *Journal of Intelligent & Robotic Systems*, 81(1), pp. 145–163.

Peidro, A., Gil, A., Marin, J.M., and Reinoso, O. (2015a). Inverse kinematic analysis of a redundant hybrid climbing robot. *International Journal of Advanced Robotic Systems*, 12:163, pp. 1–16.

Peidró, A., Marín, J.M., Gil, A., and Reinoso, Ó. (2015b). Performing nonsingular transitions between assembly modes in analytic parallel manipulators by enclosing quadruple solutions. *ASME Journal of Mechanical Design*, 137(12), p. 122302.

Peidró, A., Reinoso, O., Gil, A., Marín, J.M., and Payá, L. (2015c). A virtual laboratory to simulate the control of parallel robots. *IFAC-PapersOnLine*, 48(29), pp. 19–24.

Petuya, V., Macho, E., Altuzarra, O., Pinto, C., and Hernandez, A. (2014). Educational software tools for the kinematic analysis of mechanisms. *Comp. Appl. Eng. Education*, 22(1), pp. 72–86.

Porta, J.M., Ros, L., Bohigas, O., Manubens, M., Rosales, C., and Jaillet, L. (2014). The CUIK Suite: analyzing the motion of closed-chain multibody systems. *IEEE Robotics & Automation Magazine*, 21(3), pp. 105–114.

Ridgeway, S.C., Crane, C.D., and Duffy, J. (1996). A forward analysis of a two degree of freedom parallel manipulator. In: Lenarčič, J. and Parenti-Castelli, V., (eds.), *Recent Advances in Robot Kinematics*, pp. 431–440. Kluwer Academic Publishers.