Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-29 (2015) 019–024

A Virtual Laboratory to Simulate the Control of Parallel Robots

Adrián Peidró, Oscar 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 a virtual laboratory to simulate the dynamic control of parallel robots. The objective of the tool is to help robotics students comprehend the different singular and nonsingular methods by which parallel robots can change between different solutions to the forward kinematic problem, and to study how singularities affect the control of the robots when performing these changes. The tool is very intuitive and permits modifying the controller gains of the robots and visualizing the motion of the robot in different spaces.

Keywords: Control, Java, Parallel robots, Simulator, Singularities, Virtual laboratory

1. INTRODUCTION

Parallel robots are closed-chain mechanisms in which the *end-effector* (i.e., the body that carries a tool or a gripper to perform some task) is connected to a fixed base through two or more serial kinematic chains working in parallel. In contrast, serial robots only possess a single open kinematic chain. In general, this closed-loop architecture makes parallel robot faster, stiffer, and more precise than serial robots, which finds applications in surgical robotics, flight simulators and machine-tools, among many other (Merlet 2006).

Besides the previous advantages, parallel robots also have some drawbacks: their workspace is smaller than in serial robots of similar size, their control is more difficult due to the forward kinematic singularities, and their kinematic and dynamic analyses are generally more complex. This higher complexity is due to the closed-loop architecture of the kinematic chains of the robot. To help the robotics students better understand the complex kinematics and dynamics of parallel robots, it is necessary to use simulation tools.

There are many simulation tools that allow the students to easily analyze the kinematics of parallel robots. One of them is GIM (Petuya et al. 2011), a free educational program for intuitively studying the kinematics of general planar and spatial mechanisms. This tool is very flexible, since it permits the student to define mechanisms composed of any number of links connected through different kinds of joints (revolute, prismatic, spherical, etc). After defining the parallel robot, the student can simulate its forward kinematic problem, visualize its workspaces, or perform velocity analyses.

Another tool for studying the kinematics of parallel robots is The CUIK Suite (Porta et al. 2014). This powerful tool solves the position and path planning problems of general closed-chain multibody systems, which can have any architecture defined by the user. In particular, it is especially useful for visualizing the singularities, the configuration space, and the workspaces of parallel robots. This is an advanced research tool that should be used by master's and doctoral students.

Other educational tools are the Java Applets available at http://www.parallemic.org/JavaApplets.html, which can be used to analyze the Euler angles or study the kinematics and singularities of the 5R and 3<u>R</u>RR parallel robots. Another realistic educational simulator for studying the kinematics and singularities of the 5R robot is DexTAR Sim, which can be freely downloaded from http://www.mecademic.com/. This tool can also be used to simulate the off-line programming of this robot, or for programming a real commercially available 5R educational robot. RoboDK (http://www.robodk.com/) is another realistic free tool for simulating the kinematics and off-line programming of industrial robots, including some Delta-like parallel robots.

Although all these tools allow for the simulation of the kinematics of parallel robots (without dynamics), it is important that the students experiment also with the dynamics and control of parallel robots to understand how real (controlled) robots move and how singularities affect the control of the robot. The dynamics of parallel robots can be simulated using general-purpose tools such as ADAMS (Hajimirzaalian et al. 2010, Li and Xu 2009) or the library SimMechanics of MATLAB/Simulink (Koul et al. 2013, Li et al. 2015). SimMechanics can be used to model the dynamics of arbitrary closed-loop mechanisms, which can be combined with the control libraries of Simulink to effectively simulate the dynamic control of parallel robots. Although these tools are powerful, they are proprietary and students must learn how to configure them properly and build the simulations before simulating the control of parallel robots, which may be a difficult task that may hide the actual learning objective. Besides, they do not easily allow for the visualization of singularities at the same time that the robot is simulated, which is of great interest to understand the effect of these singularities in the control of the robot.

To help students understand the different ways to perform changes between different solutions to forward kinematics and the effects of singularities in the dynamic control of parallel robots, we present a virtual laboratory to simulate the control of parallel robots. Our virtual laboratory PAROLA

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.11.207

(PArallel RObotics LAboratory), developed with EJS (http://fem.um.es/Ejs/), consists of several Java Applets which allow the students to intuitively simulate the kinematics of some parallel robots, visualize its singularities and modify the geometry of the robots to study the relation between the design of the robot and the singularities or the workspace (Peidró et al. 2015). Now, the laboratory also allows the students to simulate the control of the 5R and 3RRR parallel robots to comprehend the importance of singularities in the control, simulating the dynamic behavior of real controlled parallel robots.

This paper is organized as follows. Section 2 reviews the forward kinematic problem of parallel robots and the singularities. Section 3 presents the tool developed to simulate the dynamic control of two parallel robots. Then, Section 4 presents some examples of the use of this tool to study some problems related to singularities. Finally, Section 5 presents the conclusions and the future work.

2. REDUCED CONFIGURATION SPACE

This section briefly reviews the forward kinematic problem of parallel robots and the singularity problem, following the model and notation introduced in (Thomas and Wenger 2011). For didactic purposes, in this paper we will deal with parallel robots with only 2 degrees of freedom (DOF); if the robot has $f > 2$ actuators, we will assume that $(f - 2)$ actuators remain constant. If we denote by (ω_1, ω_2) the joint coordinates of the robot (the coordinates that are controlled by actuators, such as motors) and by (s, t) the coordinates that define the position and orientation of the end-effector, then the forward kinematic problem consists in calculating (s, t) in terms of (ω_1, ω_2). In parallel robots, this problem has many different solutions: given a pair (ω_1, ω_2), different pairs (s, t) are obtained, which are the *assembly modes* of the robot.

One way of representing the motion capabilities of a parallel robot in a compact form consists in visualizing the *reduced configuration space*. To obtain the reduced configuration space, the joint coordinates ω_1 and ω_2 are varied in their motion ranges, solving the forward kinematic problem for each value of the joint coordinates. Then, one of the variables that define the position and orientation of the end-effector (t, for example) is plotted versus ω_1 and ω_2, for each of the obtained solutions. Plotting all the solutions yields a surface in the (ω_1, ω_2, t) space, which is the reduced configuration space. For example, Fig. 1 shows a reduced configuration space with spherical shape for didactic purposes (in general, the reduced configuration space has a more complex shape).

The reduced configuration space is very useful to plan the movements of the robot and to understand the effect of the singularities in its motion. For example, assume that the current configuration of the robot corresponds to the point A shown in Fig. 1, and the robot must move toward the configuration B, which lies in the southern hemisphere of the reduced configuration space. Solving the inverse kinematics, we determine that the configuration B can be attained with the joint coordinates (ω_1^B, ω_2^B), which are introduced into the control loop of the robot so that the actuators move the joint coordinates to these values, performing the trajectory T_1 shown in the plane (ω_1, ω_2) in Fig. 1 (which is the plane of joint coordinates, the plane in which the robot is controlled). However, although the desired joint coordinates are attained properly, the robot reaches a wrong configuration: it moves along the northern hemisphere toward the point C, which is the other solution to forward kinematics for the joint coordinates (ω_1^B, ω_2^B).

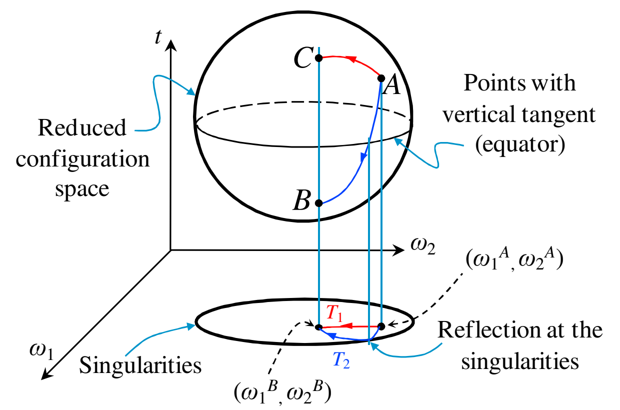

Fig. 1. A spherical reduced configuration space. The points B and C project onto the same point of the (ω_1, ω_2) plane: (ω_1^B, ω_2^B). The trajectory AB is singular.

Instead of performing the trajectory AC along the reduced configuration space, we want the robot to perform the trajectory AB, which is accomplished performing the trajectory T_2 in the (ω_1, ω_2) plane (see Fig. 1). Note that this trajectory requires crossing the points of the reduced configuration space which have a tangent that is parallel to the vertical axis t (i.e., the points of the equator of the sphere, in this example). When such points are projected onto the (ω_1, ω_2) plane, the singularities of the robot are obtained. Note that the trajectory T_2, which produces the correct transition between the configurations A and B, moves first toward the singularities and then suffers a reflection, to move toward the point (ω_1^B, ω_2^B) in the plane (ω_1, ω_2). Thus, this trajectory crosses a singularity. It is not easy to perform this *singular transition* in practice, because the robot is not controllable at singularities. This can be easily understood using the proposed simulation tool, as it will be shown in the examples in Section 4.

According to the previous example, if the robot needs to perform a transition between the points B and C to change between different assembly modes (i.e., to change between different solutions to forward kinematics for the same joint coordinates), the robot must cross a singularity, which requires performing an uncontrollable motion. However, for some robots for which the singularities form cusp points in the (ω_1, ω_2) plane, it is possible to change between different assembly modes without crossing singularities, i.e., by performing controllable movements. This situation is shown in Fig. 2, in which a trajectory T_3 that encloses the cusp point in the (ω_1, ω_2) plane, produces a transition between the points D and E (which are different solutions to the forward kinematic problem for the same joint coordinates) without crossing points of the reduced configuration space that have vertical tangent (singularities). This is possible thanks to the fold present in the reduced configuration space. For studying