Universitat Politècnica de Catalunya

Doctoral Programme

Automatic Control, Robotics and Computer Vision

Ph.D. Thesis

Parallel Robots with Unconventional Joints to Achieve Under-Actuation and Reconfigurability

Patrick Grosch

Advisor:
Federico Thomas

June 2016

Parallel Robots with Unconventional Joints to Achieve Under-Actuation and Reconfigurability

by Patrick Grosch

A thesis submitted to the Universitat Politècnica de Catalunya for the degree of Doctor of Philosophy

Doctoral Programme:
Automatic Control, Robotics and Computer Vision

This thesis has been completed at:
Institut de Robòtica i Informàtica Industrial, CSIC-UPC

Advisor:
Federico Thomas

Dissertation Committee:
Prof.
Prof.
Prof.
Prof.
Prof.

The latest version of this document is available at http://www.iri.upc.edu/people/grosch.

PARALLEL ROBOTS WITH UNCONVENTIONAL JOINTS TO ACHIEVE UNDER-ACTUATION AND RECONFIGURABILITY

Patrick Grosch

Abstract

The aim of the thesis is to define, analyze, and verify through simulations and practical implementations, parallel robots with unconventional joints that allow them to be under-actuated and/or reconfigurable. The new designs will be derived from the:

1. **6S<u>P</u>S** robot (alternatively **6U<u>P</u>S** or **6S<u>P</u>U**, depending on the implementation) when considering the spatial case (*i.e.*, robots with 3 degrees of freedom of rotation and 3 degrees of freedom of translation).

2. **S-3S<u>P</u>S** robot (alternatively **S-3U<u>P</u>S** or **S-3S<u>P</u>U**, depending on the implementation) when considering spherical robots (*i.e.*, robots with 3 degrees of freedom of rotation).

In both cases, we will see how, through certain geometric transformations, some of the standard joints can be replaced by lockable or non-holonomic joints. These substitutions permit reducing the number of legs (and hence the number of actuators needed to control the robot), without losing the robot's ability to bring its mobile platform to any position and orientation (in case of a spatial robot), or to any orientation (in case of a spherical robot), within its workspace.

The expected benefit of these new designs is to obtain parallel robots with:

1. larger working spaces because the possibility of collisions between legs is reduced, and the number of joints (with their intrinsic range limitations) is also reduced;

2. lower weight because the number of actuators and joints is reduced; and

3. lower cost because the number of actuators and controllers is also reduced.

The elimination of an actuator and the introduction of a motion constraint reduces in one the dimension of the space of allowed velocities attainable from a given configuration. As a result, it will be necessary, in general, to plan maneuvers to reach the desired configuration for the moving platform. Therefore, the obtained robots will only be suitable for applications where accuracy is required in the final position and a certain margin of error is acceptable in the generated trajectories.

Acknowledgements

I would like to thank my supervisor, Federico Thomas, for their valuable guidance and constant support.

I am grateful and indebted to Raffaele Di Gregorio (University of Ferrara, Italy) who welcomed me to work with him during a fruitful stay in Ferrara.

My gratitude also goes to Krzysztof Tchoń and Janusz Jakubiakand (Wroclaw University of Technology, Poland).

I very much appreciate the advice, assistance and encouragement that I received from my colleagues at IRI.

Finally, I would also like to acknowledge the support of the Spanish Ministry of Education and Science, under the I+D project DPI2007-60858, Explora and by Italian MIUR funds during my visit to the University of Ferrara.

Contents

List of Figures

List of Tables

Nomenclature

$\boldsymbol{SO}(3)$	Special orthogonal group $\in \boldsymbol{R}^3$.
$\boldsymbol{so}(3)$	Lie algebra of $\boldsymbol{SO}(3)$.
$[\,\cdot\,,\,\cdot\,]$	Commutator of $\boldsymbol{so}(3)$.
$[\,\cdot\,]_\times$	Isomorphism of $\boldsymbol{R}^3$ and $\boldsymbol{so}(3)$.
$\|\cdot\|$	Euclidean norm.
x	Scalar.
X	Point.
$\boldsymbol{x}$	Vector.
$\boldsymbol{O}$	Zeros vector.
$\boldsymbol{I}$	Ones vector.
$\hat{\boldsymbol{x}}$	Unit vector.
$\boldsymbol{I}$	Identity matrix or vector.
$\boldsymbol{O}$	Zero matrix or vector.
$\boldsymbol{X}_{n\times m}$	Matrix of size $n \times m$.
$\boldsymbol{X}^T$	Matrix transpose.
$\mathrm{Tr}\,\boldsymbol{X}$	Trace of matrix.
$\log \boldsymbol{R}$	Logarithm of rotation matrix R.
P	Prismatic joint.
R	Revolute joint.
U	Universal joint.
S	Spherical joint.
nS	Non-holonomic spherical joint.
bR	Lockable revolute joint.
$\underline{\mathbf{X}}$	Actuated joint.
S$\underline{\mathbf{P}}$U	(Spherical)-(Actuated Prismatic)-(Universal) kinematic chain.
nSPU	(Non-holonomic Spherical)-(Prismatic)-(Universal) kinematic chain.
FPM	Fully parallel manipulator, robot or mechanism.
DoF	Degrees-of-freedom.

1 Introduction

1.1 Motivation

A serial robot is a set of rigid bodies, or links, connected in series through actuated joints, which are typically either revolute (*i.e.*, rotational) or prismatic (*i.e.*, translational). One extremity of this serial chain of links is called the base and the other the end-effector. In a parallel robot, the end-effector (also known in this case as the moving platform) is connected to the fixed base through several serial chains. Most of the joints in a parallel robot are not actuated, and many of these passive joints have several degrees of freedom (**DoF**) (*e.g.*, spherical, universal, and planar joints).

Innocenti *et al.* [59] and Dasgupta *et al.* [20] give an introductory overview of the different parallel robot architectures. The origin of parallel robots is attributed to Gough and Stewart. Their work led to what is now known as The Gough-Stewart platform [36, 99]. In 1965, Stewart formalized the concepts that define a parallel robot (Fig. 1.1), although there are previous patents describing equivalent mechanical concepts (*e.g.*, the patent filed in 1931 for a mobile platform to support the audience in a movie theater [48], Fig. 1.2). Since then, many developments and studies have been carried out in this area. The set of articles in references [73] and [18] permit to establish a good starting point where to find solutions to the computation of the direct and inverse kinematics, the configuration space and its singularities, the manipulability and accuracy, etc., of different parallel robots.

A fully parallel robot (**FPM**) is a particular parallel robot that includes as many serial chains (known as legs) as the number of **DoF** of its moving platform. Moreover, in this case, every leg possesses only one actuated joint and no link of the legs can be linked to more than two bodies [76, 91].

The most popular spatial **FPM** is the telescoping-leg hexapod (Stewart-Gough platform) used in most motion simulators. This robot is also said to be a **6S$\underline{\mathbf{P}}$S** robot, meaning that each

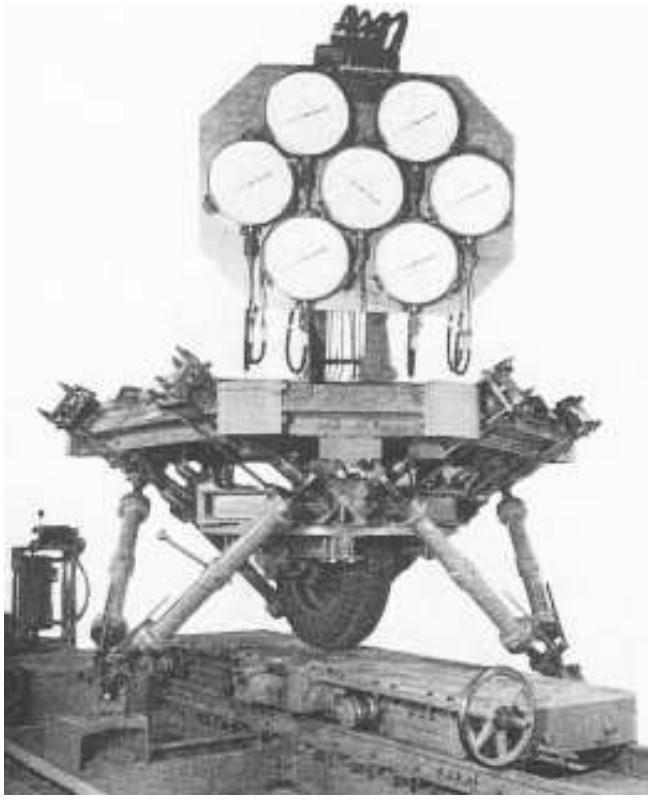


Figure 1.1: Gough platform at birth in 1954 (image from [36, 99]).

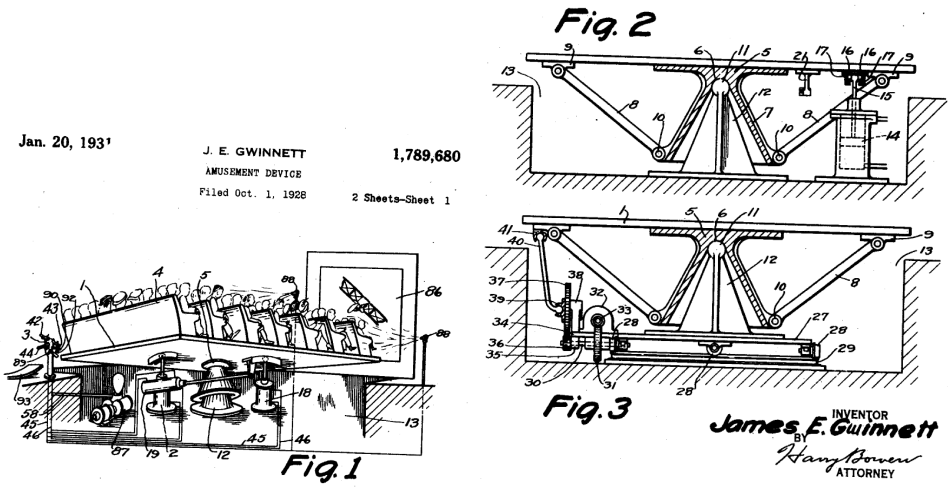


Figure 1.2: Gwinnett Amusement Device 1931 (US Patent 1,789,680, image from [48]).

of its 6 legs consists of a prismatic joint (**P** joint) connected to the fixed base and the moving platform through passive spherical joints (**S** joints). The underline is commonly used to identify

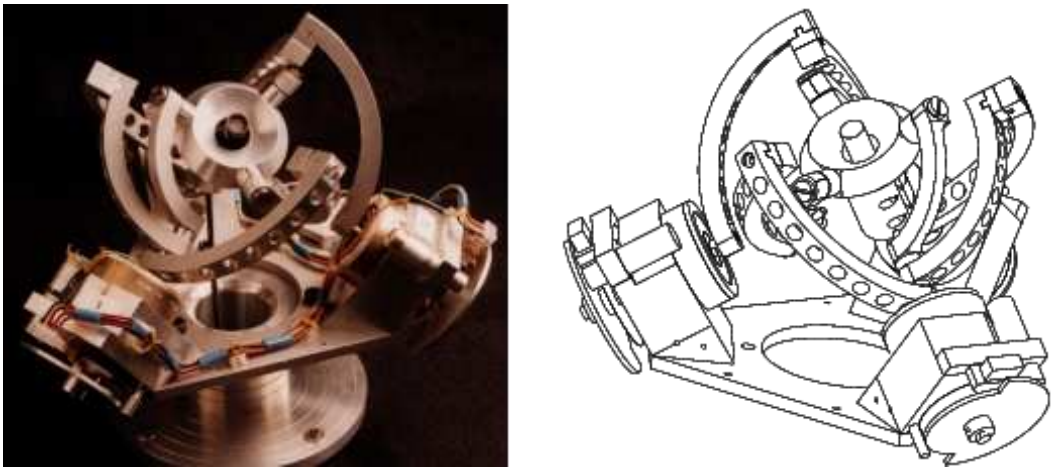


Figure 1.3: The Agile Eye is a **3-DoF 3$\underline{\mathbf{R}}$RR** spherical parallel manipulator (images from [35]).

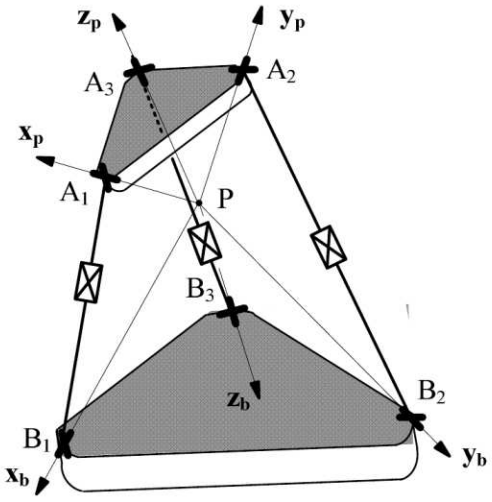


Figure 1.4: The **3U$\underline{\mathbf{P}}$U** spherical wrist manipulator (image from [24]).

the actuated joint.

The most popular spherical **FPM** probably is the **3$\underline{\mathbf{R}}$RR** robot, in which each leg is composed of three revolute joints (**R** joints) whose axes pass through the center of spherical motion [35] (Fig. 1.3). The **3U$\underline{\mathbf{P}}$U** robot also behaves as a spherical robot when the revolute axes of the universal joints (**U** joints) are properly arranged [24] (Fig. 1.4). Nevertheless, the spherical counterpart of the Stewart-Gough platform is the **S-3S$\underline{\mathbf{P}}$S** robot where the moving

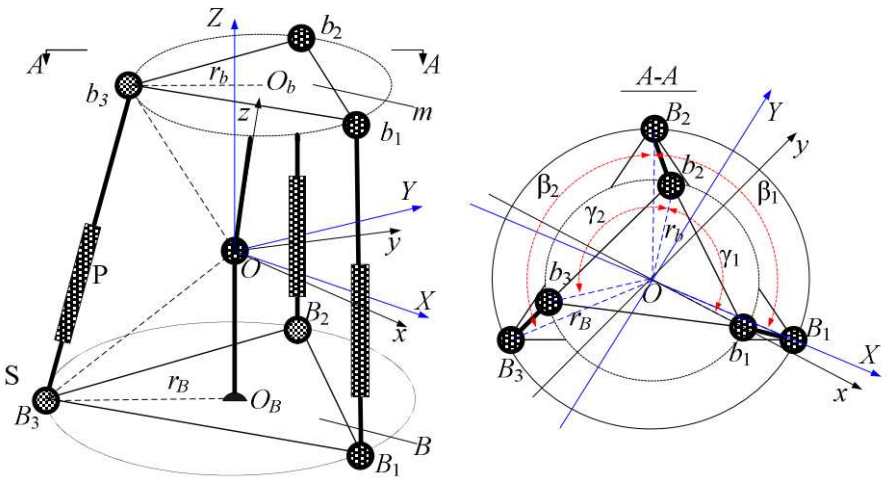


Figure 1.5: The **S-3S<u>P</u>S** robot, spherical counterpart of the Stewart-Gough platform (image from [19]).

platform is articulated with respect to the fixed base through a passive spherical joint and three **S<u>P</u>S** legs connect the base and the platform in parallel [19] (Fig. 1.5).

Throughout this thesis, we will use as a reference of spatial and spherical parallel robots the **6S<u>P</u>S** and **S-3S<u>P</u>S** robots, respectively. In other words, we will limit our study to parallel robots with **S<u>P</u>S** legs. Since these legs contain a redundant **DoF** (the rotation about the axis defined by the centers of the two spherical joints), they can be substituted by **U<u>P</u>S** or **S<u>P</u>U** legs, which is a common practice in most implementations.

FPM are used in applications where accuracy, stiffness or high speeds and accelerations are required [77]. However, one of its main drawbacks is a relatively small workspace compared to their serial counterparts. This is due mainly to the existence of potential collisions between the different elements of the robot and working range of the joints. Another feature which seriously reduces the workspace of fully-parallel manipulators is the existence of singularities. It is well known that two Jacobian matrices appear in the kinematic relations between the joint-rate and the Cartesian-velocity vectors, which are called the "inverse kinematics jacobian" and the "direct kinematics jacobian" matrices. The study of these matrices allows to define the parallel and the serial singularities, respectively [34]. They appear when two solutions of the direct kinematics (respectively inverse kinematics) meet. Parallel singularities generally appear inside the robot's workspace thus highly complicating its control unless the workspace is artificially reduced to leave these singularities out of it.

In this thesis, we propose new parallel robots that can be seen as parallel robots with **S<u>P</u>S**

legs where some of their joints have been replaced by non-holonomic or lockable joints. With this substitutions we will be able to reduce the number of legs, and hence the number of actuators, without reducing the dimension of the robot's configuration space and retaining, at the same time, some of the aforementioned advantages that make parallel robots so interesting. Reducing the number of actuators has important consequences including a reduction in robot's weight and cost, and in the possibility of collisions between legs. This opens the opportunity of enlarging the robot's workspace. Unfortunately, we will see how this reduction in the number of actuators makes these new robots unable to follow arbitrary trajectories in their configuration spaces, thereby increasing the complexity of their motion planning algorithms as they have to perform, in general, maneuvers to reach the desired pose.

The limitation to follow arbitrary trajectories is not a problem for most applications. Robots usually require a high positioning accuracy in some locations defined by the task to be performed, while the path connecting them can run inside predetermined margins. For example, in pick-and-place tasks, while high accuracies are needed in the initial and final configurations, the exact trajectory followed by the robot is unimportant, or is limited to a wide range.

We will see how replacing ordinary joints by non-holonomic or lockable joints leads to under-actuated or on-line reconfigurable robots, respectively. In both cases, the resulting robots have, in general, a larger workspace at a lower cost. The price to pay is a reduction in speed as maneuvers have to be introduced in general to either approximate arbitrary trajectories or to reach arbitrary configurations. The design of the motion planning algorithms to automatically generate these maneuvers is the major challenge faced during the development of this thesis.

To better understand the complexity of the problem, next we briefly introduce the kind of robots generated by the aforementioned substitutions.

1.1.1 Reconfigurable robots with lockable joints

A lockable revolute joint (**bR**) can be seen as a standard **R** plus a binary actuator (usually implemented using a clutch) that enables or disables the mobility of the joint at will.

If a leg is removed in a parallel robot, the moving platform is free to move and the remaining passive joints in the platform are also free to move within certain ranges. Then, if any one of these joints is locked, the system stiffness is recovered, but the moving platform would have fewer **DoF** with respect to that of the initial robot. As an alternative, instead of only locking one of the remaining passive joints, we can lock some of them alternatively. The resulting robot will still have one **DoF** less, but depending on which joints are locked and released, the robot's architecture changes.

If the lockable joints are properly placed, the union of the mobility of all possible architectures that can be generated will equal that of the original parallel robot. In other words, the

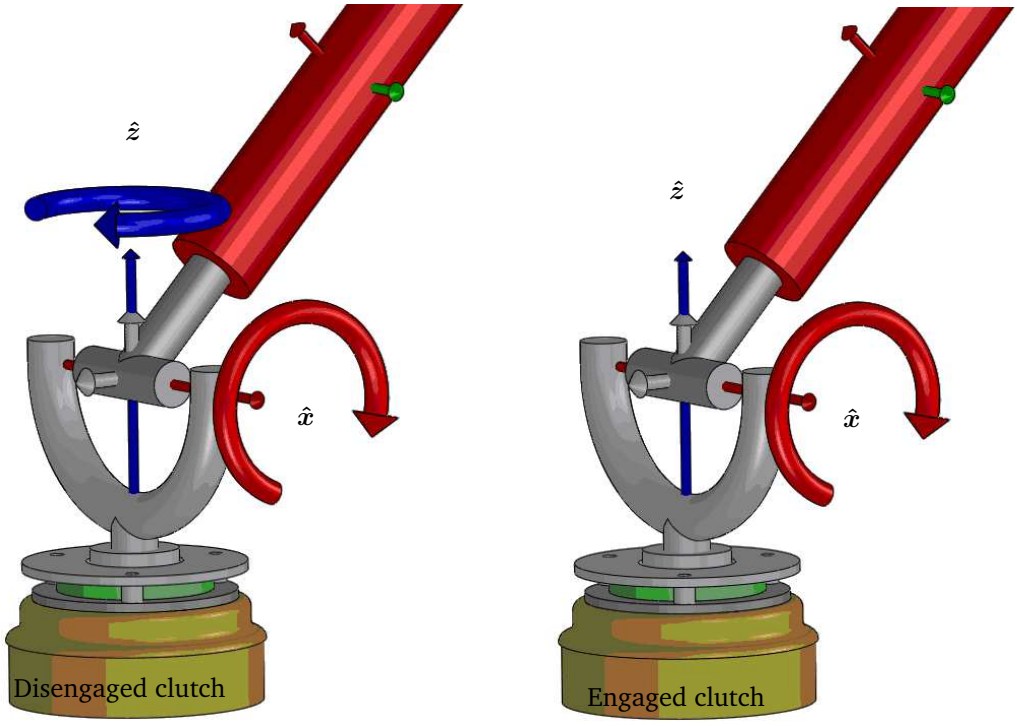


Figure 1.6: The series connection of a lockable revolute joint (**bR**), implemented using a clutch, and a ordinary revolute joint (**R**) behaves as a universal joint (**U**) if the clutch is disengaged (left), or as a revolute joint (**R**) if it is engaged (right).

robot may move in the same workspace as that of the original robot despite having one actuator less. Now, in general, the robot will have to pass through different configurations of locked and released joints to reach a desired configuration. As with non-holonomic joints, maneuvers need to be planned to reach arbitrary configurations.

As an example, consider the series connection of a lockable revolute joint and a ordinary revolute joint whose axes are perpendicular and intersecting in a point (Fig. 1.6). The first joint is lockable which can be easily implemented using an electromechanical clutch, and the second joint is passive. As a consequence, this joint behaves as a **U** if the clutch is disengaged, or as a **R** if it is engaged.

The first part of this thesis is devoted to the study of the parallel robots resulting from replacing the universal joint in their **UPS** legs by the aforementioned **bRR** set of joints. As with the proposed non-holonomic joint explained below, this permits reducing the number of legs. At least one **bR** per each leg substitution is needed to keep the robot away from collapsing, but more are needed to enable full mobility using switching sequences.

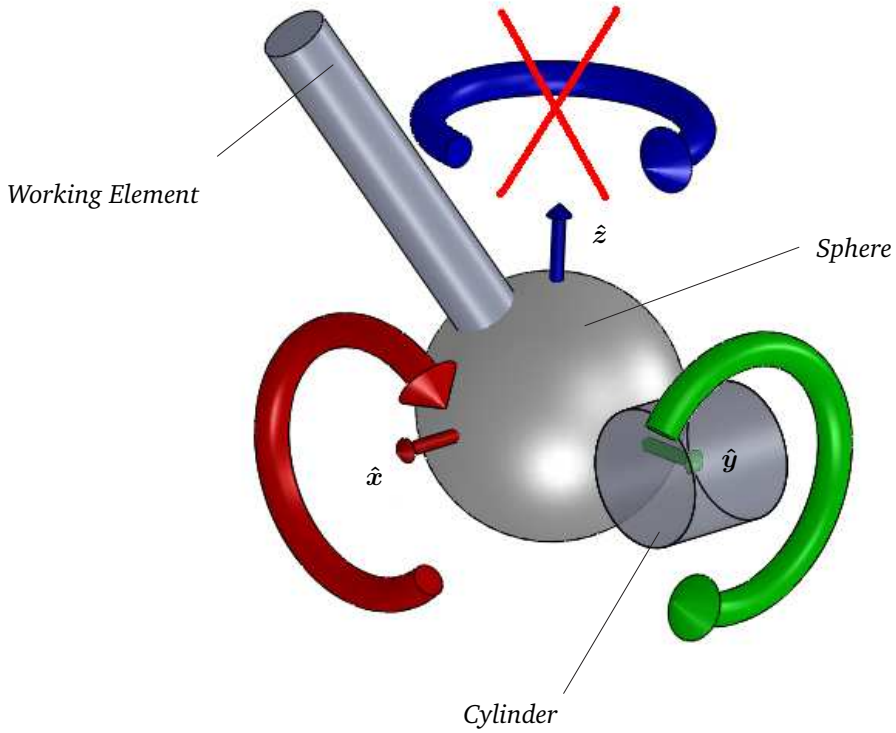


Figure 1.7: Non-holonomic spherical joint (**nS**).

1.1.2 Sub-actuated robots with non-holonomic joints

Let us considered the non-holonomic spherical joint shown in Fig. 1.7. It can be seen as a standard spherical joint where a cylinder, free to rotate about its axis, is put in contact with the sphere. According to the chosen reference frame centered on the sphere, the cylinder is placed so that its rotating axis is parallel to the x-axis, and the contact point with the sphere intersects the y-axis. Then, the possible sphere movements are as follows:

- it can rotate around the x-axis. As the sphere rotates about the x-axis, a rotation of the cylinder is induced due to friction forces at the contact point.

- it can rotate around the y-axis, pivoting around the contact point. As no torque is generated on the cylinder axis, its orientation remains unaltered.

- it cannot rotate about the z-axis. Since the cylinder cannot rotate about the z-axis, the friction forces at the contact point prevents the rotation of the sphere about this axis.

Although the resulting "non-slip" contact only allows the sphere to rotate about the x or the y axis, by combining both rotations, the sphere can attain an equivalent rotation about the z-axis. For example, a rotation of $\pi/2$ radians about the x-axis, followed by a rotation of θ radians about y-axis, and finally a rotation of $-\pi/2$ radians about the x-axis is clearly equivalent to a rotation of θ radians about the z-axis.

If a conventional spherical joint in a parallel robot is substituted by such a joint, the possible orientations between the two bodies this joint connects remain unconstrained, only the relative velocities between them are limited because of this substitution. In other words, only the trajectories connecting two arbitrary orientations between the two connected bodies are constrained. This non-holonomic constrain also introduces reaction forces that can be used to reduce the number of actuators, thus allowing designs having less actuators than the dimension of their configuration spaces.

The second part of this thesis is devoted to the study of the parallel robots resulting from replacing the spherical joint in their **SPU** legs by the aforementioned non-holonomic joint. This permits reducing the number of legs, one per each substitution.

1.2 Previous work

To the best of our knowledge, at the beginning of the development of this thesis, there were no previous results on under-actuated or reconfigurable parallel robots using non-holonomic or lockable joints, respectively. Then, in absence of directly connected previous works, next we summarize those ones that have been used, as a source of inspiration, during the development of this thesis.

1.2.1 Precursors of robots with lockable joints

The use of lockable joints in robot designs is not common. A very small number of examples can be found in the literature which can be categorized into the two groups described below.

One group uses lockable revolute joints that can be engaged with a single motor common to all joints. Then, joints can be switched between locked or actuated state. It is a good strategy to meet the design requirements of minimum weight by minimizing the number of motors and the battery size. Within this group, we can find serial robots like the LARM clutched arm described by Gu and Ceccarelli in [46, 47] (see Fig. 1.8), or like the hyper-redundant snake-like robot presented by Zhu in [107] (see Fig. 1.9), which uses gears for transmission between modules. Another two examples of hiper-redundant serial robots with lockable or clutched revolute joints are the uni-drive modular robot [64] and the 3D-Trunk [87]. The first one, in contrast to the

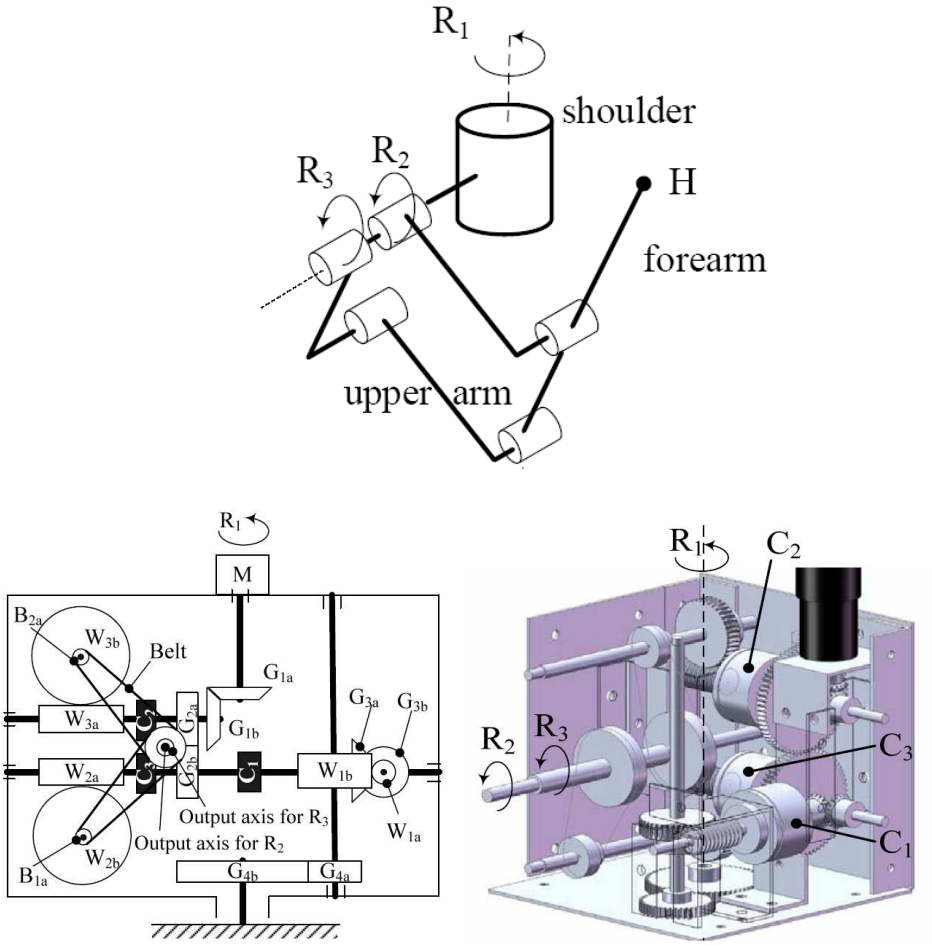


Figure 1.8: Example of serial robot with lockable revolute joints and only one motor. LARM clutched arm conceptual design (top) and mechanical design (bottom) (images from [46] and [47]).

one presented by Zhu, uses a flexible transmission shaft with clutches at each point of actuation as shown in Fig. 1.10(a). The second one, depicted in Fig. 1.10(b), exploits the use of two pairs of opposed tendons to generate torsion on all joints and, by unlocking the n^{th} clutch, to generate a rotation on the n^{th} link.

The other group is represented by the serial robots using linear lockable links. This device allows changing the robot's link lengths and the orientation of the revolute pairs axis. They were first proposed by Aghili and Parsa [1, 2] [see Fig. 1.11 (top)]. This brings two main advantages: it is possible to on-line expand the robot's workspace [see Fig. 1.11 (bottom)], and to reduce the robot's dimensions for transportation.

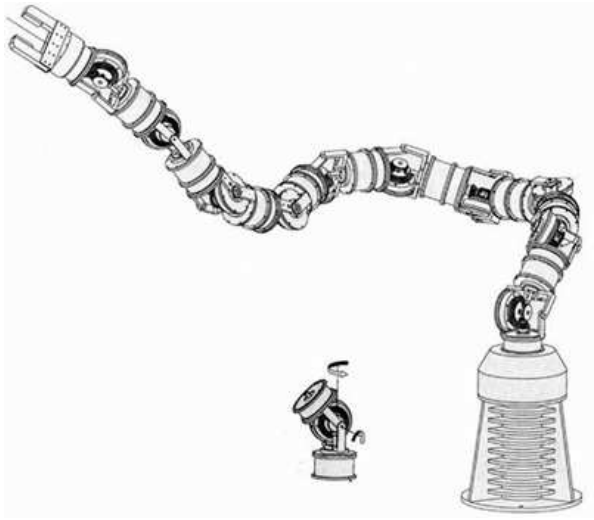


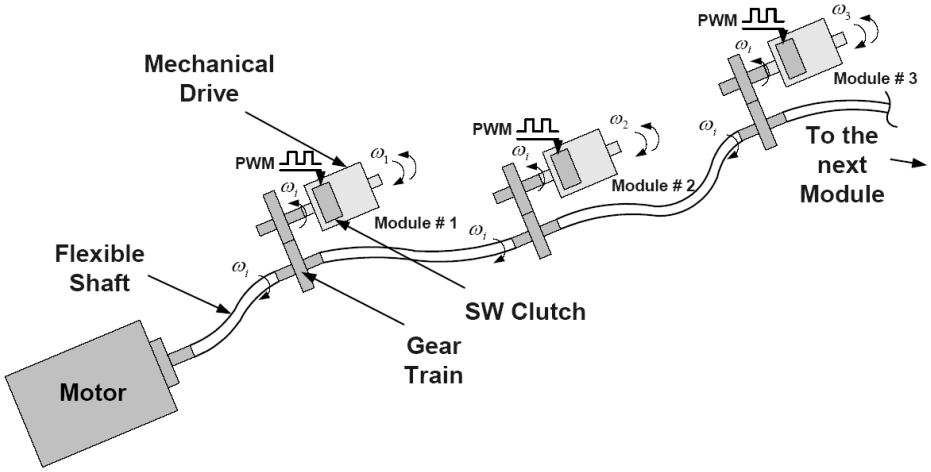
Figure 1.9: Example of serial robot with lockable revolute joints and only one motor. Modular 20-joints hyper-redundant serial robot propose by Zhu (image from [107]).

1.2.2 Precursors of robots with non-holonomic joints

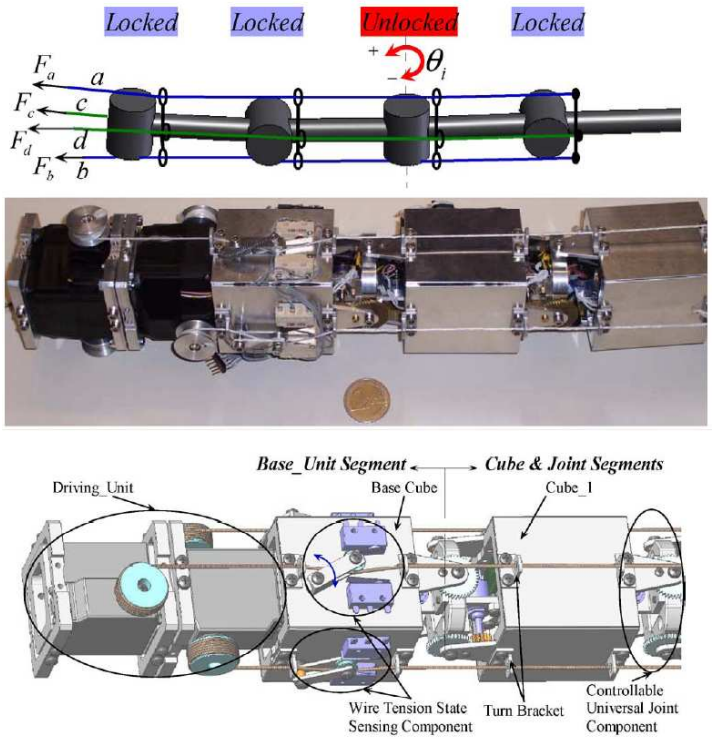
We have just seen how designing a robot manipulator with fewer actuators than the dimension of its configuration space —to reduce bulk, weight and cost— becomes feasible by introducing lockable joints. Now, we will show how this is also possible by introducing non-holonomic mechanical elements. Unfortunately, as with lockable joints, the mechanical advantages of these non-holonomic designs are usually darkened by the complexity of their control. Considerable effort has been made to clarify different aspects of non-holonomic mechanical systems [6]. A challenge in control of these systems results from a limited applicability of the feedback control, discovered by Brockett [10] and Lizárraga [72].

The joints of standard robots, either serial or parallel, implement lower kinematic pairs. An alternative to these joints are non-holonomic joints, a mechanical concept probably used for the first time in [67], which can be implemented using convex bodies rolling on spherical surfaces. Two kinds of contacts have been considered: marble rolling, when the convex body can freely roll in contact with the sphere without slipping [8], and rubber rolling, when the convex body satisfies additionally a no-twist condition [68].

In the practical implementations of non-holonomic joints, the rolling convex body is usually a disk implementing a marble rolling contact with the sphere. If the disk rolls upon the interior surface of a spherical shell, the resulting joint is said to implement the Suslov constraint [104]. Alternatively, if the contact is performed on the outer surface of the spherical shell, the resulting



(a) Top.



(b) Bottom.

Figure 1.10: Examples of two serial robots with lockable revolute joints and only one motor. Schematics of the uni-drive modular robot propose by Karbasi (top) (image from [64]), and the 3D-Trunk wire-drive system proposed by Ning and Worgotter (bottom) (image from [87]).

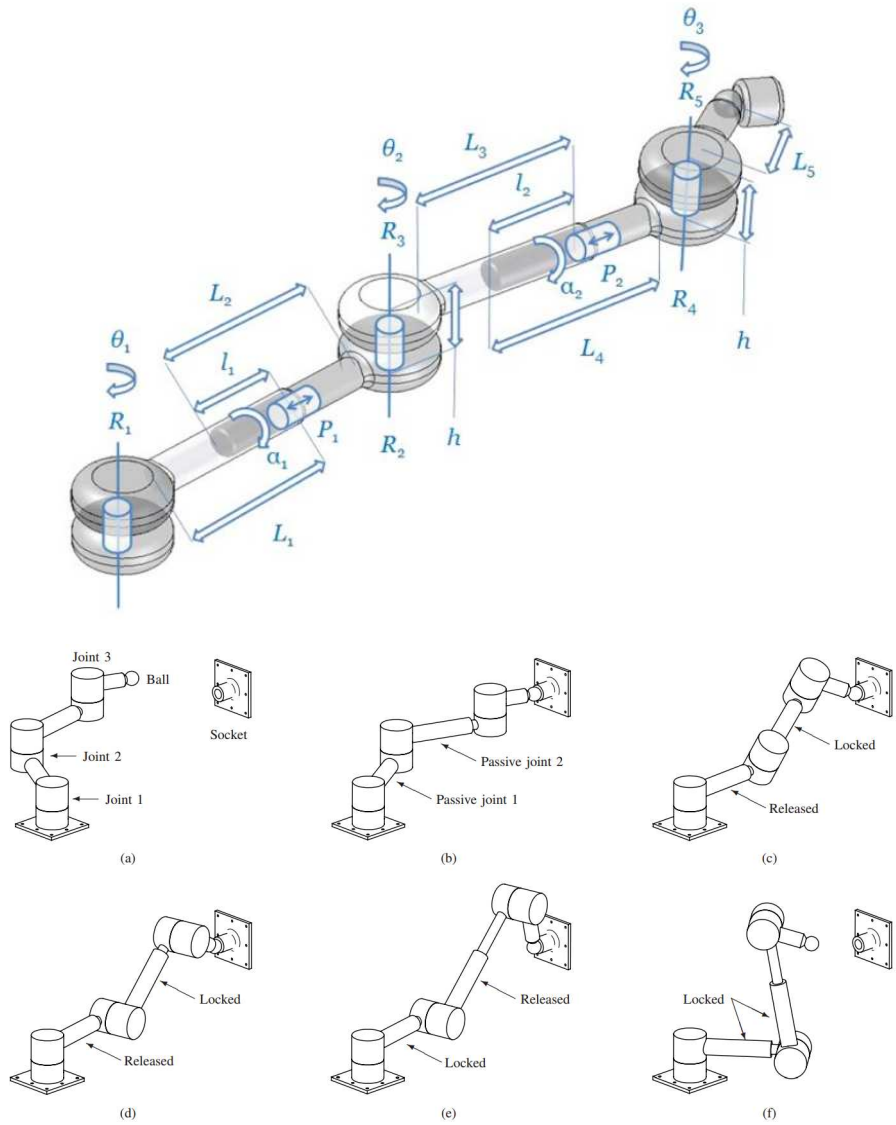


Figure 1.11: The **RRR** serial robot with lockable linear joints proposed by Aghili and Parsa. Kinematics of the reconfigurable robot (top). Typical maneuver to change its link lengths (bottom) (images from [1, 2]).

joint is said to implement the Veselova constraint [7], the kind of non-holonomic joint used throughout this thesis. Lower-mobility spatial parallel robots have become an active research topic in the field of parallel robot during the last decade because of their simple structure, low price and easy control. The dimension of the space of admissible velocities for the end-effector of this kind of parallel robots is lower than six and, if singular configurations are excluded, equal

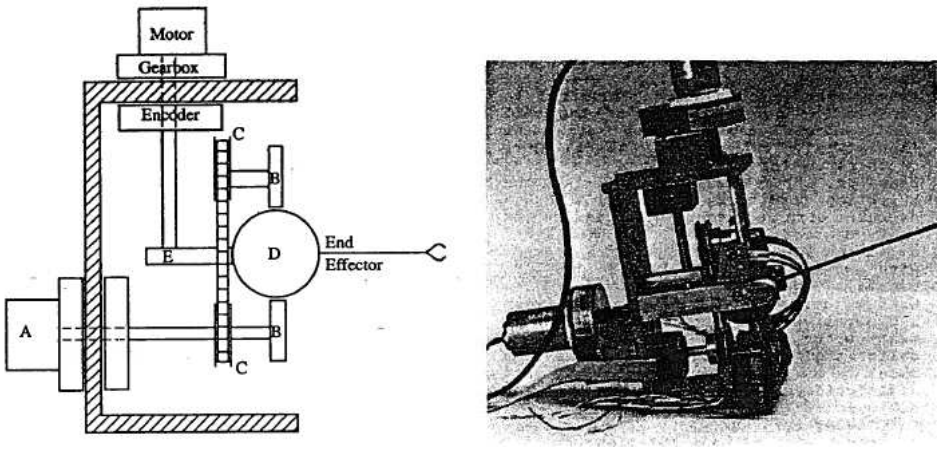


Figure 1.12: The robotic wrist proposed by Stammers *et al.* (images from [98]).

to the dimension of the tangent space of the reachable manifold. The substitution of a standard joint in a lower-mobility parallel robot by a non-holonomic joint with equivalent instantaneous kinematics has dramatic consequences: while the dimension of the space of admissible velocities for the end-effector remains the same, the dimension of the reachable space is increased. To the best of our knowledge, this idea was first used by Ben-Horin and Thomas in [4], where a three-legged parallel robot is proposed, where each leg is connected to the base through a non-holonomic joint.

Non-holonomic constraints appear in multiple areas of robotics [22, 30, 52, 94], mainly in those related to mobile robots (either terrestrial, outer-space, or underwater vehicles), or the manipulation of objects with multiple contacts. In all of these cases the non-holonomic restrictions are inherent. Angeles in [3], O'Reilly in [88], and more particularly Hennessey in [49], give a general view on how to deal with problems involving non-holonomic constraints. In particular, Hennessey presents an analysis of a unicycle moving on a sphere, an example closely related to the non-holonomic spherical joint proposed in these thesis. The difference is that in our case the moving element is the sphere and one of the control inputs affect only one of the state variables.

The literature on the use of non-holonomic devices in the design of manipulators is limited to few examples. For example, in [98], Stammers *et al.* present a robot wrist that can attain any orientation with only two motors. This is achieved by means of a friction drive, using rollers on a spherical ball to which the end effector is fixed, and by fixing the two motors to the arm. The roller axes are perpendicular to each other, which means that rotate instantaneously around the normal of the plane defined by them is not possible except by composition of rotations (see Fig. 1.12).

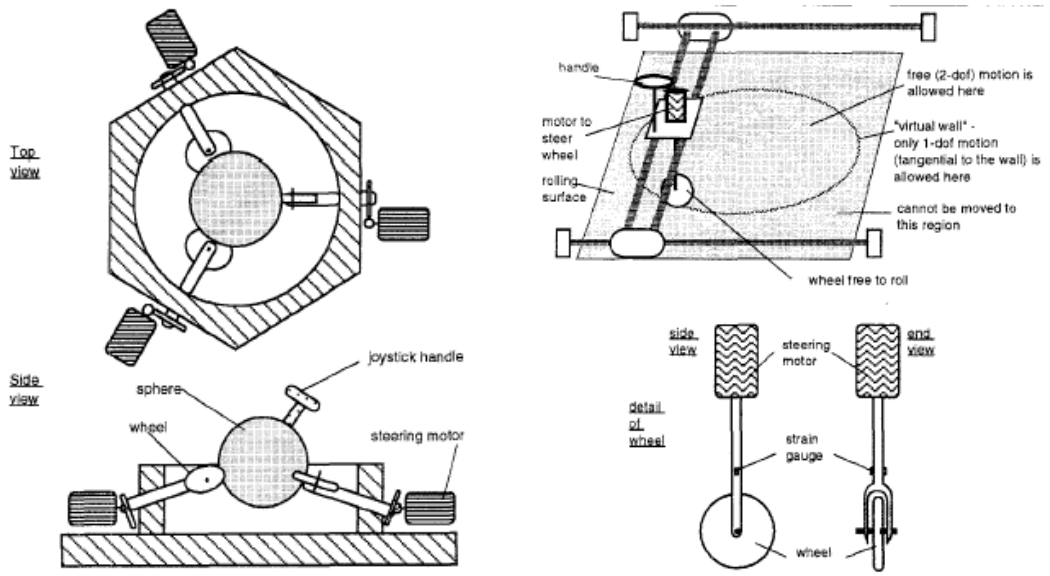


Figure 1.13: Conceptual design of a spherical Cobot (left) and a planar Cobot (right) based on non-holonomic joints (images from [90]).

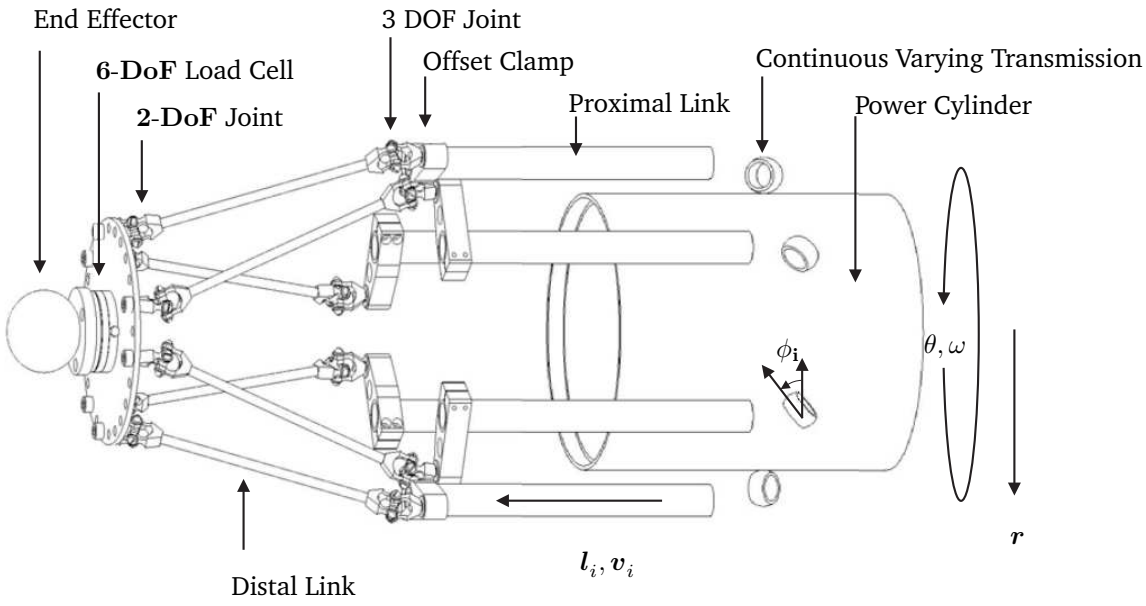


Figure 1.14: Conceptual design of a **6-DoF** Cobot Haptic Device based on non-holonomic joints (image from [33]).

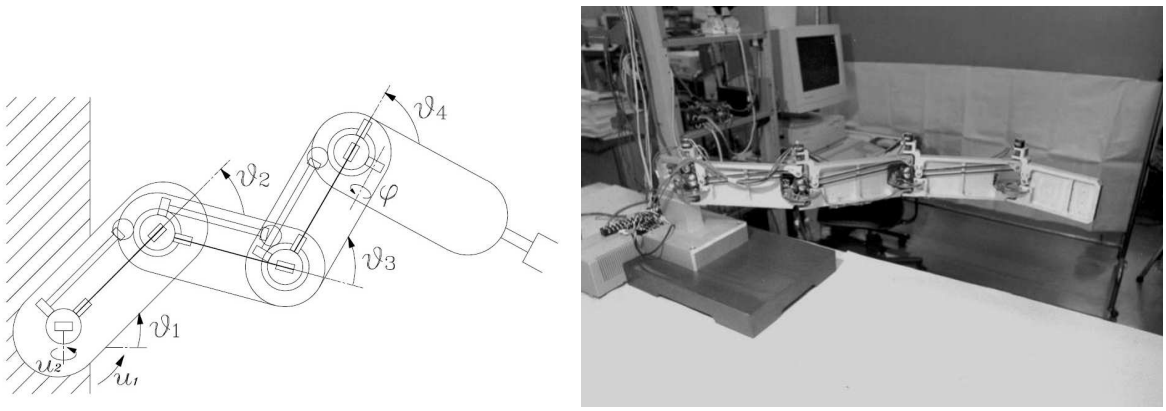


Figure 1.15: Illustration of the four-joint manipulator with only two actuators proposed by Nakamura *et al.* (left), and its implementation (right) (image from [84]).

Peshkin *et al.* [90] presented a group of haptic devices, named Cobot, with programmable constraints generated by non-holonomic joints. The non-holonomic constraints are introduced by rollers moving over different surfaces. The axis of rotation of each roller is adjustable, the axis is normal tangent plane at the contact point surface-roller. Each Cobot differ in the type of surface used. The spherical Cobot consist of a sphere with a joystick resting on three rollers. The plannar Cobot is a single roller over a plane (see Fig. 1.13). The most interesting Cobot, from this thesis point of view, is a **6-DoF FPM** with architecture **6$\underline{\mathbf{P}}$SU**, which uses the rollers in contact with a continuous rolling cylinder to actuate the prismatic joint, the non-holonomic pair is used as a continuous varying transmission (see Fig. 1.14).

Nakamura *et al.* [84] proposed a serial manipulator with n-joints which can reach any pose in its n-dimensional configuration space with only two actuators. The joints of this manipulator are coupled by non-holonomic devices, based on spheres and rollers, so that it reaches a desired pose by following a path whose computation is algorithmically equivalent to maneuvering a car with n-trailers (see Fig. 1.15).

More recently, in [4], Ben-Horin and Thomas proposed a three-legged parallel robot where each leg is connected to the base through a sphere whose motion is constrained by a roller. This parallel architecture permits to attain any position and orientation for the platform using only three prismatic actuators. It can be said that this work triggered the part of research in this thesis devoted to the use of non-holonomic joints in fully parallel robots.

1.3 Organization of the thesis

We start in Chapter 2 with a digression that can be skip on a first reading. In this digression, we take the possibility of substituting continuous actuators with binary ones to the limit. We actually prove that a single motor is enough for a robot with **6PUS** architecture to attain any pose in its workspace. Two motion strategies to solve the path-planning problem for this particular robot are proposed. Despite this robot was initially designed and analyzed as a pure intellectual exercise to exemplify the possibility of using a single continuous actuator, we believe that it could have practical interest in some applications with constraints on the robot's weight and its power consumption.

Chapter 3 is devoted to parallel robots with lockable joints. In this case the binary actuators are clutches. The considered lockable joints behave as a universal joints if the clutch is released, or as a revolute joint if it is engaged, as explained above. Due to its symmetry, an important effort is devote to the analysis, control and implementation of a parallel robot consisting of four **bRRPS** legs.

While the study of other designs of parallel robots involving the considered lockable joint will closely follow the one presented in Chapter 3, the situation is much more complicated when working with parallel robots having non-holonomic joints. This is why the following five chapters of this thesis are devoted to parallel robots with this kind of joints, analyzing separately the spatial and the spherical case.

Chapter 4 deepens in the aforementioned idea of generating under-actuated parallel robots by substituting spherical pairs by non-holonomic spherical pairs. Particular attention is also paid in this chapter to the practical implementation of non-holonomic joints. In Chapter 5, the kinetostatic analysis of an under-actuated spatial parallel robot with only three actuators is presented. This analysis is then used both in the design of this manipulator, and in its control, presented in Chapter 6. The study of the under-actuated spherical parallel robot is done in Chapters 7 and 8. In Chapter 7, first the kinematics and then the workspace of the robot are analyzed. In Chapter 8, different path planners are developed, including one that can be used as part of a controller. Finally, in Chapter 9, a recap of the contributions of the thesis and prospects for future research are given.

2

Full-mobility with only one continuous actuator

This chapter discusses the possibility of having an under-actuated parallel manipulator with **6P̲US** topology. The proposed device exploits the fact that, in some applications like work-tables in CNC machine tools, the path between the initial and final poses of the mobile platform is not assigned to reduce the number of actuators to only one. Part of the work presented in this chapter appeared in [42].

2.1 Introduction

The need of making an object move along an assigned path arises only in a limited number of applications. In most cases, the only initial and final poses of the object are assigned, whereas the path between them must just satisfy weak constraints (*e.g.*, obstacle avoidance, preventing interferences among machine components, etc.) which leave the choice of the path practically free. Such a freedom can be exploited during design to simplify the machine architecture. Work-tables of machine tools usually either perform simple translations or just lock the workpiece during cutting. Thus every time the workpiece has to be reoriented or, in general, repositioned with respect to the spindle axis either manual operations or external devices must intervene. Repositioning workpieces is a manipulation task that involves small six-dimensional workspaces, good positioning precision and high stiffness in the final configuration; it does not impose any constraint to the path between the initial and final poses. Parallel manipulators can satisfy the requirements on positioning precision and stiffness; moreover, they are specially suitable for applications that involve small workspaces. Therefore, they are natural candidates to move the work-table during workpiece repositioning.

How to exploit the free path for reducing the complexity of a manipulator destined to move the work-table during repositioning is an open problem. In this chapter, we propose an under-actuated parallel manipulator that, by exploiting the free path, is able to control the

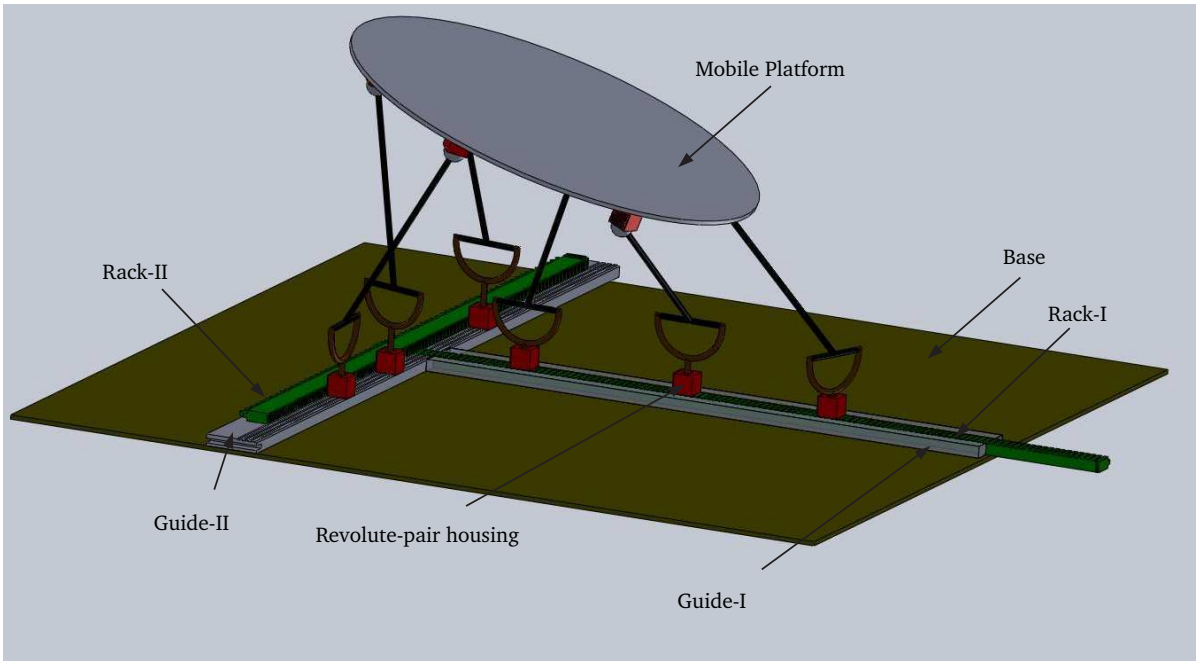


Figure 2.1: Under-actuated **6PUS**.

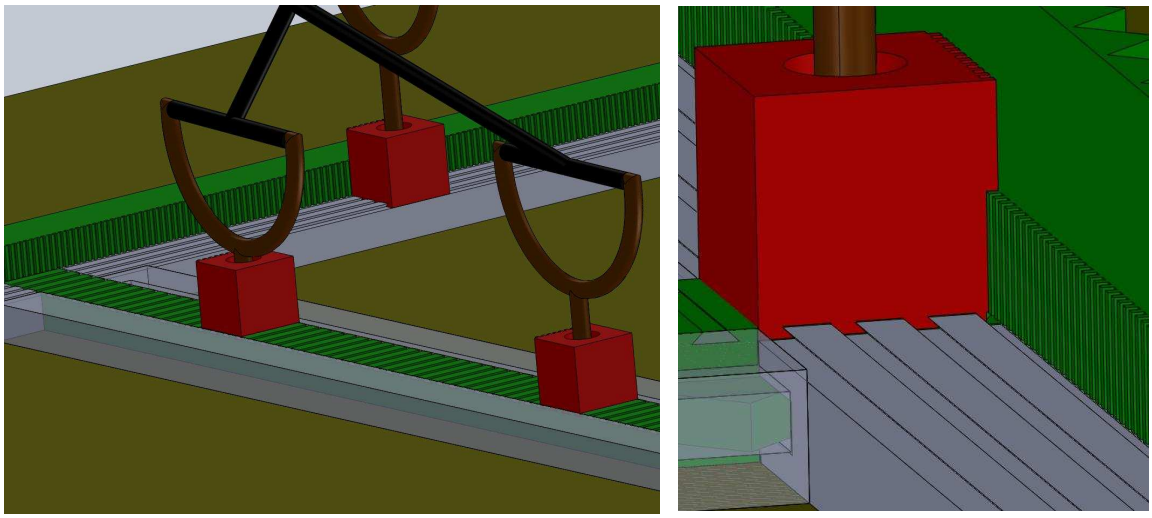


Figure 2.2: Details of the proposed under-actuated **6PUS** robot. The racks, the guides, and some of the revolute-pair housings (left). Revolute-pair housing at the intersection between the two racks (right).

mobile platform pose in a six-dimensional workspace by using only one motor. Section 2.2 describes the manipulator architecture and illustrates its operation. Section 2.3 addresses the kinetostatic analysis of the machine and gives conditions the path must satisfy to keep the mobile platform pose controllable during motion. Eventually, the conclusions are drawn in Section 2.4.

2.2 Manipulator architecture and operation

Fully parallel mechanisms with topology **6$\underline{\mathbf{P}}$US** feature a mobile platform connected to a fixed base through six in-parallel kinematic chains (legs) of type **$\underline{\mathbf{P}}$US**. Their architectures vary according to the relative disposition of the prismatic-pair sliding directions, the platform geometry, and the six fixed distances (**U**-**S** link lengths) between universal-joint center and spherical-pair center of each leg. By changing these geometric parameters, a number of **6$\underline{\mathbf{P}}$US** have been proposed in the literature (see [79, Chap. 2] for references). Boye and Pritschow [9] named them linapods. Honegger *et al.* [51] proposed the Hexaglide that has six parallel and coplanar guides. Moreover, some of the proposed architectures (Bernier *et al.* [5], and Pritschow *et al.* [92]) exhibit coincident guides for couples of prismatic pairs, and, in particular, decouple linapod Nabla 6, proposed by Bernier *et al.* [5], has three coplanar guides each carrying two sliders. The actuation of each prismatic pair is independent of the other actuations in all the linapods proposed in the literature.

Fig. 2.1 shows the proposed linapod. On the base, a single motor, through a transmission, actuates, one at a time, two racks that are constrained to slide along two mutually orthogonal guides forming a cross-shaped path (*T-paths*). The transmission is able to actuate one or the other rack by using two clutches that also act as brakes for the non-actuated rack. The racks carry suitably shaped hooks which can firmly lock revolute-pair housings (the cubes attached to the racks in Fig. 2.2(left)). In these revolute-pair housings, legs' universal joints insert one pin of their cross link so that the resulting revolute pair has the axis perpendicular to the plane of the guides. In so doing, all the universal joints have the other revolute-pair axis parallel to the plane of the guides, and their centers are constrained to lie on *T-paths* that are all parallel to the plane of the guides (*T-paths* are parallel but not coplanar). The universal-joint centers slide on these *T-paths* when the racks are moved.

On the mobile platform, the housing of the spherical pairs, which join the leg endings to the platform, are embedded in the platform.

The hooking between rack and revolute-pair housing is managed by a purely mechanical device carried either on the revolute-pair housing or on the guides. This hooking device and the hooks on the racks are conceived so that the following functional requirements are satisfied:

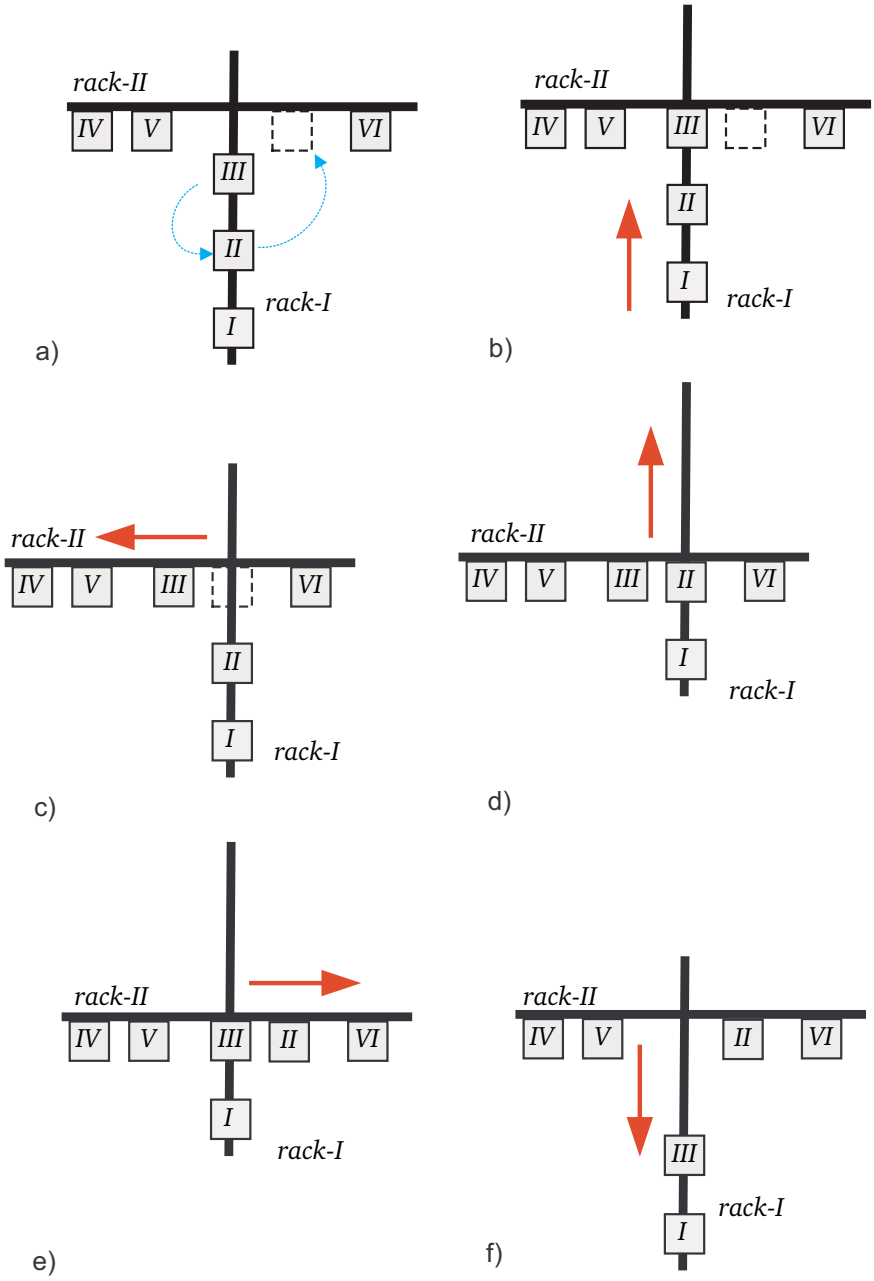


Figure 2.3: Sequence of rack motions (the numbered boxes represent the revolute-pair housings): (a) initial configuration, (b) *rack-I* was moved to make *housing-III* touch *rack-II*, (c) *rack-II* was moved to make the dotted box centered on *rack-I*, (d) *rack-I* was moved to make *housing-II* touch *rack-II*, (e) *rack-II* was moved back to its initial position, (f) final configuration obtained by moving back *rack-I* to its initial position.

1. if the revolute-pair housing is in contact with only one rack, the hooking device must provide a constraint that, combined with the constraint of rack's hooks, firmly holds the housing fixed to the rack;

2. if the revolute-pair housing is in contact with both racks (*i.e.*, at the intersection of the *T-path*), the hooking device must not provide any constraint, whereas the hooks of both racks must provide the constraints necessary to hold the housing;

3. if the revolute-pair housing is at the intersection of the *T-path* and one rack starts moving, the hooks of the moving rack must be able to tow the housing, whereas the hooks of the other rack must not forbid this towing and must be so shaped that, during the housing motion, make the hooking device intervene to provide its constraint.

Many hooking devices and complementary hooks for the racks can be easily devised to satisfy the above technical requirements. For instance, in Fig. 2.2(right), the hooks of *rack-I* are dovetail joints parallel to *rack-II*, whereas the hooks of *rack-II* are frontal teeth with rectangular cross section; moreover, the hooking device is constituted of two lateral stops fixed to *guide-I*, and of dovetail joints, identical to the ones of *rack-I*, fixed to *guide-II* [see Fig. 2.2(right)]. The many design alternatives for these equipments will not be discussed here, for the sake of conciseness.

By exploiting the above-reported properties of hooking device and racks' hooks the positions of the universal-joint centers on the *T-paths* can be about freely changed. In fact, if, for instance, *housing-II*, on *rack-I*, [Fig. 2.3(a)] must be moved to the dotted position on *rack-II* and *housing-III* must be moved to the actual position of *housing-II*, the sequence of operations shown in Fig. 2.3 can be implemented.

In general, many different rack-motion sequences lead to the same final configuration, and the number of operations to implement may decrease when the number of housing permutations increases.

2.3 Kinetostatics and constraints on the *T-path*

The inverse kinematics that, for this linapod, means the determination of the housing positions on the racks for an assigned platform pose (position and orientation) must be solved every time the platform is repositioned. This determination is straightforward once the positions of the universal-joint centers have been computed. The assigned relative pose between base and platform involves that the six *T-paths* (one per leg), the universal-joint centers must lie on, have assigned poses with respect to the spherical-pair centers embedded in the platform. Thus, for each leg, the determination of the universal-joint center's position reduces itself to compute the

intersection points between the *T-path* the universal-joint center must lie on and a sphere, with center at the spherical-pair center and radius equal to the leg length. This geometric problem has at most four solutions: the two sets of intersections between the sphere and the two sides of the **T**. At most four solutions for each leg yields at most 4^6 (*i.e.*, 4096) leg arrangements compatible with an assigned platform pose. Such a high number of inverse kinematics solutions is mainly theoretical. In fact, many line-sphere intersections will fall out of the line segments actually occupied by the *T-path's* sides. Moreover, other solutions will be excluded by the fact that two or more housings cannot be located on the same position, and that, on each rack, the hooks' sequence has a fixed pitch, which implies that the distance between couples of housings positioned on the same rack can only be multiples of the hooks' pitch. Eventually, all the leg arrangements that give a singular configuration (see below) must be excluded.

The direct kinematics of the proposed linapod consists in the determination of the platform poses compatible with an assigned disposition of the revolute-pair housings on the two racks. If the positions of the revolute-pair housings are assigned, the positions of the universal-joint centers will be assigned, too. Thus this problem reduces itself to the determination of the assembly modes of the **6US** structure (*i.e.*, two rigid bodies connected by six in-parallel **US** legs), which was broadly treated in the last two decades in connection with the direct kinematics solution of the general Stewart platform (see [79] for references). The result of these studies is that the **6US** structure can have at most forty assembly modes which can be even analytically determined [53, 56].

The singularities of the forward instantaneous kinematics are, for this linapod, the configurations where the platform can perform instantaneous motions even though the racks are locked (*i.e.*, they are uncertainty configurations of the **6US** structure). At a singularity of this type, the platform pose is not controllable, and the internal loads of one or more links of the legs are not able to equilibrate the external loads applied on the platform. Thus, they must be identified during design and avoided during operation. The uncertainty configurations of the **6US** structure have been studied by many authors, and, in the literature, both geometric and analytic conditions to identify them have been provided [23, 75, 76, 97]. The actual implementation of the proposed linapod requires that all this literature be exploited to correctly design and control it. Herein, we will only give the justification of some design choices due to the need of avoiding uncertainty configurations.

From a static point of view, in **6US** structures, an uncertainty configuration occurs when the six forces applied to the platform through the spherical pairs are not able to equilibrate any system of external loads. The fact that these forces are aligned with the leg axes (the axes connecting the centers of the universal and spherical joints for each leg) allowed the geometric classification of the singular configurations through particular arrangements of the six leg axes

[75]. All these singular arrangements satisfy at least one out of the following three geometric conditions: (a) the six axes either intersect or are parallel to a line, (b) the six axes are all parallel to a plane, and (c) the six axes are tangent to coaxial helices with the same pitch.

For the linapod under study, the possibility of locating all the revolute-pair housings on one rack would greatly improve the path planning algorithms (see below). Therefore, making this housing arrangement non-singular is important. Once all the housings are located on the same rack all the universal-joint centers lie on the same plane. And, in order to avoid the geometric conditions (a) and (b), the universal-joint centers must not be located at the same height on the rack (*i.e.*, the *T-paths* must not coincide); whereas the spherical-pair centers must be suitably distributed on the platform. The manufacturing conditions that allow condition (c) to be avoided are much more difficult to be visualized and a careful numerical check is necessary. It is worth noting that the leg arrangements with all the housings on the same rack geometrically coincide with the Hexaglide architecture [51], and the results obtained for the Hexaglide can also be usefully exploited.

A path-planning algorithm for the proposed linapod has to take into account all the above-reported kinetostatic considerations. In addition, it needs the implementation of a motion strategy for choosing the sequence of rack motions able to move from the initial platform pose to the final one. Each step of this sequence finishes with a particular arrangement (state) of the revolute-pair housings on the racks that is reached when both the racks are at rest and the actuation is about to be switched from one rack to the other. Thus, a path-planning algorithm has to determine the states' sequence by respecting the rule that the transition from one state to the successive one must be possible by moving only one rack. For instance, the motion described in Fig. 2.3 is characterized by six states and five transitions. Two different paths that have the same initial and final housing arrangements can be compared on the basis of the number of intermediate states, and, of course, the lower is the number the better is the path.

The sequence that moves only one housing from any position to any other without changing the positions of the other housings, in the final state, can be easily automated. Thus, a simple path-planning algorithm could reduce itself to implement six separate sequences each of which brings only one housing from its initial to its final position and, in the final state, does not change the positions of the housings already brought to their final positions. Such a motion strategy employs a great number of intermediate states. For instance, it is easy to realize that, in Fig. 2.3, the motion of the only *housing-II* without changing the position of *housing-III* would have required nine states, whereas the strategy reported in Fig 2.3 uses only six states to move both *housing-II* and *housing-III* to their final positions.

A much better motion strategy can be obtained by finding a state (parking state) from which any other state can be reached through a reduced number of intermediate states. The state with

all the housings located on *rack-II* and no housing at the rack intersection could be a parking state. In fact, from this parking state, a housing can be put on *rack-I*, at any position, with a sequence involving only two intermediate states, whereas only four intermediate states are required to change the position of a housing on *rack-II*. A path-planning algorithm based on this parking state, first, has to implement the sequences that bring all the housings on *rack-II* (note that the only housings located on *rack-I* in the initial state are involved in this phase); then, it has to move all the housings from the parking positions to their final positions.

2.4 Conclusions

The feasibility study of an under-actuated parallel manipulator with **6PUS** topology, destined to handle work-tables in CNC machine tools, has been presented. The proposed device exploits the fact that, in such an application, the path between the initial and final poses of the mobile platform is not assigned to reduce the number of actuators to only one. For the proposed manipulator, all the hardware critical points have been addressed. Its kinematic and static characteristics have been discussed, and the availability of the solutions to all the problems involved in its design and control has been verified. Two motion strategies that can be used in the path-planning algorithms have been proposed.

A formalization of the allowed rearrangements using group theory will probably provide a deeper insight into this path planning problem [63]. This is certainly a point that deserves further attention.

3

Parallel robots with lockable revolute joints

This chapter introduces a class of reconfigurable parallel robots consisting of a fixed base and a moving platform connected by n-serial chains, $n < 6$, having **bRR$\underline{\mathbf{P}}$S** (Lockable Revolute - Revolute-Prismatic-Spherical) topology. Only the prismatic joint is actuated and the first revolute joint in the chain can be locked or released during operation. It will be shown how the introduction of these lockable joints allow the prismatic actuators to maneuver to approximate **6-DoF** motions for the moving platform, despite having less than six actuators. An algorithm for generating these maneuvers is also described. Then, a motion planner, based on the generation of a Probabilistic Road Map, whose nodes are connected using the described maneuvers, is presented. The generated trajectories avoid singularities and possible collisions between legs. Part of the work presented in this chapter has appeared in [40].

3.1 Introduction

The Gough-Stewart platform consists of a base and a moving platform connected by six **U$\underline{\mathbf{P}}$S** (Universal-Prismatic-Spherical) legs, where the underline indicates that the prismatic joint is actuated. Thus, it is usually referenced as a **6U$\underline{\mathbf{P}}$S** platform. If a number of these **U$\underline{\mathbf{P}}$S** legs are eliminated, the mobility (remaining set of passive joints) must be reduced by as many leg substitutions to keep the platform location controllable. The resulting parallel manipulator will have a number of **DoF** equal to the number of remaining legs. Substitution of *all* the remaining **U$\underline{\mathbf{P}}$S** legs by **bRR$\underline{\mathbf{P}}$S** legs, where **bR** denotes a revolute joint lockable at any time during operation through a clutch, is one possibility for implementing this mobility reduction. To keep the mobility reduction at all times, the number of activated **bR** must be equal or greater to the number of eliminated legs. Each **bRR$\underline{\mathbf{P}}$S** will behave as a **R$\underline{\mathbf{P}}$S** chain when the **bR** joint is locked and, by properly arranging the axis of the revolute joints, as a **U$\underline{\mathbf{P}}$S** when it is not. The maximum number of leg eliminations is three, as many are the manipulators that can be

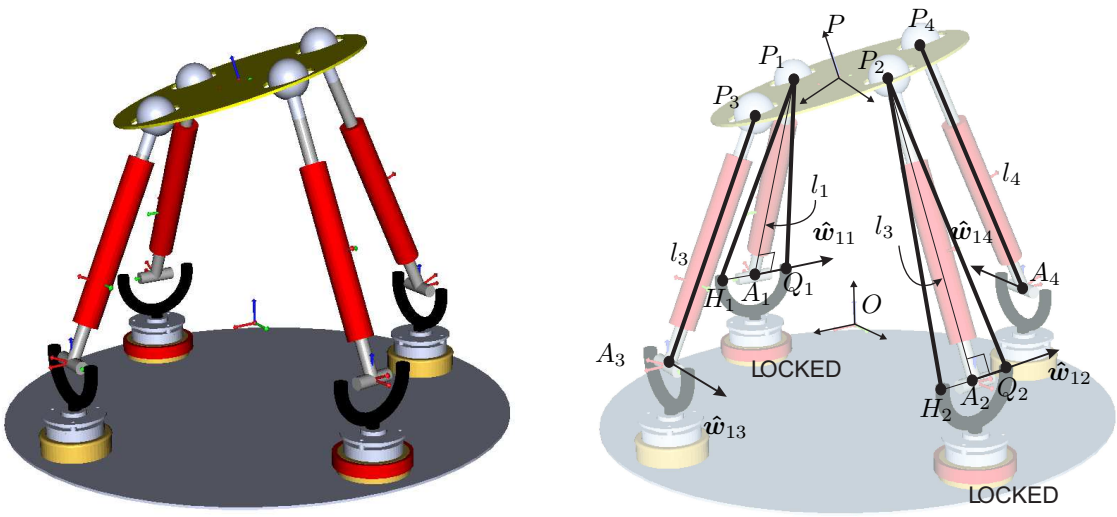


Figure 3.1: The proposed parallel robot (left) and associated notation (right). It consists of four **bRRPS** legs attached to the base through passive lockable revolute joints. Since two clutches must be engaged at any time to keep the platform rigidly linked to the base, it behaves as a reconfigurable **2RPS-2UPS** platform.

generated from the Gough-Stewart platform with this technique. Table 3.1 summarizes the family of generated manipulators. The **5bRRPS** and **4bRRPS** architectures are of interest because their motion possibilities can be increased by on-line switching the lockable joints. Two kinds of reconfigurable parallel platforms with low mechanical complexity are thus obtained. The architecture involving four legs is probably the most attractive because it uses the least number of actuators (see Fig. 3.1). This chapter is devoted to its study.

The rest of this chapter is organized as follows. Section 3.2 studies the kinematics of the proposed platform. Section 3.3 shows how to maneuver to locate the platform in any arbitrary pose. Section 3.4 shows how to generate a roadmap in the configuration space of the platform that permits to obtain paths, far from singularities and leg collisions, connecting two arbitrary poses. Section 3.5 and 3.6 describes practical aspects concerning the implemented prototype. Finally, the main results are summarized in Section 3.7.

Table 3.1: The three possible architectures for parallel robots, having **UPS** legs, using **bRRPS** legs as replacement.

Legs removed (minimum # of **bR** locked at any time)	# of legs replaced	Architecture	Robot space	Related references
1	5	**4UPS + RPS**	**5-DoF** (reconfigurable, covering **6-DoF**)	[74]
2	4	**2UPS + 2RPS**	**4-DoF** (reconfigurable, covering **6-DoF**)	[55], [73], [71]
3	3	**3RPS**	**3-DoF** (non-reconfigurable)	[57], [95]

3.2 Kinematics of the 2RPS-2UPS parallel robot

If the leg lengths of the robot in Fig. 3.2 are fixed and clutches of legs 1 and 2 are locked the resulting parallel structure is **2RS-2US**. In case the platform is removed, points P_1 and P_2 would move on circular arcs, while P_3 and P_4 would be constrained to move on spheres. With reference to this figure, P_i, $i = 1, \ldots, 4$, are the centers of the spherical pairs. Points A_i, $i = 1, \ldots, 4$, are the projections of P_i onto the revolute-pair axes adjacent to the spherical pair, P_i being its center. A_3 and A_4 are also chosen as centers of the two universal joints without losing generality. Points B_3 and B_4 are the projections of P_3 and P_4, respectively, onto the line through P_1 and P_2. The i^{th} leg length $\|\boldsymbol{p}_i - \boldsymbol{a}_i\|$ will be denoted l_i, the magnitude of the vector $(\boldsymbol{p}_2 - \boldsymbol{p}_1)$ will be denoted a, whereas the magnitudes of the vectors $(\boldsymbol{p}_j - \boldsymbol{b}_j)$ for $j = 3, 4$ will be denoted r_j. Moreover, the following unit vectors and scalars are defined

$$
\begin{aligned}
\hat{\boldsymbol{q}} &= \frac{(\boldsymbol{p}_1 - \boldsymbol{a}_2)}{\|\boldsymbol{p}_1 - \boldsymbol{a}_2\|}, & \hat{\boldsymbol{v}}_3 &= \frac{\hat{\boldsymbol{q}} \times \hat{\boldsymbol{u}}_3}{\|\hat{\boldsymbol{q}} \times \hat{\boldsymbol{u}}_3\|}, \\
\hat{\boldsymbol{h}}_1 &= \hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{v}}_1, & \hat{\boldsymbol{h}}_3 &= \hat{\boldsymbol{u}}_3 \times \hat{\boldsymbol{v}}_3, \\
\hat{\boldsymbol{h}}_2 &= \hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{v}}_2, & b_3 &= (\boldsymbol{b}_3 - \boldsymbol{p}_1) \cdot \hat{\boldsymbol{u}}_3, \\
\hat{\boldsymbol{u}}_3 &= \frac{(\boldsymbol{p}_2 - \boldsymbol{p}_1)}{a}, & b_4 &= (\boldsymbol{b}_4 - \boldsymbol{p}_1) \cdot \hat{\boldsymbol{u}}_3. \\
\hat{\boldsymbol{g}}_i &= \frac{(\boldsymbol{p}_i - \boldsymbol{a}_i)}{l_i} \quad i = 1, \ldots, 4,
\end{aligned}
$$

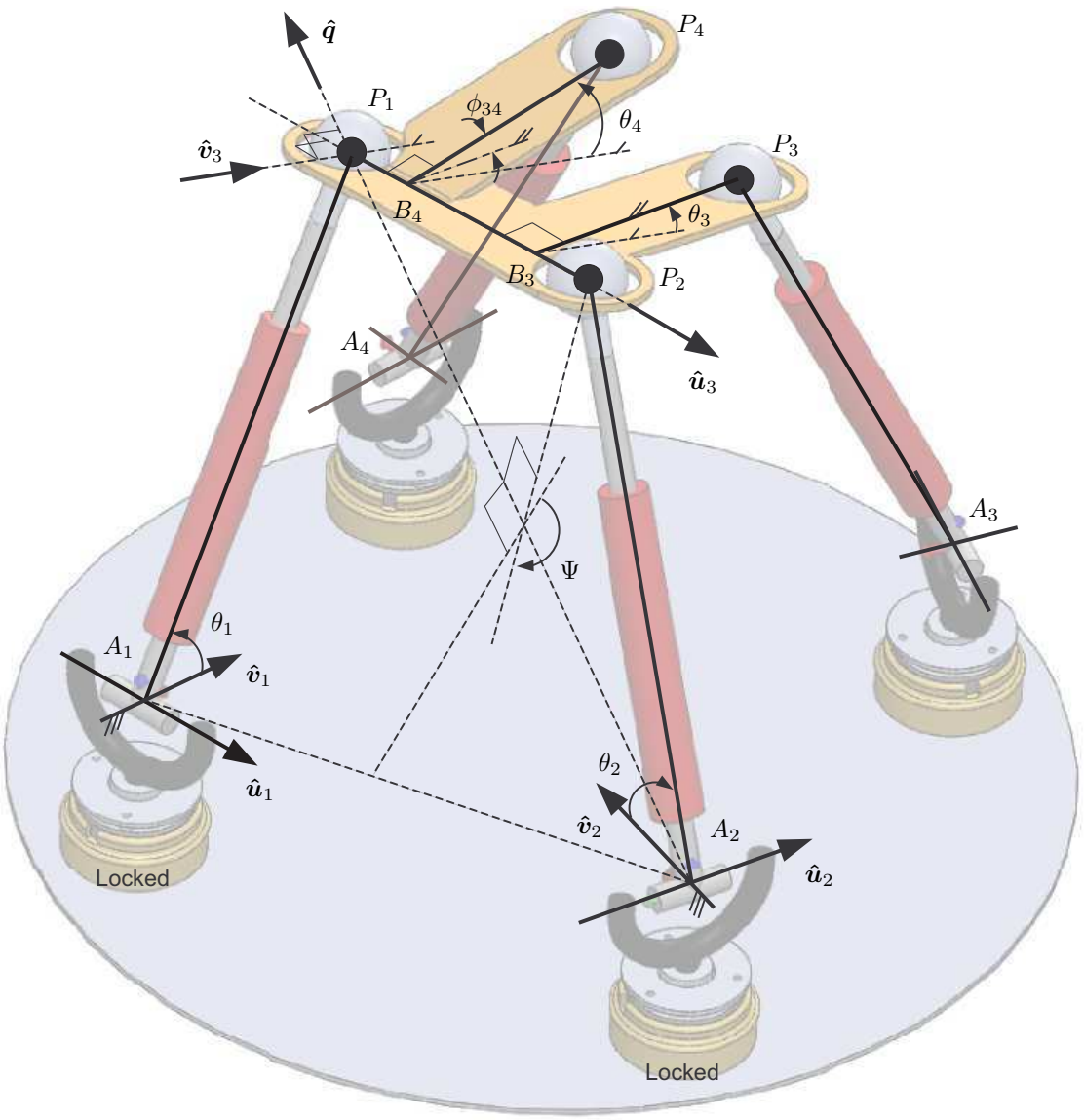


Figure 3.2: Notation associated with the **2RS-2US** structure resulting from fixing the leg lengths and locking the revolute joints centered at A_1 and A_2 for the more general case in which P_1, P_2, P_3 and P_4 are not necessarily coplanar.

3.2.1 Position analysis

The determination of the actuated-joint variables (leg lengths) for an assigned pose of the platform (the inverse kinematics problem) is straightforward. In fact, once the positions of P_i, $i = 1, \ldots, 4$, are known, the leg lengths can be immediately computed since the positions

of A_i, $i = 1, \ldots, 4$, are geometric data linked to the base reference frame (see Fig. 3.1). On the contrary, the determination of the platform pose for assigned leg lengths (the forward kinematics problem) requires the solution of the **2RS-2US** closure equations which constitute a non-linear equation system. This problem coincides with the one encountered when solving the forward kinematics of the **6-4** fully-parallel mechanism [55] since that mechanism generates an **2RS-2US** structure when the actuated joints are locked. In [55], Innocenti gave the analytical solution of this problem and showed that, in general, up to 32 platform poses may be compatible with an assigned set of leg lengths. In the following part of this subsection, the **2RS-2US** closure equations will be deduced in a form slightly different, from the one reported in [55], which is more appropriate to the analysis presented in the next subsection. With reference to Fig. 3.2 and the adopted notations, the **2RS-2US** closure equations can be written as follows:

$$(\boldsymbol{p}_2 - \boldsymbol{p}_1) \cdot (\boldsymbol{p}_2 - \boldsymbol{p}_1) = a^2, \tag{3.1}$$
$$(\boldsymbol{p}_3 - \boldsymbol{a}_3) \cdot (\boldsymbol{p}_3 - \boldsymbol{a}_3) = l_3^2, \tag{3.2}$$
$$(\boldsymbol{p}_4 - \boldsymbol{a}_4) \cdot (\boldsymbol{p}_4 - \boldsymbol{a}_4) = l_4^2, \tag{3.3}$$

where

$$\boldsymbol{p}_1 = \boldsymbol{a}_1 + l_1(c_1\hat{\boldsymbol{v}}_1 + s_1\hat{\boldsymbol{h}}_1), \tag{3.4}$$
$$\boldsymbol{p}_2 = \boldsymbol{a}_2 + l_2(c_2\hat{\boldsymbol{v}}_2 + s_2\hat{\boldsymbol{h}}_2), \tag{3.5}$$
$$\boldsymbol{p}_3 = \boldsymbol{p}_1 + b_3\hat{\boldsymbol{u}}_3 + r_3(c_3\hat{\boldsymbol{v}}_3 + s_3\hat{\boldsymbol{h}}_3), \tag{3.6}$$
$$\boldsymbol{p}_4 = \boldsymbol{p}_1 + b_4\hat{\boldsymbol{u}}_3 + r_4(c_4\hat{\boldsymbol{v}}_3 + s_4\hat{\boldsymbol{h}}_3), \tag{3.7}$$

and

$$c_4 = c_3\cos(\phi_{34}) - s_3\sin(\phi_{34}), \tag{3.8}$$
$$s_4 = c_3\sin(\phi_{34}) + s_3\cos(\phi_{34}), \tag{3.9}$$

where c_i and s_i for $i = 1, \ldots, 4$ stand for $\cos(\theta_i)$ and $\sin(\theta_i)$, respectively.

Equation (3.1) is a trigonometric c-s-linear equation that involves only c_1, c_2, s_1 and s_2. It is the closure equation of the **RSSR** loop. Equations (3.2) and (3.3) involve c_1, c_2, c_3, s_1, s_2 and s_3. By eliminating c_3 and s_3 from these equations, the resultant will contain c_1, c_2, s_1 and s_2 and can be used with equation (3.1) for a further elimination which yield an univariate polynomial equation.

3.2.2 Singularities

The configurations where the platform can perform elementary motions, even though the actuators are locked, are called *parallel singularities*. Parallel singularities are critical both from the control (the platform pose becomes no longer controllable) and the statics (some links should stand infinite internal loads) point of views. Thus, they must be avoided during operation.

When the **2RPS-2UPS** platform is at a parallel singularity, the **2RS-2US** structure obtained by locking the actuators is singular, too (*i.e.*, the structure is not rigid). Thus, by looking for the **2RS-2US** singular geometries, the parallel singularities of the associated **2RPS-2UPS** can be found.

When the **2RS-2US** structure assumes a singular geometry, the platform can perform elementary motions that must fulfill the following velocity equations, deduced by differentiating equations (3.1), (3.2), and (3.3):

$$(\dot{\boldsymbol{p}}_2 - \dot{\boldsymbol{p}}_1) \cdot \hat{\boldsymbol{u}}_3 = 0, \tag{3.10}$$

$$\dot{\boldsymbol{p}}_3 \cdot \hat{\boldsymbol{g}}_3 = 0, \tag{3.11}$$

$$\dot{\boldsymbol{p}}_4 \cdot \hat{\boldsymbol{g}}_4 = 0, \tag{3.12}$$

where

$$\dot{\boldsymbol{p}}_1 = \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1), \tag{3.13}$$

$$\dot{\boldsymbol{p}}_2 = \dot{\theta}_2 l_2 (\hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{g}}_2), \tag{3.14}$$

$$\begin{aligned}\dot{\boldsymbol{p}}_3 =& \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1) + \frac{b_3}{a}\left[\dot{\theta}_2 l_2 (\hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{g}}_2) - \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1)\right] \\ &+ \dot{\theta}_3 \left[\hat{\boldsymbol{u}}_3 \times (\boldsymbol{p}_3 - \boldsymbol{b}_3)\right] + r_3 \left[c_3 \dot{\hat{\boldsymbol{v}}}_3 + s_3 \dot{\hat{\boldsymbol{h}}}_3\right],\end{aligned} \tag{3.15}$$

$$\begin{aligned}\dot{\boldsymbol{p}}_4 =& \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1) + \frac{b_4}{a}\left[\dot{\theta}_2 l_2 (\hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{g}}_2) - \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1)\right] \\ &+ \dot{\theta}_3 \left[\hat{\boldsymbol{u}}_3 \times (\boldsymbol{p}_4 - \boldsymbol{b}_4)\right] + r_4 \left[c_4 \dot{\hat{\boldsymbol{v}}}_3 + s_4 \dot{\hat{\boldsymbol{h}}}_3\right],\end{aligned} \tag{3.16}$$

and

$$\dot{\hat{\boldsymbol{u}}}_3 = \frac{\dot{\theta}_2 l_2 (\hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{g}}_2) - \dot{\theta}_1 l_1 (\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1)}{a}, \tag{3.17}$$

$$\dot{\hat{\boldsymbol{q}}} = \frac{\dot{\theta}_1 l_1 \left[(\hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1) - (\hat{\boldsymbol{q}} \cdot \hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1)\hat{\boldsymbol{q}}\right]}{\|\boldsymbol{p}_1 - \boldsymbol{a}_2\|}, \tag{3.18}$$

$$\dot{\hat{\boldsymbol{v}}}_3 = \frac{\dot{\hat{\boldsymbol{q}}} \times \hat{\boldsymbol{u}}_3 + \hat{\boldsymbol{q}} \times \dot{\hat{\boldsymbol{u}}}_3 - \left[\hat{\boldsymbol{v}}_3 \cdot (\dot{\hat{\boldsymbol{q}}} \times \hat{\boldsymbol{u}}_3 + \hat{\boldsymbol{q}} \times \dot{\hat{\boldsymbol{u}}}_3)\right] \hat{\boldsymbol{v}}_3}{\|\hat{\boldsymbol{q}} \times \hat{\boldsymbol{u}}_3\|}, \tag{3.19}$$

$$\dot{\hat{\boldsymbol{h}}}_3 = \dot{\hat{\boldsymbol{u}}}_3 \times \hat{\boldsymbol{v}}_3 + \hat{\boldsymbol{u}}_3 \times \dot{\hat{\boldsymbol{v}}}_3, \tag{3.20}$$

which are obtained by differentiating equations (3.4)-(3.7). The introduction of (3.13) and (3.14) into (3.10) yields

$$\dot{\theta}_2 = \dot{\theta}_1 \frac{l_1(\hat{\boldsymbol{u}}_3 \cdot \hat{\boldsymbol{u}}_1 \times \hat{\boldsymbol{g}}_1)}{l_2(\hat{\boldsymbol{u}}_3 \cdot \hat{\boldsymbol{u}}_2 \times \hat{\boldsymbol{g}}_2)}. \tag{3.21}$$

In the above relationship, $\dot{\theta}_2$ is undefined when $\hat{\boldsymbol{u}}_3$, $\hat{\boldsymbol{u}}_2$, and $\hat{\boldsymbol{g}}_2$ are coplanar. The configuration where this geometric condition occurs are singularities of the internal **RSSR** loop and, in general, they are singularities of the **2RS-2US** structure, too. The introduction of (3.21) into (3.15)-(3.20) and of the resultant relationships into equations (3.11) and (3.12) yield a linear and homogeneous system of two equations in two unknowns which can be written as follows:

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_3 \end{pmatrix} = 0. \tag{3.22}$$

This linear system admits a non-null solution for $\dot{\theta}_1$ and $\dot{\theta}_3$ (*i.e.*, a singular configuration occurs for the **2RS-2US** structure) if and only if

$$m_{11}m_{22} - m_{12}m_{21} = 0. \tag{3.23}$$

The above relationship is the analytic expression of the singularity condition of the **2RS-2US** structure. It is satisfied either when the two vectors $\boldsymbol{m}_i = (m_{1i}, m_{2i})^T$, $i = 1, 2$, are parallel or when at least one of them is a null vector. The dimensionless parameters

$$k_1 = \frac{\|\boldsymbol{m}_1\|}{\|\boldsymbol{m}_2\|}, \quad k_2 = \frac{\|\boldsymbol{m}_1\| \, \|\boldsymbol{m}_2\|}{\|\boldsymbol{m}_1 \cdot \boldsymbol{m}_2\|} \tag{3.24}$$

can be used to evaluate how far from singularity a configuration is. The farthest-from-singularity configuration is the one where k_1 is equal to 1 and k_2 is equal to infinity; whereas a singular configuration occurs when at least one of the following conditions occur: (*a*) k_1 is equal to 0, (*b*) k_1 is equal to infinity, (*c*) k_2 is equal to 1. Based on these values, the following objective function, to be maximized during the platform motion, can be defined

$$n = \frac{k_1}{(k_1 - 1)^4} + (k_2 - 1). \tag{3.25}$$

The value of such a function tends to infinity when k_1 (k_2) tends to 1 (infinity); and it decreases when either k_1 (k_2) tends to 0 (1) or k_1 tends to infinity. It will be useful later, when assigning a cost to a path.

3.3 Maneuvers

Let us assume that we want to generate a trajectory connecting $\boldsymbol{X}_I = (\boldsymbol{l}_I, \boldsymbol{\phi}_I) = (l_1^I, \ldots, l_4^I, \phi_1^I, \ldots, \phi_4^I)$ to $\boldsymbol{X}_F = (\boldsymbol{l}_F, \boldsymbol{\phi}_F) = (l_1^F, \ldots, l_4^F, \phi_1^F, \ldots, \phi_4^F)$ where l_i is the length of leg i and ϕ_i is the angle formed by $\hat{\boldsymbol{w}}_{1i}$ and the x-axis of the world reference frame (see Fig. 3.1). Since the robot is not capable, in general, of reaching the final pose directly, it is necessary to introduce an intermediate one (a *via pose* $\boldsymbol{X}_V = (\boldsymbol{l}_V, \boldsymbol{\phi}_V)$) where the lockable joints are switched. The leg lengths in the via pose, $\boldsymbol{l}_V$, can be computed numerically by setting the released joints to their values in the final pose and solving a local optimization problem starting from the initial pose. This can be efficiently implemented using the Newton's method [66]. Then, the proposed maneuver consist of two clutches-switching and two legs movement detailed in Fig. 3.3.

Note that there are up to six sets of possible maneuvers connecting two given poses: one for each possible pair of locked joints. Once we have a candidate for a maneuver, and its corresponding via pose, it must be execute by driving the robot's prismatic actuators, as explained above. The simplest driving law is that consisting in linearly interpolating the leg lengths from $\boldsymbol{l}_I$ to $\boldsymbol{l}_V$, and then from $\boldsymbol{l}_V$ to $\boldsymbol{l}_F$. During this process, it might happen that the system reaches a different solution from the expected one (remind that the forward kinematics problem has no single solution). If so, the generated maneuver is not valid. This might happen mainly when the maneuver involves a path close to a singularity. For the sake of simplicity, in this case the obtained maneuver would not be considered as valid, though a more sophisticated driving law might connect the initial to the final configuration through the obtained via pose.

There is one more reason to reject a candidate for a maneuver: it leads to collisions or the joints are not kept within their valid range of motion. A complete test for collision detection can be implemented using available collision detection packages such as GJK, SOLID, V-Clip, I-Collide, etc. (see [11, p. 201] and the references therein).

Once all valid maneuvers are computed, it is reasonable to choose the one that keeps the platform as far as possible from its parallel singularities. Unfortunately, there is no proper distance to a singularity [78]. As a simplification in our particular design, the quality measure to decide whether the maneuver is close to a singularity is taken to be (3.25) evaluated in the corresponding via pose. It is assumed that the bigger this value is, the farther the via pose is from a singularity. Then, the reciprocal of this value is taken as the cost of a maneuver.

The above procedure to find the best maneuver connecting two arbitrary configurations is

summarized in pseudocode in Algorithm 3.1. Function **Candidate** implements the Newton's method that computes the leg lengths in the via pose. Function **ValidPath** verifies if the final configuration is reached by linearly interpolating the leg lengths, checks if no collisions arise, and verifies if the joints are kept within their range along the trajectory. Finally, function **Cost** assigns a cost to the maneuver based on the objective function (3.25) to a singularity of the via pose.

Algorithm 3.1: BestViaPose($\boldsymbol{X}_i$, $\boldsymbol{X}_j$)

Maneuvers ← {[1,2,3,4],[1,3,2,4],[1,4,2,3],[2,3,1,4],[2,4,1,3],[3,4,1,2]}
/* The first two indices of each **4-tuple** correspond to the locked joints */
/* during the first motion of the maneuver */
$\boldsymbol{X}_V \leftarrow$ void
for all M $\in$ Maneuvers **do**
  $[i, j, k, l] \leftarrow$ M
  $\boldsymbol{\phi}_x[i] \leftarrow \boldsymbol{\phi}_I[i]$
  $\boldsymbol{\phi}_x[j] \leftarrow \boldsymbol{\phi}_I[j]$
  $\boldsymbol{\phi}_x[k] \leftarrow \boldsymbol{\phi}_F[k]$
  $\boldsymbol{\phi}_x[l] \leftarrow \boldsymbol{\phi}_F[l]$
  $\boldsymbol{l}_x \leftarrow$ **Candidate**$(\boldsymbol{X}_I, \boldsymbol{\phi}_x, M)$
  $\boldsymbol{X}_x \leftarrow (\boldsymbol{l}_x, \boldsymbol{\phi}_x)$
  if ValidPath$(\boldsymbol{X}_I, \boldsymbol{X}_x, \boldsymbol{X}_F)$ = TRUE **then**
    /* The maneuver is valid */
    if Cost$(\boldsymbol{X}_x) <$ **Cost**$(\boldsymbol{X}_V)$ **then**
      /* **Cost**(void) returns ∞ */
      $\boldsymbol{X}_V \leftarrow \boldsymbol{X}_x$
    end if
  end if
end for
return $\boldsymbol{X}_V$

It is clear that the above algorithm might fail to find a path mainly when the initial and final poses are far apart in the configuration space of the robot. In these cases, one alternative is to subdivide the trajectory into segments whose initial and final poses can be connected using the above algorithm. Unfortunately, this simple idea might also fail. The alternative is to use a motion planner, as described in the next section.

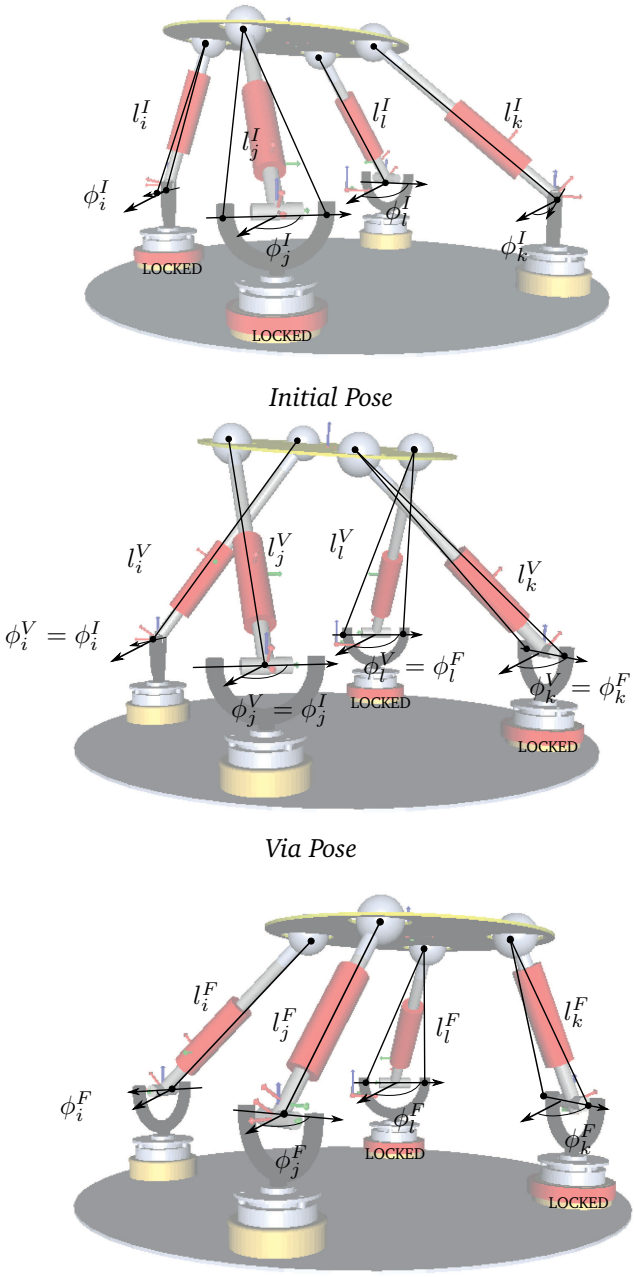


Figure 3.3: The proposed maneuver connecting two configurations, $\boldsymbol{X}_I = (\boldsymbol{l}_I, \boldsymbol{\phi}_I)$ and $\boldsymbol{X}_F = (\boldsymbol{l}_F, \boldsymbol{\phi}_F)$ passing by the via pose $\boldsymbol{X}_V = (\boldsymbol{l}_V, \boldsymbol{\phi}_V)$, with $\boldsymbol{\phi}_V = (\phi_i^I, \phi_j^I, \phi_k^F, \phi_l^F)$ and $\boldsymbol{l}_V$ to be found, is performed in three steps. Top: with clutches i and j engaged and clutches k and l disengaged, the prismatic actuators are driven from $\boldsymbol{l}_I$ to $\boldsymbol{l}_V$; Center: all four clutches are switched around, clutches k and l engaged and clutches i and j disengaged; Bottom: the prismatic actuators are driven from $\boldsymbol{l}_V$ to $\boldsymbol{l}_F$.

Algorithm 3.2: GenerateRoadmap

```
1:  for i = 1 to NumPoses do
2:     X_i ← RandomPose()
3:     Poses ← FindNeighborPoses(X_i)
4:     for all X_j ∈ Poses do
5:        X_V ← BestViaPose(X_i, X_j)
6:        ManMatrix[i,j] ← X_V
7:        CostMatrix[i,j] ← Cost(X_V)
8:        /* Cost(void) returns ∞ */
9:     end for
10: end for
```

3.4 Path planning

There are many possible approaches for implementing a motion planner but those based on Probabilistic Road Maps [65] have demonstrated their potential in many applications [11, Chapter 7]. This approach has already been successfully applied to ordinary parallel robots in [18]. Next, it is adapted to the proposed reconfigurable robot. Within this approach, the proposed robot would be subjected to a *learning phase* where its configuration space is randomly sampled. These samples are connected to their neighbors through the maneuvers, presented in the previous section, to generate a roadmap. Then, in the *query phase*, in which a path between two arbitrary poses must be found, the initial and final poses are firstly linked to their neighbors in the roadmap and, using a graph search algorithm, the best path according to a given criterion is found.

3.4.1 Generating the roadmap

The roadmap is built by sampling poses in the configuration space of the robot. When a sample is chosen, the best maneuvers to connect it to its neighboring poses previously generated are computed (see Fig. 3.4). Two poses are considered to be neighbors if the Euclidean norm between both their position and orientation components are below a given threshold. If a valid maneuver is found, its corresponding via pose is stored in an adjacent matrix together with its associated cost. If not, the stored cost will be infinite. Algorithm 3.2 gives this description in pseudocode.

To increase the density of the roadmap, it is always possible to add an intermediate configuration when two configurations fail to be connected directly through one of the six maneuvers that can be obtained using the procedure described above.

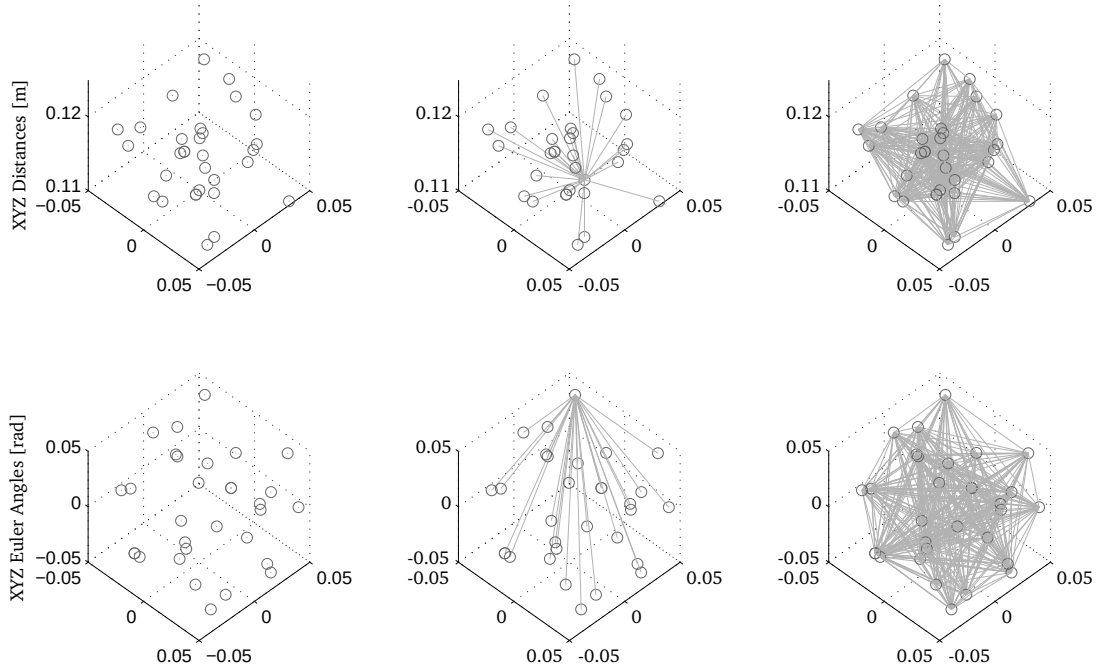


Figure 3.4: Learning phase of the proposed path planning algorithm. As a result, a roadmap is generated. Left: Random generation of poses. Center: Connection generation between neighboring poses. Right: Resulting roadmap.

3.4.2 Finding a path

If a trajectory —free from collisions and as far as possible from any singularity— connecting $\boldsymbol{X}_I$ to $\boldsymbol{X}_F$ must be generated, it is firstly necessary to connect these two poses to the previously generated roadmap (see Fig. 3.5). That is, the best maneuvers to connect them to their neighbors should be computed. Once the initial and final poses are connected to the roadmap, it is only needed to find the shortest path connecting them in terms of costs. Dijkstra's algorithm is well-suited to this end [17, p. 595]. Finally, when the path is obtained, if one exists, the corresponding maneuvers —described in terms of leg lengths settings and sequences of locked and released revolute joints— can be executed by the robot.

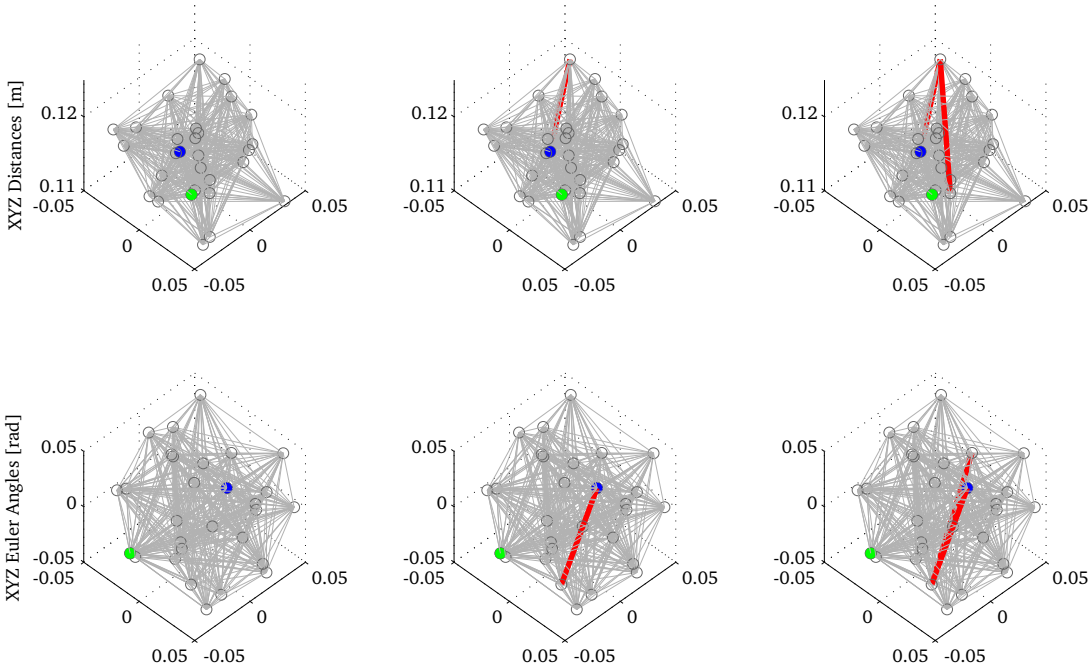


Figure 3.5: Query phase of the proposed path planning algorithm. The initial and final poses are represented in blue and green, respectively. Left: The initial and final poses are connected to the roadmap. Center: The starting pose is connected to the via pose. Right: The via pose is connected to the end pose.

3.5 Hardware implementation

After verifying the kinematic behavior of the proposed robot in simulation, the prototype in Fig. 3.6(left) was built. The base and the moving platform were made of 3mm thick nickel-plated steel plates. They are disks of 400 mm and 200 mm in diameter, respectively. The lockable revolute joints have been implemented using electromagnetic clutches manufactured by Huco-Dynatork Co. [12]. When one of this clutches is energized, the corresponding axis of rotation is disengaged, otherwise it remains locked. The actuated prismatic joints are implemented using miniature servo linear motors manufactured by Firgelli Technologies Inc. [54]. The four actuators are controlled through a USB servo card. An interesting feature of this prototype is that the legs are attached to the base and the moving platform through magnetic fixtures. This simplifies any leg rearrangement during tests. Finally, all plastic elements were manufactured using a 3D printer.

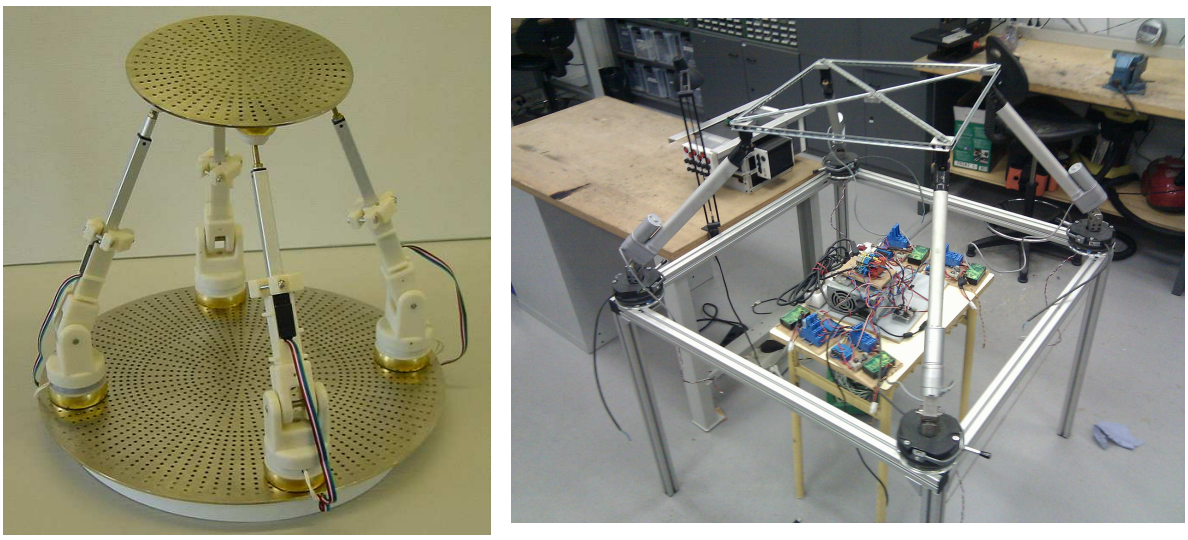


Figure 3.6: The two implemented prototypes.

It was observed that, even when all clutches were engaged, the moving platform was slightly shaky because the lockable joints were not completely stiff. This was directly connected with the diameter of the clutch and the mechanical play of the linear actuators and the clutches. As a consequence the second prototype shown in Fig. 3.6(right) was implemented, using more robust elements. It was implemented using EMTorq industrial clutchs [31] and HDLS-12.00-2.00 (12.00" Stroke, 2.00"/sec., 12V) linear servos from ServoCity [16]. While in the first version, one of the main goals was to have a robot where the **bRR** joints could be easily relocated, in the second version, the relocation objective was left a side, so more permanent anchoring between base-legs and platform-legs were used. Both, platform and base, are squared-shaped, with diagonal tensors pursuing high rigidity, build from metallic Item profiles [13]. Videos of this implementation can be found in [38] and [37].

The first video shows the **4bRRPS** reconfigurable parallel robot implementation performing a maneuver. The maneuver consists of several steps. Each step consists of changing the leg length from their maximum to minimum stroke and vice versa. Between each stroke, the clutches are switched following the sequence, ON($[2,4]$)-OFF($[1,3]$),OFF($[2,4]$)-ON($[1,3]$). The normal behavior of the platform is a rotation around the z-axis. The second video shows the special case where all leg planes, defined by $\hat{\boldsymbol{w}}_{1i}$ (see Fig. 3.1), intersect on a line and the legs move with the same vertical velocity. In this case there is a stagnation of the rotation, no matter neither the number of maneuver iterations, nor which pair of clutches are engaged. In theory they could all be engaged or not and the effect will still occur.

3.6 Software implementation

In order to verify the behavior of the proposed parallel robot and the described path planner, a simulator using MATLAB and Simulink whose output is connected to a VMRL 3D model of the robot was implemented. Using the equations presented in Section 3.2, it simulates the motion of the platform generated by applying the leg lengths settings and the sequence of switchings obtained by the path planner. A typical output of this simulator can be seen by accessing at [39].

Table 3.2: Statistics for the roadmap generated in the example

Number of random configurations	100
Intermediate configurations added	3229
Connections	
Evaluated connections	19070
Possible direct connections	4950
Failed direct connections	2275
Failed after adding one intermediate configuration	644
Established connections	9356
Manoeuvres	
Evaluated (6 per connection)	114420
Discarded	92245
Go outside joint limits	64180
Do not converge to solution	87967
Lead to collisions	201

The data used for the simulation is as follows. The diameters of the base and the platform are 0.4m and 0.2m, respectively. When the legs are extended at half their maximum extension, the platform is located at 0.3m from the base. This is taken as the home configuration. The generated roadmap has been obtained by taking 100 configurations randomly sampled in a region centered at the home configuration with $x \in [-0.04, 0.04]$, $y \in [-0.04, 0.04]$, $z \in [0.115, 0.125]$, $\theta_x \in [-0.05, 0.05]$, $\theta_y \in [-0.05, 0.05]$, and $\theta_z \in [-0.05, 0.05]$ (where distances are given in meters and the orientation angles in radians using the roll-pitch-yaw convention). When each of these configurations have been tried to be connected to all others, 2,275 connections fail (out of the 4,950 possible connections) for the six possible maneuvers. If an intermediate configuration is added in these cases, the amount of failed connections drops to 644. Due to these intermediate configurations, the total number of configurations in the roadmap is 3,229 and the total number of maneuvers checked for validity amounts to 114,420. 92,245 are discarded for different reasons (*e.g.*, out of range of the linear actuator stroke, over the limit of the maximum angle of **U** or **S** joints, etc.). Table 3.2 compiles this information. The same software is used to control

the implementation, the only variation is sending information, via a USB connection, to the real actuators and clutches instead of to the simulator.

3.7 Conclusions

By introducing legs of type $\mathbf{bRR\underline{P}S}$, where $\mathbf{bR}$ stands for a lockable revolute joint, two novel reconfigurable parallel robots of reduced mechanical complexity —the $\mathbf{5bRR\underline{P}S}$ and the $\mathbf{4bRR\underline{P}S}$— have been proposed. Moreover, the $\mathbf{4bRR\underline{P}S}$ has been studied in depth, and a practical implementation of it has been presented.

Regarding the $\mathbf{4bRR\underline{P}S}$ robot, it has been demonstrated that: (*i*) its moving platform can be moved in a six-dimensional operational space by using only four actuators that are maneuvered so that via poses, where the couple of locked $\mathbf{bR}$ pairs is changed, are introduced; (*ii*) the parallel singularities can be avoided and the maximum forces in the actuators can be reduced by suitably managing the insertion of via poses. Eventually, these theoretical results have been verified on the built prototype.

4

Parallel robots with non-holonomic joints

In this chapter, we show how to generate under-actuated manipulators by substituting non-holonomic spherical pairs (**nS** pairs) for (holonomic) spherical pair (**S** pairs) in fully-parallel manipulators. Part of the work presented in this chapter has been published in [43] and in [41].

4.1 Introduction

As already mentioned in the introduction, non-holonomic constraints arise in many different areas of robotics such as motion planning and control of mobile robots, reorientation of free-flying space robots, rolling contacts of multi-fingered hands, etc. In all these cases, the non-holonomic constraints are inherent to the problem, but there are some cases in which the artificial introduction of this kind of constraints can provide important advantages.

In pick-and-place applications of manipulators, only the initial and the final poses (position and orientation) of the end effector are assigned by the task, whereas the end-effector path between them is free. The ideal manipulator for these applications should be able to make the end effector reach any pose in the six-dimensional operational space, and, by exploiting the free fly of the end effector, it should be able to satisfy additional design conditions that reduce its hardware complexity. Joints with non-holonomic constraints do not reduce the reachable relative poses of the links connected by the joint since non-holonomic constraints have the only effect of reducing the set of paths that can be covered between two reachable relative poses. This reduction of practicable paths is accompanied by the rising of new reaction forces in the joint which can be usefully exploited to eliminate actuators. Thus, a manipulator with fewer actuators than the **DoF**[1] of its configuration space —to reduce bulk, weight and expense—

[1]The **DoF** of the configuration space, also called configuration (or finite) **DoF** [3], are the minimum number of geometric parameters necessary to uniquely identify the configuration of the mechanical system [88]. They may be different from the instantaneous **DoF**, also called velocity **DoF** [3], of the same mechanical system.

becomes feasible by introducing mechanical elements that lead to non-holonomic constraints.

As presented in the introduction, the literature on the use of non-holonomic devices in the design of manipulators is limited to few examples. Despite the difference of purpose, all mentioned examples include a surface (cylinder, sphere or plane) in contact with a roller that can freely roll without slipping laterally. This no-slip constraint is a non-holonomic constraint, a constraint relating the velocities of the surface and the roller. For the case sphere-roller, the kinematics of the assembly is equivalent to that of a unicycle on a sphere whose equations of motion can be represented by first-order differential equations [49].

Many research efforts have been made to clarify different aspects of non-holonomic mechanical systems including its controllability, stability, feedback stabilization, time-periodic control, chained form transformation, etc. but, in any case, achieving a formulation for the kinematics of the system, as compact and simple as possible, is essential to explore the applicability of all these results available in the literature.

Herein, the under-actuated parallel architecture presented by Ben-Horin and Thomas in [4] is reconsidered from a different point of view which allows to see it as a particular case of under-actuated manipulator obtained through the substitution of spherical pairs by non-holonomic pairs. Moreover, its kinetostatic analysis is reformulated so that a simple and compact formulation necessary for its design and control is obtained.

This chapter is structured as follows. Next section describes how to generate under-actuated parallel manipulators from fully parallel robots. Section 4.3 gives practical considerations on the implementation of a non-holonomic joints. Finally some conclusions are drawn in Section 4.4.

4.2 Generation of under-actuated manipulators

Two rigid bodies connected by a spherical pair can assume any relative orientation, and can move from one relative orientation to another by covering any spherical-motion path that joins the two relative orientations. Actually, the possibility of freely orientating two rigid bodies with respect to one another is not related to the possibility of performing relative rotations around axes which pass through the center of spherical motion and have any direction. In fact, a suitable sequence of finite rotations around coplanar axes that pass through the spherical-motion center can freely orientate one rigid body with respect to another. Thus, if the only free relative orientation of two rigid bodies is required, the use of a spherical pair will be redundant. The use of a kinematic pair that allows only rotations around coplanar axes that pass through a fixed point would be sufficient.

Due to frictional forces, the rolling contact between a sphere and a roller forbids the sphere rotations around the axis through the sphere center, and perpendicular to the plane defined by

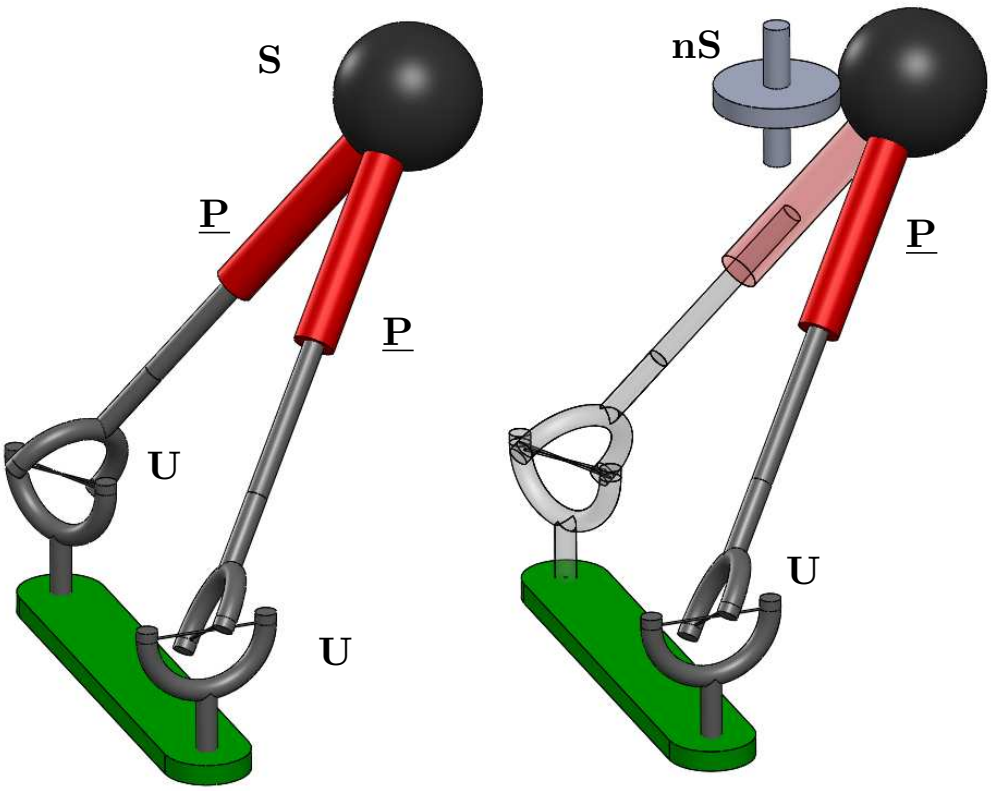


Figure 4.1: **UPnS** limb (right) generated by substituting an **nS** pair for the **S** pair in the two **UPS** limbs with coalesced **S** pairs (left).

the roller axis and the sphere center. By combining such a non-holonomic constraint with other constraints that forbid the relative translation between the sphere center and the roller axis, a non-holonomic joint will result. This joint constrains two rigid bodies: one fixed to the sphere and the other fixed to the plane, defined by the roller axis and the sphere center. Then, the resulting constrained motion permits only relative rotations around axes lying on the above-mentioned plane and passing through the sphere center. Hereafter, this type of joint will be called non-holonomic spherical pair (**nS** pair).

The constraint forces, which two rigid bodies, joined by an **nS** pair, exert on one another through the joint, can be reduced to a resultant force applied on the sphere center and a torque perpendicular to the plane defined by the roller axis and the sphere center. The torque is the static effect of the non-holonomic constraint, whereas the resultant force on the sphere center is the same static effect that an **S** pair would have generated.

From a manufacturing point of view, it is worth noting that, in an **nS** pair, the presence of

any number of roller-sphere contacts does not alter the kinetostatics of the **nS** pair, provided that all the roller axes lie on a same plane passing through the sphere center[2]. Moreover, the maximum torque transmitted through the **nS** pair, due to its frictional origin, can be fixed by suitably choosing the number of roller-sphere contacts together with the normal force transmitted through each contact.

The above discussion brings to the proposition: *i*) the substitution of a number of **nS** pairs for as many **S** pairs in a kinematic chain does not change the configuration space of that chain. That is, neither the **DoF** of the configuration space nor the reachable configurations change[3]. It only reduces the practicable paths for moving that chain from one configuration to another.

Moreover, due to the torque that arises in a **nS** pair in a parallel robot, the substitution of a number of **nS** pairs for as many **S** pairs does not change its workspace and allows the elimination of a number of actuators equal to the number of introduced **nS** pairs (*i.e.*, generates an under-actuated manipulator).

Fully-parallel manipulators feature two platforms, one mobile (end effector) and the other fixed (frame), connected to each other by means of six universal(**U**)-prismatic(**P**)-spherical(**S**) kinematic chains (**UPS** limbs) where the prismatic pairs are the only actuated pairs. In each limb, the centers of the universal joint and of the spherical pair (limb's attachment points) are points, fixed either to the end effector or to the frame, whose distance (limb length) is controlled by the actuated prismatic pair. Two or more attachment points, either in the end effector or in the frame, can coalesce into a unique point. According to the number of attachment points (no matter if they are multiple or not) in the end effector, say **p**, and in the frame, say **q**, different architectures, named **p**-**q**, are distinguished [59].

Due to the high number of **S** pairs appearing in fully parallel robots, the substitutions of **nS** pairs for **S** pairs, accompanied by as many eliminations of actuators in the prismatic pairs, can be operated in many ways in all the fully parallel architectures. By exploiting all the possible substitutions, a lot of new under-actuated parallel manipulators can be generated. It is worth noting that a passive **UPS** limb only affects the workspace borders since it has connectivity six, and, if this effect is not necessary, the elimination of the actuator in a prismatic pair could be accompanied by the elimination of the whole resulting passive **UPS** limb.

[2] In general, two rollers whose axes locate with the sphere center two different planes constrains the sphere to rotate around the intersection line between the two planes, whereas three rollers whose axes locate with the sphere center three different planes lock the sphere.

[3] The presence of non-holonomic constraints does not change the configuration **DoF** [3, 88]. It only affects the instantaneous **DoF** of the mechanism. Hereafter, the acronym **DoF** used alone will mean configuration **DoF**.

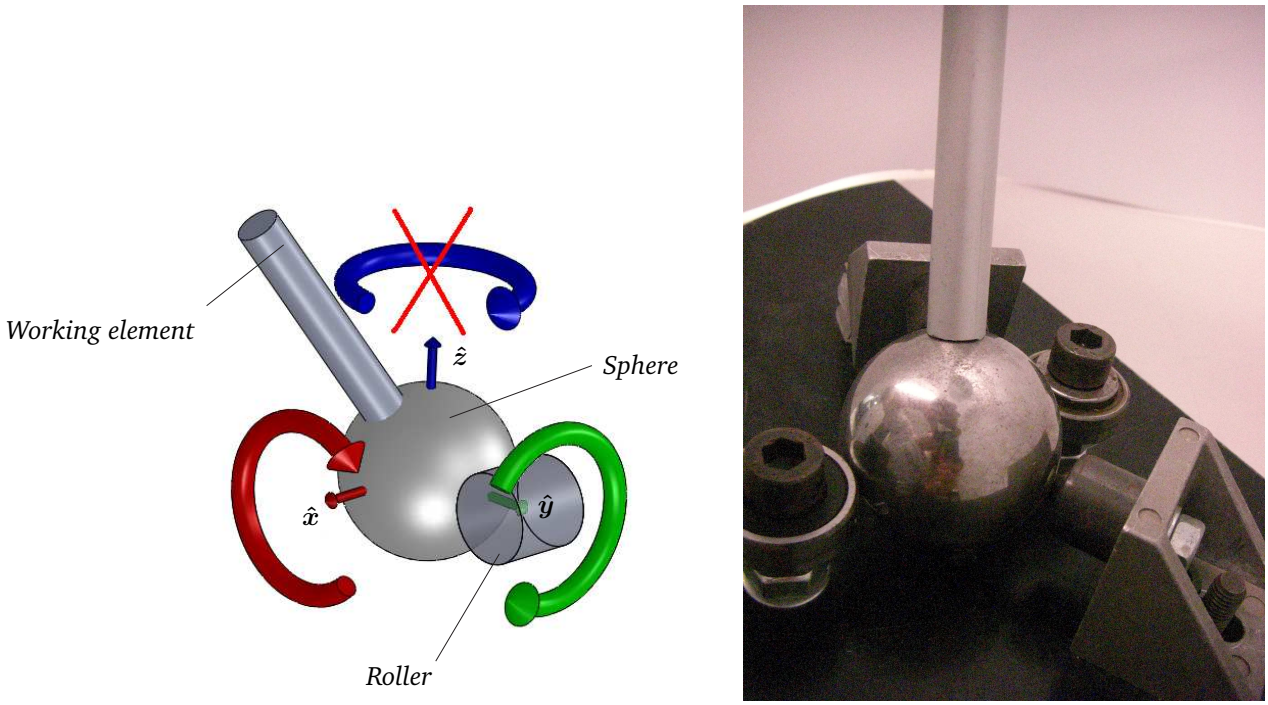


Figure 4.2: Conceptual design (left) and proof of concept (right) of the proposed non-holonomic joint.

4.3 Implementation of non-holonomic joints

The considered non-holonomic spherical joint consists of a spherical joint, an element with **3-DoF** of rotation and no translation, in contact with a roller with its rolling axis fixed and a working element attached the spherical joint (Fig. 4.2). In the case of robots with arquitecture **3nS$\underline{\text{P}}$U**, the working element is a prismatic joint and in the case of the wrist robot **nS-2S$\underline{\text{P}}$U** the working element is the platform (see Chapters 5 and 7 respectively).

To experiment and get some insight into the behavior of the proposed joint, a prove of concept was assembled as shown in Fig. 4.2. Experiments with this assembly led us to establish a set of constrains that the final design must satisfy. Namely,

1. The sphere should be secured to avoid it translating, but letting it to freely rotate.

2. The roller should be put in contact with the sphere with a high contact force. The coefficient of friction and the magnitude of the force will determine the maximum moment that the sphere can undergo before it slips with respect to the roller.

3. The location of the roller and the securing elements must be chosen so that the range of motion of the working element is maximized.

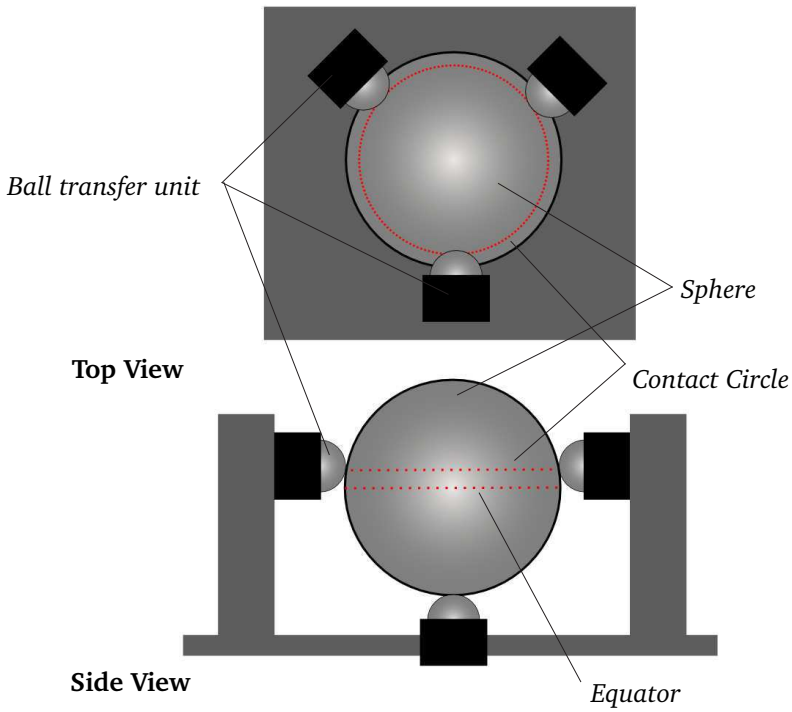


Figure 4.3: Securing the sphere within its housing.

Our design is also limited by the characteristics of the spheres commercially available. The chosen sphere is a stainless steel ball of 50 mm in diameter (RS code 687-629) drilled using an electric discharge machine to fasten the working element. The chosen securing elements are four ball transfer units (RS code 687-679) arranged as shown in Fig. 4.3. This arrangement disables sphere's translation while still allowing it to freely rotate. Observe how three contact points occur over the sphere's equator, thus preventing it from getting out of the housing.

To obtain the non-holonomic effect, we have to place a roller in contact with the sphere. One of the ball transfer units could actually be replaced by a roller. Nevertheless, this is not enough because a high contact force is needed to prevent slippage between the sphere and the roller. The high contact force on the sphere must be counteracted by the remaining ball transfer units, which is not possible as they can only support a low load in order to rotate freely. An alternative solution is to place two opposing rollers in contact with the sphere at polar points so that the counteraction on the ball transfer units is minimized. The introduction of the second roller also doubles the friction force, and in consequence doubles the allowed torque, which adds to the lateral non-slip objective of the design. Nevertheless, this option has an important drawback:

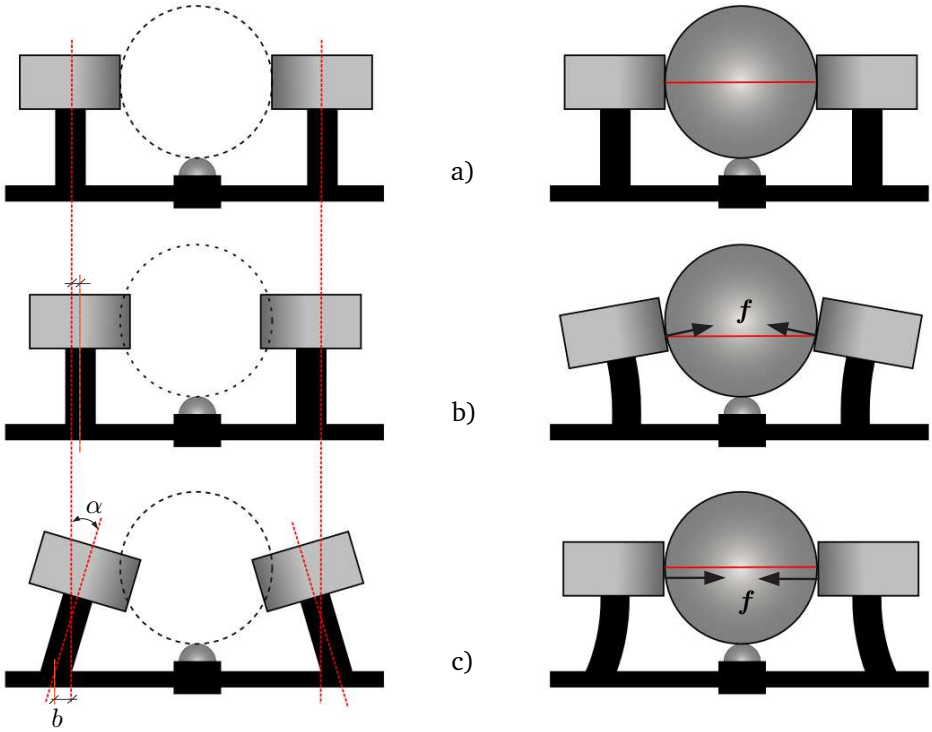


Figure 4.4: To guarantee a high contact force, a device is needed to adjust α and b.

the rollers' axes must be parallel and their contacts with the sphere must be exactly polar, which is not easy to achieve. If any of these two conditions is not satisfied, slippage at the contact points with the roller will occur, or the sphere will only rotate about a single axis. Therefore, the mechanism should include the possibility of both adjusting the orientation of the rollers' axes and their location to ensure the polarity of the contact points.

The required high contact forces between the rollers and the sphere can be obtained by taking advantage of the rollers' stems flexibility, as illustrated in Fig. 4.4. If the rollers are just touching the sphere, no forces are obviously generated [Fig. 4.4(top)]. If the rollers are too close and parallel before placing the sphere, the rollers' axes will not be parallel and the contact points will not be polar [Fig. 4.4(center)]. On the contrary, if the orientations and the anchor points of the rollers' stems can be adjusted by an angle α and a displacement b [Fig. 4.4(bottom)], the stems' rollers will bend when the sphere is placed so that the axes of the rollers become parallel and the contact points are polar. Nevertheless, since the values of α and b are very difficult to determine beforehand, a device to adjust them must be included in the design. In summary, the design, besides including adjustments for the rollers axes orientations and the location of the contact points, should also include adjustments for α and b.

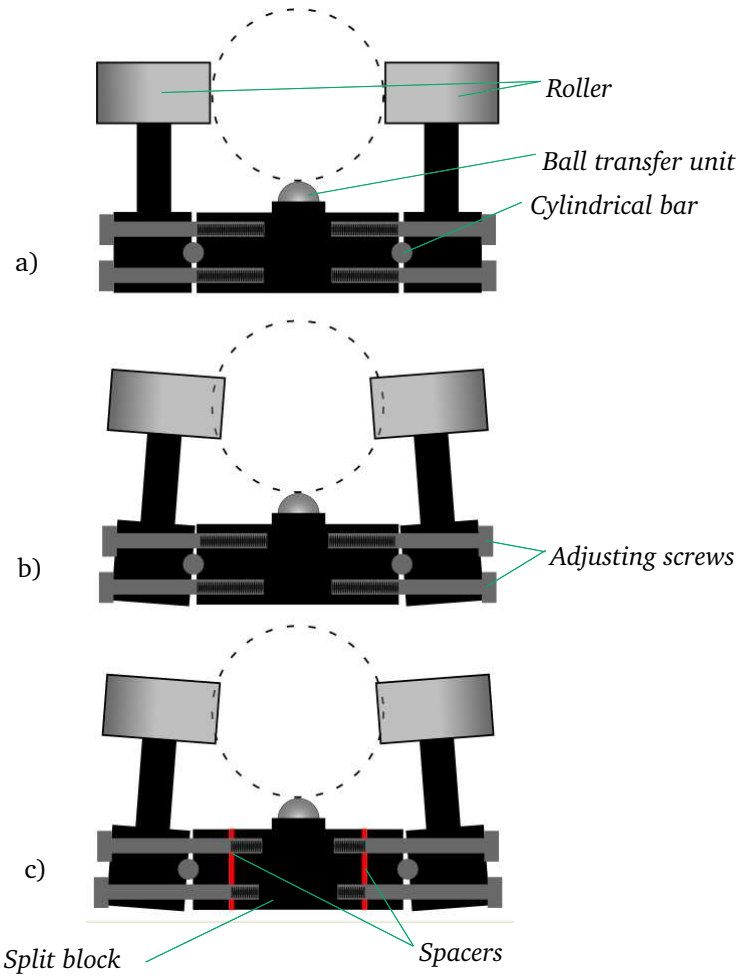


Figure 4.5: Device to adjust angle α and distance b.

Fig. 4.5 shows the device designed to adjust α and b. It consists of a central block and two side blocks to which the rollers are attached. The side blocks are fasten to the central one by eight screws, four per side. Two cylindrical bars are placed between the blocks so that they can pivot on these bars and the angles between them can then be adjusted with the screws. The center block is split into three elements so that we have the possibility to add spacers between them. Adding or subtracting spacers changes the distance b. To adjust the lateral contact points between the rollers and the sphere, we have also added the possibility of adding spacers at the housing of the ball transfer units. This will move the sphere sideways as shown in Fig. 4.6.

As a result of the above analysis, we came out with the CAD design shown in Fig. 4.7(left). Its implementation can be seen in Fig. 4.7(right).

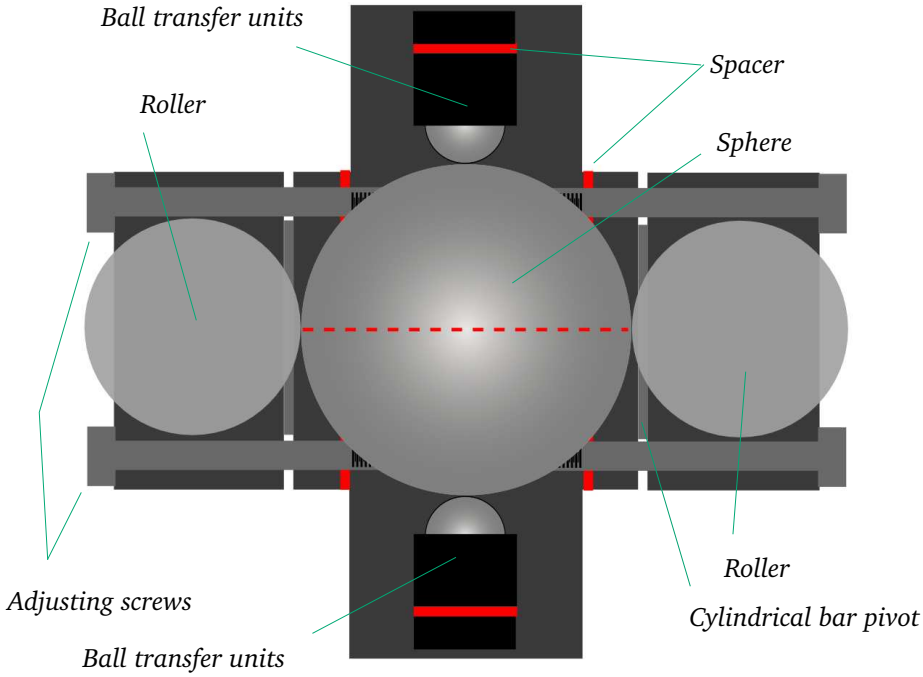


Figure 4.6: Device to adjust the contact points between the rollers and the sphere.

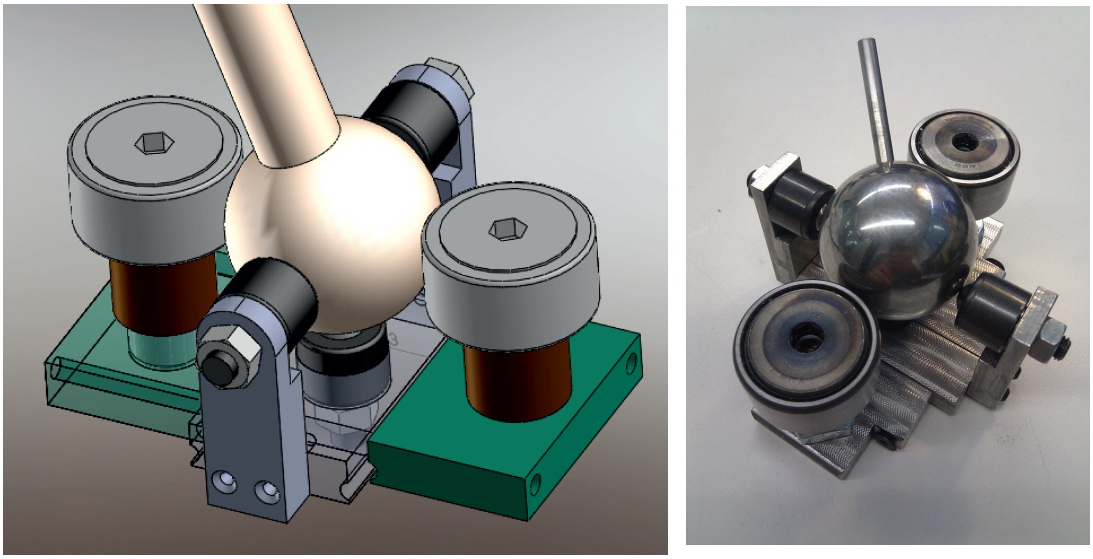


Figure 4.7: CAD design (left) and final implementation (right) of the proposed non-holonomic joint.

4.4 Conclusions

It has been shown that, if a number of non-holonomic spherical pairs replaces as many spherical pairs in a parallel manipulator, the same number of actuators can be eliminated. The resulting manipulator will keep the same workspace of the original manipulator, but it will become under-actuated.

The explained technique for generating under-actuated manipulators has been applied to fully-parallel manipulators, where many spherical pairs are present, and the elimination of an actuator in an **UPS** limb can be accompanied with the elimination of the whole limb.

5

Kinetostatics of the 3nSPU Robot

In this chapter, a simple and compact formulation for the kinetostatics of the **3nSPU** under-actuated robot is presented. This is needed both in the design of this under-actuated robot, and in its control. Part of the work presented in this chapter has appeared in [43] and in [41].

5.1 Introduction

In the previous chapter, we showed how under-actuated manipulators can be generated through the substitution of a spherical pair by a particular non-holonomic pair in ordinary (*i.e.*, not under-actuated) manipulators. In this chapter, as a case study, the under-actuated parallel architecture presented in [4] is demonstrated to be generable from an inversion of the **6-3** fully-parallel manipulator. Its kinetostatic analysis is reformulated, and a simple and compact formulation, useful for its design and control, is obtained.

This rest of this chapter is structured as follows. Section 5.2 is devoted to the case study: a compact formulation for its instantaneous kinematics and statics is obtained, and some clues for the characterization of its singularities and controllability are provided. Section 5.3 presents a numerical example of the behavior of singularities and controllability for a particular **3nSPU**. Eventually, Section 5.4 offers the conclusions.

5.2 The 3nSPU Robot

In this section, an under-actuated parallel manipulator generated from the **6-3 FPM** [Fig. 5.1(top)] is studied.

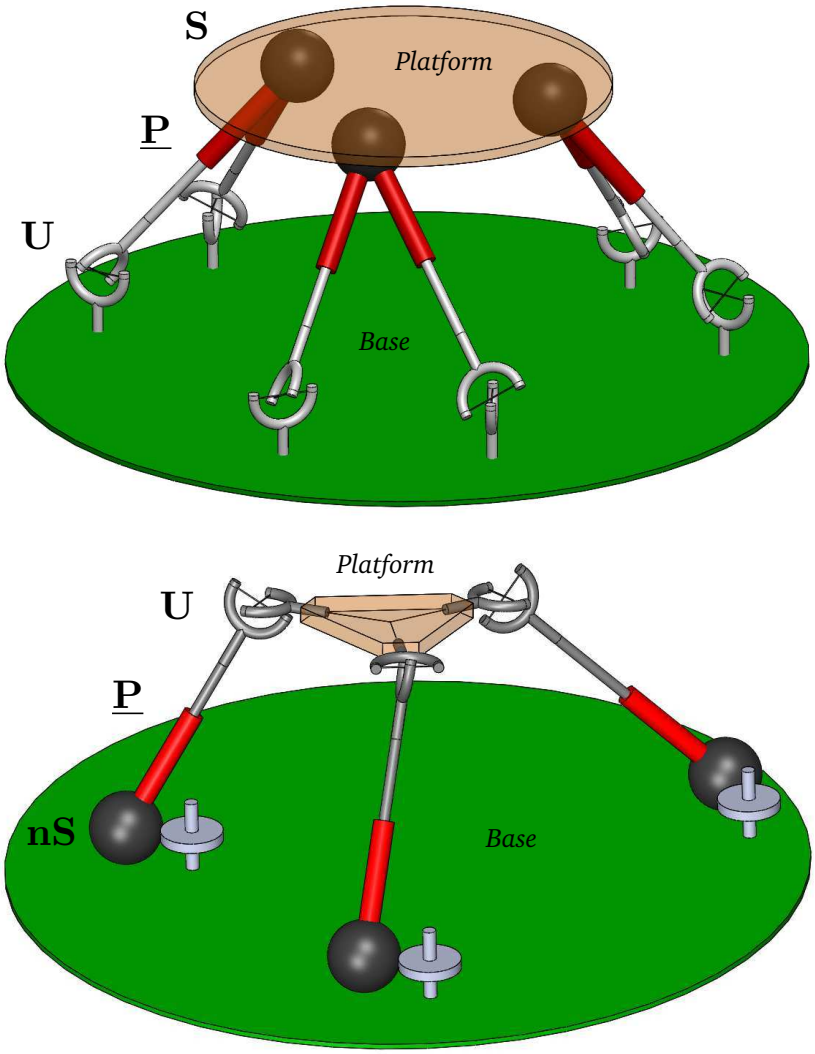


Figure 5.1: A fully parallel robot with **6-3** architecture (top), and under-actuated parallel robot with **3nSPU** architecture resulting from applying the joint substitution presented in Chapter 4 (bottom).

A fully parallel robot with **6-3** architecture features three couples of **UPS** limbs with coalesced **S** pairs in the end effector [Fig. 5.1(top)]. This architecture was proposed first by Stewart [99], in the 1965, for a flight simulator. Successively, with the renewed interest for the parallel architectures, started at the end of the eighties, it was diffusely studied. In particular, regarding the direct position analysis of the **6-3** parallel robot, Innocenti and Parenti-Castelli [57] demonstrated that at most sixteen end-effector poses correspond to a given set of limb lengths. Then, Parenti-Castelli and Di Gregorio [89] demonstrated that the end-effector

pose is uniquely determined when the value of one passive joint variable is measured besides the six limb lengths. The direct position analysis of this parallel manipulator can also be used for spatial parallel manipulators that become **3RS** structures when the actuators are locked (see, for instance, [32, 62, 93, 96]).

Starting from the **6-3** architecture, each couple of **UPS** limbs with coalesced **S** pairs [Fig. 5.1(top)] can be transformed into an **nSPU** limb without affecting the workspace of the manipulator, as explained in the previous chapter. By operating this substitution in all the three couples of **UPS** limbs together with the inversion of the end effector with the frame, the under-actuated manipulator with topology **3nSPU**, shown in Fig. 5.1(bottom), is obtained. This under-actuated manipulator is able to move the end effector in a six dimensional workspace by changing only the three limb lengths.

Regarding the direct position analysis of the obtained robot, since its configuration space has six dimensions, a number of closure equations equal to the number of unknowns can be written if, and only if, over the three limb lengths, three more passive joint variables are measured. By measuring the three joint variables of the three revolute pairs[1] not adjacent to the end effector, the closure equation system coincides with the one of the **6-3** parallel manipulator for assigned limb lengths [89], and admits at most sixteen solution for the end-effector pose. Moreover, if the joint variable of a revolute pair adjacent to the end effector is measured (or coherently assigned) too, only one end-effector pose satisfies the closure equations [89].

5.2.1 Instantaneous kinematics

Fig. 5.2 shows the i^{th} limb, $i = 1, 2, 3$, together with the notation that will be used. $\hat{\boldsymbol{w}}_{1i}$ and $\hat{\boldsymbol{w}}_{2i}$ are two any mutually orthogonal unit vectors fixed to the frame and lying on the plane defined by the roller axis and the center, A_i, of the sphere, in the roller-sphere contact. $\hat{\boldsymbol{w}}_{3i}$ and $\hat{\boldsymbol{w}}_{4i}$ are the two mutually orthogonal unit vectors of the axes of the two revolute pairs constituting the **U** joint. B_i is the center of the **U** joint. $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are the two position vectors which locate the points A_i and B_i, respectively, in a generic Cartesian reference fixed to the frame, whereas $\boldsymbol{p}$ is the position vector of an end-effector point, P, in the same Cartesian reference. θ_{ji}, for $i,\ j = 1, \ldots, 4$, is a joint variable denoting a rotation angle around the joint-axis defined by $\hat{\boldsymbol{w}}_{ji}$, for $i,\ j = 1, \ldots, 4$, and positive if counterclockwise with respect to $\hat{\boldsymbol{w}}_{ji}$. The length of the i^{th} limb is equal to $\|\boldsymbol{b}_i - \boldsymbol{a}_i\|$, and it will be denoted l_i. Moreover, the limb-axis' unit vector, $\hat{\boldsymbol{g}}_i$, and the unit vector, $\hat{\boldsymbol{h}}_i$ ($\hat{\boldsymbol{r}}_i$) normal to the plane located by the **U**-joint's revolute-pair axes (by the

[1] Each **U** joint can be seen as two revolute pairs: one adjacent to, and the other not adjacent to the end effector.

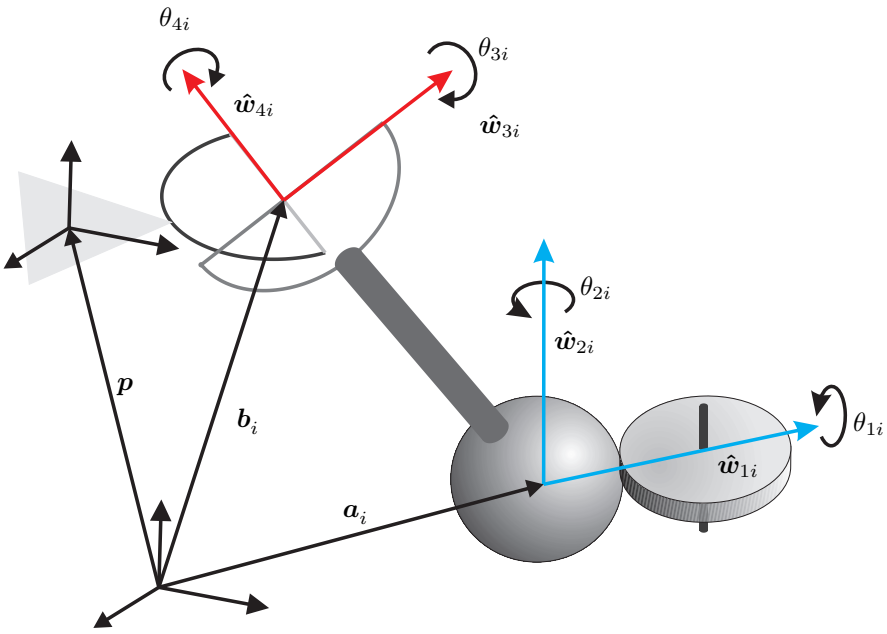


Figure 5.2: Notation associated with the i^{th} limb of the studied **3nSPU** robot.

roller axis together with the sphere center in the **nS** joint) satisfy the following relationships:

$$l_i\hat{\boldsymbol{g}}_i = \boldsymbol{b}_i - \boldsymbol{a}_i, \quad \hat{\boldsymbol{h}}_i = \hat{\boldsymbol{w}}_{3i} \times \hat{\boldsymbol{w}}_{4i} \quad \text{and} \quad \hat{\boldsymbol{r}}_i = \hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{w}}_{2i}. \tag{5.1}$$

The time differentiation of the first of the relationships (5.1) yields

$$\dot{l}_i\ \hat{\boldsymbol{g}}_i + l_i\ \dot{\hat{\boldsymbol{g}}}_i = \dot{\boldsymbol{b}}_i. \tag{5.2}$$

Since $\dot{\hat{\boldsymbol{g}}}_i = (\dot{\theta}_{1i}\ \hat{\boldsymbol{w}}_{1i} + \dot{\theta}_{2i}\ \hat{\boldsymbol{w}}_{2i}) \times \hat{\boldsymbol{g}}_i$, and $\dot{\boldsymbol{b}}_i = \dot{\boldsymbol{p}} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p})$, where $\boldsymbol{\omega}$ denotes the end-effector angular velocity, equation (5.2) can be rewritten as

$$\dot{l}_i\ \hat{\boldsymbol{g}}_i + l_i\ \Big[\dot{\theta}_{1i}\ (\hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{g}}_i) + \dot{\theta}_{2i}\ (\hat{\boldsymbol{w}}_{2i} \times \hat{\boldsymbol{g}}_i)\Big] = \dot{\boldsymbol{p}} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p}). \tag{5.3}$$

The dot products of (5.3) by $\hat{\boldsymbol{w}}_{1i}$ and $\hat{\boldsymbol{w}}_{2i}$ yield the following two scalar equations:

$$\dot{l}_i\ (\hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{1i}) + l_i\ \dot{\theta}_{2i}\ (\hat{\boldsymbol{w}}_{2i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{1i}) = \dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{w}}_{1i} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p}) \cdot \hat{\boldsymbol{w}}_{1i}, \tag{5.4}$$

$$\dot{l}_i\ (\hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{2i}) + l_i\ \dot{\theta}_{1i}\ (\hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{2i}) = \dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{w}}_{2i} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p}) \cdot \hat{\boldsymbol{w}}_{2i}. \tag{5.5}$$

On the other hand (see Fig. 5.2), the end-effector angular velocity is equal to $\sum_{j=1,4} \dot{\theta}_{ji}\ \hat{\boldsymbol{w}}_{ji}$,

whose dot product by $\hat{\boldsymbol{h}}_i$ gives the following expression

$$\boldsymbol{\omega} \cdot \hat{\boldsymbol{h}}_i = \dot{\theta}_{1i} \, (\hat{\boldsymbol{w}}_{1i} \cdot \hat{\boldsymbol{h}}_i) + \dot{\theta}_{2i} \, (\hat{\boldsymbol{w}}_{2i} \cdot \hat{\boldsymbol{h}}_i).$$

Solving (5.4) and (5.5) for $\dot{\theta}_{2i}$ and $\dot{\theta}_{1i}$, respectively, and replacing the result in the above equation, yields:

$$\begin{aligned} \boldsymbol{\omega} \cdot \hat{\boldsymbol{h}}_i = & \left(\frac{\dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{w}}_{2i} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p}) \cdot \hat{\boldsymbol{w}}_{2i} - \dot{l}_i \, \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{2i}}{l_i \, (\hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{2i})} \right) (\hat{\boldsymbol{w}}_{1i} \cdot \hat{\boldsymbol{h}}_i) \\ & + \left(\frac{\dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{w}}_{1i} + \boldsymbol{\omega} \times (\boldsymbol{b}_i - \boldsymbol{p}) \cdot \hat{\boldsymbol{w}}_{1i} - \dot{l}_i \, \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{1i}}{l_i \, (\hat{\boldsymbol{w}}_{2i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{1i})} \right) (\hat{\boldsymbol{w}}_{2i} \cdot \hat{\boldsymbol{h}}_i). \end{aligned} \tag{5.6}$$

Taking into account the vector identities

$$\begin{aligned} \hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i &= \hat{\boldsymbol{h}}_i \times (\hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{w}}_{2i}) = (\hat{\boldsymbol{w}}_{2i} \cdot \hat{\boldsymbol{h}}_i)\hat{\boldsymbol{w}}_{1i} - (\hat{\boldsymbol{w}}_{1i} \cdot \hat{\boldsymbol{h}}_i)\hat{\boldsymbol{w}}_{2i}, \\ \hat{\boldsymbol{r}}_i \times \hat{\boldsymbol{g}}_i &= \hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{w}}_{2i} \cdot \hat{\boldsymbol{g}}_i = -\hat{\boldsymbol{w}}_{1i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{2i} = \hat{\boldsymbol{w}}_{2i} \times \hat{\boldsymbol{g}}_i \cdot \hat{\boldsymbol{w}}_{1i}, \end{aligned}$$

the relationship (5.6) can be rewritten as:

$$\dot{l}_i \, \hat{\boldsymbol{g}}_i \cdot (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) = \dot{\boldsymbol{p}} \cdot (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) + \boldsymbol{\omega} \cdot \left[(\boldsymbol{b}_i - \boldsymbol{p}) \times (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) - l_i \, (\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i \right]. \tag{5.7}$$

Since $\dot{l}_i$ can also be obtained as the projection of $\dot{\boldsymbol{b}}_i$ on $\hat{\boldsymbol{g}}_i$ (see Eq. (5.2)), the following expression holds

$$\dot{l}_i = \dot{\boldsymbol{b}}_i \cdot \hat{\boldsymbol{g}}_i = \dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{g}}_i + \boldsymbol{\omega} \cdot [(\boldsymbol{b}_i - \boldsymbol{p}) \times \hat{\boldsymbol{g}}_i]. \tag{5.8}$$

Replacing expression (5.8) for $\dot{l}_i$ in (5.7), gives

$$\dot{\boldsymbol{p}} \cdot \boldsymbol{s}_i + \boldsymbol{\omega} \cdot \left[(\boldsymbol{b}_i - \boldsymbol{p}) \times \boldsymbol{s}_i - l_i (\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i \right] = 0, \tag{5.9}$$

where

$$\boldsymbol{s}_i = \hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i - \left[\hat{\boldsymbol{g}}_i \cdot (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) \right] \hat{\boldsymbol{g}}_i, \tag{5.10}$$

is the component of $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$ perpendicular to $\hat{\boldsymbol{g}}_i$.

Eventually, rewriting equations (5.8) and (5.9), for $i = 1, 2, 3$, in matrix form yields

$$\begin{pmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{O}_{3\times3} \end{pmatrix} \dot{\boldsymbol{l}} = \begin{pmatrix} \boldsymbol{G}_{3\times3} & \boldsymbol{K}_{3\times3} \\ \boldsymbol{S}_{3\times3} & \boldsymbol{J}_{3\times3} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \tag{5.11}$$

where $\boldsymbol{I}_{3\times3}$ and $\boldsymbol{O}_{3\times3}$ are the 3×3 identity and zero matrix, respectively, $\dot{\boldsymbol{l}} = (\dot{l}_1, \dot{l}_2, \dot{l}_3)$ is the

vector collecting the joint rates of the actuated joints, and

$$\boldsymbol{K}^T[i,:] = (\boldsymbol{b}_i - \boldsymbol{p}) \times \hat{\boldsymbol{g}}_i, \tag{5.12}$$

$$\boldsymbol{G}^T[i,:] = \hat{\boldsymbol{g}}_i, \tag{5.13}$$

$$\boldsymbol{J}^T[i,:] = (\boldsymbol{b}_i - \boldsymbol{p}) \times \boldsymbol{s}_i - l_i(\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i, \tag{5.14}$$

$$\boldsymbol{S}^T[i,:] = \boldsymbol{s}_i, \tag{5.15}$$

with the notation $\boldsymbol{A}^T[i,:]$ to mean the i^{th} column of matrix ${\boldsymbol{A}_{3\times 3}}^T$.

Matrix relationship (5.11) is the sought-after input-output instantaneous relationship necessary to implement the control algorithms of the **3nSPU**.

5.2.2 Statics analysis

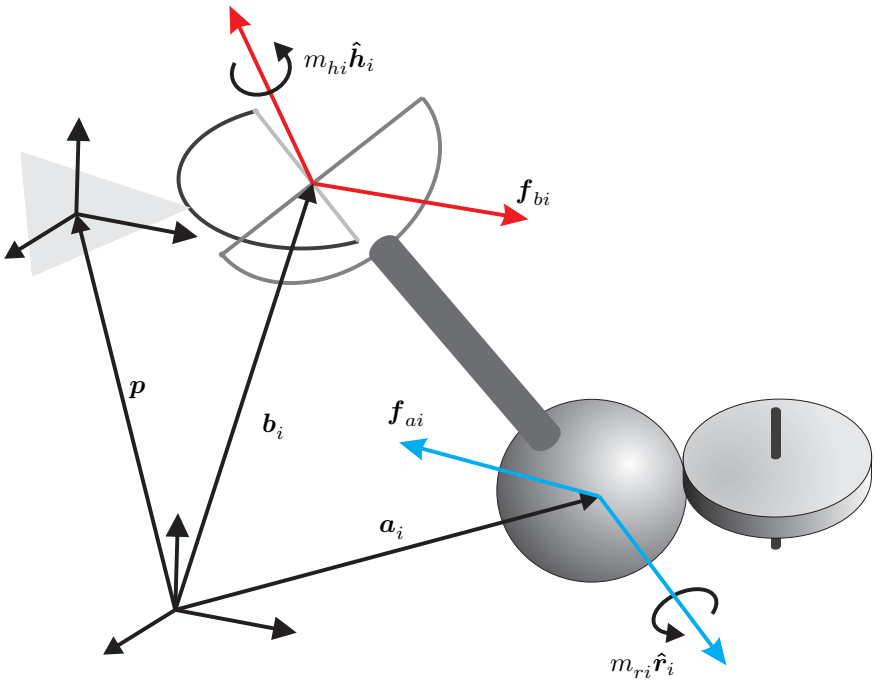


Figure 5.3: Free-body diagram associated with the i^{th} limb of the studied **3nSPU** robot.

The input-output static relationship can be immediately deduced from (5.11) through the principle of virtual work. Nevertheless, in order to highlight how the loads act upon the limbs and are transmitted through the joints, the complete static analysis of the **3nSPU** will be developed here independently of (5.11).

Fig. 5.3 shows the free-body diagram of the i^{th} limb. With reference to Fig. 5.3, the force

$\boldsymbol{f}_{bi}$ ($\boldsymbol{f}_{ai}$), applied on B_i (A_i), together with the torque $m_{hi}\ \hat{\boldsymbol{h}}_i$ ($m_{ri}\ \hat{\boldsymbol{r}}_i$) are the resultants of constraint forces exerted by the end effector (frame) on the i^{th} limb through the **U** joint (the **nS** joint). Moreover, the force $\boldsymbol{f}_{ext}$, applied on the end-effector point P, together with the torque $\boldsymbol{m}_{ext}$ will denote the resultants of the interaction forces exerted on the end effector. The force $\tau_i\ \hat{\boldsymbol{g}}_i$ will denote the axial force exerted on the upper part of the i^{th} limb by the actuator in the prismatic pair. It is worth noting that the force equilibrium, along the limb axis, of the the upper part of the i^{th} limb yields the following relationship $\tau_i = \boldsymbol{f}_{bi} \cdot \hat{\boldsymbol{g}}_i$.

With the introduced notation, the equilibrium of the forces applied on the i^{th} limb yields $\boldsymbol{f}_{bi} + \boldsymbol{f}_{ai} = \boldsymbol{O}$; whereas, taking A_i as reference point, the equilibrium of the moments applied on the same limb is:

$$m_{hi}\ \hat{\boldsymbol{h}}_i + m_{ri}\ \hat{\boldsymbol{r}}_i + l_i\ \hat{\boldsymbol{g}}_i \times \boldsymbol{f}_{bi} = \boldsymbol{O}. \tag{5.16}$$

The dot product of Eq. (5.16) by $\hat{\boldsymbol{g}}_i$, yields the relationship

$$m_{ri} = -m_{hi} \frac{\hat{\boldsymbol{h}}_i \cdot \hat{\boldsymbol{g}}_i}{\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i}, \tag{5.17}$$

whose substitution for m_{ri} in (5.16), gives

$$\frac{m_{hi}}{\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i} \left[(\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i - (\hat{\boldsymbol{h}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{r}}_i \right] + l_i\ \hat{\boldsymbol{g}}_i \times \boldsymbol{f}_{bi} = \frac{m_{hi}}{\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i} \left[\hat{\boldsymbol{g}}_i \times (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) \right] + l_i\ \hat{\boldsymbol{g}}_i \times \boldsymbol{f}_{bi} = \boldsymbol{O}, \tag{5.18}$$

where the vector identity $\hat{\boldsymbol{g}}_i \times (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) = (\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i - (\hat{\boldsymbol{h}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{r}}_i$ has been used.

The dot product of Eq. (5.18) by $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$ yields the relationship: $(\hat{\boldsymbol{g}}_i \times \boldsymbol{f}_{bi}) \cdot (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) = 0$. Such a relationship is satisfied if, and only if, $\boldsymbol{f}_{bi}$ is a linear combination of $\hat{\boldsymbol{g}}_i$ and $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$. Subtracting from $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$ its component along $\hat{\boldsymbol{g}}_i$, $\boldsymbol{s}_i$ is obtained. Since $\hat{\boldsymbol{g}}_i$ and $\boldsymbol{s}_i$ are two orthogonal vectors that span the same subspace as $\hat{\boldsymbol{g}}_i$ and $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$, $\boldsymbol{f}_{bi}$ can be expressed as follows:

$$\boldsymbol{f}_{bi} = \tau_i\ \hat{\boldsymbol{g}}_i + \tau_i^{\perp}\ \boldsymbol{s}_i. \tag{5.19}$$

Equation (5.19) can be interpreted as the equilibrium of the forces applied on the upper part of the i^{th} limb. In fact, the two forces $\tau_i\ \hat{\boldsymbol{g}}_i$ and $\tau_i^{\perp}\ \boldsymbol{s}_i$ are, respectively, the active axial and the passive shear forces applied through the actuated prismatic pair.

Replacing expression (5.19) for $\boldsymbol{f}_{bi}$ in (5.18), and taking into account that $\hat{\boldsymbol{g}}_i \times (\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i) = \hat{\boldsymbol{g}}_i \times \boldsymbol{s}_i$ yields

$$\left(\frac{m_{hi}}{\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i} + l_i\ \tau_i^{\perp} \right) (\hat{\boldsymbol{g}}_i \times \boldsymbol{s}_i) = \boldsymbol{O},$$

which is satisfied if

$$m_{hi} = -l_i\ \tau_i^{\perp}\ (\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i). \tag{5.20}$$

Using equation (5.17), eq. (5.20) can be rewritten as

$$m_{ri} = l_i \, \tau_i^{\perp} (\hat{\boldsymbol{h}}_i \cdot \hat{\boldsymbol{g}}_i). \tag{5.21}$$

Regarding the end-effector equilibrium, the equilibrium of the forces is:

$$\boldsymbol{f}_{ext} = -\sum_{i=1}^{3} \boldsymbol{f}_{bi} = -\sum_{i=1}^{3} \tau_i \, \hat{\boldsymbol{g}}_i - \sum_{i=1}^{3} \tau_i^{\perp} \, \boldsymbol{s}_i, \tag{5.22}$$

and, taking the end-effector point P as reference point, the equilibrium of the moments is:

$$\boldsymbol{m}_{ext} = -\sum_{i=1}^{3} m_{hi} \, \hat{\boldsymbol{h}}_i - \sum_{i=1}^{3} (\boldsymbol{b}_i - \boldsymbol{p}) \times \boldsymbol{f}_{bi}. \tag{5.23}$$

Substituting $\boldsymbol{f}_{bi}$, according to (5.19), and m_{hi}, according to (5.20), yields

$$\boldsymbol{m}_{ext} = -\sum_{i=1}^{3} \tau_i (\boldsymbol{b}_i - \boldsymbol{p}) \times \hat{\boldsymbol{g}}_i - \sum_{i=1}^{3} \tau_i^{\perp} [(\boldsymbol{b}_i - \boldsymbol{p}) \times \boldsymbol{s}_i - l_i \, (\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i) \hat{\boldsymbol{h}}_i] \tag{5.24}$$

Finally, equations (5.22) and (5.24), for $i = 1, 2, 3$, can be rewritten in matrix form as follows:

$$\begin{pmatrix} \boldsymbol{f}_{ext} \\ \boldsymbol{m}_{ext} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{G}_{3\times3} & \boldsymbol{K}_{3\times3} \\ \boldsymbol{S}_{3\times3} & \boldsymbol{J}_{3\times3} \end{pmatrix}^{T} \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\tau}^{\perp} \end{pmatrix}, \tag{5.25}$$

where $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$ is a vector collecting the signed magnitudes of forces applied by the actuators in the prismatic pairs, whereas $\boldsymbol{\tau}^{\perp} = (\tau_1^{\perp}, \tau_2^{\perp}, \tau_3^{\perp})$ from equation 5.19.

Matrix relationship (5.25) is the input-output static relationship of the **3nSPU**. It is worth noting that (5.11) and (5.25) satisfy the instantaneous power balance: $\boldsymbol{f}_{ext} \cdot \dot{\boldsymbol{p}} + \boldsymbol{m}_{ext} \cdot \boldsymbol{\omega} = -\boldsymbol{\tau} \cdot \dot{\boldsymbol{l}}$.

5.2.3 Singularity analysis

Singularities are manipulator configurations where the relationship (input-output instantaneous relationship) between the rates of the actuated-joint variables and the end-effector twist, ($\dot{\boldsymbol{p}}$,$\boldsymbol{\omega}$), fails [34, 108]. Three types of singularities can be distinguished [34]: (*I*) singularities of the inverse kinematic problem, (*II*) singularities of the direct kinematic problem, and (*III*) singularities both of the inverse and of the direct kinematic problems. *Type-I* singularities occur when the actuated joint rates cannot be uniquely computed for an assigned end-effector twist. Vice versa, *type-II* singularities occur when the end-effector twist cannot be uniquely determined for assigned actuated joint rates.

For the **3nSPU**, the input-output instantaneous relationship is (5.11) where the actuated

joint rates are collected in the vector $\dot{\boldsymbol{l}}$. This relationship highlights that the **3nSPU** has only three instantaneous **DoF**. Therefore, its singularity analysis can be addressed by using the scheme proposed in [25].

Regarding *type-I* singularities, provided that the assigned twist, $(\dot{\boldsymbol{p}}, \boldsymbol{\omega})$, satisfies the last three equations of system (5.11), it can be always solved. Note the system (5.11) does not model the mobility limitations due to the physical constitution of the real joints, and to the real sizes of the links. Such limitations bound the workspace and, when correctly modeled, yield *type-I* singularities.

Regarding *type-II* singularities, the equation of the singularity locus is

$$\det \begin{pmatrix} \boldsymbol{G}_{3\times3} & \boldsymbol{K}_{3\times3} \\ \boldsymbol{S}_{3\times3} & \boldsymbol{J}_{3\times3} \end{pmatrix} = 0. \tag{5.26}$$

The geometric interpretation of the above algebraic condition is not straightforward.

Nevertheless, the last three equations of system (5.11) allows the elimination of $\dot{\boldsymbol{p}}$ provided that $\det(\boldsymbol{S}_{3\times3}) = \boldsymbol{s}_1 \cdot \boldsymbol{s}_2 \times \boldsymbol{s}_3$ is different from zero. In this case, system (5.11) becomes

$$\dot{\boldsymbol{l}} = \boldsymbol{Q}\boldsymbol{\omega}, \tag{5.27}$$

where $\boldsymbol{Q}$ is the 3×3 matrix $(\boldsymbol{K}_{3\times3} - \boldsymbol{G}_{3\times3}{\boldsymbol{S}_{3\times3}}^{-1}\boldsymbol{J}_{3\times3})$ Thus, the analytic expression of the singularity locus becomes

$$\det(\boldsymbol{Q}_{3\times3}) = \boldsymbol{q}_1 \cdot \boldsymbol{q}_2 \times \boldsymbol{q}_3 = 0, \tag{5.28}$$

where the vectors $\boldsymbol{q}_i$, for $i = 1, 2, 3$, are the column vectors of matrix $\boldsymbol{Q}$. In conclusion, if the mixed product $\boldsymbol{s}_1 \cdot \boldsymbol{s}_2 \times \boldsymbol{s}_3$ is different from zero (*i.e.*, the three vectors $\boldsymbol{s}_i$, for $i = 1, 2, 3$, are neither coplanar nor null vectors), the *type-II* singularities are geometrically identified by either the coplanarity of the three vectors $\boldsymbol{q}_i$, for $i = 1, 2, 3$, or by the fact that at least one of the $\boldsymbol{q}_i$ vectors is a null vector.

If the mixed product $\boldsymbol{s}_1 \cdot \boldsymbol{s}_2 \times \boldsymbol{s}_3$ is zero, the determinant of the whole 6×6 matrix appearing in (5.26) must be considered, and geometric interpretations of (5.26) are much more difficult to provide.

The zeroing of $\boldsymbol{s}_1 \cdot \boldsymbol{s}_2 \times \boldsymbol{s}_3$ can be geometrically identified since it occurs when either (a) at least one of the $\boldsymbol{s}_i$ vectors is a null vector, or (b) the three $\boldsymbol{s}_i$ vectors are coplanar. Vector $\boldsymbol{s}_i$ (see definition (5.26)) is related to the configuration of the i^{th} limb, and it is the component of $\hat{\boldsymbol{h}}_i \times \hat{\boldsymbol{r}}_i$ perpendicular to $\hat{\boldsymbol{g}}_i$ (*i.e.*, to the limb axis).

As a consequence, condition (a) occurs when, in at least one limb, either $(a.1)$ the two unit vectors $\hat{\boldsymbol{h}}_i$ and $\hat{\boldsymbol{r}}_i$ are parallel (*i.e.*, when, in a limb, the revolute pair axes in the **U** joint are both

parallel to the plane defined by the roller axis and the sphere center in the **nS** pair), or $(a.2)$ the limb axis is the intersection line between the plane, defined by the roller axis and the sphere center in the **nS** pair, and the plane, defined by the revolute pair axes of the **U** joint. Condition $(a.2)$ is forbidden in practice by the actual sizes of joints and links. Regarding condition $(a.1)$, a very special case occurs when $\hat{\boldsymbol{h}}_i$ and $\hat{\boldsymbol{r}}_i$ are parallel in all the limbs. This occurrence makes the matrix $\boldsymbol{S}_{3\times3}$ a null matrix and allows the determinant at the left-hand side of (5.26) to be factorized as $\det(\boldsymbol{G}_{3\times3})\det(\boldsymbol{J}_{3\times3})$ where $\det(\boldsymbol{G}_{3\times3})$ is equal to $\hat{\boldsymbol{g}}_1 \cdot \hat{\boldsymbol{g}}_2 \times \hat{\boldsymbol{g}}_3$, whereas, in this case, $\det(\boldsymbol{J}_{3\times3})$ is equal to $-l_1 l_2 l_3 (\hat{\boldsymbol{r}}_1 \cdot \hat{\boldsymbol{g}}_1)(\hat{\boldsymbol{r}}_2 \cdot \hat{\boldsymbol{g}}_2)(\hat{\boldsymbol{r}}_3 \cdot \hat{\boldsymbol{g}}_3)\hat{\boldsymbol{h}}_1 \cdot \hat{\boldsymbol{h}}_2 \times \hat{\boldsymbol{h}}_3$. Thus, in this case, a *type-II* singularity occurs when either the limb axes are all parallel to a unique plane, or the $\hat{\boldsymbol{h}}_i$ vectors are coplanar, or, finally, in at least one limb, the limb axis lies on the plane defined by the roller axis and the sphere center of the **nS** pair. Moreover, it is worth noting that, in this case, the end effector performs an instantaneous translation, if neither $\det(\boldsymbol{G}_{3\times3})$ nor $\det(\boldsymbol{J}_{3\times3})$ are equal to zero (*i.e.*, out of singularity).

Regarding condition (b) (*i.e.*, the coplanarity of the $\boldsymbol{s}_i$ vectors), it occurs when the limb axes are all parallel, and in other configurations more difficult to visualize.

5.2.4 Local and global controllability

Each end-effector configuration (pose) can be modeled as a point in $\boldsymbol{R}^3 \times \boldsymbol{SO}(3)$ which is locally diffeomorphic to $\boldsymbol{R}^6$ equipped with a proper set of local coordinates: $\boldsymbol{x} = (\boldsymbol{p}^T, \boldsymbol{\eta}^T)^T$ where $\boldsymbol{\eta}$ is a three-dimensional vector collecting the values of the three orientation parameters chosen to locate end-effector's orientation.

By using the orientation parameters' rates, $\dot{\boldsymbol{\eta}}$, the end-effector's angular velocity, $\boldsymbol{\omega}$, can be expressed as:

$$\boldsymbol{\omega} = \boldsymbol{H}_{3\times3}\dot{\boldsymbol{\eta}}. \tag{5.29}$$

Relationship (5.29) allows system (5.11) to be rewritten in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{V}_{6\times6} \begin{pmatrix} \dot{\boldsymbol{l}} \\ \boldsymbol{O}_{3\times1} \end{pmatrix}, \tag{5.30}$$

with

$$\boldsymbol{V}_{6\times6} \triangleq (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_6) = \begin{pmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{H}_{3\times3}{}^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{G}_{3\times3} & \boldsymbol{K}_{3\times3} \\ \boldsymbol{S}_{3\times3} & \boldsymbol{J}_{3\times3} \end{pmatrix}^{-1}, \tag{5.31}$$

where $\boldsymbol{v}_i$ is the i^{th} 6-dimensional column vector of matrix $\boldsymbol{V}_{6\times6}$. The vectors $\boldsymbol{v}_i$ depend only on $\boldsymbol{x}$; so, they are vector fields defined on end-effector's configuration space.

Definition (5.31) allows (5.30) to be further simplified as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{v}_1 \dot{l}_1 + \boldsymbol{v}_2 \dot{l}_2 + \boldsymbol{v}_3 \dot{l}_3. \tag{5.32}$$

Relationship (5.32) states that, in the neighborhood of a generic configuration, $\boldsymbol{x}_0$, all the configurations, $\boldsymbol{x}$, reachable without maneuvering (*i.e.*, without a sequence of coordinated actions of the actuators) are so located that $(\boldsymbol{x} - \boldsymbol{x}_0) \in \text{Span}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$.

(5.32) is the relationship to be considered for discussing end-effector's ability to reach any configuration in the neighborhood of a generic configuration, $\boldsymbol{x}_0$ [11, 82]. The presence of non-holonomic constraints in the **3nSPU** manipulator might allow all neighboring configurations be reachable possibly by maneuvering. If this happens, the system would be "locally controllable" [11] at the configuration $\boldsymbol{x}_0$.

It can be shown (see [82, pp. 323-324]) that, if a system, satisfying (5.32) and at the configuration $\boldsymbol{x}_0$, first follows $\boldsymbol{v}_i$, $i \in \{1, 2, 3\}$, for a small time ε, then follows $\boldsymbol{v}_j$, $j \in \{1, 2, 3 \ : \ j \neq i\}$, for the same time ε, then -$\boldsymbol{v}_i$ for ε, and finally -$\boldsymbol{v}_j$ for ε, it will reach the following configuration of $\boldsymbol{x}_0$'s neighborhood:

$$\lim_{\varepsilon \to 0} \boldsymbol{x}(4\varepsilon) = \boldsymbol{x}_0 + \varepsilon^2 [\boldsymbol{v}_i, \boldsymbol{v}_j], \tag{5.33}$$

where $[\boldsymbol{v}_i, \boldsymbol{v}_j]$ is the 6-dimensional vector field named Lie product of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, defined as follows

$$[\boldsymbol{v}_i, \boldsymbol{v}_j] = \frac{\partial \boldsymbol{v}_j}{\partial \boldsymbol{x}} \boldsymbol{v}_i - \frac{\partial \boldsymbol{v}_i}{\partial \boldsymbol{x}} \boldsymbol{v}_j, \tag{5.34}$$

and the trailing subscript, $\boldsymbol{x} = \boldsymbol{x}_0$, indicates the point the two vector fields, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, are evaluated at.

By reiterating the same reasoning (first, on pairs of vector fields of type $\boldsymbol{v}_i$ and $[\boldsymbol{v}_j, \boldsymbol{v}_k]$, $i, j, k \in \{1, 2, 3 \ : \ i \neq j \neq k\}$, and, successively, on pairs of vector fields belonging to the set which collects all the vector fields that, in the previous iterations, were demonstrated to point from $\boldsymbol{x}_0$ toward reachable configurations), it can be demonstrated that all the vector fields obtained through Lie products of any degree of elements of the set $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ point toward configurations that are reachable by maneuvering from $\boldsymbol{x}_0$ [11]. In other words, for any reachable configuration, say $\boldsymbol{x}$, the vector $(\boldsymbol{x} = \boldsymbol{x}_0)$ belongs to the Lie algebra of $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ (The "Lie algebra" of a set of vector fields is the linear span of all Lie products, of all degrees, of vector fields belonging to that set [11].).

In our case, demonstrating that the dimension of the linear space $Span(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, [\boldsymbol{v}_1, \boldsymbol{v}_2], [\boldsymbol{v}_2, \boldsymbol{v}_3], [\boldsymbol{v}_3, \boldsymbol{v}_1])$ is six[2] is sufficient for concluding that the manipulator

[2] It is worth noting that, if the dimension of $Span(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, [\boldsymbol{v}_1, \boldsymbol{v}_2], [\boldsymbol{v}_2, \boldsymbol{v}_3], [\boldsymbol{v}_3, \boldsymbol{v}_1])$ equals six, all the Lie

is locally controllable at a given configuration since end-effector's configuration space is 6-dimensional (Chow's theorem [82]). Moreover, showing that the set of configurations where the system is locally controllable is a simply connected region is sufficient to demonstrate the existence of finite regions of end-effector's configuration space where, for any two configurations belonging to that region, at least one path exists, which the system can follow, for moving from one configuration to the other (*i.e.*, the system is "globally controllable" in that region).

According to the above discussion, the configurations where the local controllability of our manipulator is not guaranteed are the geometric locus of the roots of the following equation

$$\det[\boldsymbol{L}_{6\times 6}] = 0, \tag{5.35}$$

where

$$\boldsymbol{L}_{6\times 6} \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, [\boldsymbol{v}_1, \boldsymbol{v}_2], [\boldsymbol{v}_2, \boldsymbol{v}_3], [\boldsymbol{v}_3, \boldsymbol{v}_1]]\,. \tag{5.36}$$

The locus of the roots of (5.35) is in general a 5-dimensional variety; thus, a finite region where our manipulator is globally controllable in general exists. This statement will be verified through the following numerical example.

5.3 Example

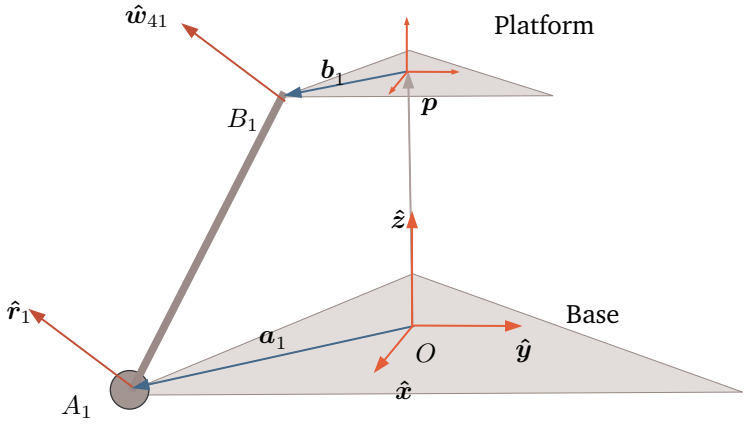


Figure 5.4: Identification of parameters in example.

According with the notation used in the Figs. 5.2 and 5.3 let us consider a **3nSPU** platform

products of any degree in $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ must belong to $Span(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, [\boldsymbol{v}_1, \boldsymbol{v}_2], [\boldsymbol{v}_2, \boldsymbol{v}_3], [\boldsymbol{v}_3, \boldsymbol{v}_1])$; thus, all the reachable configurations, $\boldsymbol{x}$, satisfy the condition $(\boldsymbol{x} - \boldsymbol{x}_0) \in Span(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, [\boldsymbol{v}_1, \boldsymbol{v}_2], [\boldsymbol{v}_2, \boldsymbol{v}_3], [\boldsymbol{v}_3, \boldsymbol{v}_1])$.

where the points A_i (B_i), $i = 1, 2, 3$, are at the vertices of an equilateral triangle fixed to the frame (to the end effector). The Cartesian reference system fixed to the frame (to the end effector) has the origin O (P) at the centroid of the equilateral triangle, the z-axis is perpendicular to the plane of the triangle and, the x-axis pass through A_3 (B_3) with positive direction from A_3 (B_3) toward O (P). In an arbitrary unit of length (**aul**), the distance of the triangle vertices A_i (B_i) from its centroid is 39.7 **aul** (11.76 **aul**). In the same unit, the geometry of the frame and of the end effector are defined by the following data (the vector without any leading superscript are measured in the reference frame, whereas the vectors with the leading superscript ${}^e(\cdot)$ are measured in the end-effector reference frame):

	Leg 1	Leg 2	Leg 3
$\boldsymbol{a}_i$	$[19.84; -34.38; 0.0]^T$	$[19.85; 34.38; 0.0]^T$	$[-39.70; 0.0; 0.0]^T$
$\hat{\boldsymbol{r}}_i$	$[0.353; -0.612; 0.707]^T$	$[0.353; 0.612; 0.707]^T$	$[-0.707; 0.0; 0.707]^T$
$\boldsymbol{b}_i^e$	$[5.883; -10.19; 0.0]^T$	$[5.884; 10.19; 0.0]^T$	$[-11.76; 0.0; 0.0]^T$
$\boldsymbol{w}_{4i}^e$	$[0.353; -0.612; 0.707]^T$	$[0.353; 0.612; 0.707]^T$	$[-0.707; 0.0; 0.707]^T$

With this manipulator geometry, the singularity locus defined by Eq. (5.26) and the root locus of Eq. (5.35) have been computed for a fixed orientation of the end effector with respect to the base reference frame.

The results of these two computations are shown in the Figs. 5.5(top) and 5.5(bottom), respectively. By comparing the singularity locus [Fig. 5.5(top)] and the root locus of Eq. (5.35) [Fig. 5.5(bottom)], a wide free-from-singularity region that is also globally controllable can be identified.

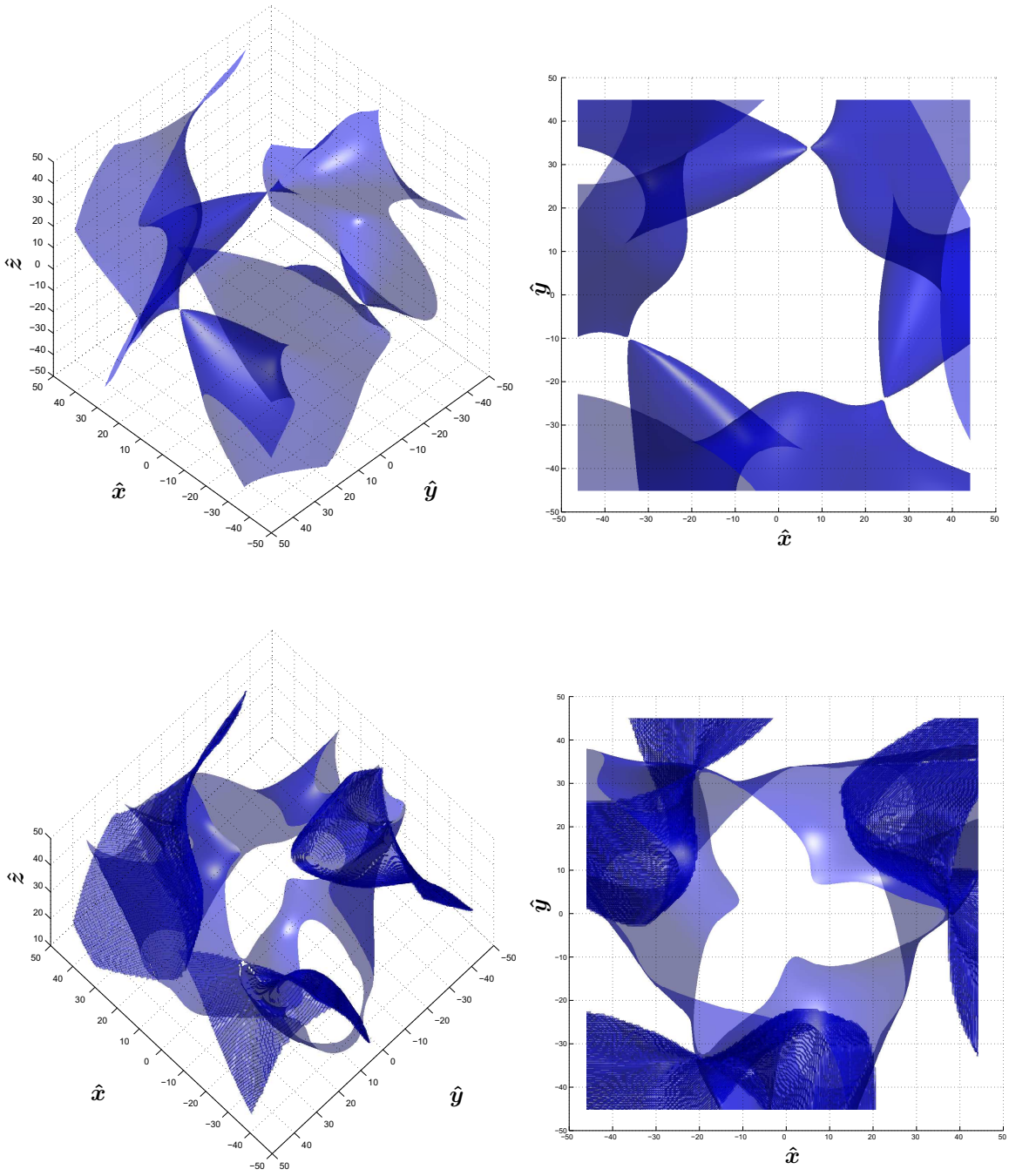


Figure 5.5: Top: Axonometric and zenithal views of the singularity loci, defined by equation (5.26). Bottom: Axonometric and zenithal views of the region, defined by equation (5.35), where the robot is not controllable. In both cases the moving platform is swept in the xyz-space from -50 to 50 **aul**, while keeping its orientation fixed to (0, 1, 0) radians, usingXZX Euler angles.

5.4 Conclusions

In this chapter, a compact formulation for the kinetostatics of the **3nSPU** robot has been presented. Using this formulation, we have identified the regions of its configuration space where it is controllable. This information is needed both in the design of this particular manipulator, and in its control. It will be the base for the path planning algorithm presented in the next chapter.

6

Motion planning for the 3nS$\underline{\text{P}}$U robot

The kinetostatics analysis of the **3nS$\underline{\text{P}}$U** robot allowed us, in the previous chapter, to prove that this robot is able to *locally* move its moving platform —excluding singular configurations— in a six-dimensional configuration space. In this chapter we go a step further by presenting a solution to the motion planning problem for this robot which can be adapted to other non-holonomic parallel robots. This chapter presents some results obtained in cooperation with professors Krzysztof Tchoń and Janusz Jakubiak (Wroclaw University of Technology, Poland), which already appeared in [101].

6.1 Introduction

The motion planning problem for the **3nS$\underline{\text{P}}$U** robot will be addressed using the endogenous configuration space approach [100], specified in [61] to the class of mechanical systems including this parallel non-holonomic robot. The solution to this motion planning problem is decomposed into two steps: first the control system representing the robot's kinematics is subject to a feedback transformation, and afterwards the end-point map of the obtained system is inverted.

This chapter is organized as follows. Section 6.2 introduces the motion planning algorithm whose performance is illustrated in Section 6.3 by a numeric example. Some conclusions are drawn in Section 6.4.

6.2 Motion planning

In the previous chapter, it is proven that the instanteneous kinematics of the **3nS$\underline{\mathbf{P}}$U** robot can be expressed in a very compact way as:

$$\begin{pmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{O}_{3\times3} \end{pmatrix} \dot{\boldsymbol{l}} = \begin{pmatrix} \boldsymbol{G}_{3\times3} & \boldsymbol{K}_{3\times3} \\ \boldsymbol{S}_{3\times3} & \boldsymbol{J}_{3\times3} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \tag{6.1}$$

where $\dot{\boldsymbol{l}} = (\dot{l}_1, \dot{l}_2, \dot{l}_3)^T$ is the vector of velocities in the actuators, $\begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix}$ is the vector of linear and angular velocities of the moving platform, $\boldsymbol{I}_{3\times3}$ and $\boldsymbol{O}_{3\times3}$ are the 3×3 identity and the zero matrix, respectively, and $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{p}, \boldsymbol{R})$, $\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{p}, \boldsymbol{R})$, $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{p}, \boldsymbol{R})$, $\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{p}, \boldsymbol{R})$ are 3×3 matrices dependent on the end-effector pose (position and orientation) $(\boldsymbol{p}, \boldsymbol{R}) \in \boldsymbol{R}^3 \times \boldsymbol{SO}(3)$ whose entries are defined as

$$\begin{aligned} \boldsymbol{K}^T[i,:] &= (\boldsymbol{b}_i - \boldsymbol{p}) \times \hat{\boldsymbol{g}}_i, & \boldsymbol{G}^T[i,:] &= \hat{\boldsymbol{g}}_i, \\ \boldsymbol{J}^T[i,:] &= (\boldsymbol{b}_i - \boldsymbol{p}) \times \boldsymbol{s}_i - l_i(\hat{\boldsymbol{r}}_i \cdot \hat{\boldsymbol{g}}_i)\hat{\boldsymbol{h}}_i, & \boldsymbol{S}^T[i,:] &= \boldsymbol{s}_i, \end{aligned} \tag{6.2}$$

where $\boldsymbol{A}[i,:]$ denotes the i^{th} row of a matrix $\boldsymbol{A}$.

Now, assuming invertibility of the whole block matrix standing on the r.h.s. of (6.1) and taking $\boldsymbol{u} = \dot{\boldsymbol{l}}$ as a control variable, the kinematics model is converted to the driftless control system

$$\dot{\boldsymbol{p}} = \boldsymbol{E}(\boldsymbol{p}, \boldsymbol{R})\boldsymbol{u}, \quad \dot{\boldsymbol{R}} = [\boldsymbol{F}(\boldsymbol{p}, \boldsymbol{R})\boldsymbol{u}]_\times \boldsymbol{R}, \tag{6.3}$$

used in [61], where $[\ \ ]_\times : \boldsymbol{R}^3 \to \boldsymbol{so}(3)$ denotes the standard Lie algebras isomorphism of $\boldsymbol{R}^3$ with the cross product and the space of skew symmetric 3×3 matrices with the matrix commutator, so that $[\boldsymbol{v} \times \boldsymbol{w}]_\times = [\boldsymbol{v}]_\times[\boldsymbol{w}]_\times - [\boldsymbol{w}]_\times[\boldsymbol{v}]_\times = [[\boldsymbol{v}]_\times, [\boldsymbol{w}]_\times]$, and

$$\begin{bmatrix} \boldsymbol{E}(\boldsymbol{p}, \boldsymbol{R}) \\ \boldsymbol{F}(\boldsymbol{p}, \boldsymbol{R}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{K} \\ \boldsymbol{S} & \boldsymbol{J} \end{bmatrix}^{-1} \Bigg|_{\text{3 first columns}}. \tag{6.4}$$

Given the control system (6.3), the motion planning problem for the parallel non-holonomic robot can be stated in the following way: compute a control function $\boldsymbol{u}(t)$ steering the system from an initial end effector pose $(\boldsymbol{p}_0, \boldsymbol{R}_0)$ to the desired one $(\boldsymbol{p}_d, \boldsymbol{R}_d)$ within a prescribed time T. More formally, setting $\boldsymbol{p}(t) = \boldsymbol{p}_{\boldsymbol{p}_0, \boldsymbol{R}_0, t}(\boldsymbol{u}(\cdot))$, $\boldsymbol{R}(t) = \boldsymbol{R}_{\boldsymbol{p}_0, \boldsymbol{R}_0, t}(\boldsymbol{u}(\cdot))$ to be the trajectory of (6.3) starting at $(\boldsymbol{p}_0, \boldsymbol{R}_0)$ and driven by the control $\boldsymbol{u}(t)$, this means that at time T the end-point map of (6.3) assumes the prescribed values $\boldsymbol{p}(T) = \boldsymbol{p}_d$ and $\boldsymbol{R}(T) = \boldsymbol{R}_d$.

Due to the complexity of the matrix entries on the r.h.s. of (6.1), the analytic form of (6.3)

is not very enlightening. To make it more tractable, two regularity assumptions will be made. First, the matrix $\boldsymbol{G}$ will be assumed invertible, resulting in the following form of the system (6.4)

$$\begin{bmatrix} \boldsymbol{E}(\boldsymbol{p},\boldsymbol{R}) \\ \boldsymbol{F}(\boldsymbol{p},\boldsymbol{R}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}^{-1} + \boldsymbol{G}^{-1}\boldsymbol{K}(\boldsymbol{J}-\boldsymbol{S}\boldsymbol{G}^{-1}\boldsymbol{K})^{-1}\boldsymbol{S}\boldsymbol{G}^{-1} \\ -(\boldsymbol{J}-\boldsymbol{S}\boldsymbol{G}^{-1}\boldsymbol{K})^{-1}\boldsymbol{S}\boldsymbol{G}^{-1} \end{bmatrix}. \tag{6.5}$$

The second assumption is the invertibility of $\boldsymbol{S}$. Under this assumption the feedback

$$\boldsymbol{u} = \boldsymbol{G}\boldsymbol{S}^{-1}(\boldsymbol{J}-\boldsymbol{S}\boldsymbol{G}^{-1}\boldsymbol{K})\boldsymbol{v}, \tag{6.6}$$

where $\boldsymbol{v} \in \boldsymbol{R}^3$ is a new control, makes the control system (6.3) equivalent to

$$\dot{\boldsymbol{p}} = \boldsymbol{S}^{-1}\boldsymbol{J}\boldsymbol{v}, \quad \dot{\boldsymbol{R}} = -[\boldsymbol{v}]_\times \boldsymbol{R}. \tag{6.7}$$

Thanks to the above regularity assumptions, the solution of the motion planning problem may be obtained in two steps: first a control $\boldsymbol{v}(t)$ solving the motion planning problem for the system (6.7) is found, and then the original control $\boldsymbol{u}(t)$ is computed using (6.6). The first step can be accomplished in accordance with the guidelines presented in [61], that will be concisely recalled below. Let $\boldsymbol{v}_\theta(t)$ be a family of control functions smoothly dependent on a parameter $\theta \in \boldsymbol{R}$, and

$$\boldsymbol{p}_t(\theta) = \boldsymbol{p}_{\boldsymbol{p}_0,\boldsymbol{R}_0,t}(\boldsymbol{v}_\theta(\cdot)), \quad \boldsymbol{R}_t(\theta) = \boldsymbol{R}_{\boldsymbol{p}_0,\boldsymbol{R}_0,t}(\boldsymbol{v}_\theta(\cdot)),$$

denote the trajectory of the system (6.7) initialized at $(\boldsymbol{p}_0, \boldsymbol{R}_0)$ and subject to the control $\boldsymbol{v}_\theta(t)$. The derivation of the motion planning algorithm for the system (6.7) relies on an assumption that there exists a control family $\boldsymbol{v}_\theta(t)$, such that the error

$$\boldsymbol{e}(\theta) = \left(\boldsymbol{p}_T(\theta) - \boldsymbol{p}_d, \log(\boldsymbol{R}_T(\theta))\boldsymbol{R}_d{}^T\right), \tag{6.8}$$

decreases to zero exponentially along with θ with a prescribed decay rate $\gamma > 0$,

$$\frac{d\boldsymbol{e}(\theta)}{d\theta} = -\gamma \boldsymbol{e}(\theta). \tag{6.9}$$

The logarithm of the rotation matrix in (6.8) is defined as $\log \boldsymbol{R} = \frac{\alpha}{2\sin\alpha}\left(\boldsymbol{R} - \boldsymbol{R}^T\right)$, where $\cos\alpha = \frac{1}{2}(\mathrm{Tr}\,\boldsymbol{R} - 1)$ and the angle of rotation $0 \le \alpha < \pi$.

To proceed, a pair of auxiliary variables will be introduced, denoted by $\boldsymbol{w}_t(\theta)$, $\boldsymbol{s}_t(\theta)$, satisfying the following dependencies

$$\boldsymbol{w}_t(\theta) = \frac{\partial \boldsymbol{p}_t(\theta)}{\partial\theta}, \quad [\boldsymbol{s}_t(\theta)]_\times = \frac{\partial \boldsymbol{R}_t(\theta)}{\partial\theta}\boldsymbol{R}_t{}^T(\theta). \tag{6.10}$$

The differentiation with respect to θ of the matrices on the r.h.s of the system (6.7) results in a collection of differential equations (for details see [61], proof of Theorem 2.1)

$$\begin{pmatrix} \dot{\boldsymbol{w}}_t(\theta) \\ \dot{\boldsymbol{s}}_t(\theta) \end{pmatrix} = \begin{bmatrix} \boldsymbol{A}_{11\theta}(t) & \boldsymbol{A}_{12\theta}(t) \\ \boldsymbol{O} & -[\boldsymbol{v}_\theta(t)]_\times \end{bmatrix} \begin{pmatrix} \boldsymbol{w}_t(\theta) \\ \boldsymbol{s}_t(\theta) \end{pmatrix} + \begin{bmatrix} \boldsymbol{B}_{1\theta}(t) \\ -\boldsymbol{I} \end{bmatrix} \frac{d\,\boldsymbol{v}_\theta(t)}{d\theta}, \tag{6.11}$$

where the entries of the matrices $\boldsymbol{A}_{11\theta}(t)$, $\boldsymbol{A}_{12\theta}(t)$ and $\boldsymbol{B}_{1\theta}(t)$ have been computed on the basis of the data provided in [43]. The assumption that $\boldsymbol{p}_0(\theta) = \boldsymbol{p}_0$ and $\boldsymbol{R}_0(\theta) = \boldsymbol{R}_0$ yields the initial conditions for (6.11) $\boldsymbol{w}_0(\theta) = \boldsymbol{O}$ and $\boldsymbol{s}_0(\theta) = \boldsymbol{O}$. With these initial conditions the solution of (6.11) at T can be represented as

$$\begin{pmatrix} \boldsymbol{w}_T(\theta) \\ \boldsymbol{s}_T(\theta) \end{pmatrix} = \int_0^T \boldsymbol{\Phi}_\theta(T,t)\boldsymbol{B}_\theta(t)\frac{d\,\boldsymbol{v}_\theta(t)}{d\theta}dt, \tag{6.12}$$

where the fundamental matrix $\boldsymbol{\Phi}_\theta(T,t)$ satisfies the evolution equation

$$\frac{\partial \boldsymbol{\Phi}_\theta(t,s)}{\partial t} = \boldsymbol{A}_\theta(t)\boldsymbol{\Phi}_\theta(t,s), \quad \boldsymbol{\Phi}_\theta(s,s) = \boldsymbol{I}_6,$$

and

$$\boldsymbol{A}_\theta(t) = \begin{bmatrix} \boldsymbol{A}_{11\theta}(t) & \boldsymbol{A}_{12\theta}(t) \\ \boldsymbol{O} & -[\boldsymbol{v}_\theta(t)]_\times \end{bmatrix}, \; \boldsymbol{B}_\theta(t) = \begin{bmatrix} \boldsymbol{B}_{1\theta}(t) \\ -\boldsymbol{I} \end{bmatrix}.$$

The integral operator in (6.12) can be regarded as a Jacobian operator of the parallel non-holonomic robot [100]. Now, it has been proved in [61] that the error vanishing formula (6.9) is tantamount to the integral equation

$$\int_0^T \boldsymbol{\Phi}_\theta(T,t)\boldsymbol{B}_\theta(t)\frac{d\,\boldsymbol{v}_\theta(t)}{d\theta}dt = -\gamma \begin{pmatrix} \boldsymbol{p}_T(\theta) - \boldsymbol{p}_d \\ \boldsymbol{r}_T(\theta) \end{pmatrix}, \tag{6.13}$$

where $[\boldsymbol{r}_T(\theta)]_\times = \log(\boldsymbol{R}_T(\theta)\boldsymbol{R}_d{}^T)$. This being so, the motion planning algorithm for the parallel non-holonomic robot is obtained by solving the equation (6.13) using a generalized inverse of the Jacobian. If the Moore-Penrose pseudo inverse is chosen, the resulting differential equation for the control function $\boldsymbol{v}_\theta(t)$ takes the following form

$$\frac{d\boldsymbol{v}_\theta(t)}{d\theta} = -\gamma \boldsymbol{B}_\theta{}^T(t)\boldsymbol{\Phi}_\theta{}^T(T,t)\boldsymbol{D}_\theta{}^{-1} \begin{pmatrix} \boldsymbol{p}_T(\theta) - \boldsymbol{p}_d \\ \boldsymbol{r}_T(\theta) \end{pmatrix}. \tag{6.14}$$

The matrix $\boldsymbol{D}_\theta = \int_0^T \boldsymbol{\Phi}_\theta(T,t)\boldsymbol{B}_\theta(t)\boldsymbol{B}_\theta{}^T(t)\boldsymbol{\Phi}_\theta{}^T(T,t)dt$, is the Gram matrix of the system (6.11). Given the system (6.14), the solution of the motion planning problem is computed by taking the

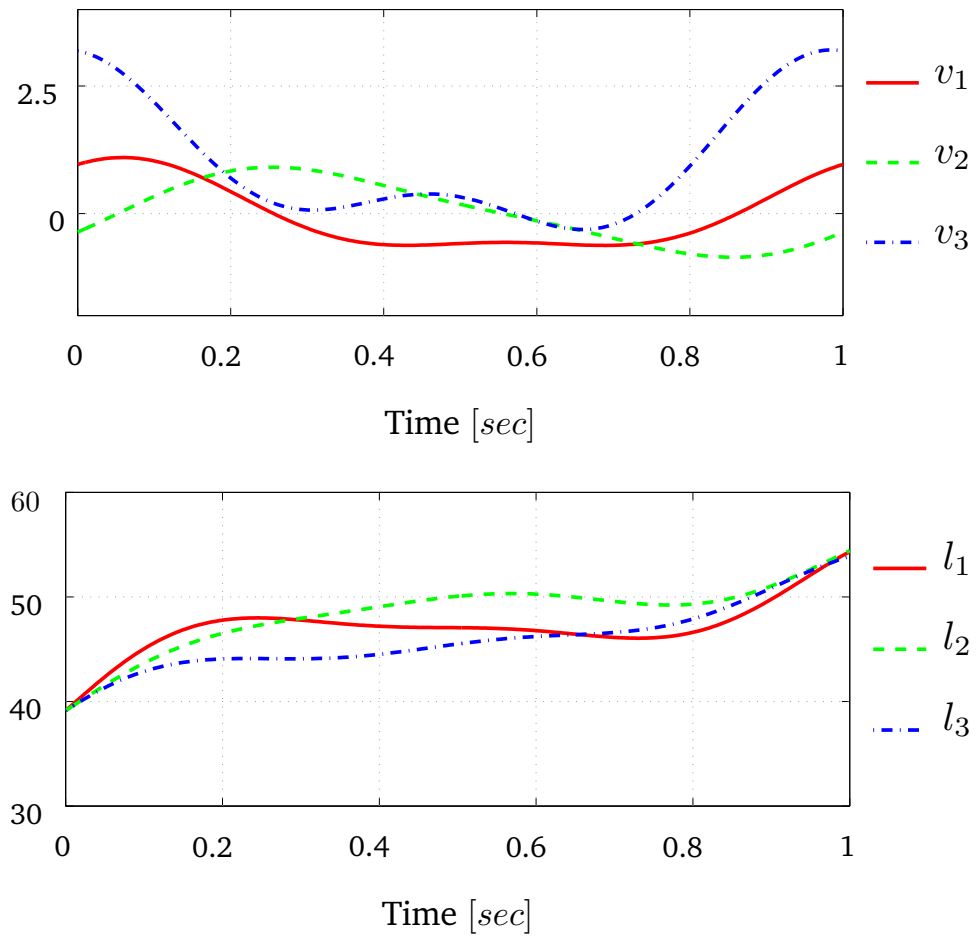


Figure 6.1: Solution of the motion planning problem for the presented example: controls $\boldsymbol{v}(t)$ and leg lengths $\boldsymbol{l}(t)$.

limit $\boldsymbol{v}(t) = \lim_{\theta \to +\infty} \boldsymbol{v}_\theta(t)$. The system (6.7) subject to the control $\boldsymbol{v}(t)$ produces a trajectory $(\boldsymbol{p}(t), \boldsymbol{R}(t))$. A suitable substitutions to the feedback equation (6.6) defines the control $\boldsymbol{u}(t)$ solving the motion planning problem for the parallel non-holonomic robot.

6.3 Example

Since the motion planning algorithm (6.14) operates in an infinite dimensional space of control functions, its computer implementation needs to be preceded by the introduction of a finite dimensional space of controls. This is done in a standard way, by representing the control function by its truncated orthogonal expansion [100]. In this work the truncated Fourier series is

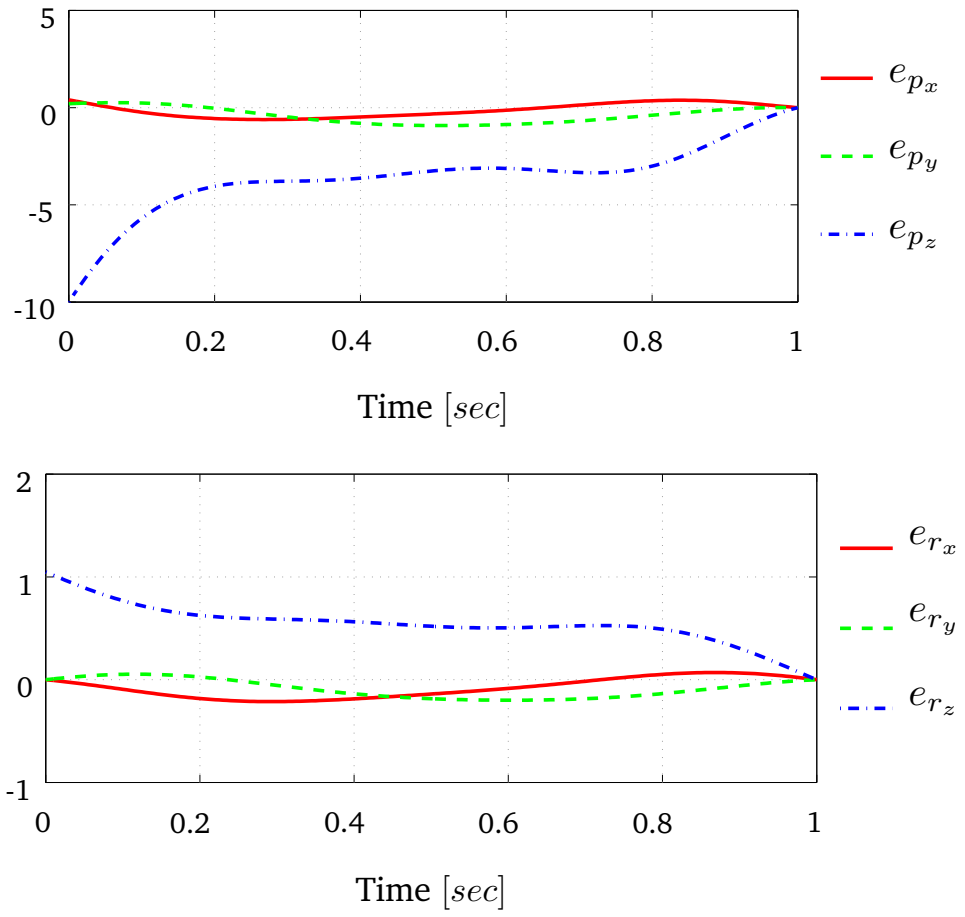


Figure 6.2: Trajectory relative to destination for the presented example: position error $\boldsymbol{e}_p = \boldsymbol{p}(t) - \boldsymbol{p}_d$ and orientation error $\boldsymbol{e}_r(t)$, where $[\boldsymbol{e}_r(t)]_\times = \log \boldsymbol{R}(t)\boldsymbol{R}_d{}^T$.

exploited, so each control $v_i(t)$, $i = 1, 2, 3$, will consist of a constant term and up to h harmonics,

$$v_i(t) = \lambda_{i,0} + \sum_{k=1}^{h} \left(\lambda_{i,2k-1} \sin \frac{2\pi}{T} kt + \lambda_{i,2k} \cos \frac{2\pi}{T} kt \right), \tag{6.15}$$

so the control is finitely parametrized by $\boldsymbol{\Lambda} = (\lambda_{1,0}, \ldots, \lambda_{1,2h}, \ldots, \lambda_{3,0}, \ldots, \lambda_{3,2h})^T \in \boldsymbol{R}^{6h+3}$. In the finite dimensional case the control family takes the form $\boldsymbol{v}_\theta(t) = \boldsymbol{P}(t)\boldsymbol{\Lambda}(\theta)$, where the block matrix $\boldsymbol{P}(t)$ aggregates the basic harmonic functions. Consequently, the differential equation (6.14) underlying the motion planning algorithm determines the control coefficients $\boldsymbol{\Lambda}$

$$\frac{d\boldsymbol{\Lambda}_\theta}{d\theta} = -\gamma \boldsymbol{J}^{\#}_{\boldsymbol{p}_0, \boldsymbol{R}_0, T}(\boldsymbol{\Lambda}_\theta) \begin{pmatrix} \boldsymbol{p}_T(\theta) - \boldsymbol{p}_d \\ \boldsymbol{r}_T(\theta) \end{pmatrix}, \tag{6.16}$$

where $\boldsymbol{J}^{\#}_{\boldsymbol{p}_0,\boldsymbol{R}_0,T}(\boldsymbol{\Lambda}_\theta)$ denotes the Moore-Penrose pseudo inverse of the $6 \times (6h+3)$ Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{p}_0,\boldsymbol{R}_0,T}(\boldsymbol{\Lambda}_\theta) = \int_0^T \boldsymbol{\Phi}_\theta(T,t)\boldsymbol{B}_\theta(t)\boldsymbol{P}(t)dt,$$

of the parallel non-holonomic robot. The differential equation (6.16) should be integrated numerically in accordance with a suitable integration scheme. In the sequel, the simplest Euler scheme will be applied leading to the following difference equation for $\boldsymbol{\Lambda}_\theta$, where $\theta = 0, 1, \ldots$

$$\boldsymbol{\Lambda}_{\theta+1} = \boldsymbol{\Lambda}_\theta - \gamma \boldsymbol{J}^{\#}_{\boldsymbol{p}_0,\boldsymbol{R}_0,T}(\boldsymbol{\Lambda}_\theta) \begin{pmatrix} \boldsymbol{p}_T(\theta) - \boldsymbol{p}_d \\ \boldsymbol{r}_T(\theta) \end{pmatrix}. \tag{6.17}$$

Performance of the motion planning algorithm will be illustrated with a numeric example. The initial position of the platform is $\boldsymbol{p}_0 = (0,0,25)^T$, while its orientation $\boldsymbol{R}_0 = RPY(0,0,-\pi/6)$ corresponds to the Roll-Pitch-Yaw angles $(0,0,-\pi/6)$. The desired end effector position and orientation $\boldsymbol{p}_d = (-0.4,-0.2,35)^T$ and $\boldsymbol{R}_d = RPY(0,0,-\pi/2)$. The initial values of control parameters have been set to 0, except for $\lambda_{11} = \lambda_{21} = \lambda_{32} = 0.5$, $\lambda_{30} = 1$. The planning time horizon $T = 1$. The algorithm has been stopped when the total error $\mathcal{E}(\theta) = \sqrt{\|\boldsymbol{p}_T(\theta) - \boldsymbol{p}_d\|^2 + \|\boldsymbol{r}_T(\theta)\|^2}$ drops below 10^{-3}. In the computations the number h of harmonics is set to two. Results of computations are shown in Figs. 6.1 and 6.2. In Fig. 6.2 the relative trajectories are shown, define as $\boldsymbol{e}_p(t) = \boldsymbol{p}(t) - \boldsymbol{p}_d$ and $[\boldsymbol{r}(t)]_\times = \log(\boldsymbol{R}(t)\boldsymbol{R}_d{}^T)$.

6.4 Conclusion

We have presented a motion planning algorithm for the **3nSPU** robot. The algorithm's synthesis has been based on an application of the endogenous configuration space approach preceded by a feedback transformation of the system (6.3). The presented results provide a novel motion planning algorithm and essentially extend the applicability of the endogenous configuration space approach.

Besides the controllability assumption, the presented algorithm requires two regularity assumptions concerned with the matrices $\boldsymbol{G}$ and $\boldsymbol{S}$. Geometrically, the regularity of $\boldsymbol{G}$ means that vectors $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$ should not be either parallel or co-planar, which is guaranteed by design unless the moving platform of the robot coincides with the base. The regularity of the matrix $\boldsymbol{S}$ does not seem to have so transparent geometric interpretation, and needs a further analysis. In principle, it is possible to devise a motion planning algorithm without these regularity assumptions, however, at the expense of increased computational complexity of the algorithm. The trade-off between regularity and complexity should be subject to further studies.

Since the motion planning algorithm relies on the Jacobian inversion, only a local con-

vergence can be guaranteed. This means that the initial vector of control parameters should be chosen with some care. It is known that in the Jacobian inversion algorithms there is a substitution of locality and the speed of convergence; this question in the context of parallel non-holonomic robots has not been addressed before. An issue traditionally raised in the context of the motion planning algorithm (6.14) refers to the choice of basic control functions and of the integration procedure. Alternative choices to those made in this work can be examined.

The motion planning algorithm proposed in this chapter can be applied only off-line. Speeding up the algorithm may be achieved by a suitable program code optimization and by using a dedicated hardware, but likely it will also require a simplification of the mathematical model of the robot's kinematics and of the motion planning algorithm. This opens up another avenue of future research.

7

The nS-2UPS non-holonomic parallel orienting robot and its kinetostatics

This chapter analyzes the kinetostatic of a non-holonomic parallel orienting robot which can maneuver to reach any orientation for its moving platform. We show how by (*a*) properly locating the actuators, and by (*b*) representing the platform orientation using Euler parameters, the analysis admits a simple bilinear formulation by introducing a local feedback transformation. Interestingly enough, we also show that the singularities introduced by this transformation coincide with the mechanical singularities of the robot system. Thus, no extra singularities are added. A complete description of the robot's workspace is presented, made-up by a singularity free space combined with all joints ranges reachability. Part of the work presented in this chapter has been published in [44].

7.1 Introduction

Consider a mechanism consisting of a sphere whose center is fixed with respect to the world and whose orientation is controlled by three prismatic actuators anchored by their ends to the sphere and the world through spherical joints, as shown in Fig. 7.1 (left). Using the mechanism nomenclature this system can be referred as a **3SPS** mechanism, or **3UPS**, more frequently used as two rotation axis of the **S** joints are always collinear. This kind of mechanism, which can be regarded as a parallel robot, have been studied by several authors due to their practical interest as a robotics wrist or, in general, as an orienting platform. The works of Innocenti and Parenti-Castelli [58], and Wohlhart [105], are usually referred as the pioneering ones on the kinematics analysis of this parallel platform.

As already mentioned in the introduction, in some applications, it is possible to substitute one of the prismatic joints by a disk that rolls without slipping with respect to the sphere as

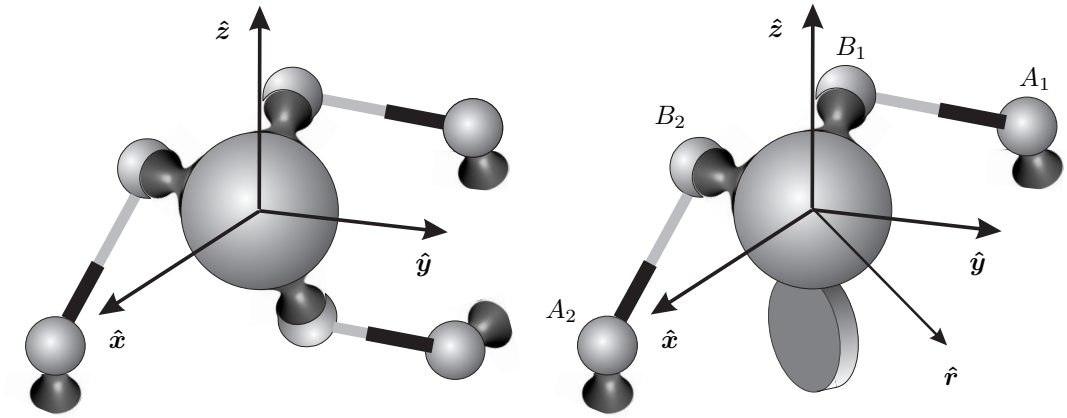


Figure 7.1: In a parallel orienting platform actuated by three prismatic joints (left), one of there joints can be substitute by a disk that rolls without slippage to obtain a non-holonomic parallel orienting platform (right).

shown in Fig. 7.1(right). This idea was first introduced by Stammers in [98], and generalized by Di Gregorio [27] following the ideas presented by Grosch *et al.* [43]. The new mechanism can be referred as a **nS-2UPS** mechanism which belongs to a family of non-holonomic parallel spherical robots studied and developed by different authors (see, for example, [26, 28, 29, 44, 60]).

If the system is analyzed at a first-order kinematic level (the dynamics of the system is not considered), it can be shown that the differential equations that describe the system can be expressed in the standard form of two-input driftless (no motion takes place for null inputs) non-holonomic system. If the dynamics of the system is introduced, the system will exhibit drift but an invertible feedback control can eliminate the dynamic parameters [21]. Therefore, the analysis of the system can be addressed as that of a two-input driftless non-holonomic system. An important class of non-holonomic systems for which a satisfactory understanding has been reached is the class of systems that can be put, by feedback transformation, in the so-called *chained form* [83]. A complete characterization of such systems (*i.e.*, necessary and sufficient conditions for the existence of a feedback transformation to chained-form) has been provided by [81], while an algorithm for finding the necessary coordinate transform has been presented in [103]. This is important in the presented problem because it has been shown that a two-input driftless non-holonomic system with up to four generalized coordinates can always be transformed in chained form [50, 80, 83]. Once in chained form, different methods can be used for motion planning. Essentially two kinds of steering inputs signals have been considered: sinusoidal and piecewise constant. While the first approach was pioneered by [83], the second

is attributed to [69].

Putting a system in chained form is not an easy task and the result is not always satisfactory. The generated feedbacks introduce, in general, singularities that lead to unfeasible control inputs (*i.e.*, infinite steering rates). Moreover, the characterization of these singularities, in the general case, is difficult due to the complexity of the generated expressions. Besides this, the standard procedures to derive chained forms assume that the number of generalized coordinates coincides with the number of degrees of freedom of the system. This leads to an important drawback when working with spatial orientations as they cannot be parameterized by only three parameters without introducing more singularities.

In this chapter, it is shown how, by properly arranging the actuators and representing the platform orientation using Euler parameters, a bilinear model can be derived and this derivation requires an endogenous feedback whose singularities coincide with the mechanical singularities of the platform. Thus, no extra singularities are added. Then, it is shown how this bilinear model admits a closed-form formula for the path planning problem by relying on linear algebra arguments.

This chapter is organized as follows. The kinematic model of the general parallel orienting platform with one non-holonomic joint and two prismatic actuators is derived in Section 7.2. Next, Section 7.3 shows how this model can be expressed in bilinear form by properly arranging the actuators. This bilinear form depends on two matrices whose properties are investigated in Section 7.5. The singularities of the system are analyzed in Section 7.4. In Section 7.6, the robot workspace is analyzed, taking into account its mechanical singularities and the ranges of all joints. Finally, Section 7.7 summarizes the main results and gives some prospects for further research.

7.2 Kinematic model of non-holonomic parallel orienting platforms

7.2.1 Notation

$\boldsymbol{R}$ 3×3 rotation matrix defining the orientation of the moving platform.

$\boldsymbol{\omega}$ vector of angular velocities.

$\hat{\boldsymbol{r}}$ unit vector of the non-holonomic constraint. Rotations about this axis are forbidden.

$\boldsymbol{a}_i$ position vector of leg attachment i to the base in the base reference frame.

$\boldsymbol{b}_i^e$ position vector of leg attachment i to the moving platform in the moving platform reference frame.

$\boldsymbol{b}_i$ position vector of leg attachment i to the moving platform in the base reference frame. $\boldsymbol{b}_i = \boldsymbol{R}\boldsymbol{b}_i^e$.

l_i length of leg i. That is, $l_i = \|\boldsymbol{b}_i - \boldsymbol{a}_i\|$.

$\hat{\boldsymbol{g}}_i$ unit vector in the direction of leg i. That is, $\hat{\boldsymbol{g}}_i = \frac{\boldsymbol{b}_i - \boldsymbol{a}_i}{l_i}$.

7.2.2 Holonomic constraints

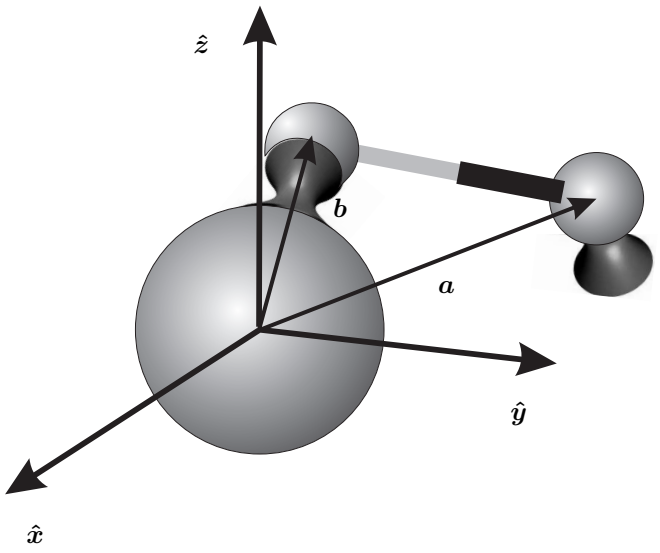


Figure 7.2: A holonomic constraint is imposed on a freely rotating sphere by attaching a prismatic actuator anchored by its ends to the rotating body and the world through spherical joints.

Let us suppose a sphere that rotates ω_x rad/s, ω_y rad/s, and ω_z rad/s, about the x, y, and z axes, respectively. The linear velocity, due to these angular velocities, of a point attached to this sphere with reference position vector $\boldsymbol{b}$ is

$$\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{b},$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$. Then, the linear velocity of this point along the direction given by the unit vector $\hat{\boldsymbol{g}}$ is

$$\dot{l} = \hat{\boldsymbol{g}} \cdot (\boldsymbol{\omega} \times \boldsymbol{b}) = \boldsymbol{\omega} \cdot (\boldsymbol{b} \times \hat{\boldsymbol{g}}). \tag{7.1}$$

Now, if we introduce a prismatic actuator anchored by its ends to the rotating sphere and the world through spherical joints, as depicted in Fig. 7.2, **1-DoF** of the rotating sphere is constrained according to (7.1), where $\hat{\boldsymbol{g}}$ is a unit vector in the direction of the actuator and $\dot{l}$, its linear velocity.

7.2.3 Non-holonomic constraints

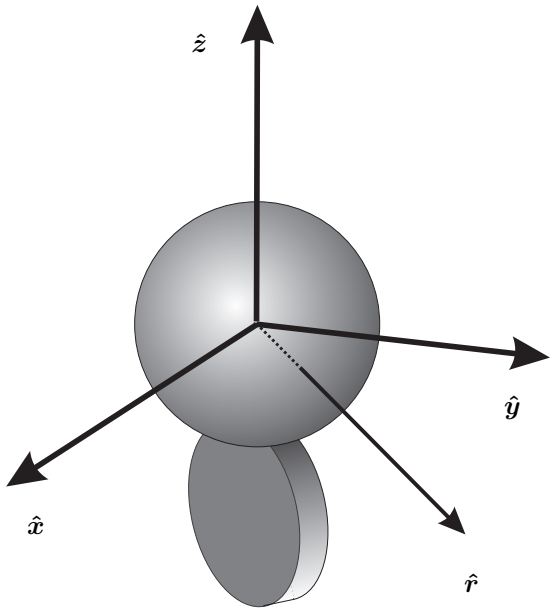


Figure 7.3: A non-holonomic constraint is imposed on a freely rotating sphere by putting in contact with it a disk that freely rolls without slipping.

Alternatively to the holonomic constraint introduced above, we can also constrain the motion of the freely rotating sphere by putting in contact with it a disk that rolls without slipping as shown in Fig. 7.3. This disk prevents the sphere to rotate about the axis oriented in the direction of the wheel. In other words,

$$\boldsymbol{\omega} \cdot \hat{\boldsymbol{r}} = 0. \tag{7.2}$$

7.2.4 Constraining the motion of a sphere

Now, let us consider the case depicted in Fig. 7.1(right) in which the rotation of the sphere is constrained by two actuated prismatic joints and a disk. In this case, the angular velocity of the

sphere must satisfy the following system of equations (see equations 7.2 and 7.1)

$$\left.\begin{array}{l} \dot{l_1} = \boldsymbol{\omega} \cdot (\boldsymbol{b}_1 \times \hat{\boldsymbol{g}}_1) \\ \dot{l_2} = \boldsymbol{\omega} \cdot (\boldsymbol{b}_2 \times \hat{\boldsymbol{g}}_2) \\ 0 = \boldsymbol{\omega} \cdot \hat{\boldsymbol{r}} \end{array}\right\},$$

which can be expressed in matrix form as

$$\boldsymbol{J}\boldsymbol{\omega} = \begin{pmatrix} \dot{l_1} \\ \dot{l_2} \\ 0 \end{pmatrix},$$

where

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{b}_1 \times \hat{\boldsymbol{g}}_1 & \boldsymbol{b}_2 \times \hat{\boldsymbol{g}}_2 & \hat{\boldsymbol{r}} \end{pmatrix}^T. \tag{7.3}$$

Since $\hat{\boldsymbol{g}}_i = (\boldsymbol{b}_i - \boldsymbol{a}_i)/l_i$ and $\boldsymbol{b}_i = \boldsymbol{R}\boldsymbol{b}_i^e$, the above expression for $\boldsymbol{J}$ can be rewritten as:

$$\boldsymbol{J} = \begin{pmatrix} 1/l_1 & 0 & 0 \\ 0 & 1/l_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_1 \times \boldsymbol{R}\boldsymbol{b}_1^e & \boldsymbol{a}_2 \times \boldsymbol{R}\boldsymbol{b}_2^e & \hat{\boldsymbol{r}} \end{pmatrix}^T. \tag{7.4}$$

Therefore,

$$\boldsymbol{\omega} = \boldsymbol{K} \begin{pmatrix} \dot{l}_1 l_1 \\ \dot{l}_2 l_2 \end{pmatrix}, \tag{7.5}$$

where

$$\boldsymbol{K} = \left[\begin{pmatrix} \boldsymbol{a}_1 \times \boldsymbol{R}\boldsymbol{b}_1^e & \boldsymbol{a}_2 \times \boldsymbol{R}\boldsymbol{b}_2^e & \hat{\boldsymbol{r}} \end{pmatrix}^T \right]^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{7.6}$$

7.3 Deriving a bilinear model

Although three is the minimum number of parameters required to describe the kinematics of a rotating rigid body, every such three-dimensional parametrization of the motion is singular. This is the case of the Euler angles and the Cayley-Rodrigues parameters. Alternatively, a non-singular parameterization is possible by using four parameters. This is the case of the Euler

parameters defined as

$$\boldsymbol{q} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \cos\frac{\phi}{2} \\ n_x \sin\frac{\phi}{2} \\ n_y \sin\frac{\phi}{2} \\ n_z \sin\frac{\phi}{2} \end{pmatrix}, \tag{7.7}$$

where $\hat{\boldsymbol{n}} = (n_x,\ n_y,\ n_z)^T$ is the equivalent axis of rotation and ϕ, the angle rotated about it. From this definition, one can easily derive the following constraint

$$\|\boldsymbol{q}\|^2 = a^2 + b^2 + c^2 + d^2 = 1. \tag{7.8}$$

See [85] and [86] for a detailed analysis of Euler parameter and their connections with other parameterizations.

It can be shown that the rotation matrix, in terms of Euler parameters, can be expressed as

$$\boldsymbol{R} = 2\begin{pmatrix} a^2 + b^2 - \frac{1}{2} & bc - ad & bd + ac \\ bc + ad & a^2 + c^2 - \frac{1}{2} & cd - ab \\ bd - ac & cd + ab & a^2 + d^2 - \frac{1}{2} \end{pmatrix}. \tag{7.9}$$

If we substitute this parametrization of $\boldsymbol{R}$ in (7.6), the result is rather awkward. Nevertheless, an important simplification is attained if the anchor points of the prismatic actuators are oriented at $\pi/2$ one from each other in their local reference frames. For example, if we set $\boldsymbol{a}_1 = (1, 0, 0)^T$, $\boldsymbol{a}_2 = (0, 1, 0)^T$, $\boldsymbol{b}_1^e = k\boldsymbol{a}_1$, and $\boldsymbol{b}_2^e = k\boldsymbol{a}_2$, the substitution of (7.9) in (7.6) yields

$$\boldsymbol{K} = \frac{2k}{\det(\boldsymbol{J})}\begin{pmatrix} -r_2(ad + bc) + r_3(ac - bd) & r_2(ad - bc) \\ r_1(ad + bc) & r_1(-ad + bc) + r_3(ab + cd) \\ -r_1(ac - bd) & -r_2(ab + cd) \end{pmatrix},$$

where $\hat{\boldsymbol{r}} = (r_1, r_2, r_3)^T$. Further simplifications are still possible by properly locating the disk. For example, if we set $\hat{\boldsymbol{r}} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right)^T$, then

$$\boldsymbol{K} = \frac{\sqrt{2}k}{\det(\boldsymbol{J})}\begin{pmatrix} -ad - bc & ad - bc \\ ad + bc & bc - ad \\ bd - ac & -ab - cd \end{pmatrix}. \tag{7.10}$$

Since the relationship between angular velocities and time derivatives of Euler parameters

is given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix}\boldsymbol{\omega}, \tag{7.11}$$

the substitution of (7.5), with the expression of $\boldsymbol{K}$ given in (7.10), in (7.11) yields

$$\dot{\boldsymbol{q}} = \begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix}\begin{pmatrix} -ad-bc & ad-bc \\ ad+bc & bc-ad \\ bd-ac & -ab-cd \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \tag{7.12}$$

where

$$u_i = \left(\frac{\sqrt{2}kl_i}{\det(\boldsymbol{J})}\right)\dot{l}_i. \tag{7.13}$$

Equation (7.13) can be seen as a transformation in the input variables. It actually represents a local feedback transformation because both $\det(\boldsymbol{J})$ and l_i depend on $\boldsymbol{q}$. Observe that this change of inputs is singular at the mechanical singularities of the platform, that is, at those orientations in which $\det(\boldsymbol{J}) = 0$. These singularities are studied in Section 7.4.

Now, let us define the transformation in the new input variables defined by

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} ad+bc & bc-ad \\ bd-ac & -ab-cd \end{pmatrix}^{-1}\begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{7.14}$$

This is also a local feedback transformation because it depends on the orientation of the platform. Those orientations for which the matrix inverse in (7.14) is not defined are singularities introduced by this transformation. These singularities are also analyzed in Section 7.4 where it is shown that they coincide with the mechanical singularities of the platform. With this input transformation, (7.12) can be rewritten as

$$\dot{\boldsymbol{q}} = \begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix}\begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

or, alternatively, as

$$\dot{\boldsymbol{q}} = (\boldsymbol{A}v_1 + \boldsymbol{B}v_2)\,\boldsymbol{q}, \tag{7.15}$$

where

$$\boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{7.16}$$

and

$$\boldsymbol{B} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{7.17}$$

This corresponds to the model of a driftless bilinear system with two inputs and four states, but it is not a minimal representation because the four states are not independent. They must satisfy (7.8). That is, $\boldsymbol{q} \in \boldsymbol{S}^3$ where $\boldsymbol{S}^3 = \{\boldsymbol{x} \in \boldsymbol{R}^3, \|\boldsymbol{x}\|^2 = 1\}$. This dependency is already implicit in (7.15). To make it explicit, let us derive (7.8) with respect to time to obtain

$$\boldsymbol{q}^T \dot{\boldsymbol{q}} = 0. \tag{7.18}$$

Then, by substituting (7.15) in (7.18), we have

$$v_1 \boldsymbol{q}^T \boldsymbol{A} \boldsymbol{q} + v_2 \boldsymbol{q}^T \boldsymbol{B} \boldsymbol{q} = 0.$$

Since the above equation must hold for any value of v_1 and v_2, it can be concluded that

$$\boldsymbol{q}^T \boldsymbol{A} \boldsymbol{q} = 0 \tag{7.19}$$

and

$$\boldsymbol{q}^T \boldsymbol{B} \boldsymbol{q} = 0, \tag{7.20}$$

but the quadratic form of a matrix is identically 0 if, and only if, the matrix is skew-symmetric, as is our case.

7.4 Singularities

The mechanical singularities of the studied platform are the set of orientations in which $\det(\boldsymbol{J}) = 0$. From (7.3), it can be concluded that they correspond to those orientations in which the vectors $\boldsymbol{a}_1 \times \boldsymbol{b}_1$, $\boldsymbol{a}_2 \times \boldsymbol{b}_2$, and $\hat{\boldsymbol{r}}$ lie on a plane. For the special configuration with bilinear formulation, the expansion of (7.4) in terms of Euler parameters permits to formulate this

geometric condition in algebraic terms as

$$d^2a(b-c)+c^2b(a-d)-b^2c(a+d)-a^2d(b+c)=0. \tag{7.21}$$

The substitution of these parameters by their definition given in (7.7) yields

$$(\cos\phi-1)(p\cos\phi+q\sin\phi+r)=0,$$

where

$$\begin{aligned}
p &= n_xn_z(1-n_y^2)+n_yn_z(1-n_x^2),\\
q &= n_y(1-n_y^2)-n_x(1-n_x^2),\\
r &= n_xn_z(1+n_y^2)+n_yn_z(1+n_x^2).
\end{aligned}$$

Then, the configuration is singular if, and only if, $\phi=0$, or

$$\phi=\operatorname{atan2}(q,p)\pm\arccos\left(\frac{-r}{\sqrt{p^2+q^2}}\right).$$

To derive the bilinear model presented in Section 7.3, two input transformations are needed that might introduce extra singularities. The first input transformation (7.13) is only singular in a mechanical singularity, so it does not introduce any new singularity. The second input transformation (7.14) is apparently more complicated but the expansion of the determinant of the matrix that depends on the configuration yields

$$-a^2d(b+c)-b^2c(a+c)+c^2b(d-a)+d^2a(c-b)=0,$$

which is identical to (7.21), so it does not introduce any new singularity either.

In Section 7.6, the singularities will be revisited in a more general setting.

7.5 A, B, and rotations in R^4

Let us define

$$\boldsymbol{C}=\boldsymbol{A}\boldsymbol{B}=\frac{1}{\sqrt{2}}\begin{pmatrix}0&1&1&0\\-1&0&0&1\\-1&0&0&-1\\0&-1&1&0\end{pmatrix}. \tag{7.22}$$

Then, it can be checked that

$$\boldsymbol{A}^2 = \boldsymbol{B}^2 = \boldsymbol{C}^2 = \boldsymbol{A}\boldsymbol{B}\boldsymbol{C} = -\boldsymbol{I}. \tag{7.23}$$

Hamilton called quadruples with these rules of multiplication a *quaternion*. Actually, (7.23) reproduces the celebrated formula that Hamilton carved into the stone of Brougham Bridge. Therefore, the real linear span of $\{\boldsymbol{I}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ is isomorphic to the real algebra of quaternions. As with standard quaternions, (7.23) determines all the possible products of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ resulting in

$$\begin{aligned} \boldsymbol{A}\boldsymbol{B} &= \boldsymbol{C}, & \boldsymbol{B}\boldsymbol{A} &= -\boldsymbol{C}, \\ \boldsymbol{B}\boldsymbol{C} &= \boldsymbol{A}, & \boldsymbol{C}\boldsymbol{B} &= -\boldsymbol{A}, \\ \boldsymbol{C}\boldsymbol{A} &= \boldsymbol{B}, & \boldsymbol{A}\boldsymbol{C} &= -\boldsymbol{B}. \end{aligned} \tag{7.24}$$

According to (7.23), it can be said that $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ behave as imaginary magnitudes. Then, it is not surprising that their matrix exponentials, defined according to the traditional power series, have simple expressions similar to Euler's formula:

$$e^{\omega \boldsymbol{A}} = \sin(\omega)\boldsymbol{A} + \cos(\omega)\boldsymbol{I}, \tag{7.25}$$

$$e^{\omega \boldsymbol{B}} = \sin(\omega)\boldsymbol{B} + \cos(\omega)\boldsymbol{I}, \tag{7.26}$$

$$e^{\omega \boldsymbol{C}} = \sin(\omega)\boldsymbol{C} + \cos(\omega)\boldsymbol{I}. \tag{7.27}$$

Then, it is not either surprising to realize that $e^{\omega \boldsymbol{A}}$, $e^{\omega \boldsymbol{B}}$ and $e^{\omega \boldsymbol{C}}$ behave as rotations in four dimensions. Indeed, since the exponential of an antisymmetric matrix is an orthogonal matrix with determinant equal to $+1$ and unit length eigenvalues, $e^{\omega \boldsymbol{A}}$, $e^{\omega \boldsymbol{B}}$ and $e^{\omega \boldsymbol{C}}$ represent rotations.

According to Cayley's factorization, a 4D rotation matrix can always be expressed as the product of two matrices of the form

$$\boldsymbol{R}^L(l_1, l_2, l_3, l_4) = \begin{pmatrix} l_1 & -l_2 & -l_3 & -l_4 \\ l_2 & l_1 & -l_4 & l_3 \\ l_3 & l_4 & l_1 & -l_2 \\ l_4 & -l_3 & l_2 & l_1 \end{pmatrix}, \tag{7.28}$$

and

$$\boldsymbol{R}^R(r_1,r_2,r_3,r_4)=\begin{pmatrix} r_1 & -r_2 & -r_3 & -r_4\\ r_2 & r_1 & r_4 & -r_3\\ r_3 & -r_4 & r_1 & r_2\\ r_4 & r_3 & -r_2 & r_1\end{pmatrix} \tag{7.29}$$

which are known as left- and right-isoclinic rotation matrices, respectively (see [102] for details on Cayley's factorization).

Now, it can observed that

$$\boldsymbol{R}^R(r_1,r_2,r_3,r_4)=\gamma_1\boldsymbol{I}+\gamma_2\boldsymbol{A}+\gamma_3\boldsymbol{B}+\gamma_4\boldsymbol{C} \tag{7.30}$$

where

$$\begin{pmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ \gamma_4\end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0\\ 0 & 0 & 0 & 1\\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0\end{pmatrix}\begin{pmatrix}r_1\\ r_2\\ r_3\\ r_4\end{pmatrix}. \tag{7.31}$$

Hence,

$$\begin{pmatrix}r_1\\ r_2\\ r_3\\ r_4\end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2}\\ 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2}\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ \gamma_4\end{pmatrix}. \tag{7.32}$$

Therefore, $\{\boldsymbol{I},\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\}$ is a basis for right-isoclinic rotations and, as a consequence, (7.25)-(7.27) represent right-isoclinic rotations.

In what follows, it is shown that three consecutive rotation around $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{A}$, covers all the rotational space in 4D. This is the key result used in the next chapter to solve the path planning problem. From (7.25) and (7.26), we obtain

$$\begin{aligned} e^{\omega_3\boldsymbol{A}}e^{\omega_2\boldsymbol{B}}e^{\omega_1\boldsymbol{A}} =&(\sin(\omega_3)\boldsymbol{A}+\cos(\omega_3)\boldsymbol{I})(\sin(\omega_2)\boldsymbol{B}+\cos(\omega_2)\boldsymbol{I})(\sin(\omega_1)\boldsymbol{A}+\cos(\omega_1)\boldsymbol{I})\\ =&\sin(\omega_3)\boldsymbol{A}\sin(\omega_2)\boldsymbol{B}\sin(\omega_1)\boldsymbol{A}+\cos(\omega_3)\boldsymbol{I}\sin(\omega_2)\boldsymbol{B}\sin(\omega_1)\boldsymbol{A}\\ &+\sin(\omega_3)\boldsymbol{A}\cos(\omega_2)\boldsymbol{I}\sin(\omega_1)\boldsymbol{A}+\cos(\omega_3)\boldsymbol{I}\cos(\omega_2)\boldsymbol{I}\sin(\omega_1)\boldsymbol{A}\\ &+\sin(\omega_3)\boldsymbol{A}\sin(\omega_2)\boldsymbol{B}\cos(\omega_1)\boldsymbol{I}+\cos(\omega_3)\boldsymbol{I}\sin(\omega_2)\boldsymbol{B}\cos(\omega_1)\boldsymbol{I}\\ &+\sin(\omega_3)\boldsymbol{A}\cos(\omega_2)\boldsymbol{I}\cos(\omega_1)\boldsymbol{I}+\cos(\omega_3)\boldsymbol{I}\cos(\omega_2)\boldsymbol{I}\cos(\omega_1)\boldsymbol{I}. \end{aligned}$$

Then, using equations (7.24), we obtain

$$\begin{aligned} e^{\omega_3 \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} =& \sin(\omega_3)\sin(\omega_2)\sin(\omega_1)\boldsymbol{B} - \cos(\omega_3)\sin(\omega_2)\sin(\omega_1)\boldsymbol{C} \\ & - \sin(\omega_3)\cos(\omega_2)\sin(\omega_1)\boldsymbol{I} + \cos(\omega_3)\cos(\omega_2)\sin(\omega_1)\boldsymbol{A} \\ & + \sin(\omega_3)\sin(\omega_2)\cos(\omega_1)\boldsymbol{C} + \cos(\omega_3)\sin(\omega_2)\cos(\omega_1)\boldsymbol{B} \\ & + \sin(\omega_3)\cos(\omega_2)\cos(\omega_1)\boldsymbol{A} + \cos(\omega_3)\cos(\omega_2)\cos(\omega_1)\boldsymbol{I}. \end{aligned}$$

That is,

$$\begin{aligned} e^{\omega_3 \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} =& (\cos(\omega_3)\cos(\omega_1) - \sin(\omega_3)\sin(\omega_1))\cos(\omega_2)\boldsymbol{I} \\ & + (\cos(\omega_3)\sin(\omega_1) + \sin(\omega_3)\cos(\omega_1))\cos(\omega_2)\boldsymbol{A} \\ & + (\sin(\omega_3)\sin(\omega_1) + \cos(\omega_3)\cos(\omega_1))\sin(\omega_2)\boldsymbol{B} \\ & + (\sin(\omega_3)\cos(\omega_1) - \cos(\omega_3)\sin(\omega_1))\sin(\omega_2)\boldsymbol{C}. \end{aligned}$$

Now, after somewhat tedious algebraic manipulations, it can be checked that:

$$\begin{aligned} e^{\omega_3 \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} =& \cos(\omega_2)\cos(\omega_3 + \omega_1)\boldsymbol{I} \\ & + \cos(\omega_2)\sin(\omega_3 + \omega_1)\boldsymbol{A} \\ & + \sin(\omega_2)\cos(\omega_3 - \omega_1)\boldsymbol{B} \\ & + \sin(\omega_2)\sin(\omega_3 - \omega_1)\boldsymbol{C}. \end{aligned} \tag{7.33}$$

Therefore, any arbitrary right-isoclinic rotation can be expressed as:

$$\gamma_1 \boldsymbol{I} + \gamma_2 \boldsymbol{A} + \gamma_3 \boldsymbol{B} + \gamma_4 \boldsymbol{C} = e^{\omega_3 \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}}. \tag{7.34}$$

Then, to find an expression for ω_1, ω_2, and ω_3, we can identify (7.33) and (7.34)

$$\begin{aligned} \gamma_1 &= \cos(\omega_2)\cos(\omega_3 + \omega_1), \\ \gamma_2 &= \cos(\omega_2)\sin(\omega_3 + \omega_1), \\ \gamma_3 &= \sin(\omega_2)\cos(\omega_3 - \omega_1), \\ \gamma_4 &= \sin(\omega_2)\sin(\omega_3 - \omega_1). \end{aligned} \tag{7.35}$$

Therefore, from the first and the second equation above, we have that

$$\omega_3 + \omega_1 = \operatorname{atan2}(\gamma_2, \gamma_1), \tag{7.36}$$

and, from the third and the fourth,

$$\omega_3 - \omega_1 = \text{atan2}(\gamma_4, \gamma_3). \tag{7.37}$$

As a consequence, the subtraction of (7.36) from (7.37) yields

$$\omega_1 = \frac{1}{2}(\text{atan2}(\gamma_2, \gamma_1) - \text{atan2}(\gamma_4, \gamma_3)). \tag{7.38}$$

Likewise, the addition of (7.36) and (7.37) yields

$$\omega_3 = \frac{1}{2}(\text{atan2}(\gamma_2, \gamma_1) + \text{atan2}(\gamma_4, \gamma_3)). \tag{7.39}$$

Now, dividing the third equation by the first one in (7.35), we obtain:

$$\frac{\gamma_3}{\gamma_1} = \frac{\sin(\omega_2)\cos(\omega_3 - \omega_1)}{\cos(\omega_2)\cos(\omega_3 + \omega_1)}.$$

That is,

$$\tan(\omega_2) = \frac{\gamma_3}{\gamma_1}\frac{\cos(\omega_3 + \omega_1)}{\cos(\omega_3 - \omega_1)}.$$

Then, replacing (7.36) and (7.37) in the above expression, we finally obtain:

$$\omega_2 = \arctan\left(\frac{\gamma_3}{\gamma_1}\frac{\cos(\text{atan2}(\gamma_2, \gamma_1))}{\cos(\text{atan2}(\gamma_4, \gamma_3))}\right). \tag{7.40}$$

The values of ω_1, ω_2 and ω_3 obtained using (7.38), (7.40), and (7.39), respectively, permit to solve equation (7.33) in closed form.

7.6 Workspace computation

A fundamental element in the design of any robot is the computation of its workspace. In what follows the workspace of the robot is determined by taking into account separately:

1. the robot singularities;
2. the stroke range of the prismatic actuators;
3. the working range of the **S** and **U** joints; and
4. the collisions between base, legs and platform.

To define the robot design, without loss of generality, $\boldsymbol{a}_1$ is set to $(1, 0, 0)$ and $\boldsymbol{a}_2$ is assumed lo lie in the xy-plane of the base reference, $\boldsymbol{b}_1^e$ will lie on the x-axis of the platform reference

and $\boldsymbol{b}_2^e$ will lie in the xy-plane of the platform reference. Moreover, we introduce the following parameters:

- α Angle between $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$. That is, $\boldsymbol{a}_2 = \boldsymbol{R}_z(\alpha)\boldsymbol{a}_1$.
- β Angle between $\boldsymbol{b}_1^e$ and $\boldsymbol{b}_2^e$. That is, $\boldsymbol{b}_2^e = \boldsymbol{R}_z(\beta)\boldsymbol{b}_1^e$.
- k Factor size between $\boldsymbol{a}_1$ and $\boldsymbol{b}_1^e$. That is, $\boldsymbol{b}_1^e = k\boldsymbol{a}_1$.

Thus, the robot design will be defined by the 4-tuplet (α, β, k, $\hat{\boldsymbol{r}}$) (see Fig. 7.4).

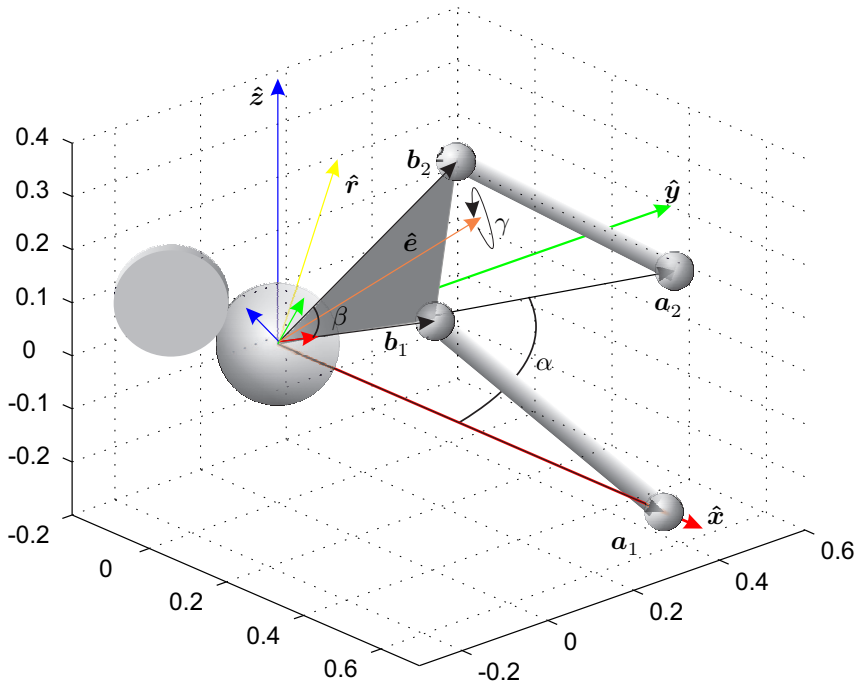


Figure 7.4: Identification of design parameters of a **nS-2UPS** robot.

7.6.1 Graphical representation of the platform orientation

To analyze the influence of the four restrictions, given above, on the robot workspace, a parametrization leading to a workspace representation that can be easily and intuitively interpreted, is needed.

There are multiple sets of parameters to represent the orientation of an object (Euler parameters, Euler angles, rotation matrix, vector-angle, etc.) and each of them has its pros and cons. Choosing a point in a plot using any of these sets of parameters and trying to imagine how the platform is oriented is, in general, far from intuitive. The situation worsens when trying to

imagine how the orientation changes when following a trajectory in the same plot. With these difficulties in mind, an ad-hoc representation of the platform orientation have been adopted for graphical purposes.

The platform orientation will be represented by a unit vector $\hat{\boldsymbol{e}}$ and an angle γ (see Fig. 7.4). $\hat{\boldsymbol{e}}$ will be the unit vector pointing from the origin to the midpoint of the segment connecting $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Since $\hat{\boldsymbol{e}}$ is a unit vector, we only need e_x and e_y to define it, all possible values for e_x and e_y will be inside the unit circle of the xy-plane. Then, we can represent the platform pose as the position of the center line of the platform, $\hat{\boldsymbol{e}}$, and the angle γ rotated about it, using the right hand convention. For example, when $\gamma = 0$, the line defined by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is parallel to the xy-plane. Thus, the scalars e_x, e_y and γ can be used to unambiguously represent the platform orientation. Like any other set of three parameters, this orientation representation has singularities. They correspond to the case in which $\hat{\boldsymbol{e}} = (0, 0, 1)^T$. In this case any value of γ makes the line defined by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ to be parallel to the xy-plane, so the orientation of the platform cannot be defined. This corresponds to the case in which the moving platform plane is orthogonal with respect to the base plane, so it is excluded from the analysis.

7.6.2 Workspace boundaries due to singularities

The singularities of the studied robot are those orientations in which, according to (7.3), $\det(\boldsymbol{J}) = 0$. To characterize this singularity locus, let us introduce two unit vectors, $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$, and two planes, Π_1 and Π_2. $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ are the normas of the planes defined by $\boldsymbol{b}_1 \times \boldsymbol{a}_1$ and $\boldsymbol{b}_2 \times \boldsymbol{a}_2$ respectively. As $\boldsymbol{a}_1$ ($\boldsymbol{a}_2$) is constant, $\hat{\boldsymbol{n}}_1$ ($\hat{\boldsymbol{n}}_2$) necessarily swivels around $\boldsymbol{a}_1$ ($\boldsymbol{a}_2$) in a fix plane, Π_1 (Π_2), as $\boldsymbol{b}_1$ ($\boldsymbol{b}_2$) varies.Thus, the normal of Π_1 (Π_2) is parallel to $\boldsymbol{a}_1$ ($\boldsymbol{a}_2$).

The set of singular orientations can be classified in three types:

1. Type I: singular orientations in which $\hat{\boldsymbol{n}}_1$ or $\hat{\boldsymbol{n}}_2$ are not defined. They occur when $\boldsymbol{a}_1 \parallel \boldsymbol{b}_1$ or $\boldsymbol{a}_2 \parallel \boldsymbol{b}_2$. They will always be present, no matter the robot design parameters.

2. Type II: singular orientations in which $\hat{\boldsymbol{n}}_1$, $\hat{\boldsymbol{n}}_2$, and $\hat{\boldsymbol{r}}$ are coplanar. To describe this type of singularities a third plane, Π_r, is introduced. It contains and swivels around $\hat{\boldsymbol{r}}$. These singularities arise when $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ are at the intersection of Π_1 with Π_r, and Π_2 with Π_r, respectively.

3. Type III: singular orientations in which $\hat{\boldsymbol{n}}_1$, $\hat{\boldsymbol{n}}_2$, and $\hat{\boldsymbol{r}}$ are also coplanar but they do not occur under the same conditions as those of Type II. They arise when:

   (a) $\hat{\boldsymbol{n}}_1 \parallel \hat{\boldsymbol{n}}_2$. Since $\boldsymbol{a}_1$ is not parallel to $\boldsymbol{a}_2$ by construction, this only occurs when the base and the platform planes coincide. This singularities always are present no matter the robot design parameters.

(b) $\hat{\boldsymbol{n}}_1 \parallel \hat{\boldsymbol{r}}$. This only occurs when $\hat{\boldsymbol{r}}$ lies in Π_1, and $\hat{\boldsymbol{n}}_1$ lies on the intersection of Π_r and Π_1.

(c) $\hat{\boldsymbol{n}}_2 \parallel \hat{\boldsymbol{r}}$. This only occurs when $\hat{\boldsymbol{r}}$ lies in Π_2, and $\hat{\boldsymbol{n}}_2$ lies on the intersection of Π_r and Π_2.

(d) $\hat{\boldsymbol{n}}_1 \parallel \hat{\boldsymbol{n}}_2 \parallel \hat{\boldsymbol{r}}$. This only occurs when $\hat{\boldsymbol{r}}$ lies in the intersection of Π_1 and Π_2, and $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ lie on the intersection of Π_1 and Π_2.

If the robot is near to a singularity, it will still withstand any external force, at least in theory, but the internal forces generated to counteract the external forces would be as high as leading to a possible slippage between the roller and the sphere, or to the break down of some mechanical elements. Thus, in practice, not only the orientations at which $\det(\boldsymbol{J}) = 0$ have to be determined, but it is also important to examine the condition number of $\boldsymbol{J}$ in the whole workspace. In general, a linear system is said to be ill-conditioned if the logarithm of its defining matrix condition number is higher than the precision of the matrix entries [3]. Thus, in our case, we will say, as a rule of thumb, that an orientation with condition number lower than 10 is well-conditioned.

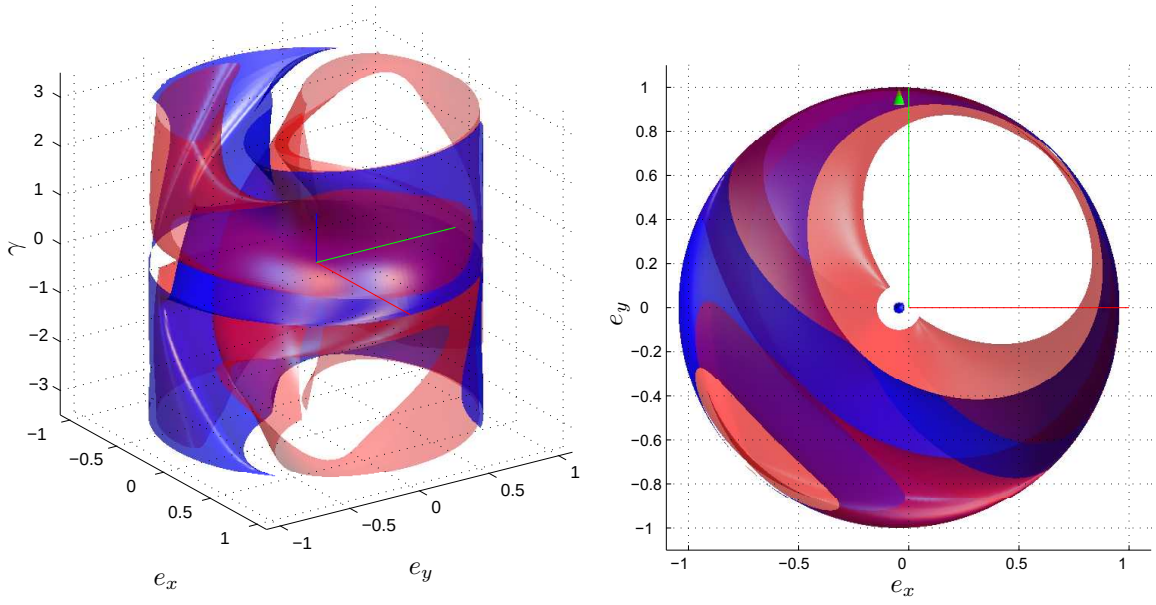


Figure 7.5: Singularity analysis of an **nS-2UPS** robot with design parameters $k = 0.5$, $\alpha = \pi/2$, $\beta = \pi/4$ and $\hat{\boldsymbol{r}} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$. The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view.

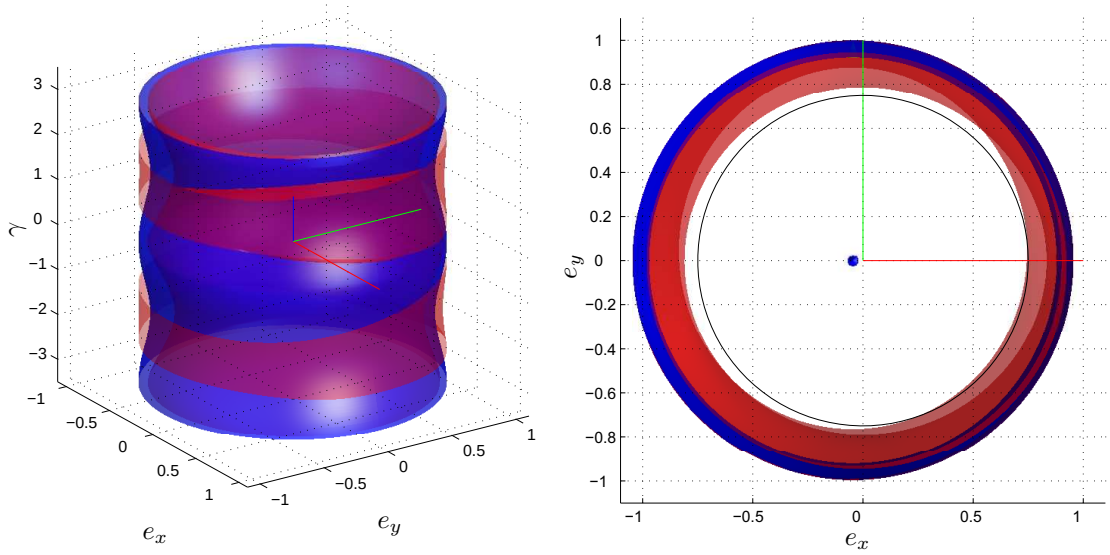


Figure 7.6: Singularity analysis of an **nS-2UPS** robot with same design parameters as those in figure 7.5 except for $\hat{\boldsymbol{r}} = (0, 0, 1)^T$. The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view.

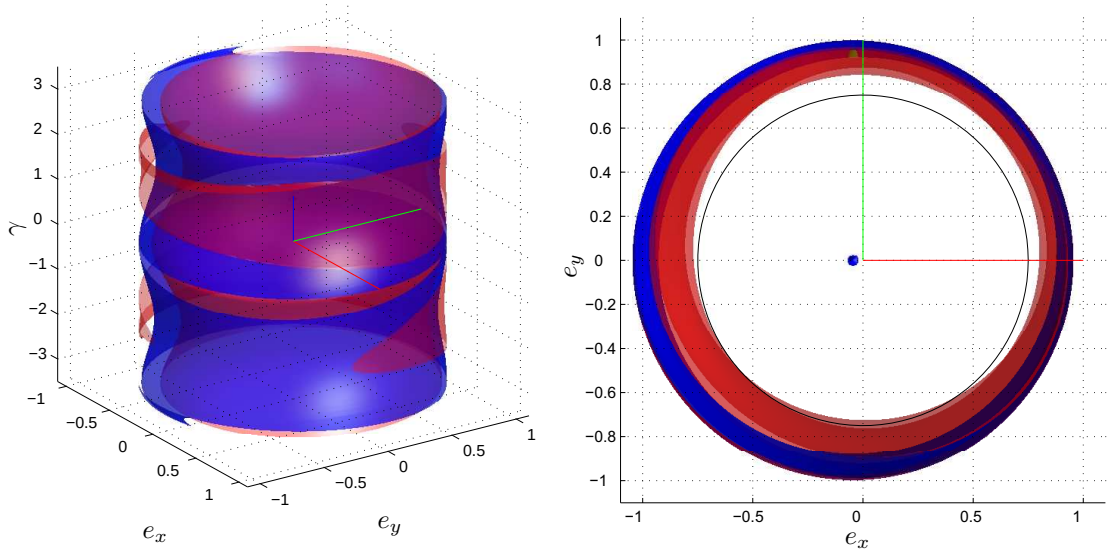


Figure 7.7: Singularity analysis of an **nS-2UPS** robot with same design parameters as those in figure 7.5 except for $\hat{\boldsymbol{r}} = (0.1, 0.1, \sqrt{0.6})^T$. The blue surfaces represent the singularity locus, and the red surfaces the orientations with condition number equal to 10. Left: Isometric view. Right: Top view.

As an example, let us set the robot design parameters to $k = 0.5$, $\alpha = \pi/2$, $\beta = \pi/4$ and $\hat{\boldsymbol{r}} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$. For this particular robot, Fig. 7.5 shows two views of the singularity locus, represented as a blue surface. In the same figure, the red surface represents those orientations in which the condition number is 10. Any pose containing or in between the red and blue surface should be avoided, they are not well-conditioned. In this particular example we can see (due to the adhoc parametrization) that the singularity locus breaks up the workspace in a very intricate way.

If no other particular objective than maximizing the workspace is considered in the design phase, the best solution is placing $\hat{\boldsymbol{r}}$ on the intersection of Π_1 and Π_2, that is, setting $\hat{\boldsymbol{r}} = (0, 0, 1)^T$. In this case, all singularities, both of type I and II, are concentrated and limited to the case in which $\boldsymbol{b}_1$, or $\boldsymbol{b}_2$, are coplanar with $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, which makes $\hat{\boldsymbol{n}}_1 \parallel \hat{\boldsymbol{r}}$ or/and $\hat{\boldsymbol{n}}_2 \parallel \hat{\boldsymbol{r}}$. To see the effect on the singularity locus, let us continue with the previous example, changing only $\hat{\boldsymbol{r}} = (0, 0, 1)^T$. For this particular design, Fig. 7.6 shows two views of the resulting singularity locus as a blue surface, and those configurations in which the condition number is 10 as a red surface. Observe how the region in which $e_z > \sqrt{1 - 0.8^2}$ can be consider as well-conditioned configurations. Therefore, setting $\hat{\boldsymbol{r}} = (0, 0, 1)^T$ leads to a very favorable design.

Another interesting aspect to study in the singularity locus is the influence on it due to small variations of $\hat{\boldsymbol{r}}$ that could come from construction or assembly inaccuracies. Performing the same singularity locus exploration as to the previous example, but with $\hat{\boldsymbol{r}} = (0.1, 0.1, \sqrt{0.6})^T$, shows an insignificant variation with respect to $\hat{\boldsymbol{r}} = (0, 0, 1)^T$. The resulting singularity locus appears in Fig. 7.7. Thus, it can be concluded that, in general, the choice of $\hat{\boldsymbol{r}} = (0, 0, 1)^T$ is a good and robust solution.

7.6.3 Workspace boundaries due to joint limits

As already pointed out, the workspace is not only determined by the robot singularities. The stroke of the prismatic actuators and the working ranges of the spherical and universal joints have to be taken into account for its complete characterization. The workspace due to the range stroke of the legs in the same example above ($k = 0.5$, $\alpha = \pi/2$, $\beta = \pi/4$ and $\hat{\boldsymbol{r}} = (0, 0, 1)^T$) can be performed by characterizing the platform poses in a cylindrical mesh and finding the leg length at each discretized pose. Surfaces are constructed for poses with equal values, values used are 1 to 0.5 for each leg. Fig. 7.8 shows four views of the surfaces, two views for each leg. The plot information on where the platform is capable of arriving with each leg, assuming that the other leg has no length restriction. Each plot shows six surfaces. Each surface represent an specific leg length. In the case in which the leg length for leg 1 is 0.5 the surface degenerates into a curve passing through $e_x = 0.9239$, $e_y = 0.3827$, and $\gamma = 0$.

The plots in Fig. 7.8 are the workspaces for each leg independent from each other. The actual

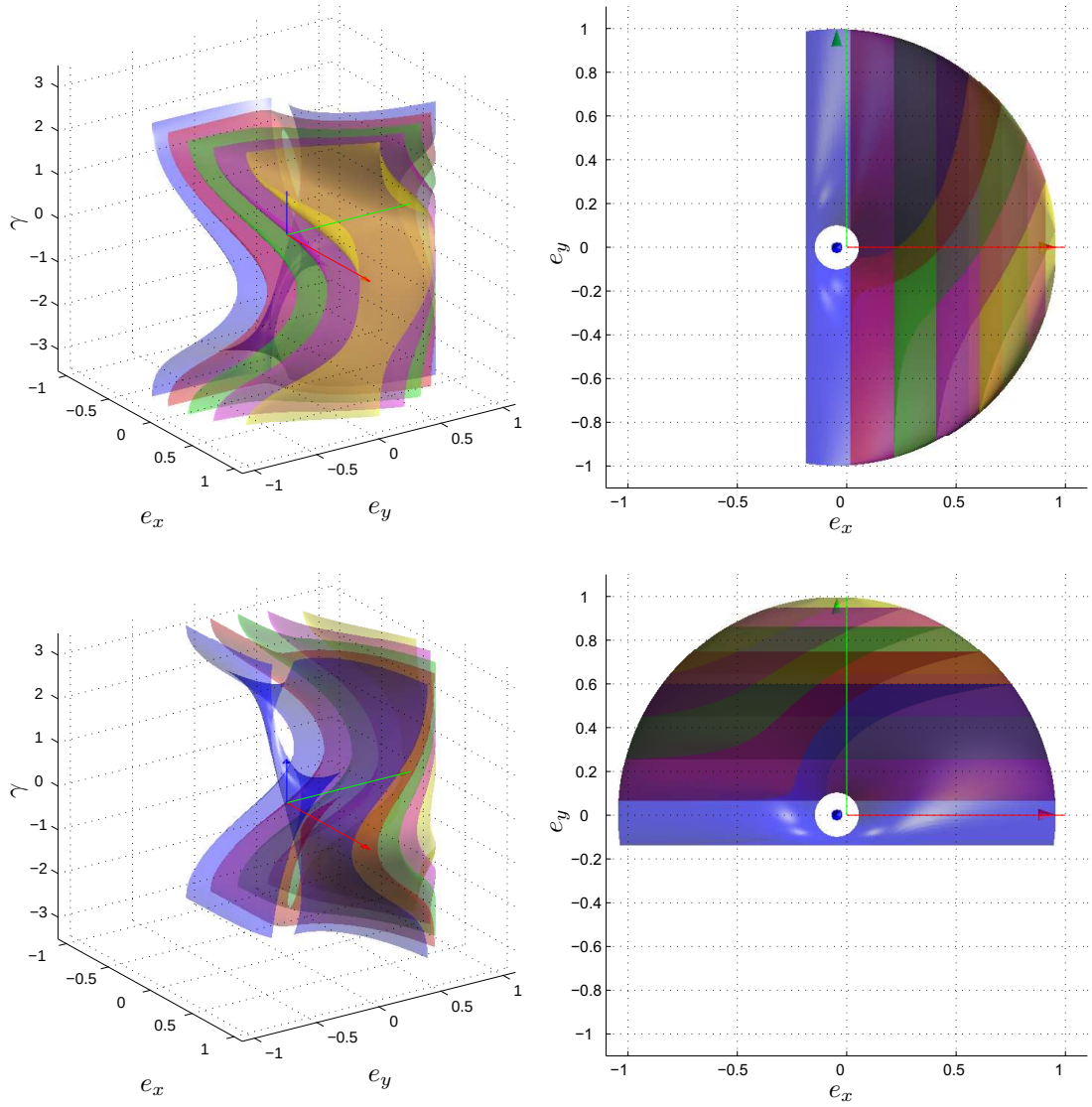


Figure 7.8: Contribution of the stroke range for each leg to the analyzed **nS-2UPS** robot workspace. Each surface corresponds to orientations in which the robot has a leg with the same length. The color code is as follows: Leg length: 1 (Blue), 0.9 (Red), 0.8 (Green), 0.7 (Magenta) and 0.6 (Yellow). Top: Leg 1 (Left: Isometric view. Right:Top view). Bottom: Leg 2 (Left: Isometric view. Right:Top view).

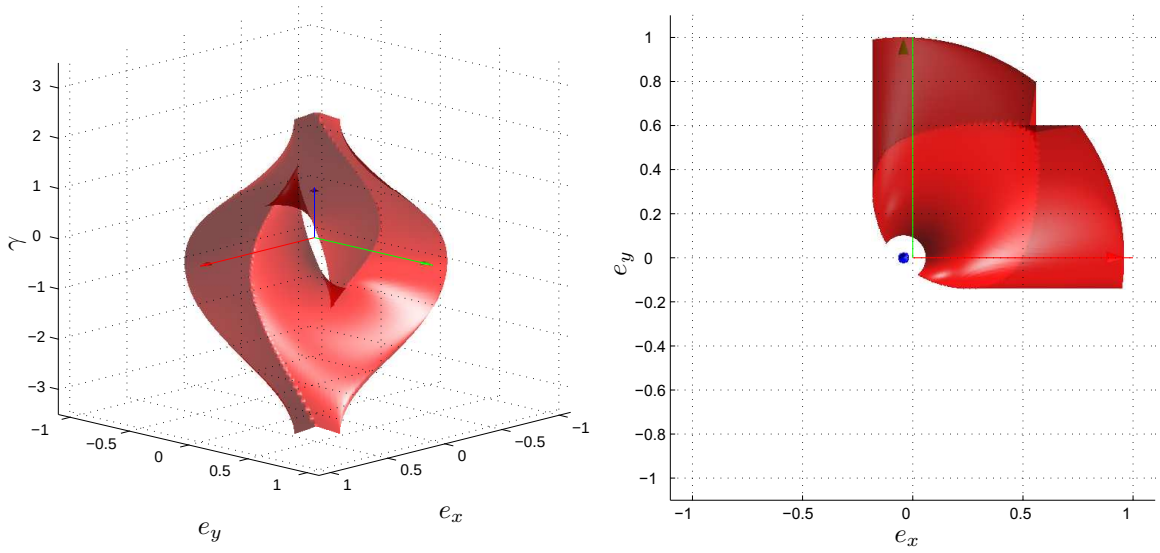


Figure 7.9: Workspace limits of analyzed **nS-2UPS** robot due to the stroke ranges of both legs. Both legs lengths are in the range $[0.5, 1]$. Left: Isometric view. Right: Top view.

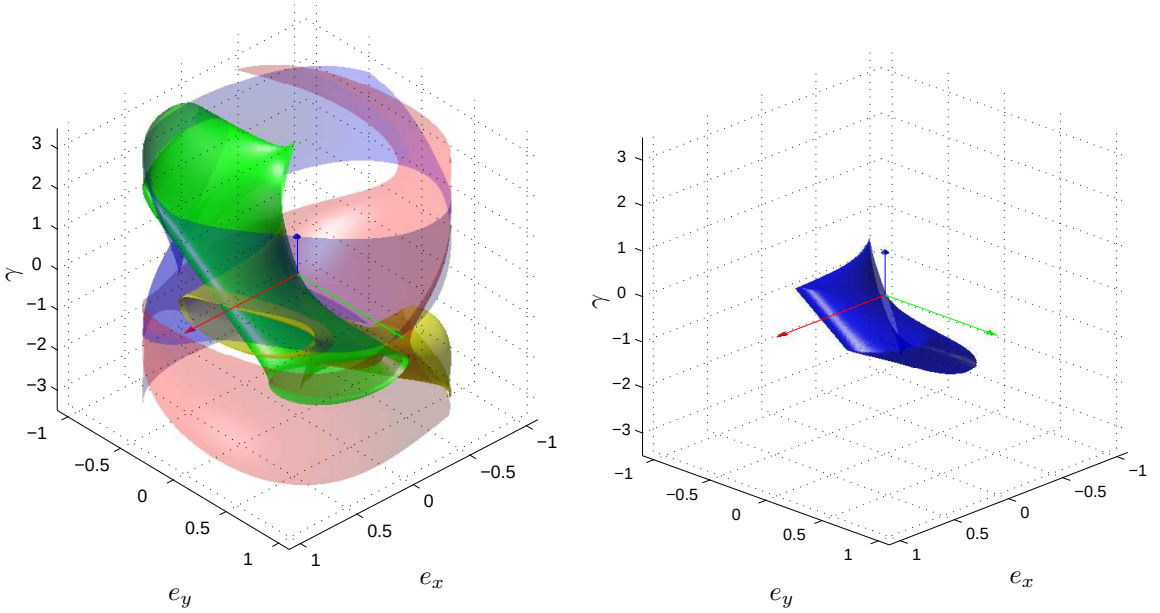


Figure 7.10: Workspace limits of analyzed **nS-2UPS** robot due to the joint ranges. All four joints working ranges are assumed to be cones with 70^o of aperture. Left: the four regions resulting from computing the limit for each joint independently. Right: Intersection of all four regions.

robot workspace is the intersection of the contribution of each leg. Fig. 7.9 shows the result.

Another factor that limits the workspace is the working range of the universal and spherical joints. Typically, universal joints have a working range of 35 degrees measure from their center line. In other words, they can move inside a 70 degree cone. Spherical joints have an equivalent work range although they have an additional degree of freedom. They freely spin around the axis of the prismatic actuator. Thus, for our analysis, they can be treated as universal joints. Next, a pose is chosen to be the reference pose. At this orientation all the joints are centered at the middle of their working range. Repeating the same procedure as in the case of leg ranges, the volumes containing feasible poses for each joint are found, where the surface of the volume are poses at the limit of the joint range. The feasibility of a pose is determined by doing the dot product of the vector along the leg, $\hat{\boldsymbol{g}}_i$, at the actual pose and the vector along the leg, $\hat{\boldsymbol{g}}_i$, at the reference pose. The process is done for the four joints [see Fig. 7.10(left)], and then an intersection of the four volumes is found [see Fig. 7.10(right)]. The obtained volume contains all the feasible orientations of the **nS-2UPS** robot, without violating the joints ranges.

Finally, we should find the limits of the workspace due to collisions between platform, legs and non-holonomic restriction, but in this particular case the limits due to joint ranges are so restrictive that there is no possible collision inside the allowed region. Thus, it is not necessary to perform this computation. Moreover, it is interesting to observe that the limits of the workspace due to the joint ranges is a subset of the one due to the stroke ranges of the actuators, and also to the one due to the mechanical singularities.

7.7 Conclusions

We have shown how designing a parallel orienting platform with only two actuators becomes feasible by introducing mechanical elements that lead to non-holonomic constraints. The advantages of the presented design might seem dubious when facing the necessity of introducing a path planner to generate the required maneuvers to reach a target from a given initial configuration. Nevertheless, in the next chapter, we present a set of alternative geometric techniques to the dominating Lie algebraic methods applicable to our particular design, like the one presented in this chapter, whose simplicity make the proposed robot architecture a real alternative in some applications where the reduction of bulk, weight or cost is a must.

8

Motion planning for the nS-2UPS parallel orienting robot

Any system with two inputs and up to four generalized coordinates can always be transformed into chained form. Then, since the **nS-2UPS** non-holonomic robot has two inputs (its leg lengths) and three generalized coordinates (its orientation parameters), its kinematics can be formulated in chained form. Given a system in chain form, its path planning problem can be solved using well-established procedures (see [70] and the references therein) which means the path planning problem for the analyzed **nS-2UPS** parallel orienting robot can be solved using one of these procedures, as Jakubiak did in [60] for a particular configuration of the system. In this case, Jakubiak used control functions given by truncated trigonometric series, as we already did in Chapter 6 for the **3nSPU** robot. Nevertheless, the use of these procedures requires a good understanding of sophisticated methods in non-linear control whose technicalities have proven a challenge to many practitioners who are not familiar with them. As an alternative, geometric path planners have been proposed, for example, in [29]. The main advantage of this kind of path planners is that they are based on elementary kinematics arguments. However, they generate stepped maneuvers, that is, maneuvers with intermediate instants with zero velocity that guide the moving platform to the desired orientation. This chapter goal is to present a geometric path planner able to steer the robot to the desired orientation in a continuous maneuver (maneuvers with no intermediate instants with zero velocity). Part of the work presented in this chapter has appeared in [45].

8.1 Introduction

The path planning problem for the studied non-holonomic parallel robot can be decomposed into the following two steps: (1) first solve the planning problem considering only the sphere

and the disk that constrains its motion, and then (2) obtain the required motion for the prismatic joints in the legs using the inverse kinematics of the robot. Separating both problems, instead of considering both at once, leads to an important simplification. However, this does not go without a price as the first step does not take into account the robot singularities. This is not an important drawback if a workspace free from singularities is defined beforehand (see Section 7.6).

Based on a rather old result on linear time-varying systems, this chapter shows that there are infinitely many differentiable paths connecting two arbitrary configurations in $\boldsymbol{SO}(3)$ such that the instantaneous axis of rotation along the path rest on a fixed plane. This theoretical result leads to a practical path planner for non-holonomic parallel orienting robots. A path planner which is a closed-form solution and whose derivation requires no other tools than ordinary linear algebra. To present this result, we start with a path planner based on three-move maneuvers, and then we proceed by progressively reducing the number of moves to one, thus providing a unified treatment with respect to previous geometric path planners.

This chapter is organized as follows. The first approach, the three-move maneuver, is presented in Section 8.2. It takes advantage of the bi-linear formulation presented in Section 7.3. It consists of three consecutive rotations around two orthogonal axes on the plane defined by the non-holonomic constraint. The second approach, presented in Section 8.3, is a generalization of the first one. It is based on quaternion formulation instead of the bi-linear formulation, which overrides the robot configuration restrictions assumed for the development of the bi-linear formulation. Section 8.4 presents the third approach, a two-move maneuver. It uses only two consecutive constant velocity rotations around two axes non-necessarily orthogonal on the plane defined by the non-holonomic constraint. Finally, in Section 8.5, a one-move maneuver is presented. It uses a rotation about a varying axis that rests on the plane defined by the non-holonomic restriction. To compare the procedures, a detailed example is developed in Section 8.6. A description of the experimental testbed, where the derived path planner have been verified, is presented in Section 8.7. We conclude in Section 8.8 with a summary of the main points.

8.2 Three-move maneuver

It has been proved in Section 7.3 that, by properly arranging the actuators of an **nS-2UPS** parallel orienting robot, its instantaneous kinematics is governed by the simple differential equation

$$\dot{\boldsymbol{q}} = (\boldsymbol{A}v_1 + \boldsymbol{B}v_2)\boldsymbol{q}, \tag{8.1}$$

where v_1 and v_2 stand for generalized inputs. Remember that the actuated leg lengths

velocities, the actual inputs, can be expressed as:

$$\begin{pmatrix} \dot{l}_1 \\ \dot{l}_2 \end{pmatrix} = \frac{\det(\boldsymbol{J})}{\sqrt{2}k} \begin{pmatrix} 1/l_1 & 0 \\ 0 & 1/l_2 \end{pmatrix} \begin{pmatrix} ad+bc & bc-ad \\ bd-ac & -ab-cd \end{pmatrix}^{-1} \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{8.2}$$

Now, let us suppose that the generalized inputs are constant, then (8.1) becomes a linear differential equation which can be easily integrated yielding

$$\boldsymbol{q}(t) = e^{(v_1 \boldsymbol{A} + v_2 \boldsymbol{B})t} \boldsymbol{q}_I,$$

where $\boldsymbol{q}_I$ stands for the initial orientation of the platform. Even if v_1 and v_2 vary with time, the above equation might still be used as a good approximation of the evolution of the system for small values of t.

Thus, for piecewise constant generalized inputs, the motion of the orienting platform can be easily computed. Let us introduce a maneuver consisting in a sequence of actuations in which during Δt seconds $v_1 = k_1$ and $v_2 = 0$, then during Δt seconds $v_1 = 0$ and $v_2 = k_2$ and, finally, during Δt seconds $v_1 = k_3$ and $v_2 = 0$. The configuration reached by the moving platform, after this maneuver, can be expressed as:

$$\boldsymbol{q}_F = e^{\omega_3 \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} \boldsymbol{q}_I, \tag{8.3}$$

where $\omega_i = k_i \Delta t$. Then, if we compare (8.3) with (7.33), it can be concluded that this simple maneuver permits to reach any desired configuration by finding the proper values of ω_i, $i = 1, 2, 3$. To this end, we first need to find the right-isoclinic rotation that drives the moving platform from $\boldsymbol{q}_I = (a_I, b_I, c_I, d_I)^T$ to $\boldsymbol{q}_F = (a_F, b_F, c_F, d_F)^T$, that is, the set of parameters r_1, r_2, r_3 and r_4 that satisfies

$$\begin{pmatrix} r_1 & -r_2 & -r_3 & -r_4 \\ r_2 & r_1 & r_4 & -r_3 \\ r_3 & -r_4 & r_1 & r_2 \\ r_4 & r_3 & -r_2 & r_1 \end{pmatrix} \begin{pmatrix} a_I \\ b_I \\ c_I \\ d_I \end{pmatrix} = \begin{pmatrix} a_F \\ b_F \\ c_F \\ d_F \end{pmatrix},$$

which can be rewritten as

$$\begin{pmatrix} a_I & -b_I & -c_I & -d_I \\ b_I & a_I & -d_I & c_I \\ c_I & d_I & a_I & -b_I \\ d_I & -c_I & b_I & a_I \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} a_F \\ b_F \\ c_F \\ d_F \end{pmatrix}. \tag{8.4}$$

Then, substituting (7.32) in (8.4), we obtain

$$\begin{pmatrix} a_I & \frac{\sqrt{2}}{2}(b_I - c_I) & -d_I & \frac{\sqrt{2}}{2}(b_I + c_I) \\ b_I & \frac{\sqrt{2}}{2}(-a_I - d_I) & c_I & \frac{\sqrt{2}}{2}(-a_I + d_I) \\ c_I & \frac{\sqrt{2}}{2}(a_I - d_I) & -b_I & \frac{\sqrt{2}}{2}(-a_I - d_I) \\ d_I & \frac{\sqrt{2}}{2}(b_I + c_I) & a_I & \frac{\sqrt{2}}{2}(-b_I + c_I) \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} a_F \\ b_F \\ c_F \\ d_F \end{pmatrix}.$$

Solving this linear system yields

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} a_I & b_I & c_I & d_I \\ \frac{b_I - c_I}{\sqrt{2}} & -\frac{a_I + d_I}{\sqrt{2}} & \frac{a_I - d_I}{\sqrt{2}} & \frac{b_I + c_I}{\sqrt{2}} \\ -d_I & c_I & -b_I & a_I \\ \frac{b_I + c_I}{\sqrt{2}} & -\frac{a_I - d_I}{\sqrt{2}} & -\frac{a_I + d_I}{\sqrt{2}} & -\frac{b_I - c_I}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} a_F \\ b_F \\ c_F \\ d_F \end{pmatrix}.$$

Finally, substituting these values of γ_i, $i = 1, \ldots, 4$, in (7.38), (7.40), and (7.39), we get the values of ω_1, ω_2, and ω_3, respectively, that define the proposed maneuver. Once these values are known, the platform orientation, as a function of time, can be expressed as

$$\boldsymbol{q}(t) = \begin{cases} e^{\omega_1 t \boldsymbol{A}} \boldsymbol{q}_I, & 0 \leq t < \Delta t \\ e^{\omega_2 \frac{\Delta t - t}{\Delta t} \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} \boldsymbol{q}_I, & \Delta t \leq t \leq 2\Delta t \\ e^{\omega_3 \frac{2\Delta t - t}{\Delta t} \boldsymbol{A}} e^{\omega_2 \boldsymbol{B}} e^{\omega_1 \boldsymbol{A}} \boldsymbol{q}_I, & 2\Delta t < t \leq 3\Delta t \end{cases} \tag{8.5}$$

As an example, let us consider an **nS-2UPS** platform with the design parameters $\boldsymbol{a}_1 = (1, 0, 0)^T$, $\boldsymbol{a}_2 = (0, 1, 0)^T$, $\boldsymbol{b}_1^e = k\boldsymbol{a}_1$, $\boldsymbol{b}_2^e = k\boldsymbol{a}_2$, $\hat{\boldsymbol{r}} = [\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}} \ 0]$ and $k = 0.5$. We want to drive the moving platform from the initial orientation given by $\boldsymbol{q}_I = (0.9537, 0.2300, -0.1835, 0.0624)^T$ to the final orientation given in quaternion form by $\boldsymbol{q}_F = (0.9537, 0.1835, -0.2300, -0.0624)^T$. The first step is to use equation (7.31) to conclude that $(\gamma_1, \gamma_2, \gamma_3, \gamma_4) = (0.9900, 0.0000, -0.0998, 0.0993)$. Then, using equations (7.38), (7.40) and (7.39), we find that $(\omega_1, \omega_2, \omega_3) = (-1.1793, 0.1413, 1.1793)$. Finally, we can obtain the platform orientation as a function of time using (8.5), and the input leg lengths velocities using equation (8.2). The results are presented in Fig. 8.1.

In this example, it is important to observe that, despite the initial and final poses are not far apart, the leg lengths span from 0.6 to 1.18 units. There are very few examples of linear actuators able of performing such long strokes. This problem is alleviated using the maneuvers proposed next.

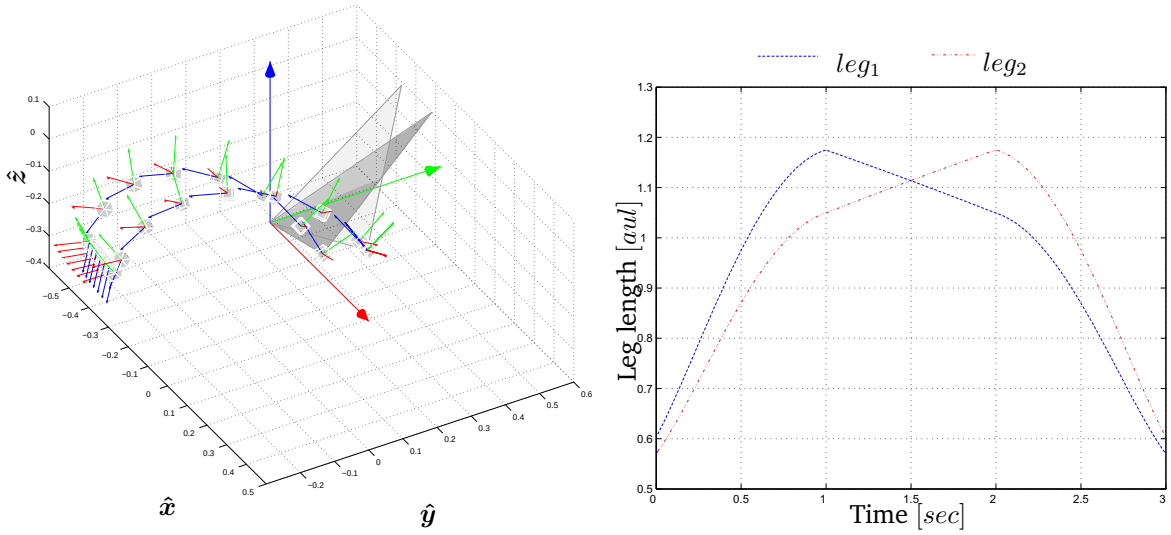


Figure 8.1: Example of a three-move maneuver platform motion, and the corresponding leg lengths as a function of time. The light and dark gray triangles represent the platform in its initial and final configurations, respectively.

8.3 Alternative three-move maneuver

Recalling the assumptions under which the bilinear formulation was developed (Section 7.3), we had that $\hat{\boldsymbol{r}} = [\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}} \ 0]$. Nevertheless, as absolute rotations[1] are reference-independent, a maneuver can be found in the quaternion-space for a particular $\hat{\boldsymbol{r}}$ and then use the maneuver with a change of reference that aligns the taken $\hat{\boldsymbol{r}}$ with the desired direction. Then, in what follows, it is assumed that $\hat{\boldsymbol{r}}$ is parallel to the z-axis.

Using a quaternion representation, the path planning problem can be restated as follows: given the initial orientation, $\boldsymbol{q}_I = a_I + b_I\hat{\boldsymbol{i}} + c_I\hat{\boldsymbol{j}} + d_I\hat{\boldsymbol{k}}$, and final orientation, $\boldsymbol{q}_F = a_F + b_F\hat{\boldsymbol{i}} + c_F\hat{\boldsymbol{j}} + d_F\hat{\boldsymbol{k}}$, find the time varying quaternion $\boldsymbol{q}(t)$ such that $\boldsymbol{q}_F = \boldsymbol{q}(t_F)\boldsymbol{q}_I$, where $\boldsymbol{q}(t)$ would have the following form

$$\boldsymbol{q}(t) = \cos\left(\frac{\theta(t)}{2}\right) + \left(\left(\cos\left(\alpha(t)\right)\hat{\boldsymbol{i}} + \sin\left(\alpha(t)\right)\hat{\boldsymbol{j}}\right)\cos(\beta(t)) + \sin(\beta(t))\hat{\boldsymbol{k}}\right)\sin\left(\frac{\theta(t)}{2}\right).$$

From the results obtained in the previous section, a three-move maneuver is the result of

[1]The absolute rotation, $\boldsymbol{q}_T$, is the rotation needed to rotate the platform from the initial orientation, $\boldsymbol{q}_I$, to the final orientation, $\boldsymbol{q}_F$, $\boldsymbol{q}_T = \boldsymbol{q}_F\boldsymbol{q}_I{}^*$ where $\boldsymbol{q}^*$ is the conjugate quaternion of $\boldsymbol{q}$

composing three constant rotations. That is,

$$\begin{aligned} \boldsymbol{q}_F &= \boldsymbol{q}_3\boldsymbol{q}_2\boldsymbol{q}_1\boldsymbol{q}_I, \\ \boldsymbol{q}_F{\boldsymbol{q}_I}^* &= \boldsymbol{q}_3\boldsymbol{q}_2\boldsymbol{q}_1, \\ \boldsymbol{q}_T &= \boldsymbol{q}_3\boldsymbol{q}_2\boldsymbol{q}_1, \end{aligned} \tag{8.6}$$

where the vector components of $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and $\boldsymbol{q}_3$ are orthogonal to $\hat{\boldsymbol{r}}$. Using a XYX sequence of rotations, we have that

$$\boldsymbol{q}_1 = \cos(\theta_1/2) + \sin(\theta_1/2)\hat{\boldsymbol{i}}, \quad \boldsymbol{q}_2 = \cos(\theta_2/2) + \sin(\theta_2/2)\hat{\boldsymbol{j}}, \quad \boldsymbol{q}_3 = \cos(\theta_3/2) + \sin(\theta_3/2)\hat{\boldsymbol{i}}.$$

Then, the time varying quaternion can be expressed as:

$$\boldsymbol{q}(t)=\begin{cases} \left(\cos\dfrac{t\theta_1}{2\Delta t}+\sin\dfrac{t\theta_1}{2\Delta t}\hat{\boldsymbol{i}}\right), & 0 \le t < \Delta t \\ \left(\cos\dfrac{(t-\Delta t)\theta_2}{2\Delta t}+\sin\dfrac{(t-\Delta t)\theta_2}{2\Delta t}\hat{\boldsymbol{j}}\right)\boldsymbol{q}_1, & \Delta t \le t \le 2\Delta t \\ \left(\cos\dfrac{(t-2\Delta t)\theta_3}{2\Delta t}+\sin\dfrac{(t-2\Delta t)\theta_3}{2\Delta t}\hat{\boldsymbol{i}}\right)\boldsymbol{q}_2\boldsymbol{q}_1, & 2\Delta t \le t \le 3\Delta t \end{cases} \tag{8.7}$$

where $\boldsymbol{q}(3\Delta t)$=$\boldsymbol{q}_T$. To determine the three unknowns, ω_1, ω_2 and ω_3, we rewrite (8.6) as

$$\boldsymbol{q}_3\boldsymbol{q}_2 = \boldsymbol{q}_T{\boldsymbol{q}_1}^* = (a_T + b_T\hat{\boldsymbol{i}} + c_T\hat{\boldsymbol{j}} + d_T\hat{\boldsymbol{k}}){\boldsymbol{q}_1}^*,$$

where

$$\begin{aligned} a_T &= a_Ia_F + b_Ib_F + c_Ic_F + d_Id_F, \\ b_T &= -b_Ia_F + a_Ib_F + d_Ic_F - c_Id_F, \\ c_T &= b_Id_F + a_Ic_F - d_Ib_F - c_Ia_F, \\ d_T &= c_Ib_F - d_Ia_F + a_Id_F - b_Ic_F. \end{aligned}$$

the following system of equations is obtained

$$\begin{aligned}
\cos(\theta_2/2)\cos(\theta_3/2) &= b_T \sin(\theta_1/2) + a_T \cos(\theta_1/2),\\
\cos(\theta_2/2)\sin(\theta_3/2) &= b_T \cos(\theta_1/2) - a_T \sin(\theta_1/2),\\
\sin(\theta_2/2)\cos(\theta_3/2) &= c_T \cos(\theta_1/2) - d_T \sin(\theta_1/2),\\
\sin(\theta_2/2)\sin(\theta_3/2) &= c_T \sin(\theta_1/2) + d_T \cos(\theta_1/2),
\end{aligned}$$

whose solution can be expressed as:

$$\theta_1 = \operatorname{atan2}\left(\frac{b_T c_T - a_T d_T}{a_T c_T + b_T d_T}\right), \tag{8.8}$$

$$\theta_2 = \arccos(b_T^2 + a_T^2 - c_T^2 - d_T^2), \tag{8.9}$$

$$\theta_3 = \operatorname{atan2}\left(\frac{a_T d_T + b_T c_T}{a_T c_T - b_T d_T}\right). \tag{8.10}$$

Section 8.6 presents an example of this three-move maneuver and compares it to the other methods presented below.

8.4 Two-move maneuver

The second method is inspired on the alternative three-move maneuver presented in the previous section. Instead of using three rotations over fixed vectors (XYX), it finds two constant rotations, given by $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, whose vector component are in the plane whose normal is $\hat{\boldsymbol{r}}$. Given the initial and final orientations, $\boldsymbol{q}_I$ and $\boldsymbol{q}_F$, respectively, then

$$\begin{aligned}
\boldsymbol{q}_F &= \boldsymbol{q}_2\boldsymbol{q}_1\boldsymbol{q}_I,\\
\boldsymbol{q}_F\boldsymbol{q}_I{}^* &= \boldsymbol{q}_2\boldsymbol{q}_1,\\
\boldsymbol{q}_T &= \boldsymbol{q}_2\boldsymbol{q}_1,
\end{aligned} \tag{8.11}$$

where $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ have the form

$$\boldsymbol{q}_i = \cos(\theta_i/2) + (\sin(\alpha_i)\hat{\boldsymbol{i}} + \cos(\alpha_i)\hat{\boldsymbol{j}})\sin(\theta_i/2).$$

Then, the two-move time function of the platform orientation will be given by

$$\boldsymbol{q}(t)=\begin{cases}\left(\cos\dfrac{t\theta_1}{2\Delta t}+\left(\sin(\alpha_1)\hat{\boldsymbol{i}}+\cos(\alpha_1)\hat{\boldsymbol{j}}\right)\sin\dfrac{t\theta_1}{2\Delta t}\right), & 0\leq t<\Delta t\\ \left(\cos\dfrac{(t-\Delta t)\theta_2}{2\Delta t}+\left(\sin(\alpha_2)\hat{\boldsymbol{i}}+\cos(\alpha_2)\hat{\boldsymbol{j}}\right)\sin\dfrac{(t-\Delta t)\theta_2}{2\Delta t}\right)\boldsymbol{q}_1, & \Delta t\leq t\leq 2\Delta t\end{cases} \tag{8.12}$$

where $\boldsymbol{q}(2\Delta t)=\boldsymbol{q}_T$. To determine the four unknowns θ_1, θ_2, α_1 and α_2, equation (8.11) must be solved. Rewriting this equation as

$$\boldsymbol{q}_2=\boldsymbol{q}_T{\boldsymbol{q}_1}^*,$$

the following system of equations is obtained

$$\begin{aligned}\cos(\theta_2/2) =& a_T\cos(\theta_1/2)+c_T\sin(\theta_1/2)\cos(\alpha_1)+b_T\sin(\theta_1/2)\sin(\alpha_1),\\ \sin(\theta_2/2)\sin(\alpha_2) =& b_T\cos(\theta_1/2)+d_T\sin(\theta_1/2)\cos(\alpha_1)-a_T\sin(\theta_1/2)\sin(\alpha_1),\\ \sin(\theta_2/2)\cos(\alpha_2) =& c_T\cos(\theta_1/2)-a_T\sin(\theta_1/2)\cos(\alpha_1)-d_T\sin(\theta_1/2)\sin(\alpha_1),\\ 0 =& d_T\cos(\theta_1/2)-b_T\sin(\theta_1/2)\cos(\alpha_1)+c_T\sin(\theta_1/2)\sin(\alpha_1).\end{aligned}$$

Since we have three equations (actually one equation above is redundant) and four unknowns, we can take one of the variables as a parameter and express the rest of variables as a function of it. Taking α_1 as a parameter, we have:

$$\begin{aligned}\theta_1 =& f_1(\alpha_1)\\ =& 2\,\mathrm{atan2}\left(\frac{d_T}{b_T\cos(\alpha_1)-c_T\sin(\alpha_1)}\right),\end{aligned} \tag{8.13}$$

$$\begin{aligned}\alpha_2 =& f_2(\alpha_1,\theta_1)\\ =& \mathrm{atan2}\left(\frac{b_T\cos(\theta_1/2)+d_T\sin(\theta_1/2)\cos(\alpha_1)-a_T\sin(\theta_1/2)\sin(\alpha_1)}{c_T\cos(\theta_1/2)-a_T\sin(\theta_1/2)\cos(\alpha_1)-d_T\sin(\theta_1/2)\sin(\alpha_1)}\right),\end{aligned} \tag{8.14}$$

$$\begin{aligned}\theta_2 =& f_3(\alpha_1,\theta_1,\alpha_2)\\ =& 2\,\mathrm{atan2}\left(\frac{(c_T\cos(\theta_1/2)-a_T\sin(\theta_1/2)\cos(\alpha_1)-d_T\sin(\theta_1/2)\sin(\alpha_1))/\cos(\alpha_2)}{(a_T\cos(\theta_1/2)+c_T\sin(\theta_1/2)\cos(\alpha_1)+b_T\sin(\theta_1/2)\sin(\alpha_1))}\right).\end{aligned} \tag{8.15}$$

Now, an additional criterion can be introduced to find a solution. In Section 8.6, an example is presented where the value taken for α_1 is the one that minimizes $\theta_1^2+\theta_2^2$.

8.5 One-move maneuver (continuous maneuver)

For the general angular velocity function $\boldsymbol{\omega}(t) = [\omega_x(t)\ \omega_y(t)\ \omega_z(t)]$ the differential equation $\dot{\boldsymbol{R}}(t) = [\boldsymbol{\omega}(t)]_\times \boldsymbol{R}(t)$ has no closed-form solution (it is a non integrable system). Nevertheless, for a constant angular velocity $\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]$, there is a close-form solution, the one given by the Rodrigues' rotation formula

$$\boldsymbol{R}(t) = \boldsymbol{I} + \frac{\sin(\theta)}{\theta} t[\boldsymbol{\omega}]_\times + \frac{1-\cos(\theta)}{\theta^2}(t[\boldsymbol{\omega}]_\times)^2 \quad \text{where} \quad [\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

It represents a rotation about $\hat{\boldsymbol{\omega}}$ the angle $\theta = \|\boldsymbol{\omega}\|\, t$.

Since in our case the angular velocity component, $\omega_z(t)$, must be 0, the differential equation to solve in quaternion form is

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x(t) & -\omega_y(t) & 0 \\ \omega_x(t) & 0 & 0 & \omega_y(t) \\ \omega_y(t) & 0 & 0 & -\omega_x(t) \\ 0 & -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix} \boldsymbol{q}(t). \tag{8.16}$$

The obvious question that comes to mind is if there are time varying functions $\omega_x(t)$ and $\omega_y(t)$ for which the differential equation (8.16) can be integrated in closed-form.

Reading Hennessey's [49] work, we find an equivalent system as the one used in this work, a non-holonomic motion of a disc over a sphere. In particular Hennessey's work describes circular paths of the disc-sphere contact point. Paths obtained by varying constantly the angular velocity axis. With this circular paths in mind, which are integrable, an orientation time function is found and used to build a new type of maneuver.

It was proved in [106] that, if $\boldsymbol{\omega}(t)$ in (8.16) satisfies the differential equation

$$[\dot{\boldsymbol{\omega}}(t)]_\times = \boldsymbol{N}_1\, [\boldsymbol{\omega}(t)]_\times - [\boldsymbol{\omega}(t)]_\times\, \boldsymbol{N}_1, \tag{8.17}$$

then the solution to (8.16), in matrix form, can be expressed as:

$$\boldsymbol{R}(t) = \exp(t\boldsymbol{N}_1) \exp(t\boldsymbol{N}_2)\, \boldsymbol{R}(0),$$

where $\boldsymbol{N}_1$ and $\boldsymbol{N}_2 = [\boldsymbol{\omega}(0)]_\times - \boldsymbol{N}_1$, are 3×3 skew-symmetric matrices. It can be seen that the one-move rotation, at each instant t, is equivalent to two simultaneous constant angular rotations, one with angle $t\,\|\boldsymbol{n}_1\|$ around vector $\hat{\boldsymbol{n}}_1$ and the other with angle $t\,\|\boldsymbol{n}_2\|$ around vector

$\hat{\boldsymbol{n}}_2$ (where $[\boldsymbol{n}_1]_\times = \boldsymbol{N}_1$ and $[\boldsymbol{n}_2]_\times = \boldsymbol{N}_2$). The solution to (8.16), in quaternion form, can be expressed as:

$$\boldsymbol{q}(t) = \exp\left(t \left\|\boldsymbol{n}_1\right\| \hat{\boldsymbol{n}}_1\right) \exp\left(t \left\|\boldsymbol{n}_2\right\| \hat{\boldsymbol{n}}_2\right) \boldsymbol{q}(0),$$

where

$$\exp\left(t \left\|\boldsymbol{n}_i\right\| \hat{\boldsymbol{n}}_i\right) = \boldsymbol{q}\left(t \left\|\boldsymbol{n}_i\right\|, \hat{\boldsymbol{n}}_i\right) = \cos\left(t \left\|\boldsymbol{n}_i\right\|/2\right) + \sin\left(t \left\|\boldsymbol{n}_i\right\|/2\right) \left(n_{i_x}\hat{\boldsymbol{i}} + n_{i_y}\hat{\boldsymbol{j}} + n_{i_z}\hat{\boldsymbol{k}}\right)/\left\|\boldsymbol{n}_i\right\|.$$

With some elementary algebra which is not reproduced here, it can be verified that

$$\boldsymbol{N}_1 = \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

is a particular solution of (8.17). Indeed replacing $\boldsymbol{N}_1$ in equation (8.17) yields the following system of equations:

$$\dot{\omega}_y = \omega\, \omega_x,$$
$$\dot{\omega}_x = -\omega\, \omega_y,$$

whose integration yields

$$\omega_x = A\cos(\omega t + \omega_0),$$
$$\omega_y = A\sin(\omega t + \omega_0). \tag{8.18}$$

Then,

$$\boldsymbol{\omega}(0) = \begin{bmatrix} A\cos\omega_0 & A\sin\omega_0 & 0 \end{bmatrix},$$

and, as a consequence,

$$\boldsymbol{N}_2 = \begin{bmatrix} 0 & \omega & A\sin\omega_0 \\ -\omega & 0 & -A\cos\omega_0 \\ -A\sin\omega_0 & A\cos\omega_0 & 0 \end{bmatrix}.$$

In order to use the above result to solve the problem of rotating the moving platform from $\boldsymbol{q}_I$ to $\boldsymbol{q}_F$, we can scale the time variable, t, so that the maneuver is completed at $t = 1$. Then, at $t = 1$, we have that

$$\boldsymbol{q}_F = \boldsymbol{q}(\omega, \hat{\boldsymbol{k}})\boldsymbol{q}(\delta, \hat{\boldsymbol{p}})\boldsymbol{q}_I,$$

where

$$\hat{\boldsymbol{p}} = \frac{1}{\sqrt{\omega^2 + A^2}} (A\cos\omega_0, A\sin\omega_0, -\omega)^T, \tag{8.19}$$

$$\delta = \sqrt{\omega^2 + A^2}. \tag{8.20}$$

In other words, the goal is to find ω, ω_0, and A such that

$$\boldsymbol{q}(\omega, \hat{\boldsymbol{k}})\ \boldsymbol{q}(\delta, \hat{\boldsymbol{p}}) = \boldsymbol{q}_F\ \boldsymbol{q}_I{}^* = \boldsymbol{q}_T. \tag{8.21}$$

If we set $\boldsymbol{q}_T = (a, b, c, d)$, it can be checked that $\boldsymbol{q}(\omega, \hat{\boldsymbol{k}})^*\ \boldsymbol{q}_T$ can be expressed, as a function of ω, as

$$\begin{aligned}
a'(\omega) &= a\cos\frac{\omega}{2} + d\sin\frac{\omega}{2},\\
b'(\omega) &= b\cos\frac{\omega}{2} + c\sin\frac{\omega}{2},\\
c'(\omega) &= c\cos\frac{\omega}{2} - b\sin\frac{\omega}{2},\\
d'(\omega) &= d\cos\frac{\omega}{2} - a\sin\frac{\omega}{2}.
\end{aligned}$$

Since, according to (8.21), $\boldsymbol{q}(\delta, \hat{\boldsymbol{p}}) = \boldsymbol{q}(\omega, \hat{\boldsymbol{k}})^*\ \boldsymbol{q}_T$, we have that

$$\cos\frac{\delta}{2} = a'(\omega), \tag{8.22}$$

$$\frac{A}{\delta}\cos\omega_0\sin\frac{\delta}{2} = b'(\omega), \tag{8.23}$$

$$\frac{A}{\delta}\sin\omega_0\sin\frac{\delta}{2} = c'(\omega), \tag{8.24}$$

$$-\frac{\omega}{\delta}\sin\frac{\delta}{2} = d'(\omega). \tag{8.25}$$

Now, observe that (8.22) and (8.25) depend only on ω and δ. From (8.22), we have that

$$\delta = \pm 2\arccos\big(a'(\omega)\big). \tag{8.26}$$

Moreover, equation (8.22) can be rewritten as $\sin(\delta/2) = \pm\sqrt{1 - [a'(\omega)]^2}$. Then, dividing this expression by (8.25), we conclude that $-\delta/\omega = \pm\sqrt{1 - [a'(\omega)]^2}/d'(\omega)$. In other words,

$$\delta = \mp\omega\frac{\sqrt{1 - [a'(\omega)]^2}}{d'(\omega)}. \tag{8.27}$$

Therefore, equating (8.26) and (8.27) yields the following transcendental equation in ω

$$2d'(\omega)\arccos\big(a'(\omega)\big) + \omega\sqrt{1-[a'(\omega)]^2} = 0. \tag{8.28}$$

Unfortunately, as it is usually the case for transcendental equations, no explicit solution has been found for (8.28). Thus, we have to rely at this point on a numerical method.

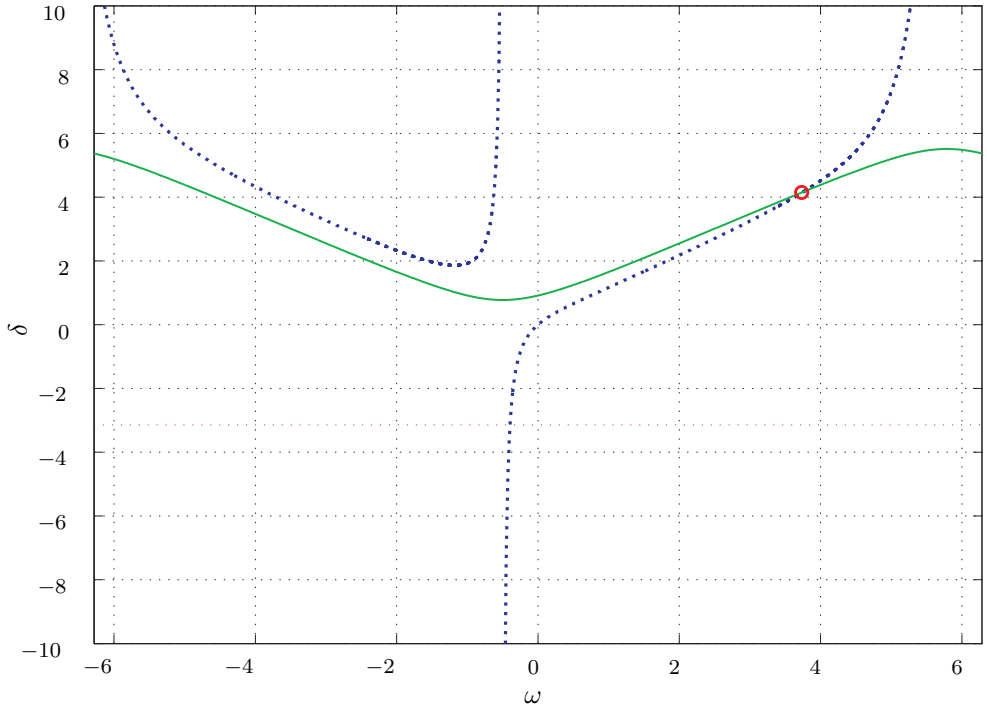


Figure 8.2: Typical example of plots representing the positive branches of (8.27) and (8.26) shown in dashed blue and solid green lines, respectively.

If we plot δ as a function of ω using (8.26) and (8.27), the intersection of both curves will correspond to the sought solutions. Fig. 8.2 depicts a typical example of the obtained plots.

Observe that if (δ, ω) is a solution of (8.26) and (8.27), then $(-\delta, \omega)$ is a solution as well, but they both correspond to the same physical motion. This simply accounts for the double covering of $\boldsymbol{SO}(3)$ when using Euler parameters.

Finally, with the obtained solutions for ω and δ, it is concluded from (8.20) that

$$A = \sqrt{\delta^2 - \omega^2}, \tag{8.29}$$

and, from (8.24) and (8.23), that

$$\omega_0 = \arctan\left(\frac{c'(\omega)}{b'(\omega)}\right). \tag{8.30}$$

In conclusion, the robot orientation as a function of time can simply be expressed as

$$\boldsymbol{q}(t)=\boldsymbol{q}\left(\frac{t}{\Delta t}\omega, \hat{\boldsymbol{k}}\right)\ \boldsymbol{q}\left(\frac{t}{\Delta t}\delta, \hat{\boldsymbol{p}}\right)\ \boldsymbol{q}_I,\ 0 \leq t \leq \Delta t. \tag{8.31}$$

8.6 Example

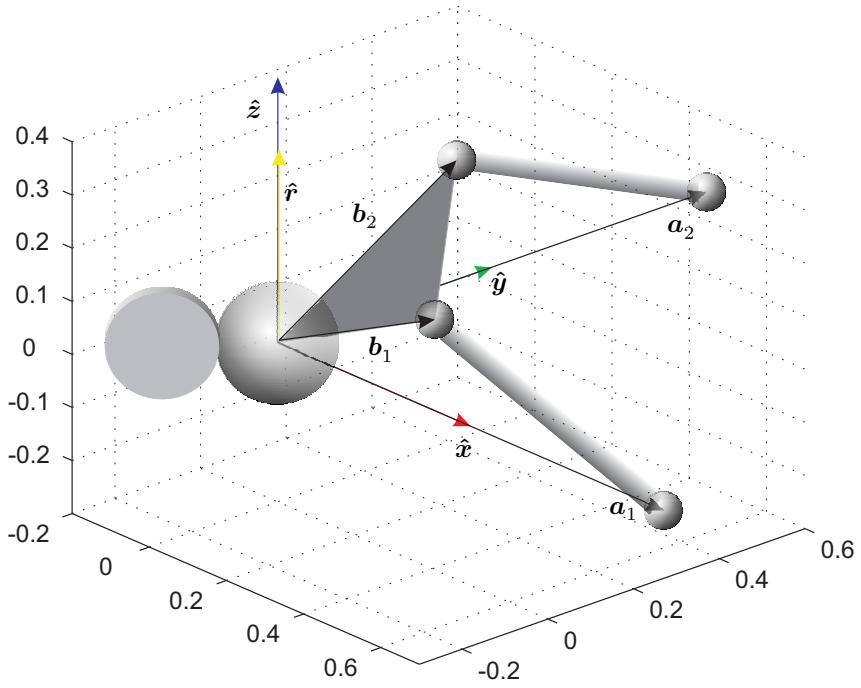


Figure 8.3: Location of the joints and the disk for the **nS-2UPS** robot used in the example.

To get some insight into the three methods and compare them, an example is used. Let us consider the non-holonomic orienting robot shown in Fig. 8.3. The center of the sphere is located at the origin, the spherical joints attached to the base are centered at $\boldsymbol{a}_1 = (1,0,0)^T$ and $\boldsymbol{a}_2 = (0,1,0)^T$, and those attached to the moving platform, in the reference orientation, at $\boldsymbol{b}_1^e = (0.5,0,0)^T$ and $\boldsymbol{b}_2^e = (0,0.5,0)^T$. Due to the non-holonomic constraint, the sphere cannot rotate about $\hat{\boldsymbol{r}}$, which is assumed to be aligned with the z-axis, as above.

Let us also assume that the initial and final orientations of the moving platform are given by

$$\boldsymbol{q}_I = 0.9355 + 0.0233\hat{\boldsymbol{i}} - 0.2188\hat{\boldsymbol{j}} + 0.2765\hat{\boldsymbol{k}},$$
$$\boldsymbol{q}_F = 0.9664 + 0.2543\hat{\boldsymbol{i}} + 0.0335\hat{\boldsymbol{j}} - 0.0188\hat{\boldsymbol{k}},$$

then

$$\boldsymbol{q}_T = \boldsymbol{q}_F \boldsymbol{q}_I{}^* = (0.8974, 0.2102, 0.3136, -0.2283).$$

8.6.1 Three-move maneuver

The three-move maneuver results from applying equations (8.8), (8.9) and (8.10). This yields

$$\omega_1 = 0.8594, \qquad \omega_2 = 0.7967, \qquad \omega_3 = -0.3993.$$

A representation of the robot motion following the resulting trajectory can be seen in Fig. 8.5(top), and for the legs movement see Fig. 8.6.

8.6.2 Two-move maneuver

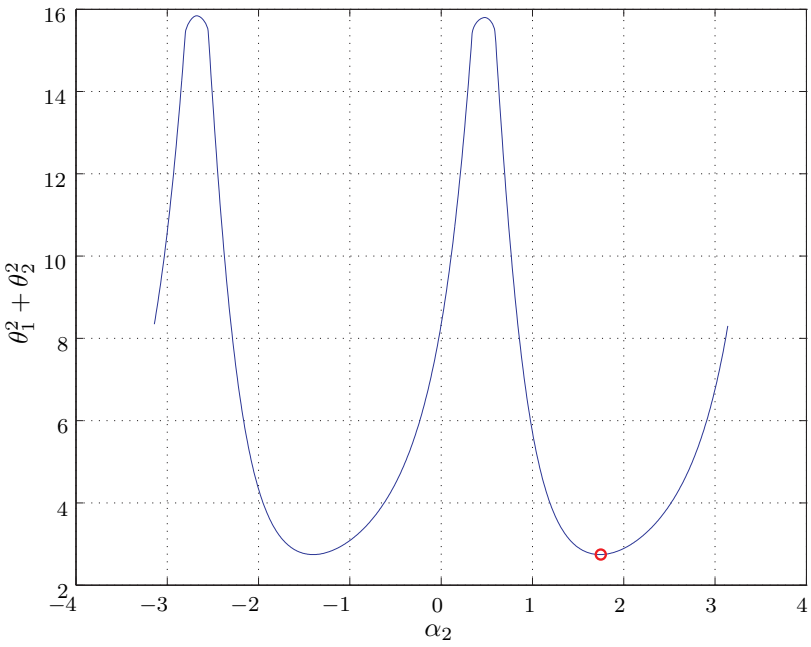


Figure 8.4: The minimum of $\theta_1^2 + \theta_2^2$ as a function of α_2 is attained at 1.7384. This value determines the two-move maneuver used in the example.

To obtain the two-move maneuver, the first step is to compute a set of values for α_1, θ_1 and θ_2 in the range $-\pi < \alpha_2 < \pi$ using equations (8.13), (8.14) and (8.15). The second step, since we can arbitrarily choose α_2, we pick out the value that minimizes $\theta_1^2 + \theta_2^2$. This minimum is attained at $\alpha_2 = 1.7384$ (see Fig. 8.4). For this value of α_2, the remaining variables of the two-move maneuver yield $\theta_1 = 1.1713$, $\alpha_1 = 2.5829$, and $\theta_2 = -1.1708$. This values define the two-move maneuver, a rotation around $\mathbf{n}_1 = (-0.848, 0.530, 0)^T$ an angle $\alpha_1 = 2.5829$, followed by a rotation around $\mathbf{n}_2 = (-0.167, 0.986, 0)^T$ an angle $\alpha_2 = 1.7384$. A representation of the robot motion following the resulting trajectory can be seen in Fig. 8.5(middle), and the legs movement can be seen in Fig. 8.6.

8.6.3 One-move maneuver

Following the procedure detailed in Section 8.5, the first move consists in plotting δ as a function of ω using both (8.26) and (8.27) for $\boldsymbol{q}_T = (0.8974, 0.2102, 0.3136, -0.2283)$. The result is plotted in Fig. 8.2. The intersection of both curves occurs at

$$\delta = 4.1578, \quad \omega = 3.7496.$$

Then, the substitution these values in (8.29) and (8.30) yields

$$A = 1.7965, \quad \omega_0 = -0.8945.$$

A representation of the motion followed by the robot along the resulting path can be seen in Fig. 8.5(bottom), and for the legs movement can be see in Fig. 8.6.

8.6.4 Comparing the three path planners

Fig. 8.5 shows the motion generated by the three path planners. On the left column, we have light gray and dark gray triangles representing the moving platform in its initial and final orientation, respectively. The sequence of small reference frames illustrate the path followed by the $\mathbf{S_{b1}}$ joint center. This is the motion generated as seen from the base reference frame. If we fix the observer to the moving reference frame, the motion followed by the disk on the sphere is better appreciated. This is represented on the right column of the figure.

All three trajectories behave well and quite similarly in this example. A greater difference is observed when translating the generated motions of the platform into a variation of the two leg lengths, $l_i = \|\boldsymbol{b}_i - \boldsymbol{a}_i\| = \|\boldsymbol{R}\boldsymbol{b}_i^e - \boldsymbol{a}_i\|$, $i = 1, 2$. The result is represented in Fig. 8.6. The one-move maneuver generates a differentiable path. It can lead to faster motions because the generated trajectory does not contain any zero-velocity points, thus making the use of the

studied parallel robot possible in a larger range of applications. Finally, it is worth noting that, although using the one-move path planner the time variation of the legs' lengths are differentiable, the trajectory followed by the point of contact between the disk and the sphere might contain cusp points. These points play a fundamental role in non-holonomic path planners [70]. An important result is that they are here automatically generated.

It must be pointed out that any of the presented maneuvers can be used to go directly from the initial to the final pose, or they can be used to move to intermediate poses. Fig. 8.7 shows the leg movements introducing intermediate poses. The maneuvers, were calculated using the same path planner: the two-move maneuver. In this particular example the legs maximum extension has been reduced by adding intermediate poses, something useful in those cases where the stroke needed to perform the maneuver is out of the physical range of the actuator.

A final thought on the generated maneuvers: all three methods behave well in this example, but this is no necessarily true in general. In some examples one method could behave better than the other. Future work aims for combinations of the presented maneuvers that will give an optimal solution.

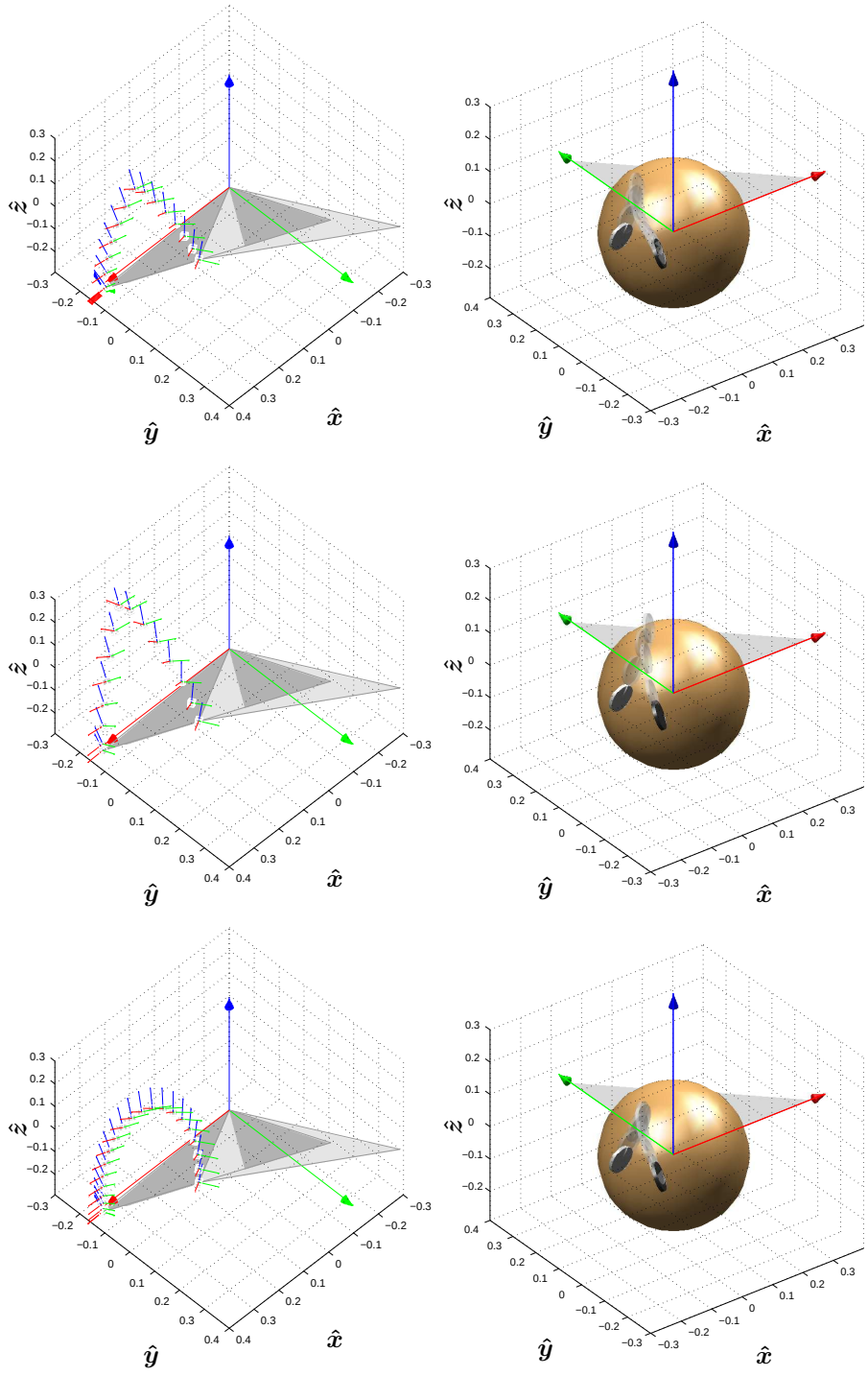


Figure 8.5: Example of the motions generated by the three described path planners shown with respect to both the reference frame of the base and that of the moving platform (see text for details).

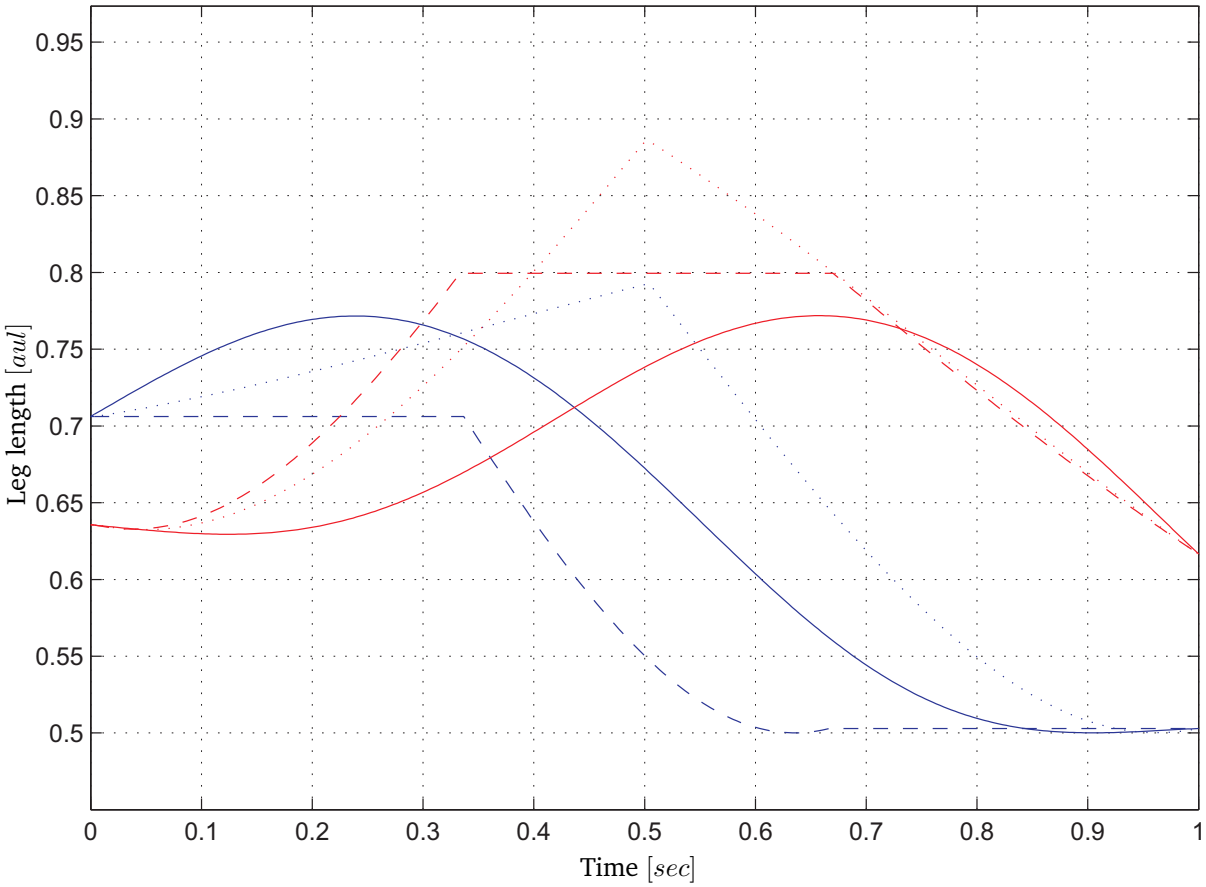


Figure 8.6: Temporal variation of the leg lengths (L_1 in blue and L_2 in red) for the three generated trajectories, while rotating from $\boldsymbol{q}_I$ to $\boldsymbol{q}_F$. The leg lengths for the three-move maneuver are represented in dashed lines, for the two-move maneuver, in dotted lines, and for the single-move maneuver, in solid lines.

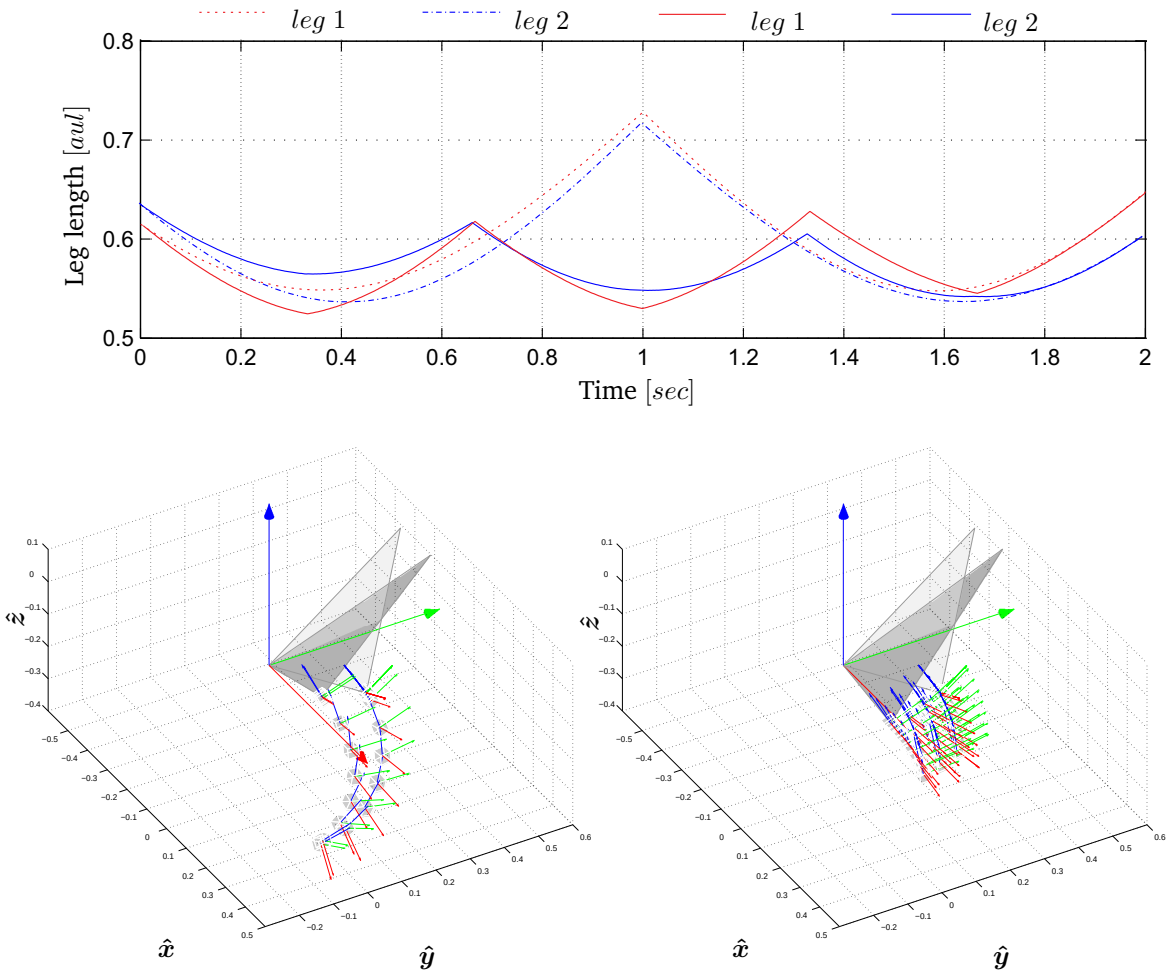


Figure 8.7: Top: leg lengths along a two-move maneuver (in solid lines), and the result of introducing two intermediate points (in dashed lines). Bottom: representation of the platform motion along both planned trajectories.

8.7 Hardware and software implementation

Figure 8.8: Implemented experimental testbed.

To validate the presented path planners, the testbed shown in Fig. 8.8 has been implemented. It features a redimensionable structure, two legs with **SPS** topology and a moving platform fixed, to a non-holonomic joint.

The structure is constructed with aluminum profiles with the idea of being modular, **S** joints and the non-holonomic joint can be placed where desired. Modularity also includes the ability to re-orient the non-holonomic constraint to point in any direction.

Initially, each **SPS** leg had as prismatic actuators a miniature Firgelli linear actuators with

a stroke of 100 mm [54]. Due to their slowness and inaccuracies, they have been replaced by Dynamixel $MX64$ motors with USB communication [15]. The legs are attached to the base through spherical joints based on spherical bearings housed in 3D printed plastic receptacles that enables the relocation of the joints.

The non-holonomic joint is based on a steel ball. It has been perforated using an electric discharge machine. Its motion is constrained by two rollers arranged in opposing positions from the center. Three free-rolling spheres are used to keep the joint centered in the plane perpendicular to the line between the contact points of the rollers (see Section 4.3 for a detailed explanation of the design and construction of the non-holonomic joint).

The orientation of the platform is measured using a Phidget spatial $3/3/3$ sensor that communicates via USB [14]. This board has a 3-axis accelerometer, gyroscope and compass. Motors and sensor are interfaced through USB using a software developed for interfacing with Matlab. The software enables reading the platform orientation and the leg lengths, and commanding the motors.

8.8 Conclusions

The motion planning problem of an **nS-2UPS** parallel orienting robot has been solved in this chapter using a quaternion formulation. First, we solve the problem by introducing a three-move maneuver. Then, we observe that it can be simplified to a two-move maneuver. Finally, we arrive at the rather surprising result that only a continuous maneuver is needed, in general, to reach an arbitrary final configuration. That is, the analyzed orienting robot cannot follow arbitrary paths connecting two orientations, but it is possible in general to find smooth paths in closed-form connecting two arbitrary orientations. We do not know of any other non-trivial non-holonomic system where such kind of closed-form solution have been found for its path planing problem.

The continuous maneuver has been developed from a particular solution to a differential equation. This means that there are probably infinitely many L^∞ paths connecting two arbitrary orientations in $\boldsymbol{SO}(3)$ that satisfy the non-holonomic constraint. This opens the possibility of optimizing the path according to some criterion, or even the possibility of finding paths with closed-form formulas for their defining parameters, instead of relying on a numerical method.

The presented path planners are open-loop methods. However, it would be desirable to construct the input as a function of the system state to compensate for noises and errors in the system. Path planners that generate maneuvers cannot be translated into control systems in an obvious way. The situation changes with the presented single-move path planner. This is certainly a point that deserves further attention. The result would be a practical algorithm

for planning and controlling the motions of the studied platform that can help to achieve all its potential benefits. The presented ideas seem to be applicable to other non-holonomic mechanical systems whose orientation has to be controlled.

9 Conclusions

9.1 Contributions

The aim of the thesis has been to define, analyze and implement parallel robots with unconventional joints so that the resulting mechanical systems are under-actuated and/or reconfigurable. It has been shown how these new mechanical systems pose challenging problems in path-planning and control, thus leading to a set of new “benchmarks" in the area of non-holonomic and reconfigurable systems. This expands the range of examples for these kind of systems, which was limited in the past to systems like vehicles, possibly with arms, or vehicles dragging n-carts in series, some of them having more academic than practical interest.

The new proposed designs have been derived from the Gough-Stewart platform, to which, through certain geometric transformations, some its joints are replaced by lockable joints or non-holonomic joints. These substitutions permit reducing the number of legs (and hence of actuators) without losing the robot’s ability to bring its mobile platform to any position and orientation within its workspace. In general, these new designs have:

1. larger working space compared to the Gough-Stewart platform from which they are derived, as the possibility of collisions between legs is reduced; and

2. lower weight and cost due to the reduction in the number of legs and actuators.

These advantages do not come without a cost: it is necessary, in all cases, to plan maneuvers to reach the desired position and orientation for the moving platform. Therefore, the obtained robots will only be suitable for applications where no high speeds are needed, accuracy is required in the final pose, and a certain margin of error is acceptable in the generated trajectories. These applications include most positioning and point-to-point tasks.

It can be said that the main contribution of this thesis has been to open a new line of research in parallel robots. Up to our knowledge, the derived publications during the elaboration of this

thesis contain the first research results in the use of non-holonomic and/or lockable joints in the area of parallel robots.

The analysis of each of the derived new robots has been conducted with the greatest possible generality. This has included in most cases:

1. the computation of its direct and inverse kinematics (both in position and velocity);
2. the analysis of its singularities;
3. the computation of its workspace;
4. the design and implementation of a motion planning algorithm; and
5. the implementation of a prototype, whenever the budget has allowed us, to validate the theoretical model.

In all studied cases, it has been possible to design an algorithm to automatically generate the necessary maneuvers to connect two arbitrary configurations. The nature of the presented algorithms ranges from pure geometrical to entirely analytical. Indeed, while the generation of trajectories for parallel platforms with lockable joints is a geometrical and combinatorial problem, the same for platforms with non-holonomic joints requires the use of differential geometry tools.

The following publications have been derived from this work:

- P. Grosch, R. Di Gregorio and F. Thomas. "Generation of under-actuated parallel robots with non-holonomic articulations and kinetostatic analysis of a case-study", Proc. of the ASME International Design Engineering Technical Conference, pp. 979-986, 2009.
- P. Grosch, R. Di Gregorio and F. Thomas. "A one-motor full-mobility 6-PUS manipulator", ROMANSY 18 Robot Design, Dynamics and Control, Vol. 524 of CSIM Courses and Lectures, pp. 49-56, 2010.
- P. Grosch, R. Di Gregorio and F. Thomas. "Generation of under-actuated manipulators with non-holonomic articulations from ordinary manipulators", Journal of Mechanisms and Robotics, vol. 2, no. 1, pp. 11005-11012, 2010.
- P. Grosch, R. Di Gregorio, J. Lopez and F. Thomas. "Motion planning for a novel reconfigurable parallel manipulator with lockable revolute articulations", Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2010), pp. 4697-4702, 2010.

- K. Tchoń, J. Jakubiak, P. Grosch and F. Thomas. "Motion planning for parallel robots with non-holonomic joints", Latest Advances in Robot Kinematics, J. Lenarcic and M. Husty (editors), Springer, pp. 115-122, 2012.

- P. Grosch and F. Thomas. "A bilinear formulation for the motion planning of non-holonomic parallel orienting platforms", Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), pp. 953-958, 2013.

- P. Grosch and F. Thomas. "Geometric path planning without maneuvers for non-holonomic parallel orienting robots", accepted in IEEE Robotics and Automation Letters, 2016.

The interest of other research groups in our ideas has led to collaborations with Prof. R. Di Gregorio, from the University of Ferrara, and Prof. K. Tchoń, from the University of Wroclaw, as reflected in the authorship of some of the above publications. Their help during the elaboration of this thesis is again gratefully acknowledged.

9.2 Prospects for future research

Many open problems detailed throughout this thesis might be the subject of interesting prospects for future research, but we think that two of them deserve particular attention.

The **4bRRPS** parallel platform has been proved to be able to move its moving platform in a six-dimensional operational space by using only four actuators. Maneuvers essentially consist, in this case, in changing the set of locked joints at via configurations. It have been shown that parallel singularities can be avoided and the maximum forces in the actuators can be reduced by suitably inserting these via configurations. We conjecture that, by properly locating the joints, it would be possible to avoid all singularities. If this conjecture is eventually proved, this parallel platform would attain a remarkable place in the universe of parallel robots.

The **nS-2UPS** parallel orienting platform has been proved to be able to move its moving platform in the three-dimensional space of orientations by using only two actuators. Based on a rather old result on linear time-varying systems, it has been shown that a single-step maneuver allows this robot, in general, to move from one configuration to any other in its workspace by following an infinitely differentiable path. The consequences of this fact have not been fully explored yet. We think that its application not only to path-planning but also to control offers the opportunity of achieving all the potentialities of this interesting platform without relying on intricate techniques.

References

[1] F. Aghili and K. Parsa. Design of a Reconfigurable Space Robot with Lockable Telescopic Joints. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4608 – 4614, 2006.

[2] F. Aghili and K. Parsa. A Reconfigurable Robot With Lockable Cylindrical Joints. *IEEE Transactions on Robotics*, 25:785 – 797, 2009.

[3] J. Angeles. Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms. *Mechanical Engineering Series. Berlin, Germany: Springer-Verlag*, 2003.

[4] P. Ben-Horin and F. Thomas. A Nonholonomic 3-DoF Parallel Robot. *Advances in Robot Kinematics: Analysis and Design, Springer Verlag*, pages 111 – 118, 2008.

[5] D. Bernier, J. M. Castelain, and X. Li. A New Parallel Structure with Six Degrees of Freedom. *In Proceedings of the 9th World Congress on the Theory of Machines and Mechanisms*, pages 8 – 12, 1995.

[6] A. M. Bloch. Nonholonomic Mechanics and Control. *Interdisciplinary Applied Mathematics*, 24, 2003.

[7] A. V. Borisov and I. S. Mamaev. Chaplygin's Ball. The Suslov Problem and Veselova's Problem. Integrability and Realization of Constraints. *Nonholonomic Dynamical Systems. Integrability, Chaos, Strange Attractors. Moscow-Izhevsk: Institute of Computer Science*, pages 118 – 130, 2002.

[8] A. V. Borisov and I. S. Mamaev. Rolling of a Rigid Body on Plane and Sphere. Hierarchy of Dynamics. *Regular and Chaotic Dynamics*, 7:201 – 218, 2002.

[9] T. Boye and G. Pritschow. New Transformation and Analysis of a n-DoF Linapod with Six Struts for Higher Accuracy. *Robotica*, 23:555 – 560, 2005.

[10] R. W. Brockett. Asymptotic Stability and Feedback Stabilization. *Differential Geometric Control Theory, Boston*, 27:181 – 191, 1983.

[11] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. Principles of Robot Motion: Theory, Algorithms, and Implementations. *Intelligent Robotics and Autonomous Agents series, MIT Press, Boston*, 2005.

[12] Huco-Dynatork Co. World Leading Manufacturers of Small Precision Couplings. www.huco.com/, 2015.

[13] Item Co. Building Kit Systems for Industrial Applications. http://www.item24.co.uk/, 2015.

[14] Phidgets Co. Products for USB Sensing and Control. http://www.phidgets.com/, 2015.

[15] Robotis Co. Commercializing Personal Robots. http://www.robotis.com/, 2015.

[16] Servo City Co. Mechanical Components for use in Robotics. http://www.servocity.com/, 2015.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. *Second Edition, MIT Press*, 2001.

[18] J. Cortés and T. Siméon. Probabilistic Motion Planning for Parallel Mechanisms. *IEEE International Conference on Robotics and Automation*, 3:4354 – 4359, 2003.

[19] G. Cui and Y. Zhang. Kinetostatic Modeling and Analysis of a New 3-DoF Parallel Manipulator. *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1 – 4, 2009.

[20] B. Dasgupta and T.S. Mruthyunjaya. The Stewart Platform Manipulator: a Review. *Mechanism and Machine Theory*, 35:15 – 40, 2000.

[21] A. De Luca and G. Oriolo. Modelling and Control of Nonholonomic Mechanical Systems. *CISM International Centre for Mechanical Sciences (Chapter 7)*, pages 277 – 342, 1995.

[22] A. De Luca, G. Oriolo, and P. Robuffo-Giordano. Image-Based Visual Servoing Schemes for Nonholonomic Mobile Manipulators. *Robotica*, 25:131 – 145, 2007.

[23] R. Di Gregorio. Singularity-Locus Expression of a Class of Parallel Mechanisms. *Robotica*, 20:323 – 328, 2002.

[24] R. Di Gregorio. Statics and Singularity Loci of the 3-UPU Wrist. *IEEE Transactions on Robotics*, 2004.

[25] R. Di Gregorio. An Exhaustive Scheme for the Singularity Analysis of Three-DoF Parallel Manipulators. *In Proc. of the 17th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD*, 2008.

[26] R. Di Gregorio. Kinematic Analysis of the (nS)-2SPU Underactuated Parallel Wrist. *ASME Journal of Mechanisms and Robotics*, 4, 2012.

[27] R. Di Gregorio. Position Analysis and Path Planning of the S-(nS)PU-SPU and S-(nS)PU-2SPU Underactuated Wrists. *ASME Journal of Mechanisms and Robotics*, 4, 2012.

[28] R. Di Gregorio. Type Synthesis of Underactuated Wrists Generated from Fully-Parallel Wrists. *ASME Journal of Mechanical Design*, 4, 2012.

[29] R. Di Gregorio. Position Analysis, Path Planning, and Kinetostatics of Single-Loop RU-(nS)PU Wrists. *ASME Journal of Mechanisms and Robotics*, 74:117Ű133, 2014.

[30] I. Duleba and W. Khefifi. Velocity Space Approach to Motion Planning of Nonholonomic Systems. *Robotica*, 25:359 – 366, 2007.

[31] EMTorq. Industrial Electric Clutches and Brakes. http://www.emtorq.com/, 2015.

[32] N. Farhat, V. Mata, À. Page, and F. Valero. Identification of Dynamic Parameters of a 3-DoF RPS Parallel Manipulator. *Mechanism and Machine Theory*, 43:1 – 17, 2008.

[33] E. Faulring, J. Colgate, and M. Peshkin. A High Performance 6-DoF Haptic Cobot. *IEEE International Conference on Robotics and Automation*, 2004.

[34] C. Gosselin and J. Angeles. Singularity Analysis of Closed-Loop Kinematic Chains. *IEEE Transactions on Robotics and Automation*, 6:281 – 290, 1990.

[35] C. Gosselin and J. F. Hamel. The Agile Eye: a High-Performance Three-Degree-of-Freedom Camera-Orienting Device. *Proc. IEEE International Conference on Robotics and Automation (San Diego, CA)*, 1:781 – 786, 1994.

[36] V. E. Gough and S. G. Whitehall. Universal Tyre Test Machine. *Proc. Ninth International Technical Congress F.I.S.I.T.A.*, pages 117 – 137, 1962.

[37] P. Grosch. 4bRRPS Robot Maneuver Special Case - Rotation Stagnation (video). https://youtu.be/6MSQhsyw6uM, 2015.

[38] P. Grosch. 4bRRPS Robot Pose Rotation Around z-axis (video). https://youtu.be/6MSQhsyw6uM, 2015.

[39] P. Grosch. 4bRRPS Robot Simulator (video). http://youtu.be/EXSNdvIOI9Y, 2015.

[40] P. Grosch, R. Di Gregorio, J. Lopez, and F. Thomas. Motion Planning for a Novel Reconfigurable Parallel Manipulator with Lockable Revolute Articulations. *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2010)*, pages 4697 – 4702, 2010.

[41] P. Grosch, R. Di Gregorio, and F. Thomas. Generation of Under-Actuated Parallel Robots with Non-Holonomic Articulations and Kinetostatic Analysis of a Case-Study. *Proc. of the ASME International Design Engineering Technical Conference*, pages 979–986, 2009.

[42] P. Grosch, R. Di Gregorio, and F. Thomas. A One-Motor Full-Mobility 6-PUS Manipulator. *ROMANSY 18 Robot Design, Dynamics and Control. CSIM Courses and Lectures*, 524:49 – 56, 2010.

[43] P. Grosch, R. Di Gregorio, and F. Thomas. Generation of Under-Actuated Manipulators with Non-Holonomic Articulations from Ordinary Manipulators. *Journal of Mechanisms and Robotics*, 2:11005 – 11012, 2010.

[44] P. Grosch and F. Thomas. A Bilinear Formulation for the Motion Planning of Non-Holonomic Parallel Orienting Platforms. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, pages 953–958, 2013.

[45] P. Grosch and F. Thomas. Geometric Path Planning Without Maneuvers for Non-Holonomic Parallel Orienting Robots. *IEEE Robotics and Automation Letters*, 2016 accepted.

[46] H. Gu and M. Ceccarelli. Simulation of Combined Motions for a 1-DoF Clutched Robotic Arm. *International Conference on Mechatronics and Automation*, pages 3721 – 3726, 2009.

[47] H. Gu, N. E. Nava, and M. Ceccarelli. Dynamics Simulation of Operation for a Clutched Arm. *XVIII Congreso Nacional de Ingeniería Mecánica, España, A.E.I.M.*, pages 3721 – 3726, 2010.

[48] J. E. Gwinnett. Amusement Device: First Concept of Parallel Robot. *US patent office US1789680*, 1931.

[49] M. P Hennessey. Visualizing the Motion of a Unicycle on a Sphere. *International Journal of Modelling and Simulation*, 26:69 – 79, 2006.

[50] H. Hermes. Distributions and the Lie Algebras their Bases can Generate. *Proceedings of the American Mathematical Society*, 106:555 – 565, 1989.

[51] M. Honegger, A. Codourey, and E. Burdet. Adaptive Control of the Hexaglide, a 6 DoF Parallel Manipulator. *Proceedings of International Conference on Robotics and Automation*, 1:543 – 548, 1997.

[52] I. I. Hussein and A. M. Bloch. Optimal Control of Underactuated Nonholonomic Mechanical Systems. *IEEE Transactions on Automatic Control*, 53:668 – 682, 2008.

[53] M. L. Husty. An Algorithm for Solving the Direct Kinematics of General Stewart-Gough Platforms. *Mechanism and Machine Theory*, 31:365 – 379, 1996.

[54] Firgelli Technologies Inc. Innovator in the Micro-Motion. http://www.firgelli.com/, 2015.

[55] C. Innocenti. Direct Kinematics in Analytical Form of the 6-4 Fully-Parallel Mechanism. *ASME Journal of Mechanical Design*, 117:89 – 95, 1995.

[56] C. Innocenti. Forward Kinematics in Polynomial Form of the General Stewart Platform. *Journal of Mechanical Design*, 123:254 – 260, 2001.

[57] C. Innocenti and V. Parenti-Castelli. Direct Position Analysis of the Stewart Platform Mechanism. *Mechanism and Machine Theory*, 25:611 – 621, 1990.

[58] C. Innocenti and V. Parenti-Castelli. Echelon Form Solution of Direct Kinematics for the General Fully-Parallel Spherical Wrist. *Mechanism and Machine Theory*, 28:553 – 561, 1993.

[59] C. Innocenti and V. Parenti-Castelli. Exhaustive Enumeration of Fully Parallel Kinematic Chains. *Proc. of the 1994 ASME Int. Winter Annual Meeting*, 55:1135 – 1141, 1994.

[60] J. Jakubiak, W. Magiera, and K. Tchoń. Control and Motion Planning of a Non-Holonomic Parallel Orienting Platform. *ASME Journal of Mechanisms and Robotics*, 7, 2015.

[61] J. Jakubiak, K. Tchoń, and W. Magiera. Motion Planning in Velocity Affine Mechanical Systems. *International Journal of Control*, 83:1965 – 1974, 2010.

[62] C. Y. Ji, T. C. Chen, and Y. L. Lee. Investigation of Kinematic Analysis and Applications for a 3-RRPS Parallel Manipulator. *Journal of the Chinese Society of Mechanical Engineers*, 28:623 – 632, 2007.

[63] D. Joyner. Adventures in Group Theory: Rubik's Cube, Merlin's Machine and other Mathematical Toys. *The Johns Hopkins University Press, Baltimore (USA).*, 40:34 – 54, 2002.

[64] H. Karbasi. Uni-Drive Modular Robots. *Phd. Thesis, University of Waterloo, Canada*, 2002.

[65] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12:566 – 580, 1996.

[66] C. T. Kelley. Solving Nonlinear Equations With Newton's Method. *Society for Industrial and Applied Mathematics (SIAM), Philadelphia*, 2003.

[67] A. P. Kharlamov and M. P. Kharlamov. Nonholonomic Joint. *Mekh. Tverd. Tela (Rigid Body Mechanics), NAS of Ukraine*, 27:1 – 7, 1995.

[68] J. Koiller and K. Ehlers. Rubber Rolling over a Sphere. *Regular and Chaotic Dynamics*, 12:177 – 200, 2007.

[69] G. Lafferriere. A General Strategy for Computing Steering Controls of Systems Without Drift. *IEEE Conference on Decision and Control, Brighton, UK*, pages 1115 – 1120, 1991.

[70] J. P. Laumond, S. Sekhavat, and F. Lamiraux. Guidelines in Nonholonomic Motion Planning for Mobile Robots. *Robot Motion Planning and Control, Lecture Notes in Control and Information Science, Springer*, 129:1 – 53, 1998.

[71] Y. Li, Y. Song, Z. Feng, and C. Zhang. Complete Jacobian Matrix of a Class of Incompletely Symmetrical Parallel Mechanisms with 4-DoF. *Chinese Journal of Mechanical Engineering (Jixie Gongcheng Xuebao)*, 43, 2007.

[72] D. A. Lizárraga. Obstructions to the Existence of Universal Stabilizers for Smooth Control Systems. *Mathematics of Control, Signals, and Systems (MCSS)*, 16:255 – 277, 2004.

[73] Y. Lu and J. Xu. Simulation Solving/Modifying Velocity and Acceleration of a 4UPS+SPR Type Parallel Machine Tool During Normal Machining of a 3D Free-Form Surface. *The International Journal of Advanced Manufacturing Technology*, 42:7 – 8, 2009.

[74] Y. Lu, M. Zhang, Y. Shi, and J. Yu. Kinematics and Statics Analysis of a Novel 4-DoF 2SPS+2SPR Parallel Manipulator and Solving its Workspace. *Robotica*, 27:771 – 778, 2009.

[75] J. M. McCarthy. Geometric Design of Linkages. *Interdisciplinary Applied Mathematics, Springer-Verlag, New York*, 11, 2000.

[76] J. P. Merlet. Singular Configurations of Parallel Manipulators and Grassmann Geometry. *The International Journal of Robotics Research*, 8:45 – 56, 1989.

[77] J. P. Merlet. Les Robots Paralléles. *Hermes, 2nd Edition*, 1997.

[78] J. P. Merlet. Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots. *ASME Journal of Mechanical Design*, 128:199 – 206, 2006.

[79] J. P. Merlet. Parallel Robots. *Dordrecht, The Netherlands: Springer, 2nd edition*, 128, 2006.

[80] R. M. Murray. Control of Nonholonomic Systems using Chained Forms. *Fields Institute Communications*, 1:219 – 245, 1993.

[81] R. M. Murray. Nilpotent Bases for a Class of Nonintegrable Distributions with Applications to Trajectory Generation for Nonholonomic Systems. *Mathematics of Control, Signals, and Systems*, 7:58 – 75, 1994.

[82] R. M. Murray, Z. Li, and S. S. Sastry. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.

[83] R. M. Murray and S. S. Sastry. Nonholonomic Motion Planning: Steering Using Sinusoids. *IEEE Transactions on Automatic Control*, 38:700 – 716, 1993.

[84] Y. Nakamura, W. Chung, and O. J. Sørdalen. Design and Control of the Nonholonomic Manipulator. *IEEE Transactions on Robotics and Automation*, 17:48 – 59, 2001.

[85] P. E. Nikravesh, O. K. Kwon, and R. A. Wehage. Euler Parameters in Computational Kinematics and Dynamics, Part 1. *ASME Journal of Mechanisms Transmissions and Automation in Design*, 107:358 – 365, 1985.

[86] P. E. Nikravesh, O. K. Kwon, and R. A. Wehage. Euler Parameters in Computational Kinematics and Dynamics, Part 2. *ASME Journal of Mechanisms Transmissions and Automation in Design*, 107:366 – 369, 1985.

[87] K. Ning and F. Worgotter. A Novel Concept for Building a Hyper-Redundant Chain Robot. *IEEE International Conference on Robotics and Automation ICRA*, 25:1237 – 1248, 2009.

[88] O. M. O'Reilly. Intermediate Dynamics for Engineers. *Cambridge University Press, New York, NY.*, 2008.

[89] V. Parenti-Castelli and R. Di Gregorio. Closed-Form Solution of the Direct Kinematics of the 6-3 type Stewart Platform using one extra Sensor. *Meccanica*, 31:705 – 711, 1996.

[90] M. Peshkin, J. E. Colgate, and C. Moore. Passive Robots and Haptic Displays Based on Nonholonomic Elements. *IEEE International Conference on Robotics and Automation*, pages 551 – 556, 1996.

[91] F. Pierrot, P. Dauchez, and A. Fournier. HEXA: a Fast Six-DoF Fully-Parallel Robot. *Proc. Fifth International Conference on Advanced Robotics ICARŠ91*, 2:1158 – 1163, 1991.

[92] G. Pritschow, C. Eppler, and W. D. Lehner. Highly Dynamic Drives for Parallel Kinematic Machines with Constant Arm Length. *In Proc. 1st Int. Colloq., Collaborative Research Centre*, 562:199 – 211, 2002.

[93] N. M. Rao and K. M. Rao. Dimensional Synthesis of a Spatial 3-RPS Parallel Manipulator for a Prescribed Range of Motion of Spherical Joints. *Mechanism and Machine Theory*, 44:477 – 486, 2009.

[94] E. A. Shammas, H. Choset, and A. A. Rizzi. Towards a Unified Approach to Motion Planning for Dynamic Underactuated Mechanical Systems with Non-holonomic Constraints. *The International Journal of Robotics Research*, 26:1075 – 1124, 2007.

[95] A. Sokolov and P. Xirouchakis. Kinematics of a 3-DoF Parallel Manipulator with an R-P-S Joint Structure. *Robotica*, 23:207 – 217, 2005.

[96] A. Sokolov and P. Xirouchakis. Dynamics Analysis of a 3-DoF Parallel Manipulator with R-P-S Joint Structure. *Mechanism and Machine Theory*, 42:541 – 557, 2007.

[97] B. M. St-Onge and C. M. Gosselin. Singularity Analysis and Representation of the General Gough-Stewart Platform. *The International Journal of Robotics Research*, 19:271 – 288, 2000.

[98] C. W. Stammers, P. H. Prest, and C. G. Mobley. A Friction Drive Robot Wrist: Electronic and Control Requirements. *Mechatronics*, 2:391 – 401, 1992.

[99] D. Stewart. A Platform with Six Degrees of Freedom. *Proceedings of the Institution of Mechanical Engineers*, 180:371 – 378, 1965.

[100] K. Tchoń and J. Jakubiak. Endogenous Configuration Space Approach to Mobile Manipulators: a Derivation and Performance Assessment of Jacobian Inverse Kinematics Algorithms. *International Journal of Control*, 76:1387 – 1419, 2003.

[101] K. Tchoń, J. Jakubiak, P. Grosch, and F. Thomas. Motion Planning for Parallel Robots with Non-Holonomic Joints. *Latest Advances in Robot Kinematics, J. Lenarcic and M. Husty (editors), Springer*, pages 115 – 122, 2012.

[102] F. Thomas. Approaching Dual Quaternions From Matrix Algebra. *IEEE Transactions on Robotics*, 30:1037 – 1048, 2014.

[103] D. Tilbury, R. M. Murray, and S. S. Sastry. Trajectory Generation for the n-Trailer Problem, Using Goursat Normal Form. *IEEE Transactions on Automatic Control*, 40:802 – 819, 1995.

[104] V. Vagner. Geometrical Interpretation of the Motion of Non-Holonomic Dynamical Systems. *Proceedings of the Seminar on Vector and Tensor Analysis*, 5:301 – 327, 1941.

[105] K. Wohlhart. Displacement Analysis of the General Spherical Stewart Platform. *Mechanism and Machine Theory*, 29:581 – 589, 1994.

[106] M. Y. Wu. Some New Results in Linear Time-Varying Systems. *IEEE Transactions on Automatic Control*, 20:159 – 161, 1975.

[107] H. Zhu. Single Motor Driven Hyper-Redundant Manipulator. *http://www.imdl.gatech.edu/haihong/Arm/Arm.html*, 2003.

[108] D. Zlatanov, R. G. Fenton, and B. Benabib. A Unifying Framework for Classification and Interpretation of Mechanism Singularities. *ASME Journal of Mechanical Design*, 117:566 – 572, 1995.