

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral Programme

AUTOMATIC CONTROL, ROBOTICS AND COMPUTER VISION

Ph.D. Thesis

NUMERICAL COMPUTATION AND AVOIDANCE  
OF MANIPULATOR SINGULARITIES

**Oriol Bohigas Nadal**

Advisors:  
Lluís Ros and Montserrat Manubens

February 2013

Universitat Politècnica de Catalunya

Doctoral Programme:  
Automatic Control, Robotics and Computer Vision

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

Advisors:  
Lluís Ros and Montserrat Manubens

Author:  
Oriol Bohigas Nadal

# NUMERICAL COMPUTATION AND AVOIDANCE OF MANIPULATOR SINGULARITIES

**Oriol Bohigas Nadal**

## Abstract

This thesis develops general solutions to two open problems of robot kinematics: the exhaustive computation of the singularity set of a manipulator, and the synthesis of singularity-free paths between given configurations. Obtaining proper solutions to these problems is crucial, because singularities generally pose problems to the normal operation of a robot and, thus, they should be taken into account before the actual construction of a prototype. The ability to compute the whole singularity set also provides rich information on the global motion capabilities of a manipulator. The projections onto the task and joint spaces delimit the working regions in such spaces, may inform on the various assembly modes of the manipulator, and highlight areas where control or dexterity losses can arise, among other anomalous behaviour. These projections also supply a fair view of the feasible movements of the system, but do not reveal all possible singularity-free motions. Automatic motion planners allowing to circumvent problematic singularities should thus be devised to assist the design and programming stages of a manipulator.

The key role played by singular configurations has been thoroughly known for several years, but existing methods for singularity computation or avoidance still concentrate on specific classes of manipulators. The absence of methods able to tackle these problems on a sufficiently large class of manipulators is problematic because it hinders the analysis of more complex manipulators or the development of new robot topologies. A main reason for this absence has been the lack of computational tools suitable to the underlying mathematics that such problems conceal. However, recent advances in the field of numerical methods for polynomial system solving now permit to confront these issues with a very general intention in mind.

The purpose of this thesis is to take advantage of this progress and to propose general robust methods for the computation and avoidance of singularities on non-redundant manipulators of arbitrary architecture. Overall, the work seeks to contribute to the general understanding on how the motions of complex multibody systems can be predicted, planned, or controlled in an efficient and reliable way.



# Acknowledgements

I would like to thank my supervisors, Lluís Ros and Montserrat Manubens, for their valuable guidance and unconditional support, and Josep Maria Porta, whose tremendous help and advice has shaped many parts of this thesis.

I am deeply grateful and indebted to Michael E. Henderson and Dimiter Zlatanov, who welcomed me to work with them during several months in what became very fruitful research stays.

I also very much appreciate the advice, assistance and encouragement that I received from friends, colleagues, and family during my work.

This work has been partially supported by an FPI grant (BES-2008-006696) of the Spanish Ministry of Education and Science under the research projects:

**DPI2007-60858** CUIK+: Analysis and Motion Planning of Complex Robotic Systems.

**DPI2010-18449** CUIK++: An Extension of Branch-and-Prune Techniques for Motion Analysis and Synthesis of Complex Robotic Systems.

The latest version of this thesis can be obtained through <http://www.iri.upc.edu/people/obohigas/>.



# Contents

|   |            |
|---|------------|
| <b>Abstract</b>   | <b>i</b>   |
| <b>Acknowledgements</b>   | <b>iii</b> |
| <b>1 Introduction</b>   | <b>1</b>   |
| 1.1 Motivation . . . . .  | 1          |
| 1.2 Objectives and Scope . . . . .                              | 5          |
| 1.3 Outlook at the Dissertation . . . . .                       | 6          |
| <b>I Computation of Singularities</b>                           | <b>9</b>   |
| <b>2 Singularity Types</b>                                      | <b>11</b>  |
| 2.1 Forward and Inverse Singularities . . . . .                 | 11         |
| 2.2 A Geometric Interpretation of Singularities . . . . .       | 14         |
| 2.2.1 Input, Output, and C-space Singularities . . . . .        | 14         |
| 2.2.2 Singularities as Critical Points . . . . .                | 16         |
| 2.3 Zlatanov’s Six Types . . . . .                              | 20         |
| 2.4 The 3-slider Mechanism . . . . .                            | 25         |
| <b>3 Numerical Computation of Singularity Sets</b>              | <b>29</b>  |
| 3.1 Solution Approach . . . . .                                 | 29         |
| 3.2 Formulating the Equations of the Singularity Sets . . . . . | 31         |
| 3.2.1 Obtaining the Assembly Constraints . . . . .              | 31         |
| 3.2.2 Obtaining the Velocity Equation . . . . .                 | 34         |
| 3.3 Isolating the Singularity Sets . . . . .                    | 37         |
| 3.3.1 Reduction to a Simple Quadratic Form . . . . .            | 37         |
| 3.3.2 Initial Bounding Box . . . . .                            | 38         |
| 3.3.3 A Branch-and-prune Method . . . . .                       | 38         |
| 3.4 Visualizing the Singularity Sets . . . . .                  | 41         |
| 3.5 Examples . . . . .  | 44         |
| 3.5.1 The 3-RRR Manipulator . . . . .                           | 44         |
| 3.5.2 The Stewart Platform . . . . .                            | 52         |
| 3.5.3 A Double-loop Manipulator . . . . .                       | 56         |
| <b>4 Workspace Determination</b>                                | <b>63</b>  |
| 4.1 Related Work . . . . .                                      | 63         |
| 4.2 The Workspace and its Boundaries . . . . .                  | 64         |
| 4.2.1 Jacobian Rank Deficiency Conditions . . . . .             | 65         |
| 4.2.2 Barrier Analysis . . . . .                                | 67         |
| 4.3 Issues of Continuation Methods . . . . .                    | 69         |
| 4.4 An Alternative Approach . . . . .                           | 72         |
| 4.4.1 Joint Limit Constraints . . . . .                         | 72         |
| 4.4.2 Equations of the Singularity Set . . . . .                | 74         |

|            |   |            |
|------------|---|------------|
| 4.4.3      | Equation Solution and Boundary Identification . . . . .     | 77         |
| 4.5        | Examples . . . . .  | 79         |
| 4.5.1      | Multi-component Workspaces . . . . .                        | 79         |
| 4.5.2      | Hidden Barriers . . . . .                                   | 81         |
| 4.5.3      | Degenerate Barriers . . . . .                               | 85         |
| 4.5.4      | Very Complex Manipulators . . . . .                         | 89         |
| <b>II</b>  | <b>Avoidance of Singularities</b>                           | <b>95</b>  |
| <b>5</b>   | <b>A General Singularity-free Path Planner</b>              | <b>97</b>  |
| 5.1        | Previous Approaches . . . . .                               | 98         |
| 5.2        | Equations of the Singularity-free C-space . . . . .         | 98         |
| 5.3        | Exploring the Singularity-free C-space for a Path . . . . . | 101        |
| 5.3.1      | Defining a Chart . . . . .                                  | 101        |
| 5.3.2      | Constructing an Atlas . . . . .                             | 102        |
| 5.3.3      | Focusing on the Path to the Goal . . . . .                  | 105        |
| 5.3.4      | The Planner Algorithm . . . . .                             | 105        |
| 5.4        | Examples . . . . .  | 108        |
| 5.4.1      | A Three-dimensional Example . . . . .                       | 108        |
| 5.4.2      | The 3-RRR Manipulator . . . . .                             | 110        |
| <b>6</b>   | <b>Adding Further Constraints</b>                           | <b>115</b> |
| 6.1        | Problem Statement . . . . .                                 | 116        |
| 6.2        | Equations of the Wrench-feasible C-space . . . . .          | 120        |
| 6.2.1      | Formulating the Wrench-feasible C-space . . . . .           | 120        |
| 6.2.2      | Properties of $\mathcal{M}$ . . . . .                       | 122        |
| 6.3        | Examples . . . . .  | 124        |
| 6.3.1      | The Stewart Platform . . . . .                              | 125        |
| 6.3.2      | Cable-driven Hexapods . . . . .                             | 126        |
| 6.4        | Additional Constraints and Robot Architectures . . . . .    | 131        |
| <b>III</b> | <b>Closing Remarks</b>                                      | <b>135</b> |
| <b>7</b>   | <b>Conclusions</b>  | <b>137</b> |
| 7.1        | Summary of Contributions . . . . .                          | 137        |
| 7.2        | Future Research Directions . . . . .                        | 139        |
| <b>A</b>   | <b>List of Publications</b>                                 | <b>143</b> |
|            | <b>References</b>   | <b>145</b> |
|            | <b>Author Index</b>   | <b>157</b> |



# Figures

|      |  |    |
|------|--|----|
| 1.1  | Loss of motion control at a singular configuration of a 3-RPR manipulator . . . . .  | 2  |
| 1.2  | Singular and non-singular configurations of an Hexaglide manipulator . . . . .       | 3  |
| 1.3  | Different topologies of robot manipulators . . . . .                                 | 6  |
| 2.1  | Inclusion relationships of the main types of singularities . . . . .                 | 15 |
| 2.2  | Examples of C-space singularities . . . . .  | 16 |
| 2.3  | Critical points of the projection map from the sphere to $\mathbb{R}^2$ . . . . .    | 17 |
| 2.4  | Interpretation of input and output singularities . . . . .                           | 18 |
| 2.5  | Trajectories through output singularities . . . . .                                  | 19 |
| 2.6  | The 3-slider and 4-bar mechanisms . . . . .  | 21 |
| 2.7  | C-space and singularities of the 3-slider mechanism . . . . .                        | 26 |
| 3.1  | The singularity set is a general algebraic variety . . . . .                         | 30 |
| 3.2  | Geometric elements intervening in the assembly of joints . . . . .                   | 33 |
| 3.3  | Linear relaxations of a parabola and a hyperbolic paraboloid . . . . .               | 39 |
| 3.4  | Progression of the numerical algorithm on two examples . . . . .                     | 42 |
| 3.5  | Visualizing the singularity set through portraits . . . . .                          | 43 |
| 3.6  | A planar 3-RRR manipulator . . . . .   | 45 |
| 3.7  | Output portrait of the 3-RRR manipulator . . . . .                                   | 48 |
| 3.8  | Input portrait of the 3-RRR manipulator . . . . .                                    | 49 |
| 3.9  | Slices of the output portrait of the 3-RRR manipulator . . . . .                     | 51 |
| 3.10 | The Stewart platform . . . . .   | 52 |
| 3.11 | Forward singularity locus of the INRIA left hand at a constant orientation . . . . . | 55 |
| 3.12 | A 2-DOF double-loop planar manipulator . . . . .                                     | 56 |
| 3.13 | C-space of the double-loop planar manipulator . . . . .                              | 57 |
| 3.14 | Singularity sets of the double-loop planar manipulator (i) . . . . .                 | 58 |
| 3.15 | Singularity sets of the double-loop planar manipulator (ii) . . . . .                | 58 |
| 3.16 | Singularity sets of the double-loop planar manipulator (iii) . . . . .               | 59 |
| 3.17 | Singularity sets of the double-loop planar manipulator (iv) . . . . .                | 59 |
| 4.1  | Singularity types in the workspace . . . . .   | 66 |
| 4.2  | Workspace of a planar 3R manipulator . . . . .                                       | 67 |
| 4.3  | Performance of a continuation method on workspace determination . . . . .            | 71 |
| 4.4  | Joint limit constraints for angular and distance variables . . . . .                 | 73 |
| 4.5  | Mechanical limits on universal and spherical joints . . . . .                        | 74 |
| 4.6  | Boundary identification process . . . . .  | 77 |
| 4.7  | A version of the planar double-butterfly linkage . . . . .                           | 80 |
| 4.8  | Reachable workspace of the double-butterfly linkage . . . . .                        | 82 |
| 4.9  | Boundary identification process on the double-butterfly linkage . . . . .            | 83 |
| 4.10 | Constant orientation workspace of a Stewart platform . . . . .                       | 84 |
| 4.11 | The 3-UPS/S and 3-RRR spherical platforms . . . . .                                  | 86 |
| 4.12 | Workspace of a 3-UPS/S platform . . . . .  | 88 |
| 4.13 | Constant torsion workspace of a 3-UPS/S platform . . . . .                           | 89 |
| 4.14 | The workspace of the Agile Eye has degenerate barriers . . . . .                     | 90 |

|      |   |     |
|------|---|-----|
| 4.15 | A one-parameter family of workspaces leading to the workspace of the Agile Eye .          | 91  |
| 4.16 | A 15-link mechanism . . . . .   | 92  |
| 4.17 | Position workspaces of the 15-link manipulator . . . . .                                  | 92  |
|      |   |     |
| 5.1  | A smooth manifold modelling the singularity-free C-space . . . . .                        | 100 |
| 5.2  | Main steps of the higher-dimensional continuation method . . . . .                        | 102 |
| 5.3  | The process of polytope-based chart construction . . . . .                                | 103 |
| 5.4  | Progression of the continuation method on the surface of a torus . . . . .                | 104 |
| 5.5  | A fictitious three-dimensional C-space . . . . .  | 108 |
| 5.6  | Two paths computed on the fictitious three-dimensional C-space . . . . .                  | 109 |
| 5.7  | Workspace, singularities, and working modes of a 3- <u>RRR</u> planar manipulator . . .   | 111 |
| 5.8  | Singularity-free path for a 3- <u>RRR</u> manipulator at a constant orientation . . . . . | 112 |
| 5.9  | Several steps of a singularity-free motion of a 3- <u>RRR</u> manipulator . . . . .       | 113 |
|      |   |     |
| 6.1  | Two hexapod manipulators with rigid and cable legs . . . . .                              | 116 |
| 6.2  | Applications of hexapod manipulators . . . . .  | 118 |
| 6.3  | Different manipulation scenarios . . . . .  | 119 |
| 6.4  | The ellipsoid of possible leg forces . . . . .  | 121 |
| 6.5  | Two wrench-feasible paths computed for a Stewart platform . . . . .                       | 125 |
| 6.6  | CAD model of a cable-driven hexapod . . . . .   | 127 |
| 6.7  | Slices of the wrench-feasible C-space of the NIST Robocrane . . . . .                     | 129 |
| 6.8  | Two wrench-feasible paths for the NIST Robocrane . . . . .                                | 130 |
| 6.9  | A wrench-feasible path for the NIST Robocrane . . . . .                                   | 131 |
| 6.10 | Example of alternative serial kinematic chains used in parallel robots . . . . .          | 132 |

# Tables

|     |  |     |
|-----|--|-----|
| 2.1 | The six singularity types exemplified with the 3-slider and 4-bar mechanisms . . . | 22  |
| 3.1 | Unit twists of lower-pair joints . . . . .   | 35  |
| 3.2 | Joint coordinates of the considered 3-RRR manipulators . . . . .                   | 46  |
| 3.3 | Parameters used on computing the singularity set of the Stewart platform . . . .   | 53  |
| 3.4 | Performance data on the reported examples of singularity set computation . . . .   | 62  |
| 4.1 | Performance data on computing the considered workspace examples . . . . .          | 79  |
| 4.2 | Parameters used on computing the workspace of the Stewart platform . . . . .       | 85  |
| 6.1 | Parameters of the NIST Robocrane manipulator . . . . .                             | 128 |



# 1

## Introduction

Singularity analysis is a central topic of robot kinematics. It has as a goal the study of certain configurations, termed singular or critical, where important changes take place in the kineto-static performance of a manipulator. Motion control or dexterity losses can arise, and there may appear unresolvable or uncontrollable end-effector forces, among other effects (Figs. 1.1 and 1.2). The study of singularities is thus motivated by a desire to avoid these configurations, but it may be helpful to operate close to them sometimes, such as when handling heavy objects, drilling, or fine-positioning, where extreme force or motion transformation ratios are often required.

While many commercial manipulators are designed not to include singularities in their workspace, this is not always possible, and the appearance of manipulators with increasingly complex motions makes the availability of general tools for singularity analysis essential to assist the robot design and programming processes. In this context, the ability to identify all singular configurations, and to plan manipulator motions avoiding them, become crucial issues to be solved.

The main objective of this thesis is to develop reliable methods to solve such issues on the very large class of lower-pair manipulators of general architecture. The emphasis is placed on mechanisms involving closed kinematic chains, because they are those exhibiting the whole range of critical phenomena and present a more intricate analysis, and the attention is restricted to the non-redundant case, which allows a simpler, more symmetric presentation of results.

### 1.1 Motivation

In a recent editorial [5], McCarthy states that the ability to analyze and design mechanisms and robotic systems of increasing complexity has strongly depended on the ability to solve the associated systems of polynomial equations. It can be expected, he follows, that current

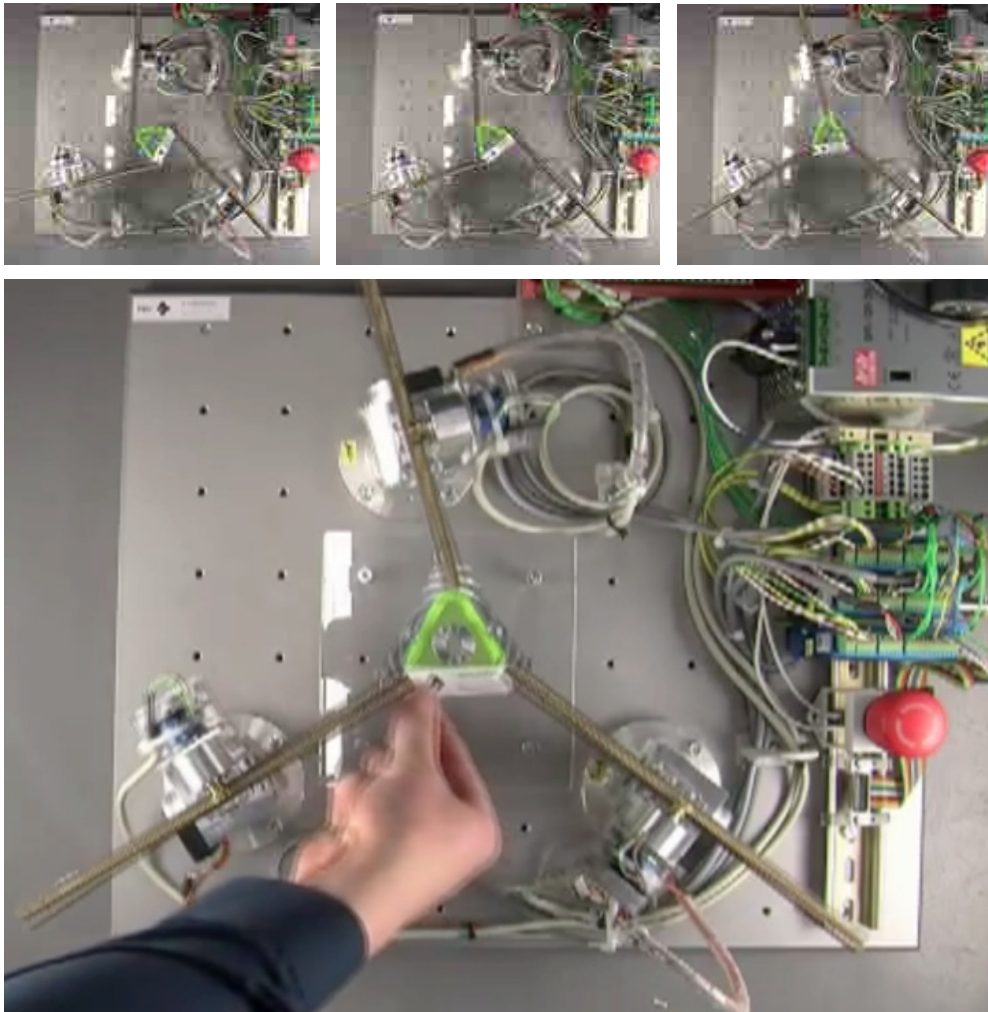


Figure 1.1: Loss of motion control at a singular configuration. The snapshots are taken from an experiment carried out at the Vorarlberg University of Applied Sciences, in which a 3-RPR manipulator is moved to a singular configuration (top sequence) so that the operator can feel the shakiness of the configuration (bottom picture). The actuator velocities do not determine the platform velocities in this point and there is a dual loss in the capacity to withstand external forces. The manipulator gets locked in this experiment, and it has to be taken to a regular configuration manually to resume normal functioning. The full sequence of the experiment is available in [1].

advances in computer algebra and numerical continuation methods will advance research in mechanisms and robotics in the coming years. The problems of singularity computation and avoidance are clear examples of this situation. Although a proper definition of singularity existed for several years, it is only now, after some recent results on polynomial system solving, that such problems become tackleable by extending the current state of the art.

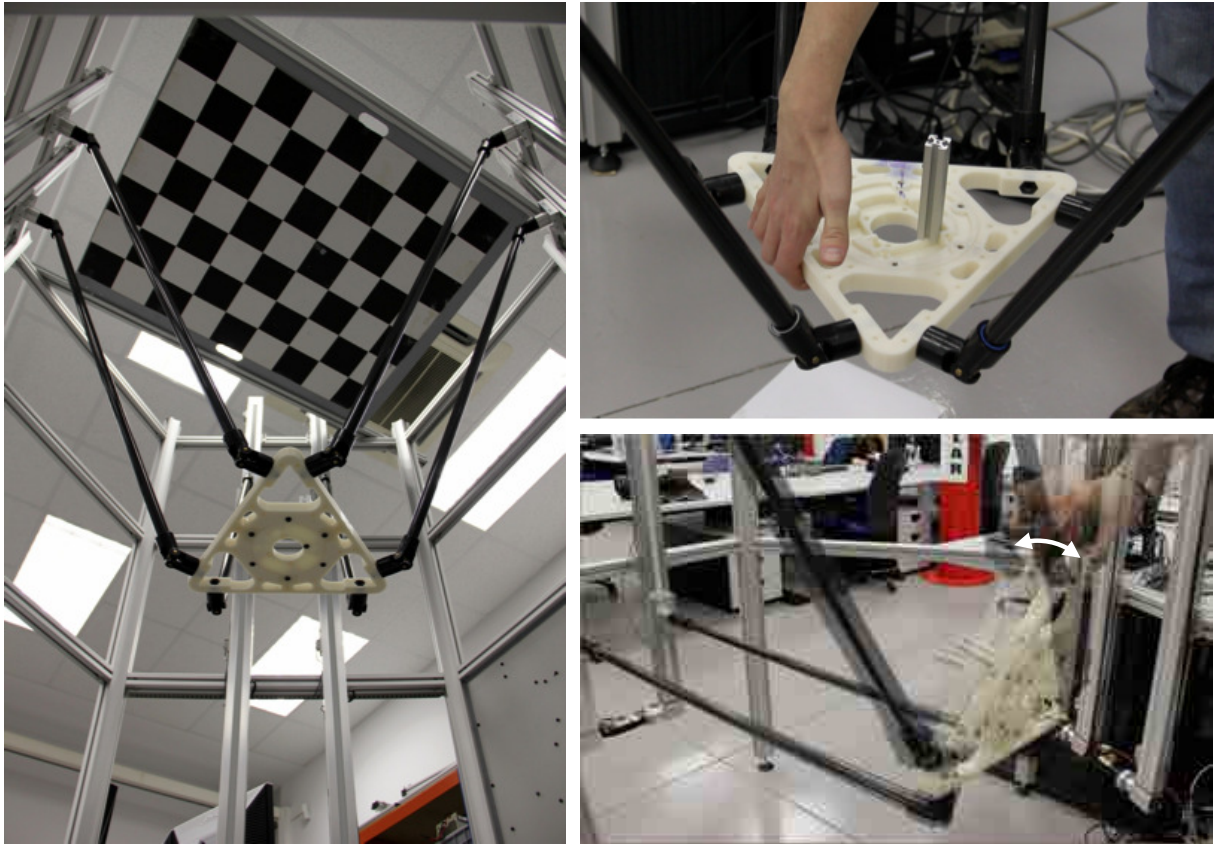


Figure 1.2: Hexaglide manipulator constructed at the Institut de Robòtica i Informàtica Industrial [2]. These robots are typically used as high-speed CNC milling machines [3,4]. When locking the actuators, the platform is stiff in a non-singular configuration (top right), but it shows uncontrollable motion in a singular one (bottom right).

Investigation on singularity analysis in the context of robot kinematics started to gain momentum in the early 1980s. Initially, most works were restricted to specific classes of manipulators, specially those with serial kinematic chains, and towards the mid-1990s an approach allowing to characterize all singularities of a manipulator with an arbitrary architecture had not been completed yet. The lack of a standard definition of singularity did not allow the development of general methods to analyze more complex manipulators.

The earliest attempt to provide a general framework to determine and classify all singular configurations can be attributed to Gosselin and Angeles in 1990, who proposed the use of input/output velocity equations to define the well-known Type I and Type II singularities, where the velocity of the end-effector does not determine the velocities of the actuators, and *vice versa* [6]. The approach was sound, but neglected the role played by passive joint velocities,

and it was later found that further singularity types existed that could not be framed into their formalism [7–11]. This observation led Zlatanov to define singular configurations in a more general way, as those where the forward (FIKP) or the inverse (IIKP) instantaneous kinematic problems, understood as the computation of the *overall* configuration velocity given the input or output velocities, becomes undetermined [12]. Zlatanov’s characterization of singularity is probably the most systematic and general one proposed so far in the literature, and accommodates, as special cases, the earlier Type I/II singularities, and subtle singularities, such as constraint [13, 14] or architecture singularities [15, 16]. By that time, this allowed to identify whether a specific configuration of an arbitrary manipulator was singular, but it was clear that for theoretical and practical reasons it was important to have a means of obtaining the whole set of singular configurations, and to be able to plan manipulator motions avoiding some or all of its subsets. The rigorous definition of kinematic singularity provided by Zlatanov certainly laid the ground for a systematic singularity analysis, but Zlatanov himself pointed out that significant computational problems remained to be settled before the emergence of a general method able to solve the associated systems of equations [12, Chapter 7].

Since then, a number of works dealing with singularity set computation or avoidance have appeared [17, 18], specially in the context of parallel manipulators. Relevant work in this respect is due to Bonev [19], who studied the singularity sets of several parallel platforms; Mayer St-Onge et al. [20] or Li et al. [21], who provided an analytic form of the six-dimensional singularity set of the general Stewart platform; Merlet [22], who presented a formal-numerical approach for in-workspace singularity detection; Dasgupta and Mruthyunjaya [23] or Sen et al. [24], who proposed path-deformation or variational techniques for singularity-free path planning; or Jiang and Gosselin [25], who obtained a method for computing the maximal singularity-free orientation workspace for the Stewart platform. Also with application to various forms of such platform, Borràs et al. discovered a number of leg rearrangements that leave the singularity set invariant, allowing to evolve existing designs with a known singularity locus to newer designs that are much easier to construct [26–29]. The advances in the field are significant, but all previous works restrict their attention to narrowly-defined classes of manipulators, and approaches providing a general way to compute or avoid the singularities have not been given yet.

The characteristics of each problem pose different challenges and thus demand distinct approaches. Because the singularity set typically has the structure of a general algebraic variety, the computation of all singularities requires a method to obtain all real solutions of a system of polynomial equations, even if such solutions exhibit several connected components, positive-dimensional continua, or non-smoothnesses like bifurcations, sharpnesses, or dimension changes. In contrast, the avoidance of singularities requires a method able to de-



termine a singularity-free path connecting given configurations, whenever one exists, taking into account that the whole path must lie in a non-parameterizable variety implicitly defined by a system of nonlinear equations, usually embedded in a highly-dimensional ambient space. Whereas branch-and-prune methods constitute the ideal basis from which to confront the former problem [30–32], the latter can be tackled by extending recently-developed tools of higher-dimensional continuation [33–35].

## 1.2 Objectives and Scope

The goal of this thesis is to provide reliable methods for solving the following open problems of robot kinematics, for manipulators of general architecture:

- **Computation of singularities:** For a given manipulator with designated input and output coordinates, compute the singularity set exhaustively and represent it in a meaningful way.
- **Avoidance of singularities:** Given two feasible configurations of the manipulator, determine whether there exists a configuration-space path connecting them that does not traverse the singularity set at any point.

For the purpose of this work a manipulator will be a multibody system with designated input and output coordinates, i.e., with a specification of which joints are actuated, and which variables describe the intended task-space functionality. The various bodies, or links, of the system will be assumed to be rigid and interconnected through lower-pair joints, and the connection pattern may be arbitrary, including closed kinematic chains in particular.

Depending on the space where the links operate, such systems are usually referred to as *planar*, *spherical*, or *spatial* manipulators. While planar and spherical manipulators usually have three degrees of freedom (two displacements and one orientation, and three orientations, respectively), spatial manipulators usually have six (three displacements and three orientations). A number of different architectures may be encountered in practice (Fig. 1.3), and our results will be applicable to all them indistinctively. However, for the sake of clarity, this work will focus on non-redundantly actuated mechanisms, where the number of inputs, and also the outputs, is equal to the number of degrees of freedom to be governed. The results should be extendable to the redundant case as well, but the developments have a simple symmetry in the non-redundant case that becomes obscured when redundant actuation is possible.

Finally, we note that collision constraints between the various links of the manipulator will be neglected in the whole study. Such constraints add considerable complexity to the problems under investigation and it is plausible to deal with them *a posteriori* in a generate-and-test



Figure 1.3: Left to right and top to bottom: The DLR LWR arm and hand [36], a 3(P)RRR planar reconfigurable manipulator [37], the tripteron mechanism [38], a commercial Stewart platform [39], a spherical parallel manipulator for the rapid orientation of a camera [40], the exechon X700 parallel kinematics machine [41], the Kawasaki Delta YF03N manipulator [42], the rotary hexapod MicARH for micropositioning [43], and the IRI hexacrane [44].

fashion, using collision checkers like those employed in probabilistic path planning methods. Mechanical joint limits, however, will be naturally accommodated by our approach.

### 1.3 Outlook at the Dissertation

The dissertation is divided into three parts. Part I is devoted to the description of the different singularities that can be encountered, and to the development of methods for their exhaustive

computation. The methods allow to obtain the different singularity sets, and to generate representations of the various workspace of a manipulator. It is shown how the representations are rich in information, but do not reveal all possible singularity-free movements of a manipulator. Part II overcomes this issue by providing tools to determine singularity-free paths between given configurations, or to determine the whole singularity-free region that is achievable from a given configuration. Finally, Part III provides the conclusions of the thesis, a list of derived publications, and references to previous work in the literature. The contents of each chapter are next summarized:

**Chapter 1** explains the motivation of this research and provides its main objectives and scope.

**Chapter 2** gives a complete view of the main singularity types and the mathematical conditions that characterize the configurations of each type. Using concepts from differential geometry and calculus, we give a geometric interpretation to these conditions, and discuss the kinematic consequences of traversing each singularity subset. The definition of Zlatanov's six lower-level singularity types is then recalled, and all singularities are illustrated on a simple manipulator that admits an analytical approach.

**Chapter 3** presents a method to compute any of the singularity sets defined in the previous chapter. The method allows to isolate any the sets at the desired accuracy and is very general, since it can be applied to any non-redundant manipulator. To do so, we formulate the systems of equations characterizing the singularity sets using quadratic polynomials only, and then exploit the special form of such polynomials to define a branch-and-prune scheme able to compute all singularity points. Since the singularity sets are usually defined in a highly-dimensional space, we also provide hints on how to represent these sets in a meaningful and informative way.

**Chapter 4** extends the developments in Chapters 2 and 3 to allow the computation and representation of the various workspaces of a manipulator. It is shown how the computation of certain generalised singularities leads to identifying the boundary of a workspace, and the various motion barriers and mobility losses present in its interior. A detailed map of the workspace is obtained as a result, indicating all interior and exterior regions and the motion impediments that separate them. Mechanical joint limits are taken into account in the calculations, since they can greatly modify the shape of the workspace. The chapter also compares the proposed method with previous methods of a similar generality.

**Chapter 5** provides a singularity-free path planning method. Given two configurations of the manipulator, the method attempts to connect them through a path that does not collide with the singularity set at any point. The method can be applied to non-redundant manipulators of

general architecture, and it always returns a path whenever one exists at a given resolution, or determines path non-existence otherwise. The crucial idea is that the singularity-free subsets of the configuration space can be placed in correspondence with a smooth manifold, so that by maneuvering through such manifold one can perform all possible singularity-free movements of the manipulator, and hence detect the desired paths between the query configurations. To guarantee a minimum clearance with respect to the singularity set, the determinant of a Jacobian matrix is forced to remain smaller than a given threshold, which is a suitable criterion when only the multibody topology of the underlying mechanism is available.

**Chapter 6** shows that, when further mechanical properties are known, more meaningful clearance measures can be taken into account. For simplicity, the chapter studies the particular case of Stewart or cable-driven hexapods, explaining how it is possible to force all configurations to remain wrench-feasible on the computed paths, i.e., able to counteract a given wrench applied to the platform, subject to bounded six-dimensional perturbations. Although the overall study is particular, hints are given on how to extend the developments to also deal with other manipulator architectures or relevant clearance measures.

**Chapter 7** finally summarizes the contributions of the thesis and outlines possible future research directions.

**Part I**

**Computation of Singularities**



# 2

## Singularity Types

The literature on singularity analysis is rich, but one often perceives a lack of familiarity with the range of critical phenomena that can arise in a manipulator, and on the kind of analysis required to detect such phenomena. To help reversing this trend, and to provide necessary background for the rest of the thesis, this chapter presents a detailed view of the main singularity types, their interpretation, and the mathematical conditions allowing to characterize them. We shall formulate systems of equations for all types, which will be useful in Chapter 3 to numerically compute the configurations of each type independently.

The presentation relies on the rigorous mathematical definitions given in [12], which depart from a system of linear equations characterizing the feasible velocities of the manipulator at a given configuration. Unlike in other approaches, this system accounts for the *whole* velocity state of the manipulator, which is crucial to detect all possible singularity types that can be encountered in practice. The definition of singularity and the mathematical conditions that characterize the singularity set are then given, and systems of equations encoding the main singularity types are derived (Section 2.1). A geometric interpretation of singularities and the kinematic consequences of traversing each singularity type are then provided (Section 2.2), and the definition of Zlatanov's six lower-level singularity types is recalled (Section 2.3). Following each definition, a system of equations allowing to obtain the configurations of the corresponding type are derived, and the application of these systems to a simple manipulator is illustrated in the end (Section 2.4).

### 2.1 Forward and Inverse Singularities

Every configuration of a manipulator can be described by a tuple  $\mathbf{q}$  of scalar generalized coordinates, and the set of all possible configurations is called the *configuration space*, or C-space, of the manipulator. For manipulators with closed kinematic chains, not all values of  $\mathbf{q}$  correspond

to feasible configurations, and the C-space is given by the solution set of a system of non-linear equations,

$$\Phi(\mathbf{q}) = \mathbf{0}, \quad (2.1)$$

which expresses the assembly constraints imposed by the joints [45]. Here,  $\Phi : \mathcal{Q} \rightarrow \mathcal{E}$  is a smooth map, where  $\mathcal{Q}$  and  $\mathcal{E}$  are  $n_q$ - and  $n_e$ -dimensional manifolds. The C-space of the manipulator is thus the set

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

The Jacobian matrix  $\Phi_{\mathbf{q}} = [\partial\Phi_i/\partial q_j]$  is usually full rank at all points  $\mathbf{q} \in \mathcal{C}$  except on a subset  $\mathcal{G} \subset \mathcal{C}$  where  $\mathcal{C}$  may lose the manifold structure. Thus,  $\mathcal{C} \setminus \mathcal{G}$  is a smooth manifold of dimension  $n = n_q - n_e$ , which is also taken as the dimension of the C-space.

Moreover, the possible instantaneous motions of the manipulator at a configuration  $\mathbf{q}$  can be characterized by a system of linear equations,

$$\mathbf{L} \mathbf{m} = \mathbf{0}, \quad (2.2)$$

where  $\mathbf{L}$  is a matrix that depends on the configuration  $\mathbf{q}$ , and  $\mathbf{m}$  is the so-called *velocity vector* of the manipulator, which globally characterizes its velocity state and takes the form  $\mathbf{m} = [\mathbf{m}_u^\top, \mathbf{m}_v^\top, \mathbf{m}_p^\top]^\top$ , where  $\mathbf{m}_u$ ,  $\mathbf{m}_v$ , and  $\mathbf{m}_p$  encode the output, input and passive velocities respectively. Typically,  $\mathbf{m}_u$  provides the velocity of a point or the angular velocity, or the twist, of an end-effector body,  $\mathbf{m}_v$  provides the actuated-joint velocities of the manipulator, and  $\mathbf{m}_p$  encompasses the remaining velocity coordinates. Such a system, called the *velocity equation* in [12], can be obtained for any manipulator by differentiation of Eq. (2.1) (provided that  $\mathbf{q}$  includes the input and output coordinates), or using twist loop equations, or by any other means, and therefore it can be used for the practical identification of singularities.

In general, the instantaneous kinematic analysis of a manipulator addresses two main problems:

- The *forward* instantaneous kinematics problem (FIKP): find the velocity vector,  $\mathbf{m}$ , when the input velocity  $\mathbf{m}_v$  is given.
- The *inverse* instantaneous kinematics problem (IIKP): find the velocity vector,  $\mathbf{m}$ , when the output velocity  $\mathbf{m}_u$  is given.

Note that, contrarily to what is assumed elsewhere [6], in both cases it is required to find *all* velocity components of  $\mathbf{m}$ , not just those referring to the output or input velocities, respectively. Following [12], a configuration  $\mathbf{q}$  is defined as *nonsingular* when both the FIKP and the IIKP have unique solutions for any input or output velocity, and as *singular* otherwise. The set  $\mathcal{S}$  of



all singular configurations is called the *singularity set* of the manipulator, and a configuration where the FIKP (resp. IIKP) does not have a unique solution for any input (resp. output) velocity will be called a *forward* (resp. *inverse*) singularity.

To characterize the forward and inverse singularity subsets, consider now the partitions  $\mathbf{m} = [\mathbf{m}_v^T, \mathbf{m}_y^T]^T$  and  $\mathbf{m} = [\mathbf{m}_u^T, \mathbf{m}_z^T]^T$ , where  $\mathbf{m}_y$  and  $\mathbf{m}_z$  encompass the velocities remaining in  $\mathbf{m}$  after the removal of  $\mathbf{m}_v$  and  $\mathbf{m}_u$ , respectively. Let also  $L_v$  and  $L_u$  be the submatrices of  $L$  containing the columns corresponding to the input and output velocities, and  $L_y$  and  $L_z$  those obtained by removing  $L_v$  and  $L_u$  from  $L$ . This allows writing Eq. (2.2) as either

$$L_v \mathbf{m}_v + L_y \mathbf{m}_y = \mathbf{0}, \quad (2.3)$$

or

$$L_u \mathbf{m}_u + L_z \mathbf{m}_z = \mathbf{0}. \quad (2.4)$$

Since the manipulator is assumed to be non-redundant, both the number of coordinates in  $\mathbf{m}_v$  and  $\mathbf{m}_u$  is equal to the global mobility of the kinematic chain, defined as the dimension  $n$  of the C-space. In particular, the number of columns of  $L$  exceeds the number of rows by exactly  $n$  and  $L_y$  and  $L_z$  are square matrices [12]. Thus, for configurations  $\mathbf{q}$  on which  $L_y$  and  $L_z$  are nonsingular, we can write Eq. (2.3) and (2.4) in the equivalent form

$$\mathbf{m}_y = -L_y^{-1} L_v \mathbf{m}_v, \quad (2.5)$$

$$\mathbf{m}_z = -L_z^{-1} L_u \mathbf{m}_u, \quad (2.6)$$

which provide the solution to the FIKP and the IIKP of the manipulator. However, Eqs. (2.5) and (2.6) only hold whenever  $L_y$  and  $L_z$  are full rank, and only in this case the input and output velocities  $\mathbf{m}_v$  and  $\mathbf{m}_u$  will determine unique values for the remaining velocities  $\mathbf{m}_y$  and  $\mathbf{m}_z$ . This must be so because, when  $L_y$  is rank-deficient at  $\mathbf{q}$ , Eq. (2.3) yields, for a given value of  $\mathbf{m}_v$ , either no solution or infinitely-many solutions for  $\mathbf{m}_y$ , in which case it is not possible to determine the velocity  $\mathbf{m}$  of the manipulator by specifying the velocities  $\mathbf{m}_v$  of the actuators. When  $L_z$  is rank-deficient at  $\mathbf{q}$ , Eq. (2.4) reveals an analogous relationship between  $\mathbf{m}_u$  and  $\mathbf{m}$ .

Following these observations, note that a configuration  $\mathbf{q} \in \mathcal{C}$  is *singular* if, and only if, either  $L_y$  or  $L_z$  is rank deficient at  $\mathbf{q}$ , and the singularity set  $\mathcal{S}$  can be obtained as the union of the solution sets of the following systems of equations:

$$\left. \begin{array}{l} \Phi(\mathbf{q}) = \mathbf{0} \\ L_y \xi = \mathbf{0} \\ \|\xi\|^2 = 1 \end{array} \right\}, \quad (2.7) \quad \left. \begin{array}{l} \Phi(\mathbf{q}) = \mathbf{0} \\ L_z \xi = \mathbf{0} \\ \|\xi\|^2 = 1 \end{array} \right\}. \quad (2.8)$$

The first equation in each system constrains  $\mathbf{q}$  to be a feasible configuration of the manipulator, and the second and third equations enforce the existence of a non-zero vector  $\xi$  in the kernel of the corresponding matrix. Note that  $\|\xi\|^2$  can be any consistent norm, for instance  $\xi^T D \xi$  where  $D$  is a diagonal matrix with the proper physical units. There is no need for the norm to be invariant with respect to change of frame or units. In short, the condition  $\|\xi\|^2 = 1$  only serves to guarantee that  $\xi$  is not  $\mathbf{0}$ . Clearly, the points  $\mathbf{q}$  satisfying the left (resp. right) system are the forward (resp. inverse) singularities.

## 2.2 A Geometric Interpretation of Singularities

Several changes take place when traversing a singularity, both at the infinitesimal and finite motion levels. An easy way to understand such changes is provided here using concepts of differential geometry and calculus.

### 2.2.1 Input, Output, and C-space Singularities

A simple differentiation of Eq. (2.1) with respect to time provides the velocity equation of the manipulator, Eq. (2.2), as

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} = \mathbf{0},$$

where it becomes apparent that the feasible velocities at each configuration are given by the vectors  $\dot{\mathbf{q}}$  belonging to the kernel of  $\Phi_{\mathbf{q}}$ . Assuming the partitions  $\mathbf{q} = [\mathbf{v}^T, \mathbf{y}^T]^T$  and  $\mathbf{q} = [\mathbf{u}^T, \mathbf{z}^T]^T$ , where  $\mathbf{v}$  and  $\mathbf{u}$  are vectors of  $n_v$  input and  $n_u$  output coordinates, respectively, and  $\mathbf{y}$  and  $\mathbf{z}$  encompass the corresponding remaining coordinates, the previous equation can also be written as

$$\Phi_{\mathbf{v}} \dot{\mathbf{v}} + \Phi_{\mathbf{y}} \dot{\mathbf{y}} = \mathbf{0},$$

or as

$$\Phi_{\mathbf{u}} \dot{\mathbf{u}} + \Phi_{\mathbf{z}} \dot{\mathbf{z}} = \mathbf{0}.$$

In these equations,  $\Phi_{\mathbf{q}}$ ,  $\Phi_{\mathbf{v}}$ ,  $\Phi_{\mathbf{u}}$ ,  $\Phi_{\mathbf{y}}$ , and  $\Phi_{\mathbf{z}}$  play the role of  $L$ ,  $L_v$ ,  $L_u$ ,  $L_y$ , and  $L_z$  in Eqs. (2.2)–(2.4). Hereafter the  $\mathbf{v}$ - and  $\mathbf{u}$ -spaces will be denoted by  $\mathcal{V}$  and  $\mathcal{U}$  respectively.

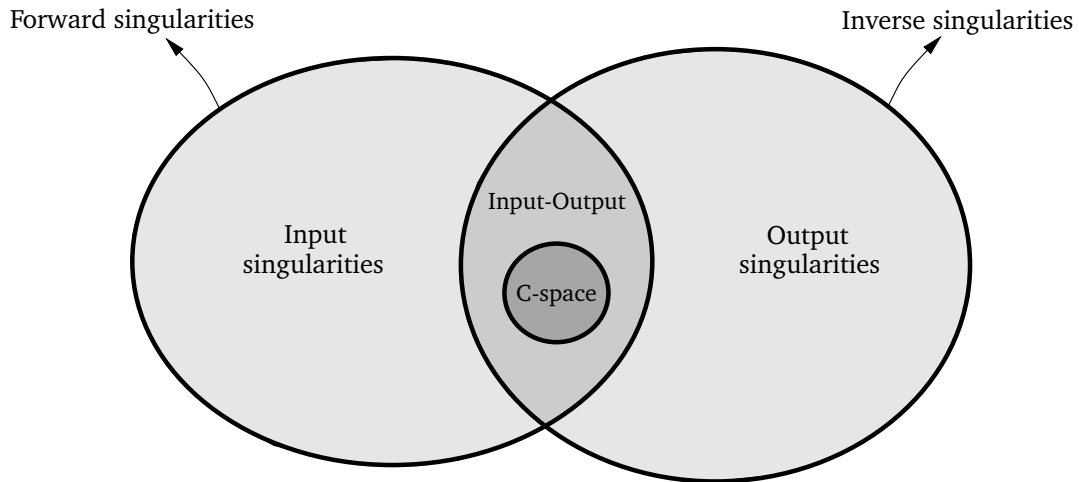


Figure 2.1: Inclusion relationships of C-space, input, and output singularities, relative to forward and inverse singularities.

Clearly, a point  $\mathbf{q}$  is a forward or an inverse singularity when either  $\Phi_{\mathbf{y}}$  or  $\Phi_{\mathbf{z}}$  is rank deficient. The previous two equations reveal that, in such configurations, the locking of the input or output coordinates yields an infinitesimally flexible, or shaky, mechanism, as the examples in Figs. 1.1 and 1.2 demonstrate.

Such a degenerate behaviour has further physical consequences, which can be better appreciated by classifying the points in  $\mathcal{S}$  according to the geometric cause of rank deficiency of  $\Phi_{\mathbf{y}}$  or  $\Phi_{\mathbf{z}}$ . Three types of singular configurations can be distinguished according to this criterion:

- *C-space singularities*: the points  $\mathbf{q} \in \mathcal{G} \subset \mathcal{C}$ , where the whole Jacobian  $\Phi_{\mathbf{q}}$  is rank deficient, so that both the FIKP and IIKP become unsolvable independently of the choice of input and output coordinates.
- *Input singularities*: the points  $\mathbf{q} \in \mathcal{C} \setminus \mathcal{G}$  where  $\Phi_{\mathbf{y}}$  is rank deficient, so that the FIKP becomes unsolvable.
- *Output singularities*: the points  $\mathbf{q} \in \mathcal{C} \setminus \mathcal{G}$  where  $\Phi_{\mathbf{z}}$  is rank deficient, so that the IIKP becomes unsolvable.

Note that when  $\Phi_{\mathbf{q}}$  is rank deficient,  $\Phi_{\mathbf{y}}$  and  $\Phi_{\mathbf{z}}$  are rank deficient too, meaning that the forward singularities contain both the input and C-space singularities, and the inverse singularities contain both the output and C-space singularities. However, a configuration can be both an input and an output singularity without being a C-space singularity, as the inclusion relationships depicted in Fig. 2.1 illustrate.

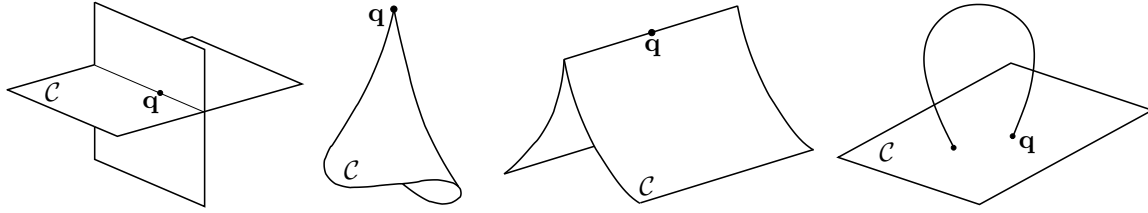


Figure 2.2: Examples of C-space singularities.

Geometrically, C-space singularities correspond to points  $\mathbf{q}$  where  $\mathcal{C}$  may lose the manifold structure, such as bifurcations, cusps, ridges, or dimension changes (Fig. 2.2). Since  $\Phi_{\mathbf{q}}$  is rank deficient at such points, the tangent space to  $\mathcal{C}$  does not exist. On input and output singularities, contrarily,  $\Phi_{\mathbf{q}}$  is full rank and  $\mathcal{C}$  has an  $n$ -dimensional tangent space, but this space has a special position, as shown in Section 2.2.2 below.

The tangent space to the C-space, wherever it exists, can be defined as the kernel of the Jacobian matrix  $\Phi_{\mathbf{q}}$ , i.e.,  $T_{\mathbf{q}}(\mathcal{C} \setminus \mathcal{G}) = \ker(\Phi_{\mathbf{q}})$ . Thus, for points in  $\mathcal{C} \setminus \mathcal{G}$ , the feasible motions lie on a tangent space of the same dimension as the C-space,  $n$ , and the mobility of the manipulator is, therefore, the same as the number of inputs or outputs. However, even though the tangent space only exists at the points of  $\mathcal{C} \setminus \mathcal{G}$ , the equation  $\Phi_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0}$  still holds for any point of  $\mathcal{C}$ . In particular, for those points in  $\mathcal{G}$ , the feasible motions lie in a space of higher dimension than  $n$ , because  $\Phi_{\mathbf{q}}$  is not full rank, and there is an increase in the instantaneous mobility of the manipulator. This explains why in C-space singularities, regardless of which  $n$  inputs or outputs are chosen, the motion of the manipulator remains undetermined. In general, we can say that the feasible velocities of the manipulator are those in the Zariski tangent space to  $\mathcal{C}$ , defined as the vector space spanned by the vectors in the kernel of  $\Phi_{\mathbf{q}}$  [12].

### 2.2.2 Singularities as Critical Points

C-space, input, and output singularities can be interpreted as critical points of specific maps, which allows to understand the physical consequences of traversing each singularity type geometrically.

To see this, let us recall the definition of critical points and values. Let  $\Gamma : \mathcal{M} \rightarrow \mathcal{N}$  be an arbitrary smooth map between two manifolds  $\mathcal{M}$  and  $\mathcal{N}$ . Then, the differential of  $\Gamma$  at  $\mathbf{q}$  is the map between tangent spaces  $\Gamma_{\mathbf{q}} : T_{\mathbf{q}}\mathcal{M} \rightarrow T_{\Gamma(\mathbf{q})}\mathcal{N}$  induced by the Jacobian matrix  $\Gamma_{\mathbf{q}} = [\frac{\partial \Gamma_i}{\partial q_j}(\mathbf{q})]$ . A point  $\mathbf{q} \in \mathcal{M}$  is said to be a *critical point* of  $\Gamma$  if the differential  $\Gamma_{\mathbf{q}}$  is not surjective at  $\mathbf{q}$ , i.e., if its Jacobian has rank less than  $\dim(\mathcal{N})$  at  $\mathbf{q}$ . On these points,  $\Gamma_{\mathbf{q}}(T_{\mathbf{q}}\mathcal{M})$  does not span the tangent space to  $\mathcal{N}$  entirely. The image of a critical point  $\mathbf{q} \in \mathcal{M}$  is a *critical value* of  $\Gamma$ , and the points of  $\mathcal{N}$  which are not critical values are called *regular values*.

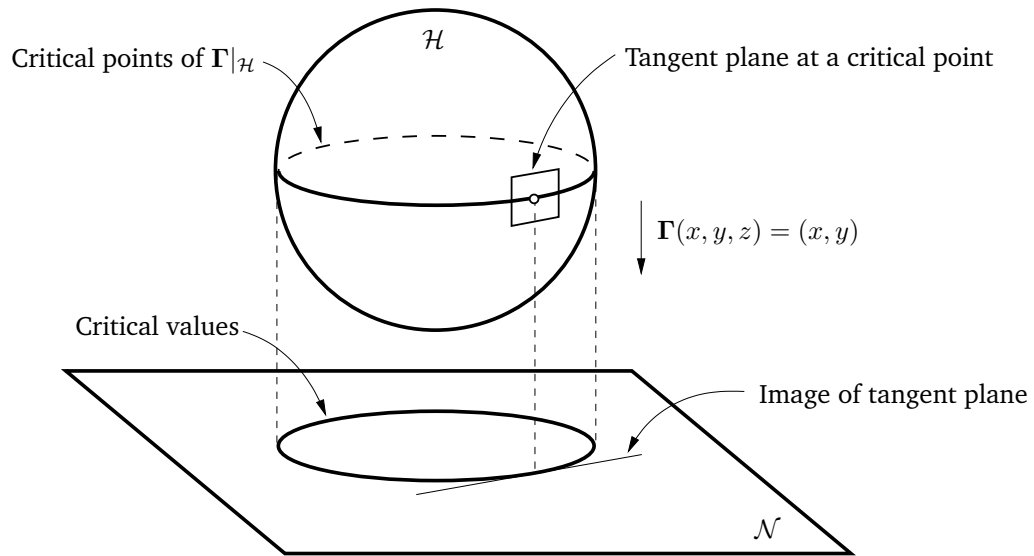


Figure 2.3: Example of critical points. Here  $\mathcal{H}$  is a sphere in  $\mathcal{M} = \mathbb{R}^3$  and  $\mathcal{N} = \mathbb{R}^2$ . The map  $\Gamma$  is the projection from  $\mathcal{M}$  to  $\mathcal{N}$ , so that  $\Gamma(x, y, z) = (x, y)$ . The figure shows the critical points and values of the projection map with domain restricted to  $\mathcal{H}$ . For most points in  $\mathcal{H}$ , the image of the tangent plane to the sphere covers the  $x$ - $y$  plane. However, at the equator the tangent plane projects down to a line in the  $x$ - $y$  plane. Figure adapted from [46].

In our case,  $\mathcal{C}$  is defined implicitly as the level set  $\Phi(\mathbf{q}) = \mathbf{0}$  of the smooth map between manifolds  $\Phi : \mathcal{Q} \rightarrow \mathcal{E}$ . It may happen that  $\mathbf{0} \in \mathcal{E}$  is not a regular value of  $\Phi$  and, hence, the set  $\mathcal{G}$  of C-space singularities is in fact the set of critical points of  $\Phi$  that map onto  $\mathbf{0}$ . If such singularities arise,  $\mathcal{C}$  is not a smooth manifold because the tangent space to  $\mathcal{C}$  does not exist in them. However, leaving out these critical points, for example by restricting the domain of  $\Phi$ , the set  $\mathcal{C} \setminus \mathcal{G}$  is indeed a smooth manifold where the tangent space exists everywhere.

To understand input and output singularities, note that one can also speak of the critical points of a map  $\Gamma$  with domain restricted to a submanifold  $\mathcal{H} \subseteq \mathcal{M}$ . These are the points  $\mathbf{q} \in \mathcal{H}$  for which  $\Gamma_{\mathbf{q}}(T_{\mathbf{q}}\mathcal{H})$  does not span  $T_{\Gamma(\mathbf{q})}\mathcal{N}$  entirely, as illustrated in Fig. 2.3, and they can be detected using the following result. A point  $\mathbf{q} \in \mathcal{C} \setminus \mathcal{G}$  is a critical point of  $\Gamma$  with domain restricted to  $\mathcal{C} \setminus \mathcal{G}$  if, and only if, the matrix

$$(\Phi, \Gamma)_{\mathbf{q}} = \begin{bmatrix} \Phi_{\mathbf{q}} \\ \Gamma_{\mathbf{q}} \end{bmatrix}$$

has rank less than  $\dim(\mathcal{E}) + \dim(\mathcal{N})$  at  $\mathbf{q}$  [46]<sup>1</sup>.

<sup>1</sup>The proof in [46] contains a flaw that is easy to repair.

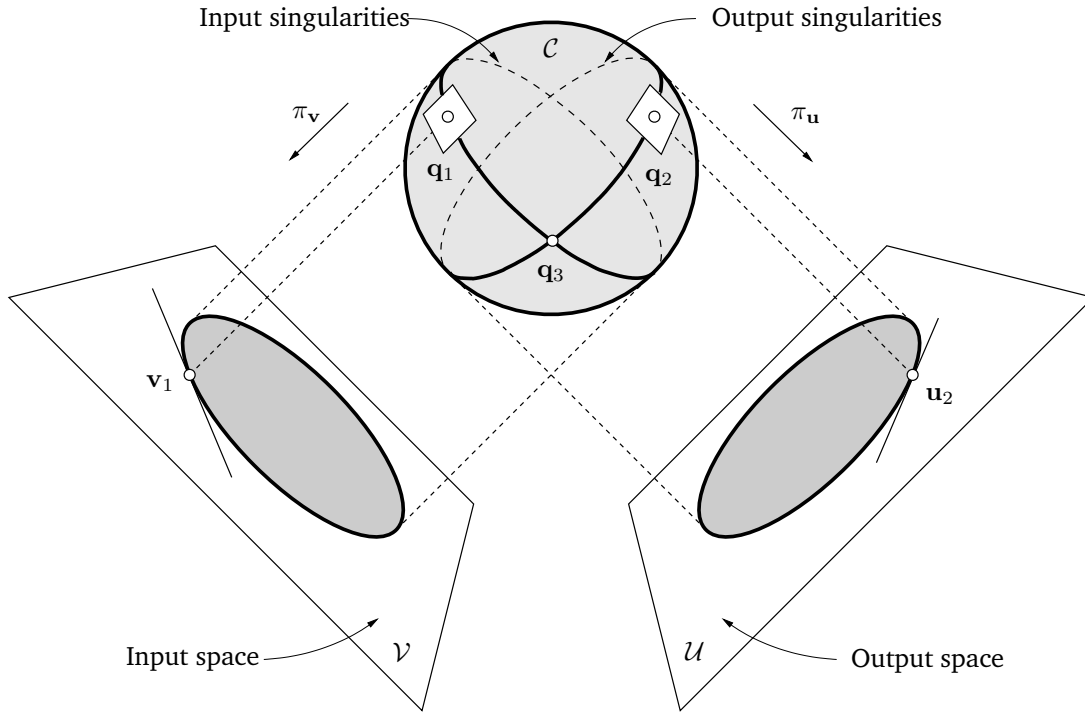


Figure 2.4: Interpretation of input and output singularities when  $\mathcal{Q} = \mathbb{R}^3$ ,  $\mathcal{C}$  is a sphere, and  $\mathcal{V}$  and  $\mathcal{U}$  are two coordinate planes of  $\mathbb{R}^3$ . In the figure,  $\mathbf{q}_1$  and  $\mathbf{q}_2$  correspond to an input and an output singularity, respectively, and  $\mathbf{q}_3$  is both an input and an output singularity. A feasible velocity vector at  $\mathbf{v}_1$  or  $\mathbf{u}_2$  does not locally determine a unique velocity vector in  $\mathcal{C}$ .

Consider, for instance, the output singularities. By the previous result, the critical points of the projection map  $\pi_{\mathbf{u}} : \mathcal{C} \setminus \mathcal{G} \rightarrow \mathcal{U}$  onto the  $\mathbf{u}$  coordinates, i.e.  $\pi_{\mathbf{u}}(\mathbf{z}, \mathbf{u}) = \mathbf{u}$ , with domain restricted to  $\mathcal{C} \setminus \mathcal{G}$ , are the points  $\mathbf{q}$  for which

$$(\Phi, \pi_{\mathbf{u}})_{\mathbf{q}} = \left[ \begin{array}{c|c} \Phi_{\mathbf{z}} & \Phi_{\mathbf{u}} \\ \hline \mathbf{0} & \mathbf{I}_{n_u} \end{array} \right]$$

is rank deficient, where  $\mathbf{I}_{n_u}$  denotes the  $n_u \times n_u$  identity matrix. Observe that  $(\Phi, \pi_{\mathbf{u}})_{\mathbf{q}}$  is rank deficient whenever  $\Phi_{\mathbf{z}}$  is rank deficient, which proves that the output singularities are the mentioned critical points. The same reasoning can be applied to conclude that the input singularities correspond to the critical points of the projection map  $\pi_{\mathbf{v}} : \mathcal{C} \setminus \mathcal{G} \rightarrow \mathcal{V}$  onto the  $\mathbf{v}$  variables. In both cases, the tangent space to  $\mathcal{C} \setminus \mathcal{G}$  has a special position, so that its projection onto the  $\mathcal{U}$  or  $\mathcal{V}$  spaces does not span the tangent space to  $\mathcal{U}$  or  $\mathcal{V}$  at the corresponding point.

This is easy to see when  $\mathcal{Q} = \mathbb{R}^{n_q}$ , in which case  $\mathcal{C}$  can be regarded as a subset of  $\mathbb{R}^{n_q}$  (Fig. 2.4). In such a situation, input singularities correspond to points  $\mathbf{q}$  where the tangent space

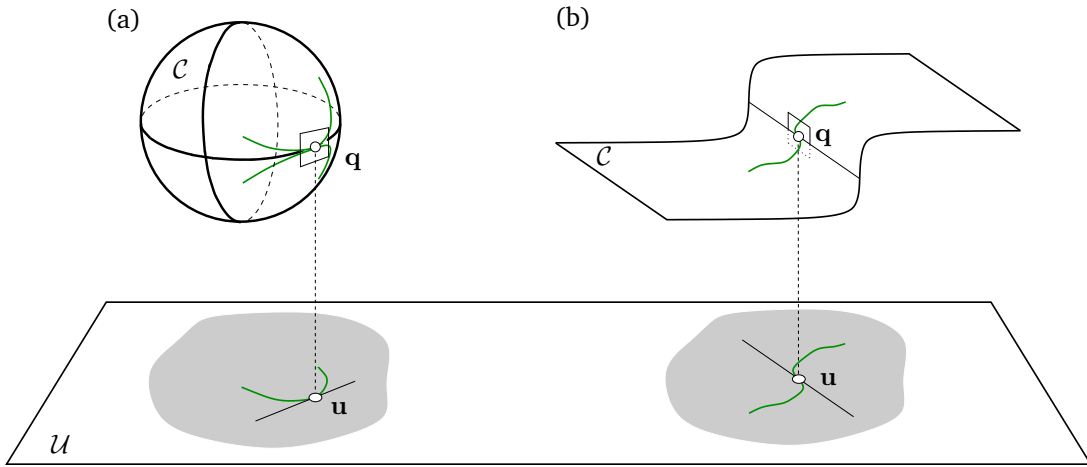


Figure 2.5: Two examples of output singularities seen as critical points of the projection onto the output space. A trajectory on the output space through a critical value does not correspond to a unique smooth trajectory on the C-space on the left. However, a smooth trajectory on the output space may correspond to a unique smooth trajectory on the C-space, as shown on the right.

to  $\mathcal{C}$  projects onto  $\mathcal{V} = \mathbb{R}^{n_v}$  as a linear space of dimension lower than  $n_v$ , and output singularities are the points where the tangent space to  $\mathcal{C}$  projects onto  $\mathcal{U} = \mathbb{R}^{n_u}$  as a subspace of dimension lower than  $n_u$ . Whereas input singularities yield controllability issues because a feasible vector  $\dot{\mathbf{v}}$  does not determine a unique vector  $\dot{\mathbf{q}}$ , output singularities correspond to mobility losses of the end-effector because, independently of the value of  $\dot{\mathbf{q}}$ ,  $\dot{\mathbf{u}}$  is always restricted to a linear subspace of smaller dimension. Note, in addition, that the input and output singularities delimit the set of possible values for the input and output coordinates,  $\mathbf{v}$  and  $\mathbf{u}$ , respectively, a fact that will be proven in Chapter 4 and will be pretty useful in the context of workspace determination.

The Implicit Function Theorem [47] provides further insight as to the advantages of avoiding each singularity type. Observe first that, as a consequence of the theorem, if  $\Phi_{\mathbf{z}}$  is full rank at  $\mathbf{q}_0 = [\mathbf{z}_0^T, \mathbf{u}_0^T]^T \in \mathcal{C}$ , it is possible to find a function  $\mathbf{z} = \mathbf{F}(\mathbf{u})$  relating  $\mathbf{z}$  with the remaining  $\mathbf{u}$  variables, satisfying  $\Phi(\mathbf{F}(\mathbf{u}), \mathbf{u}) = \mathbf{0}$ . Thus, the  $\mathbf{u}$  variables can be used as a local parameterization of  $\mathcal{C}$  around  $\mathbf{q}_0$ , implying that arbitrary values in a neighborhood of  $\mathbf{u}_0$  have a corresponding  $\mathbf{z}$  satisfying  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$ . This means that a smooth trajectory  $\mathbf{u}(t) \subset \mathcal{U}$  through a point  $\mathbf{u}_0$  will locally correspond to a unique smooth trajectory  $\mathbf{q}(t)$  on  $\mathcal{C}$  through the corresponding  $\mathbf{q}_0$ , or, in other words, a tracking of the output will be sufficient to predict the overall motion of the manipulator. In a similar way, whenever  $\Phi_{\mathbf{y}}$  is full rank at  $\mathbf{q}_0 = [\mathbf{y}_0^T, \mathbf{v}_0^T]^T$ , a smooth trajectory  $\mathbf{v}(t)$  through  $\mathbf{v}_0$  will locally determine a unique smooth trajectory  $\mathbf{q}(t)$  on  $\mathcal{C}$ , so that the overall movement of the manipulator will be controllable through the inputs.

This one-to-one correspondence between the input or output trajectories, on the one hand, and the manipulator trajectory, on the other hand, is not guaranteed at a singular configuration, as one can already see in the example of Fig. 2.5(a). However, since the Implicit Function Theorem only gives sufficient conditions to guarantee the uniqueness and smoothness of  $\mathbf{q}(t)$ , points with  $\Phi_y$  or  $\Phi_z$  singular can in principle exist where  $\mathbf{v}(t)$  or  $\mathbf{u}(t)$  determine a unique smooth trajectory  $\mathbf{q}(t)$ . An example of this situation is given in Fig. 2.5(b). Such points are not exempt of problems though, since  $\dot{\mathbf{v}}$  or  $\dot{\mathbf{u}}$  will still not determine  $\dot{\mathbf{q}}$  in them.

## 2.3 Zlatanov's Six Types

We have seen that the singularities are those configurations where either the FIKP or the IIKP of the manipulator become undetermined. However, depending on the kinematic cause of the degeneracy, six substantially different types of singularities were recognized by Zlatanov [12]. These are *redundant input* (RI), *redundant output* (RO), *impossible input* (II), *impossible output* (IO), *redundant passive motion* (RPM), and *increased instantaneous mobility* (IIM) singularities. Each of the six type definitions describes a different change in the kinematic properties of the manipulator.

When the mechanism is in a singularity of RO or IO (RI or II) type, the output (input) is indeterminate or restricted. In an RPM type singularity, the passive motion of the mechanism is indeterminate, which may create problems such as interference with other links and obstacles. And in an IIM type configuration the instantaneous motion of the manipulator is indeterminate, no matter which  $n$  parameters are being controlled. It is therefore desirable to know whether a given configuration belongs to one of the types, or to compute all possible configurations of a specific type. To this end, for each of the six singularity types we first recall its definition based on the velocity equation (2.2), and then derive a system of equations that characterizes the configurations of that type.

Recall that  $L_y$  and  $L_z$  are the submatrices of  $L$  obtained by removing the columns corresponding to the input and output, respectively, and let also  $L_P$  denote the submatrix obtained by removing both the columns corresponding to the input and output velocities. In the following, we shall use the 3-slider and the 4-bar mechanisms of Fig. 2.6 to illustrate the different singularity types encountered. Each of these mechanisms has one degree of freedom and, unless otherwise stated, the input and output velocities are those of points  $A$  and  $B$ ,  $v_A$  and  $v_B$ , for the 3-slider mechanism, and the angular velocities of links  $AB$  and  $DC$ ,  $\omega_A$  and  $\omega_D$ , for the 4-bar mechanism.



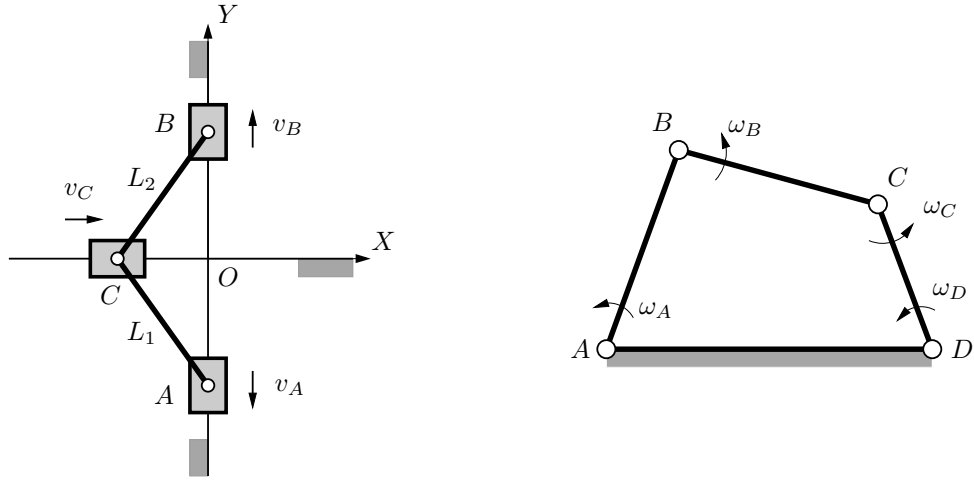


Figure 2.6: Left: A 1-DOF mechanism with three sliders. The prismatic joints at  $A$  and  $B$  are on a line perpendicular to the axis of the prismatic joint at  $C$ . Right: A 4-bar mechanism. The angular velocities indicated refer to relative motions, e.g.,  $\omega_B$  is the angular velocity of link  $BC$  relative to link  $AB$ .

### Redundant Input

A configuration is a singularity of RI type if there exist an input velocity vector  $\mathbf{m}_v \neq \mathbf{0}$  and a vector  $\mathbf{m}_p$  that satisfy the velocity equation (2.2) for  $\mathbf{m}_u = \mathbf{0}$ , i.e., such that

$$\mathbf{L}_z \begin{bmatrix} \mathbf{m}_v \\ \mathbf{m}_p \end{bmatrix} = \mathbf{0}$$

with  $\mathbf{m}_v \neq \mathbf{0}$ . Since such a vector exists whenever there exists a unit vector with  $\mathbf{m}_v \neq \mathbf{0}$ ,  $\mathbf{q}$  is a singularity of RI type if, and only if, the system of equations

$$\left. \begin{array}{l} \Phi(\mathbf{q}) = \mathbf{0} \\ \mathbf{L}_z \xi = \mathbf{0} \\ \|\xi\|^2 = 1 \end{array} \right\} \quad (2.9)$$

is satisfied for some value of  $\xi = [\mathbf{m}_v^\top, \mathbf{m}_p^\top]^\top$  with  $\mathbf{m}_v \neq \mathbf{0}$ .

Two examples of such singularities are provided in Table 2.1, first column. In the top configuration, the instantaneous input  $v_A$  can have any value, while the velocity of point  $C$  must be zero and, thus, point  $B$  cannot move. Similarly, in the bottom configuration the output link  $DC$  cannot move, since the velocity of point  $C$  must be zero, while the instantaneous input,  $\omega_A$ , can have any value.

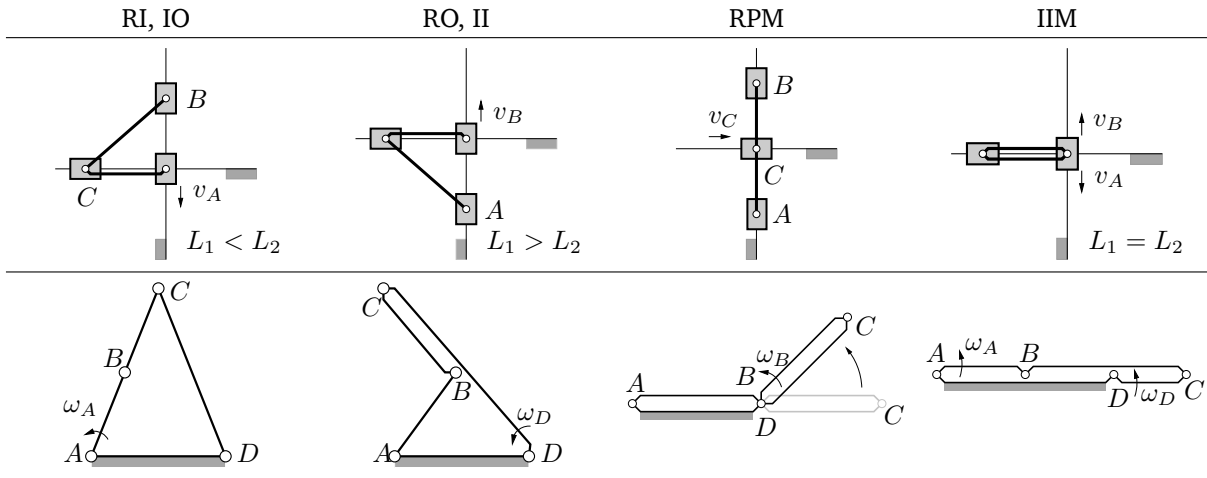


Table 2.1: The six singularity types exemplified with the 3-slider and 4-bar mechanisms.

### Redundant Output

A configuration is a singularity of RO type if there exist an output velocity vector  $\mathbf{m}_u \neq \mathbf{0}$  and a vector  $\mathbf{m}_p$  that satisfy the velocity equation for  $\mathbf{m}_v = \mathbf{0}$ , i.e., such that

$$L_y \begin{bmatrix} \mathbf{m}_u \\ \mathbf{m}_p \end{bmatrix} = \mathbf{0}$$

with  $\mathbf{m}_u \neq \mathbf{0}$ . Following a similar reasoning as above,  $\mathbf{q}$  is of RO type if, and only if, it satisfies the equations

$$\left. \begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ L_y \xi &= \mathbf{0} \\ \|\xi\|^2 &= 1 \end{aligned} \right\} \quad (2.10)$$

for some value of  $\xi = [\mathbf{m}_u^T, \mathbf{m}_p^T]^T$  with  $\mathbf{m}_u \neq \mathbf{0}$ .

The 3-slider and the 4-bar mechanisms in the second column of Table 2.1 are shown in a singularity of RO type. On the former, the instantaneous output  $v_B$  can have any value while point  $A$  must have zero velocity. The same happens on the latter, where the input link  $AB$  is locked while the instantaneous output,  $\omega_D$ , can have any value.

### Impossible Output

A configuration is a singularity of IO type if there exists a vector  $\mathbf{m}_u \neq \mathbf{0}$  in the output-velocity space for which the velocity equation cannot be satisfied for any combination of  $\mathbf{m}_v$  and  $\mathbf{m}_p$ . In

other words, this means that there is a nonzero vector  $[\mathbf{m}_u^\top, \mathbf{0}^\top, \mathbf{0}^\top]^\top$  that cannot be obtained by projection of any vector  $[\mathbf{m}_u^\top, \mathbf{m}_v^\top, \mathbf{m}_p^\top]^\top$  belonging to the kernel of  $L$ .

In order to derive the system of equations for this type, let  $B = [\mathbf{b}_1, \dots, \mathbf{b}_r]$  be a matrix whose columns form a basis of the kernel of  $L$ . Then, all vectors  $[\mathbf{m}_u^\top, \mathbf{0}^\top, \mathbf{0}^\top]^\top$  that can be obtained by projection of some vector of the kernel of  $L$  are those in the image space of the linear map given by the matrix

$$A = \begin{bmatrix} I_{n_u} & \mathbf{0} \end{bmatrix} B.$$

Thus, a singular configuration is of IO type if the map is not surjective, i.e., if  $A$  is rank deficient. In this situation it can be seen that there exists a nonzero vector  $\mathbf{m}_u^*$  in the kernel of  $A^\top$  and, hence, a vector  $[\mathbf{m}_u^{*\top}, \mathbf{0}^\top, \mathbf{0}^\top]^\top$  in the kernel of  $B^\top$ . Such a vector is orthogonal to all vectors  $\mathbf{b}_1, \dots, \mathbf{b}_r$ , so it must belong to the image of  $L^\top$ . In conclusion, there must exist a nonzero vector  $\mathbf{m}_u^*$  satisfying

$$L^\top \zeta = \begin{bmatrix} \mathbf{m}_u^{*\top} & \mathbf{0} & \mathbf{0} \end{bmatrix}^\top$$

for some vector  $\zeta$ , which can be chosen of unit norm. Therefore, a configuration  $\mathbf{q}$  is an IO type singularity if, and only if, it satisfies

$$L^\top \zeta = \left. \begin{array}{l} \Phi(\mathbf{q}) = \mathbf{0} \\ \left[ \mathbf{m}_u^{*\top} \quad \mathbf{0} \quad \mathbf{0} \right]^\top \\ \|\zeta\|^2 = 1 \end{array} \right\} \quad (2.11)$$

with  $\mathbf{m}_u^* \neq \mathbf{0}$ . For all solutions of this system, the obtained value of  $\mathbf{m}_u^*$  corresponds to a non-feasible output at the corresponding configuration.

The examples in the first column of Table 2.1 are also singularities of IO type since, in both configurations, any nonzero output is impossible.

### Impossible Input

A configuration is a singularity of II type if there exists an input velocity vector  $\mathbf{m}_v \neq \mathbf{0}$  for which the velocity equation cannot be satisfied for any combination of  $\mathbf{m}_u$  and  $\mathbf{m}_p$ . Following similar observations as for the IO type, a configuration  $\mathbf{q}$  is a singularity of II type if, and only if, there exists a nonzero vector  $\mathbf{m}_v^*$  such that

$$L^\top \zeta = \begin{bmatrix} \mathbf{0} & \mathbf{m}_v^{*\top} & \mathbf{0} \end{bmatrix}^\top$$

for some vector  $\zeta$ , which can also be chosen of unit norm. Thus, a configuration  $\mathbf{q}$  will be a singularity of II type if, and only if, it satisfies

$$\left. \begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ \mathbf{L}^T \zeta &= \begin{bmatrix} \mathbf{0} & \mathbf{m}_v^{*\top} & \mathbf{0} \end{bmatrix}^T \\ \|\zeta\|^2 &= 1 \end{aligned} \right\} \quad (2.12)$$

with  $\mathbf{m}_v^* \neq \mathbf{0}$ .

The 3-slider and the 4-bar mechanisms in the second column of Table 2.1 are also singularities of II type since any nonzero input is impossible in these configurations.

### Redundant Passive Motion

A configuration is a singularity of RPM type if there exists a vector  $\mathbf{m}_p$  that satisfies the velocity equation for  $\mathbf{m}_v = \mathbf{0}$  and  $\mathbf{m}_u = \mathbf{0}$ , i.e., such that

$$\mathbf{L}_P \mathbf{m}_p = \mathbf{0}$$

with  $\mathbf{m}_p \neq \mathbf{0}$ . This will happen when the kernel of  $\mathbf{L}_P$  is nonzero and, thus, the following system of equations

$$\left. \begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ \mathbf{L}_P \xi &= \mathbf{0} \\ \|\xi\|^2 &= 1 \end{aligned} \right\} \quad (2.13)$$

encodes all RPM type singularities  $\mathbf{q}$ .

Two examples of these singularities are provided in the third column of Table 2.1. On the 3-slider mechanism, both the input  $A$  and the output  $B$  must have zero velocity, while the velocity of point  $C$  can be nonzero. A 4-bar mechanism with a kite geometry, as shown in the table, can collapse so all joints lie on a single line and  $B$  and  $D$  coincide. If the input and output are the velocities at joints  $A$  and  $C$ ,  $\omega_A$  and  $\omega_C$ , the mechanism can move from the configuration shown in grey, maintaining zero-velocity both at the input and the output joints. Only nonzero velocity is present at the passive joints  $B$  and  $D$ . Hence, both mechanisms are shown in a singularity of RPM type.

### Increased Instantaneous Mobility

A configuration is a singularity of IIM type if  $\mathbf{L}$  is rank deficient. In fact, these are configurations where the instantaneous mobility is greater than the number of degrees of freedom. The

definition directly allows to write the system of equations

$$\left. \begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ \mathbf{L}^\top \xi &= \mathbf{0} \\ \|\xi\|^2 &= 1 \end{aligned} \right\}, \quad (2.14)$$

which will be satisfied for some  $\xi$  by a configuration  $\mathbf{q}$  if, and only if, it is a singularity of IIM type. These correspond to the C-space singularities, where both the FIKP and IIKP become indeterminate for any choice of input or output on the given velocity variables.

The mobility of the 3-slider and the 4-bar mechanisms in the fourth column of Table 2.1 increases from 1 to 2 at the shown configurations and, thus, they exhibit a singularity of IIM type.

## 2.4 The 3-slider Mechanism

To exemplify how the previous systems of equations can be used to obtain the configurations of each singularity type, consider the 3-slider mechanism in Fig. 2.6. Let  $(x_P, y_P)$  denote the coordinates of points  $P = \{A, B, C\}$  relative to the reference frame  $OXY$  in the figure, and let  $L_1$  and  $L_2$  be the lengths of the connector links. Clearly, a configuration of the mechanism can be shortly described by the tuple  $\mathbf{q} = \{y_A, y_B, x_C\}$  because  $x_A = x_B = y_C = 0$  in any configuration. Since the distances from  $A$  to  $B$  and from  $B$  to  $C$  must be kept equal to  $L_1$  and  $L_2$ , Eq. (2.1) is

$$\left. \begin{aligned} y_A^2 + x_C^2 &= L_1^2 \\ y_B^2 + x_C^2 &= L_2^2 \end{aligned} \right\}, \quad (2.15)$$

from which we realize that the C-space corresponds to the intersection of two cylinders in the space of  $y_A$ ,  $y_B$ , and  $x_C$ .

The velocity equation in (2.2) could now be obtained using the revolute- and prismatic-joint screws [12], but a more compact expression can here be derived by properly differentiating the equations in (2.15). Taking  $v_A$  and  $v_B$  as the input and output velocities, the differentiation yields

$$\mathbf{L} \cdot \mathbf{m} = \begin{bmatrix} 0 & 2y_A & 2x_C \\ 2y_B & 0 & 2x_C \end{bmatrix} \cdot \begin{bmatrix} v_B \\ v_A \\ v_C \end{bmatrix} = \mathbf{0},$$

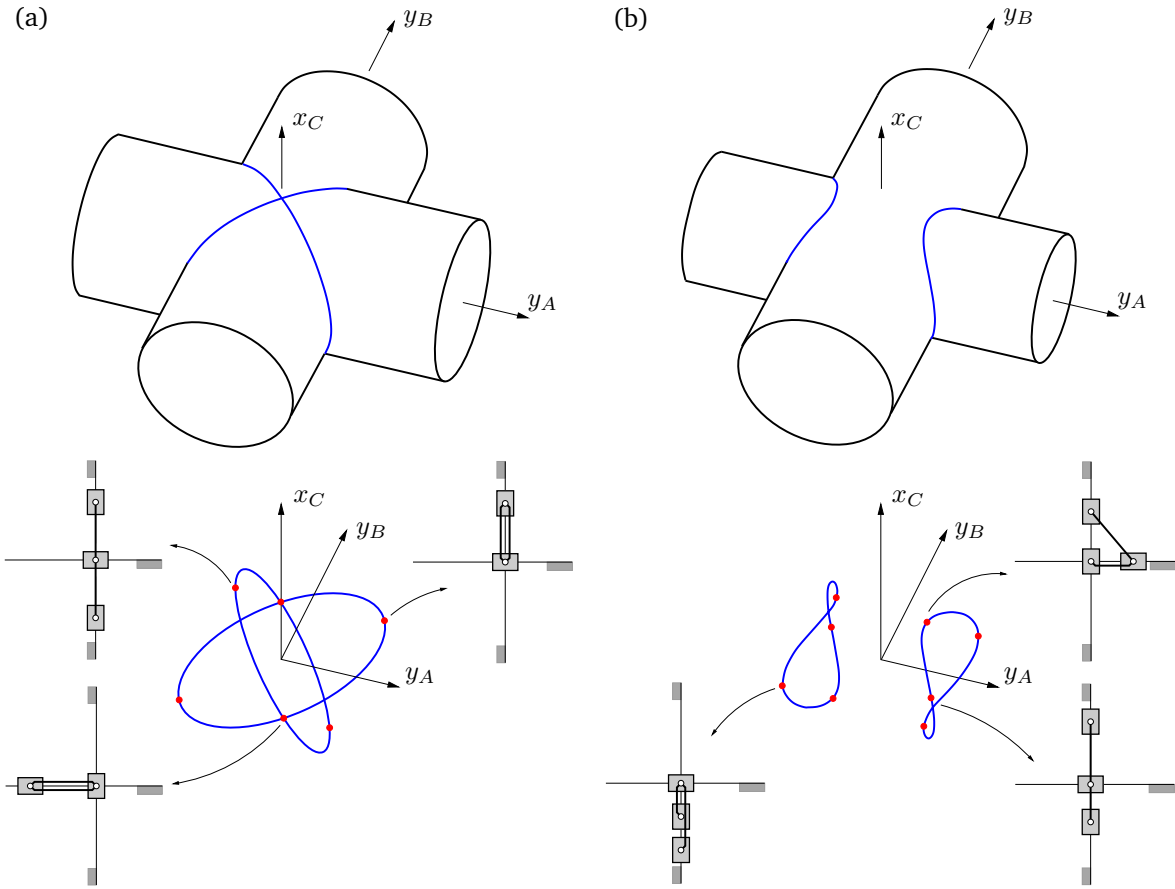


Figure 2.7: Configuration space (in blue) and singularities (red dots) of the 3-slider mechanism for  $L_1 = L_2$  (a) and  $L_1 > L_2$  (b) with some examples of singular configurations depicted. In this mechanism, the configuration space corresponds to the intersection of two cylinders at right angles.

from where we readily obtain

$$\mathbf{L}_y = \begin{bmatrix} 0 & 2x_C \\ 2y_B & 2x_C \end{bmatrix},$$

$$\mathbf{L}_z = \begin{bmatrix} 2y_A & 2x_C \\ 0 & 2x_C \end{bmatrix},$$

$$\mathbf{L}_P = \begin{bmatrix} 2x_C \\ 2x_C \end{bmatrix},$$

which allow defining any of the systems in Eqs. (2.7)–(2.14).

These systems can be solved analytically in this case. For example, if  $L_1 = L_2 = 1$ , the C-space has a single connected component composed of two ellipses intersecting on the  $x_C$  axis [Fig. 2.7(a)], and the solutions of the systems in Eqs. (2.7) and (2.8) reveal that the singularity

set has six isolated configurations, marked in red in Fig. 2.7(a) bottom, with the following values of  $\mathbf{q}$ :

$$\begin{array}{ll} (0, 0, 1), & (0, 0, -1), \\ (1, 1, 0), & (1, -1, 0), \\ (-1, -1, 0), & (-1, 1, 0). \end{array}$$

All of these configurations satisfy both systems, so they are all forward and inverse singularities because both the FIKP and the IIKP are indeterminate on them and, thus, the control of the input or the output velocity does not determine the overall motion of the mechanism. It turns out, moreover, that the four configurations with  $x_C = 0$  satisfy Eqs. (2.11), (2.12) and (2.13), meaning that they are singularities of IO, II, and RPM type. The other two configurations, which lie on the  $x_C$  axis, are singularities of RI, RO, and IIM type because they satisfy Eqs. (2.9), (2.10) and (2.14). These two configurations are in fact C-space singularities. In this case, the C-space self-intersects at these points, and presents a bifurcation that allows to change the mode of operation from both sliders moving on the same side of the horizontal axis,  $y_A y_B \geq 0$ , to one slider moving on each side,  $y_A y_B \leq 0$ .

The topology of the C-space changes when  $L_1 \neq L_2$ , since the C-space no longer presents any bifurcation, and it is instead formed by two connected components [Fig. 2.7(b)]. By solving Eqs. (2.7) and (2.8) for  $L_1 = 1$  and  $L_2 = 0.8$ , for example, eight singularities are obtained:

$$\begin{array}{ll} (1, 0.8, 0), & (-1, -0.8, 0), \\ (-1, 0.8, 0), & (0.6, 0, -0.8), \\ (1, -0.8, 0), & (-0.6, 0, 0.8), \\ (0.6, 0, 0.8), & (-0.6, 0, -0.8). \end{array}$$

As before, the configurations with  $x_C = 0$  are singularities of IO, II, and RPM type, but the other four configurations are of RO and II type, and there are no singularities of RI or IIM type. To change the operation mode from  $y_A \geq 0$  to  $y_A \leq 0$  the mechanism has to be disassembled in this case.

It must be noted that if a singularity identification were attempted by means of an input-output velocity equation, for instance by using the equation  $y_A v_A = y_B v_B$  which holds for all configurations, the singularities with  $x_C = 0$  would not be detected.





# 3

## Numerical Computation of Singularity Sets

This Chapter provides a method to compute any of the singularity sets defined in Chapter 2. The method is very general and applicable to virtually any relevant mechanism with lower-pair joints. It allows the complete singularity set to be obtained with the desired accuracy, and each of its singularity types to be computed independently. To this end, Section 3.2 provides a general way of formulating Eqs. (2.7)–(2.14) in an amenable form, and Section 3.3 describes the main steps allowing to compute the solution set of any of the systems. Because of the high number of configuration variables typically involved in the systems, the singularity set is often defined in a highly-dimensional space, and Section 3.4 provides hints on how to represent this set in a meaningful way. Finally, Section 3.5 demonstrates the method’s performance, and the complexities of the singularity set, with the analysis of several manipulators.

### 3.1 Solution Approach

A complete solution of non-linear systems like those in Eqs. (2.7)–(2.14) typically involves formulating the systems algebraically, to then take advantage of some suitable symbolic or numerical technique for polynomial system solving. Unfortunately, a good solution to both algebraization and resolution, treated as independent problems, does not necessarily lead to an efficient solution of the original problem. Our aim has been on finding a good combination of algebraization and resolution, so that the whole process becomes easy to understand and implement, yet computationally efficient in practice.

Common root-finding approaches for polynomial equations can be classified into algebraic-geometric, continuation, or branch-and-prune techniques. Within the first group of techniques, the most usual ones apply elimination methods to reduce the system to a minimal set of resultant polynomials, so that the roots of these polynomials, once backsubstituted into intermediate equations, yield all solutions to the original problem [48, 49]. Continuation methods, in con-

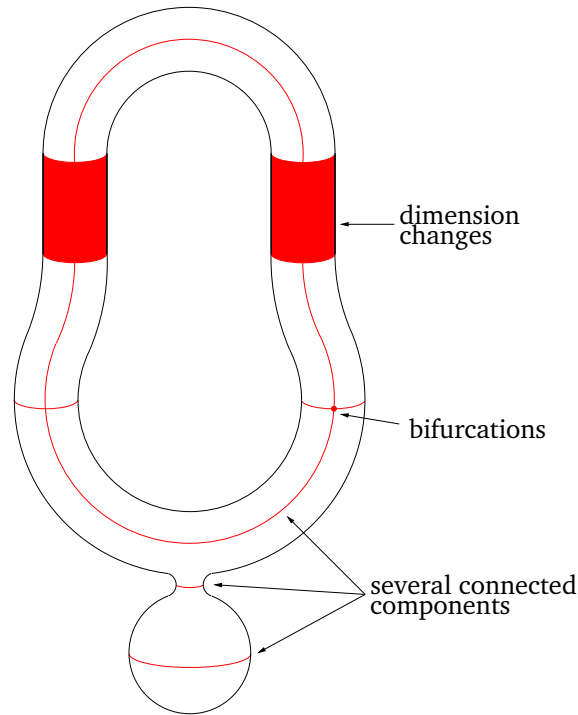


Figure 3.1: The singularity set is an algebraic variety that may contain bifurcations, dimension changes, or several connected components, even if the  $\mathcal{C}$ -space is smooth everywhere.

strast, begin with an initial system whose solutions are known, and then transform it gradually to the system whose solutions are sought, tracking all solution paths along the way [50–52]. Instead, branch-and-prune methods proceed by exclusion. They start with an initial bounding box encompassing the solutions, and then use relaxed versions of the equations to iteratively eliminate portions of the box that cannot contain a solution. The process is repeated until a fine-enough box approximation of the solution set is obtained [31, 53, 54].

To determine the proper kind of approach in our case, we must bear in mind that the structure of the singularity set  $\mathcal{S}$  can be very general. Generically,  $\mathcal{S}$  has codimension 1 with respect to the  $\mathcal{C}$ -space, so we may have to deal with singular sets of dimension 1, 2, or even higher. Elimination and continuation methods are quite general, but they are typically designed to solve systems with isolated or one-dimensional solutions at most. While it is true that continuation methods have been extended to characterize positive-dimensional solutions [55, Part III], they do not directly provide an explicit representation of all solution points in general. Only higher-dimensional continuation techniques would in principle be appropriate [35], but they should be fed with a solution point for each connected component of  $\mathcal{S}$ , and they are still not robust enough to the many non-smoothnesses that  $\mathcal{S}$  may eventually present. Note that, even if  $\mathcal{C}$

is smooth and contains only one connected component,  $\mathcal{S}$  may include multiple connected components, bifurcations, or even dimension changes, among other phenomena (Fig. 3.1). In contrast, branch-and-prune methods can isolate solutions of any dimension with several connected components, and since they are very robust to non-smoothnesses, they appear ideal for our purposes.

With regard to the formulation, note that the structure of all systems in Eqs. (2.7)–(2.14) is very similar. The first line is always Eq. (2.1), because all solution points must correspond to feasible configurations of the manipulator. The second line always involves  $\mathbf{L}$  or one of its sub-matrices, and the third line constrains the norm of some vector. Therefore, the main issue is how to formulate the assembly constraints and the velocity equation in a general way. We next show that, by using a particular choice of configuration coordinates, the formulation of these equations leads to a conceptually simple branch-and-prune technique for solving any of the Eqs. (2.7)–(2.14). The formulation closely follows that of reference point coordinates in multibody dynamics [31, 45], which produces simple polynomials with little manipulation, in comparison to other formulations departing from loop closure constraints on relative joint displacements [45, Sec. 2.2.1], or from point-to-point distances [56].

## 3.2 Formulating the Equations of the Singularity Sets

### 3.2.1 Obtaining the Assembly Constraints

To formulate Eq. (2.1), let us assume that our manipulator has  $n_b$  links and  $n_j$  joints, labelled  $L_1, \dots, L_{n_b}$ , and  $J_1, \dots, J_{n_j}$ , respectively, where  $L_1$  is supposed to be the ground link. We furnish every link  $L_i$  with a local reference frame,  $\mathcal{F}_i$ , letting  $\mathcal{F}_1$  act as the absolute frame. We will write  $\mathbf{a}^{\mathcal{F}_i}$  to indicate that the components of a vector  $\mathbf{a} \in \mathbb{R}^3$  are provided in the basis of  $\mathcal{F}_i$ , and we will assume that vectors with no superscript are expressed in the basis of  $\mathcal{F}_1$ , unless otherwise stated. Then, the pose of each link in the manipulator can be specified by the pair  $(\mathbf{r}_i, \mathbf{R}_i)$ , where  $\mathbf{r}_i \in \mathbb{R}^3$  is the position of the origin of  $\mathcal{F}_i$  in frame  $\mathcal{F}_1$ , and  $\mathbf{R}_i \in SO(3)$  is a  $3 \times 3$  rotation matrix expressing the orientation of  $\mathcal{F}_i$  relative to  $\mathcal{F}_1$ . Note that the link poses cannot be arbitrary though, since they must fulfill the assembly constraints imposed by the joints. We next provide such constraints for the revolute, prismatic, universal, and spherical joints, since these are those typically arising in most manipulators. Other lower pairs can be accommodated in a similar way however.

If  $J_i$  is a revolute joint connecting links  $L_j$  and  $L_k$ , we consider a point,  $P_i$ , on the axis of the joint, and a unit vector  $\mathbf{d}_i$  along the same axis. Then, the assembly constraints of this joint can

be formulated as follows:

$$\mathbf{r}_j + \mathbf{R}_j \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \mathbf{p}_i^{\mathcal{F}_k}, \quad (3.1)$$

$$\mathbf{R}_j \mathbf{d}_i^{\mathcal{F}_j} = \mathbf{R}_k \mathbf{d}_i^{\mathcal{F}_k}, \quad (3.2)$$

where  $\mathbf{p}_i$  is the position vector of  $P_i$ . The first equation forces the links to be placed so that the point  $P_i$ , seen as attached to  $\mathcal{F}_j$  coincides with the same point, seen as attached to  $\mathcal{F}_k$ , and the second equation does a similar identification for the  $\mathbf{d}_i$  vector [Fig. 3.2(a)].

If  $J_i$  is prismatic, we consider two different points on the axis of the joint,  $P_i$  and  $Q_i$ , and a unit vector  $\mathbf{d}_i$  aligned with the joint. In this case, the assembly constraints are equivalent to forcing  $Q_i$  to lie on the line of  $L_j$  defined by  $P_i$  and  $\mathbf{d}_i$ , while keeping the relative orientation between  $L_j$  and  $L_k$  fixed to a constant offset [Fig. 3.2(b)]. These conditions are equivalent to

$$\mathbf{r}_j + \mathbf{R}_j \mathbf{p}_i^{\mathcal{F}_j} + d_i \mathbf{R}_j \mathbf{d}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \mathbf{q}_i^{\mathcal{F}_k}, \quad (3.3)$$

$$\mathbf{R}_j = \mathbf{R}_{jk} \cdot \mathbf{R}_k, \quad (3.4)$$

where  $d_i$  is the linear displacement of the joint and  $\mathbf{R}_{jk}$  is a constant rotation matrix providing the relative orientation between links  $L_j$  and  $L_k$ .

If  $J_i$  is universal, the points  $P_i$  at the center of the joint must coincide, and the unit vectors  $\mathbf{a}_i$  and  $\mathbf{b}_i$  defining the rotation axes must be orthonormal [Fig. 3.2(c)], which translates into

$$\mathbf{r}_j + \mathbf{R}_j \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \mathbf{p}_i^{\mathcal{F}_k}, \quad (3.5)$$

$$\mathbf{R}_j \mathbf{a}_i^{\mathcal{F}_j} \cdot \mathbf{R}_k \mathbf{b}_i^{\mathcal{F}_k} = 0. \quad (3.6)$$

If  $J_i$  is spherical, the center points  $P_i$  on each link must coincide, while  $L_k$  can freely rotate with respect to  $L_j$  [Fig. 3.2(d)]. Thus, the valid poses for the two links verify

$$\mathbf{r}_j + \mathbf{R}_j \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \mathbf{p}_i^{\mathcal{F}_k}. \quad (3.7)$$

Since the coordinates of  $\mathbf{p}_i$ ,  $\mathbf{q}_i$ , and  $\mathbf{d}_i$  in the corresponding local frames are a priori known, the only unknowns appearing in the previous equations are the poses of the links and the displacements of the prismatic joints. The entries of the rotation matrices, though unknown,

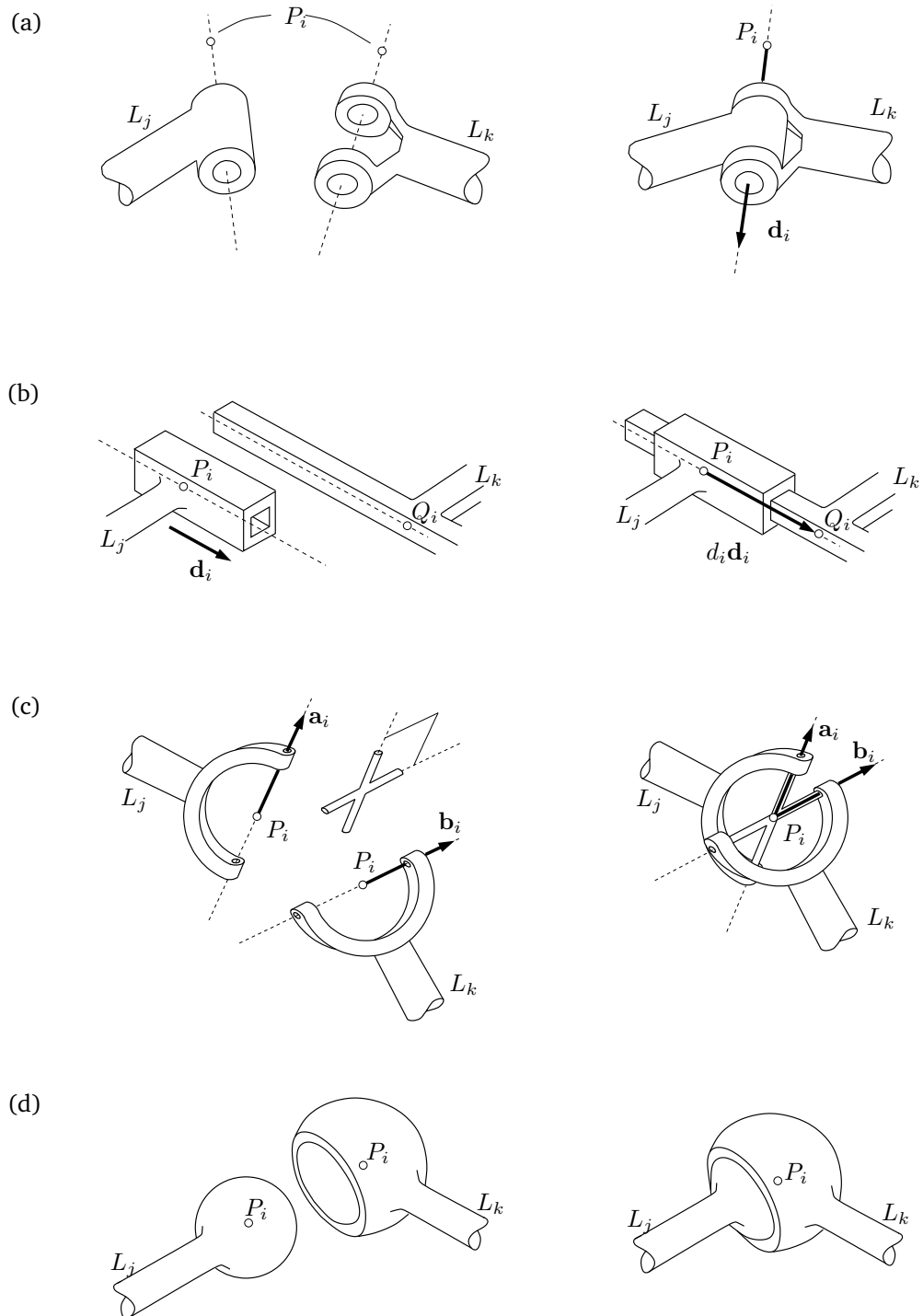


Figure 3.2: Geometric elements intervening in the assembly of (a) revolute, (b) prismatic, (c) universal, and (d) spherical joints.

are not independent, since if  $\mathbf{R}_i = [\mathbf{s}_i, \mathbf{t}_i, \mathbf{w}_i]$ , then it must be

$$\|\mathbf{s}_i\|^2 = 1, \quad (3.8)$$

$$\|\mathbf{t}_i\|^2 = 1, \quad (3.9)$$

$$\mathbf{s}_i \cdot \mathbf{t}_i = 0, \quad (3.10)$$

$$\mathbf{s}_i \times \mathbf{t}_i = \mathbf{w}_i, \quad (3.11)$$

in order for  $\mathbf{R}_i$  to represent a valid rotation.

In our case, thus, Eq. (2.1) is the system formed by Eqs. (3.1)–(3.11) established for all joints and links of the manipulator, and  $\mathbf{q}$  is the vector encompassing the variables in  $\mathbf{r}_i$  and  $\mathbf{R}_i$  of all links, and  $d_i$  for all prismatic joints. Note however that, since  $L_1$  is the ground link,  $\mathbf{r}_1 = \mathbf{0}$  and  $\mathbf{R}_1$  is the identity matrix.

### 3.2.2 Obtaining the Velocity Equation

Sometimes, as in planar manipulators [57], the input and output coordinates directly appear in the formulation of the assembly constraints, or they can be easily introduced. In these cases, Eq. (2.2) can be obtained by differentiation, as described in Section 2.2.1, but in general it is usually easier to formulate Eq. (2.2) using the tools of Screw Theory [12, 58]. In favor of generality, thus, we shall here show how to formulate Eq. (2.2) using the twist loop equations.

It is known that, for a joint  $J_i$  between two links, the velocity of one link with respect to the other is given by a linear combination of six-dimensional unit twists [12, 58, 59]. Table 3.1 provides the form of these twists and the mentioned linear combination for each of the joints herein considered. As shown, revolute and prismatic joints only involve one unit twist, which is multiplied by the joint velocity  $\omega_i$  in the linear combination. Instead, since universal and spherical joints are equivalent to a combination of two and three revolute joints, they involve linear combinations of two and three twists respectively, multiplied by their corresponding joint velocities.

Any set of feasible joint velocities must be in accordance with the kinematic constraints of the mechanism. For that, the sum of the twists along any closed loop should be zero. If the joints in a loop are numbered from 1 to  $l$ , this condition can be written as

$$\omega_1 \hat{\mathbf{S}}_1 + \omega_2 \hat{\mathbf{S}}_2 + \dots + \omega_l \hat{\mathbf{S}}_l = \sum_{i=1}^l \omega_i \hat{\mathbf{S}}_i = \mathbf{0}, \quad (3.12)$$

where  $\omega_i$  is the joint velocity related to the  $i$ -th unit twist. This condition must be satisfied for all loops of the manipulator, but only the equations relative to a maximal set of independent

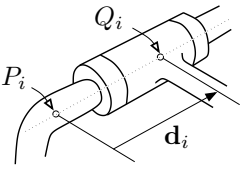
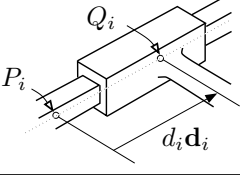
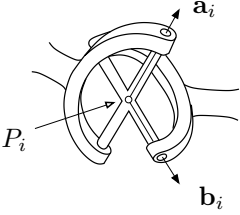
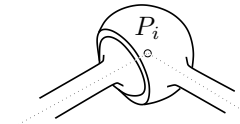
| Pair      | Shape   | Unit twists   | Linear combination  |
|-----------|---|---|---|
| Revolute  |    | $\hat{\mathbf{S}}_i = \begin{bmatrix} \mathbf{p}_i \times \mathbf{d}_i \\ \mathbf{d}_i \end{bmatrix}$   | $\omega_i \hat{\mathbf{S}}_i$   |
| Prismatic |    | $\hat{\mathbf{S}}_i = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{0} \end{bmatrix}$   | $\omega_i \hat{\mathbf{S}}_i$   |
| Universal |    | $\hat{\mathbf{S}}_{i,a} = \begin{bmatrix} \mathbf{p}_i \times \mathbf{a}_i \\ \mathbf{a}_i \end{bmatrix}$<br>$\hat{\mathbf{S}}_{i,b} = \begin{bmatrix} \mathbf{p}_i \times \mathbf{b}_i \\ \mathbf{b}_i \end{bmatrix}$  | $\omega_{i,a} \hat{\mathbf{S}}_{i,a} + \omega_{i,b} \hat{\mathbf{S}}_{i,b}$                                       |
| Spherical |  | $\hat{\mathbf{S}}_{i,x} = \begin{bmatrix} \mathbf{p}_i \times \mathbf{e}_x \\ \mathbf{e}_x \end{bmatrix}$<br>$\hat{\mathbf{S}}_{i,y} = \begin{bmatrix} \mathbf{p}_i \times \mathbf{e}_y \\ \mathbf{e}_y \end{bmatrix}$<br>$\hat{\mathbf{S}}_{i,z} = \begin{bmatrix} \mathbf{p}_i \times \mathbf{e}_z \\ \mathbf{e}_z \end{bmatrix}$ | $\omega_{i,x} \hat{\mathbf{S}}_{i,x} + \omega_{i,y} \hat{\mathbf{S}}_{i,y} + \omega_{i,z} \hat{\mathbf{S}}_{i,z}$ |

Table 3.1: Unit twists and their linear combination for the considered joints. Vectors  $\mathbf{d}_i$ ,  $\mathbf{a}_i$ , and  $\mathbf{b}_i$  are those defined in Section 3.2.1 for each joint. Vector  $\mathbf{p}_i$  provides the position of point  $P_i$  of the joint, relative to  $\mathcal{F}_1$ . Without loss of generality, we can suppose that the spherical joint is instantaneously equivalent to a chain of three revolute joints with axes coincident in  $P_i$ , aligned with the vectors  $\mathbf{e}_x = [1, 0, 0]^T$ ,  $\mathbf{e}_y = [0, 1, 0]^T$ , and  $\mathbf{e}_z = [0, 0, 1]^T$ .

loops are necessary, since the remaining ones are linearly dependent of these. If the manipulator contains  $c$  independent loops, we will have  $6c$  equations that impose a necessary and sufficient condition for the feasibility of the joint velocities, including the actuated ones,  $\mathbf{m}_v$ , and the passive ones,  $\mathbf{m}_p$ .

The output velocities,  $\mathbf{m}_u$ , can be written as a linear function of the joint velocities if each of the output velocities is a component of the twist  $\hat{T}$  of the end-effector with respect to the ground link  $L_1$ . If we choose any path of links and joints connecting  $L_1$  to the end-effector, and number

the joints in this path from 1 to  $k$ , we can write

$$\hat{\mathbf{T}} = \omega_1 \hat{\mathbf{S}}_1 + \dots + \omega_k \hat{\mathbf{S}}_k = \sum_{i=1}^k \omega_i \hat{\mathbf{S}}_i. \quad (3.13)$$

Therefore, the velocity equation is the system formed by Eq. (3.12) for a set of independent loops, and Eq. (3.13) relative to a path connecting the ground link to the end-effector. Identifying terms with Eq. (2.2), it can be seen that  $\mathbf{m}$  is the vector containing the joint velocities  $\omega_i$  and the components of the twist  $\hat{\mathbf{T}}$ , and that  $\mathbf{L}$  is the corresponding coefficients matrix. Since the manipulator is non-redundant, if it has  $N$  1-DOF joints and mobility  $n$ , then it must have  $n$  actuated joints,  $N - n$  passive joints, and  $n$  output variables, meaning that the size of  $\mathbf{m}$  is  $N + n$  and, thus,  $\mathbf{L}$  is an  $N \times (N + n)$  matrix.

The components of  $\mathbf{L}$  are given by the unit twists, and since they depend on the configuration  $\mathbf{q}$ , it will be necessary to add the expressions in the third column of Table 3.1 relating the components of the twists with those of  $\mathbf{q}$ . Even though these expressions do not include  $\mathbf{q}$  explicitly, the vectors  $\mathbf{d}_i$ ,  $\mathbf{a}_i$ ,  $\mathbf{b}_i$ , and  $\mathbf{p}_i$  can be obtained from the components of  $\mathbf{q}$  by introducing the following linear relations:

$$\mathbf{d}_i = \mathbf{R}_j \mathbf{d}_i^{\mathcal{F}_j}, \quad (3.14)$$

$$\mathbf{a}_i = \mathbf{R}_j \mathbf{a}_i^{\mathcal{F}_j}, \quad (3.15)$$

$$\mathbf{b}_i = \mathbf{R}_j \mathbf{b}_i^{\mathcal{F}_j}, \quad (3.16)$$

$$\mathbf{p}_i = \mathbf{r}_j + \mathbf{R}_j \mathbf{p}_i^{\mathcal{F}_j}. \quad (3.17)$$

Note that the previous developments already provide a general methodology to formulate Eqs. (2.7)–(2.14), which allows to address their solution in order to find the different singularity sets they characterize. Each of the systems involve some of the following variables:

- Variables in  $\mathbf{q}$  from the formulation of the assembly constraints:  $\mathbf{r}_i$ ,  $\mathbf{R}_i$ , and  $d_i$ .
- The components of matrix  $\mathbf{L}$ , which are components of the unit twists  $\hat{\mathbf{S}}_i$ :  $\hat{s}_{i,j}$ .
- Vectors needed to express the components of  $\mathbf{L}$ :  $\mathbf{d}_i$ ,  $\mathbf{a}_i$ ,  $\mathbf{b}_i$ , and  $\mathbf{p}_i$ .
- Vectors  $\xi$  or  $\zeta$  needed to establish the singularity condition.
- Vectors  $\mathbf{m}_u^*$  or  $\mathbf{m}_v^*$  in the case of IO or II type singularities.

It must be added, however, that the proposed formulation can be usually simplified by eliminating some of the variables and equations involved. For instance, many of the variables in  $\mathbf{r}_i$  can be eliminated if closed kinematic chains are present in the manipulator, through a process



explained in detail in [31]. Moreover, in hybrid-chain manipulators it is also possible to use a method to eliminate passive joint velocities, thus reducing the size of the velocity equation [12], and allowing more compact systems of equations. Both the general formulation and these simplifications will be illustrated in Section 3.5 on specific examples.

### 3.3 Isolating the Singularity Sets

To isolate the solution sets of Eqs. (2.7)–(2.14) with the numerical technique described in this Section, two requirements must be fulfilled: the equations must adopt the form of a polynomial system of quadratic equations, and an initial box bounding all solutions of the system must be defined. Then, the numerical procedure exploits the special form of the equations to iteratively remove portions of the box that contain no solution, obtaining a box approximation of the solution set at the desired accuracy.

#### 3.3.1 Reduction to a Simple Quadratic Form

From the previous formulation, observe that all equations in the systems are polynomial, involving only linear, bilinear, or quadratic terms. In other words, if  $x_i$  and  $x_j$  refer to any two variables, the involved monomials are only of the form  $x_i$ ,  $x_i x_j$ , or  $x_i^2$ .

Let us define the changes of variables

$$p_k = x_i^2, \quad (3.18)$$

$$b_k = x_i x_j, \quad (3.19)$$

for each quadratic and bilinear monomial, respectively. This allows to convert any of the systems into the expanded form:

$$\left. \begin{aligned} \Lambda(\mathbf{x}) &= 0 \\ \Omega(\mathbf{x}) &= 0 \end{aligned} \right\}, \quad (3.20)$$

where  $\mathbf{x}$  is a vector encompassing the original and the newly-defined variables,  $\Lambda(\mathbf{x}) = 0$  is a subsystem of linear equations in  $\mathbf{x}$ , and  $\Omega(\mathbf{x}) = 0$  is a subsystem of quadratic equations of the form of (3.18) or (3.19). Equation (3.20) involves more equations and variables than the original system, but the simpler structure of its equations is beneficial to the branch-and-prune strategy defined in Section 3.3.3.

In the systems of Eqs. (2.9)–(2.12), there is a vector that must be different from zero, but since the technique can also handle non-strict inequalities, as explained below, the latter condition can be enforced by setting  $\|\mathbf{m}_v\|^2 \geq \epsilon$ , for systems (2.9) and (2.12), and  $\|\mathbf{m}_u\|^2 \geq \epsilon$ ,

for systems (2.10) and (2.11), where  $\epsilon$  is a sufficiently small value. By using these inequalities, whose terms are also quadratic, some solutions might be neglected, but  $\epsilon$  can be made arbitrarily small, reducing the set of missed solutions to a negligible size.

### 3.3.2 Initial Bounding Box

An advantage of the proposed formulation is that it is straightforward to define conservative interval bounds for all solutions of Eq. (3.20), since:

- Simple feasibility intervals for the components of  $\mathbf{r}_i$  can be derived from the link dimensions.
- The allowed displacement  $d_i$  of the prismatic joints is usually restricted by design or can be derived from the link dimensions.
- The columns of  $\mathbf{R}_i$  are restricted to be unit vectors by Eqs. (3.8)–(3.11), so their components can only take values in the  $[-1, 1]$  interval.
- The components of the unit twists,  $\hat{s}_{i,j}$ , can be derived from the intervals of other variables using the expressions of Table 3.1.
- The variables that refer to the components of the unit vectors  $\mathbf{d}_i$ ,  $\mathbf{a}_i$ , and  $\mathbf{b}_i$  can only take values in the  $[-1, 1]$  interval.
- The interval for the components of  $\mathbf{p}_i$  can be derived using Eq. (3.17).
- The entries of  $\xi$  and  $\zeta$  are bounded in the range  $[-1, 1]$  because they are unit vectors.
- The feasibility intervals for the entries of  $\mathbf{m}_u^*$  and  $\mathbf{m}_v^*$  in Eqs. (2.11) and (2.12) can be obtained by mapping the known intervals using  $\mathbf{L}_u^T \zeta = \mathbf{m}_u^*$  and  $\mathbf{L}_v^T \xi = \mathbf{m}_v^*$ .
- Intervals for the  $p_k$  and  $b_k$  variables can be derived by simple interval operations using Eqs. (3.18) and (3.19).

In conclusion, from the Cartesian product of all such intervals it is possible to define an initial rectangular box  $\mathcal{B}$  bounding all solutions of Eq. (3.20).

### 3.3.3 A Branch-and-prune Method

The algorithm for solving Eq. (3.20) recursively applies two operations on  $\mathcal{B}$ : box *shrinking* and box *splitting*. Using box shrinking, portions of  $\mathcal{B}$  containing no solution are eliminated by narrowing some of its defining intervals. This process is repeated until either the box is reduced

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

The crucial operation in this scheme is box shrinking, which is implemented as follows. Note first that the solutions falling in some sub-box  $\mathcal{B}_c \subseteq \mathcal{B}$  must lie in the linear variety defined by  $\Lambda(\mathbf{x}) = 0$ . Thus, we may shrink  $\mathcal{B}_c$  to the smallest possible box bounding this variety inside  $\mathcal{B}_c$ . The limits of the shrunk box along each dimension  $x_i$  can be found by solving the following two linear programs:

**LP1:** Minimize  $x_i$ , subject to:  $\Lambda(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$ ,

**LP2:** Maximize  $x_i$ , subject to:  $\Lambda(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$ .

However, observe that  $\mathcal{B}_c$  can be further reduced, because the solutions must also satisfy all equations  $p_k = x_i^2$  and  $b_k = x_i x_j$  in  $\Omega(\mathbf{x}) = 0$ . These equations can be taken into account by noting that, if  $[g_i, h_i]$  denotes the interval of  $\mathcal{B}_c$  along dimension  $x_i$ , then:

- The portion of the parabola  $p_k = x_i^2$  lying inside  $\mathcal{B}_c$  is bound by the triangle  $A_1 A_2 A_3$ , where  $A_1$  and  $A_2$  are the points where the parabola intercepts the lines  $x_i = g_i$  and  $x_i = h_i$ , and  $A_3$  is the point where the tangent lines at  $A_1$  and  $A_2$  meet [Fig. 3.3(a)].
- The portion of the hyperbolic paraboloid  $b_k = x_i x_j$  lying inside  $\mathcal{B}_c$  is bound by the tetrahedron  $B_1 B_2 B_3 B_4$ , where the points  $B_1, \dots, B_4$  are obtained by lifting the corners of the rectangle  $[g_i, h_i] \times [g_j, h_j]$  vertically to the paraboloid [Fig. 3.3(b)].

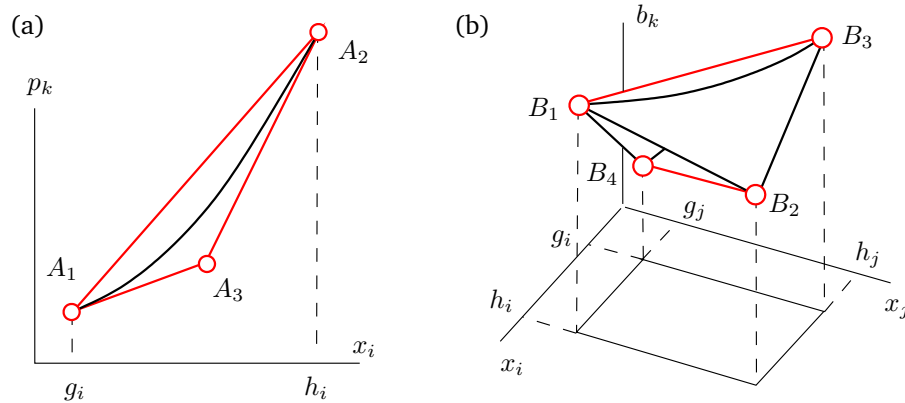


Figure 3.3: Polytope bounds within box  $\mathcal{B}_c$ .

---

**Algorithm 3.1:** The top-level search scheme.

---

**Solve-system**( $\mathcal{B}, \mathbf{F}, \sigma, \rho$ )  
**input** : The initial bounding box,  $\mathcal{B}$ , the functions defining the singularity set to compute,  $\mathbf{F}$ , and the parameters  $\sigma$  and  $\rho$ .  
**output**: A list  $\mathbf{B}$  of solution boxes.

```

1  $\mathbf{B} \leftarrow \emptyset$ 
2  $\mathbf{P} \leftarrow \{\mathcal{B}\}$ 
3 while  $\mathbf{P} \neq \emptyset$  do
4    $\mathcal{B}_c \leftarrow \text{EXTRACT}(\mathbf{P})$ 
5   while not  $\text{IS-VOID}(\mathcal{B}_c)$  and  $\text{SIZE}(\mathcal{B}_c) > \sigma$  and  $\frac{V_c}{V_p} \leq \rho$  do
6      $V_p \leftarrow \text{VOLUME}(\mathcal{B}_c)$ 
7      $\text{SHRINK-BOX}(\mathcal{B}_c, \mathbf{F})$ 
8      $V_c \leftarrow \text{VOLUME}(\mathcal{B}_c)$ 
9   if not  $\text{IS-VOID}(\mathcal{B}_c)$  then
10     if  $\text{SIZE}(\mathcal{B}_c) \leq \sigma$  then
11        $\mathbf{B} \leftarrow \mathbf{B} \cup \{\mathcal{B}_c\}$ 
12     else
13        $(\mathcal{B}_1, \mathcal{B}_2) \leftarrow \text{SPLIT-BOX}(\mathcal{B}_c)$ 
14        $\mathbf{P} \leftarrow \mathbf{P} \cup \{\mathcal{B}_1, \mathcal{B}_2\}$ 
15 RETURN( $\mathbf{B}$ )
```

---

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

Algorithm 3.1 gives the top-level pseudo-code of the process. As input, it receives the box  $\mathcal{B}$ , the list  $\mathbf{F}$  containing the equations in Eq. (3.20) describing the singularity set to compute, and two threshold parameters  $\sigma$  and  $\rho$ . As output, it returns a list  $\mathbf{B}$  of “solution boxes” that enclose all points of the solution set. The functions  $\text{VOLUME}(\mathcal{B})$  and  $\text{SIZE}(\mathcal{B})$  compute the volume and the length of the longest side of  $\mathcal{B}$ , respectively. These and other low-level procedures of straightforward implementation are left unspecified.

Initially, two lists are set up in lines 1 and 2: an empty list  $\mathbf{B}$  of solution boxes, and a list  $\mathbf{P}$  of boxes to be processed, containing  $\mathcal{B}$ . A **while** loop is then executed until  $\mathbf{P}$  gets empty (lines 3–14), by iterating the following steps. Line 4 extracts one box from  $\mathbf{P}$ . Lines 5–8 repeatedly reduce this box as much as possible, via the  $\text{SHRINK-BOX}$  function, until either the box is an empty set ( $\text{IS-VOID}(\mathcal{B}_c)$  is true), or it cannot be significantly reduced ( $V_c/V_p > \rho$ ), or it becomes small enough ( $\text{SIZE}(\mathcal{B}) \leq \sigma$ ). In the latter case, the box is considered a solution for the problem. If a box is neither a solution nor empty, lines 13 and 14 split it into two sub-boxes and add them to  $\mathbf{P}$  for further processing. The  $\text{SHRINK-BOX}$  procedure in line 7 takes as input the box  $\mathcal{B}_c$  to shrink, and the list of equations  $\mathbf{F}$ . The procedure collects all linear equations, all half planes

approximating the parabolic equations, and, finally, all half planes approximating the hyperbolic equations. Then uses these constraints to reduce every dimension of the box by solving the linear programs indicated above, which possibly give tighter bounds for the corresponding intervals.

As it turns out, the previous algorithm explores a binary tree of boxes whose internal nodes correspond to boxes that have been split at some time, and whose leaves are either solution or empty boxes. The collection of all solution boxes, which is returned as output in line 15, is said to form a *box approximation* of the solution set of Eq. (3.20), because its boxes form a discrete envelope of the set, whose accuracy can be adjusted through the  $\sigma$  parameter (Fig. 3.4 illustrates such approximations on two examples).

Notice that the algorithm is complete, in the sense that it will succeed in isolating all solution points of Eq. (3.20) accurately, provided that a small-enough value for  $\sigma$  is used. Detailed properties of the algorithm, including an analysis of its completeness, correctness, and convergence order, can be found in [31].

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

## 3.4 Visualizing the Singularity Sets

Even though we have a means to compute  $S$ , a non-trivial issue is how to represent this set in a meaningful way, suitable to the needs of a robot designer. Because of the high number of configuration variables typically involved in  $\mathbf{q}$ ,  $S$  is often defined in a highly-dimensional space, so that the use of 2- or 3-D projections becomes inevitable to understand its structure. An enlightening choice, as done for instance in [19, 21, 60, 61], is to project  $S$  onto the output space  $\mathcal{U}$ , since this space encodes the end-effector motion and is easier to interpret. On such a projection, points corresponding to inverse singularities indicate a loss of instantaneous degrees of freedom relative to the  $\mathbf{u}$  variables, and, as shown in Chapter 4, they provide the boundaries and interior barriers of the workspace relative to such variables. Similarly,  $S$  can be projected onto the input space  $\mathcal{V}$ , as done for example in [62–64], where the forward singularities delimit

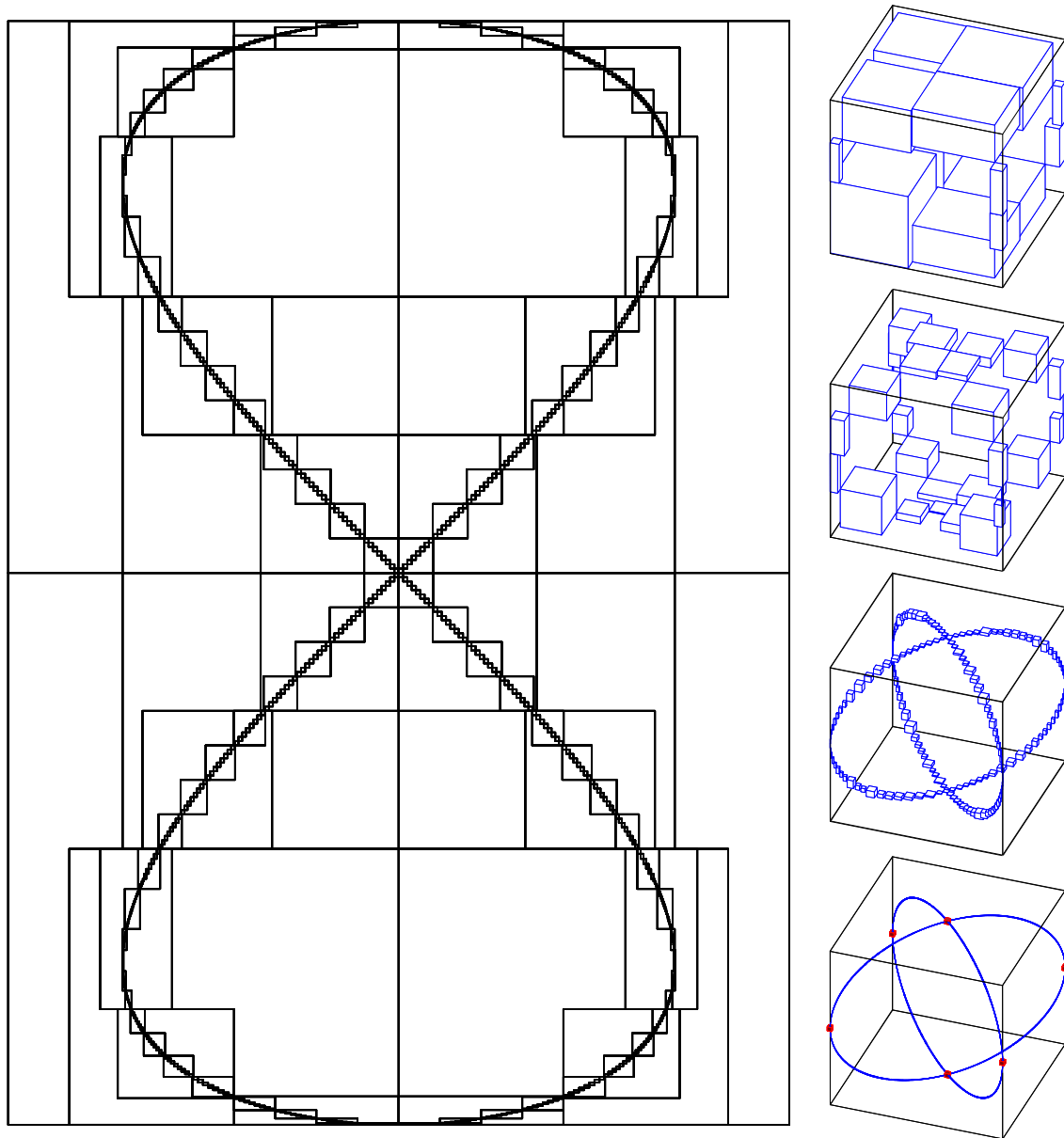


Figure 3.4: Left: progression of the algorithm on computing the lemniscate curve of Geronno, defined by the equation  $y^4 = (y^2 - x^2)$ . We show the initial box, together with the intermediate and final box approximations generated by the algorithm. Right: progression of the algorithm on computing the C-space of the 3-slider mechanism of Section 2.4 [solution of Eq. (2.15)] for  $L_1 = L_2$ . From top to bottom the sequence shows four stages of the computation, in blue, with the singularities of the mechanism [solutions of Eqs. (2.7) and (2.8)] shown overlaid in the last plot, in red. The boxes obtained for such singularities are magnified for clarity, because the value of  $\sigma$  used to compute them was too small to allow their appreciation.

the motion range that should be reachable by the actuators. Both the  $\mathcal{V}$  and  $\mathcal{U}$  spaces get partitioned into several regions after such projections, and it is possible to decide what regions correspond to feasible configurations of the manipulator by selecting a point in each region and solving  $\Phi(\mathbf{v}, \mathbf{y}) = \mathbf{0}$  or  $\Phi(\mathbf{u}, \mathbf{z}) = \mathbf{0}$ , with  $\mathbf{v}$  or  $\mathbf{u}$  fixed to the selected point, using the numerical method described in Section 3.3.

The resulting diagrams, which we refer to as *singularity portraits*, convey much global information on the motion capabilities of the manipulator because (Fig. 3.5):

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

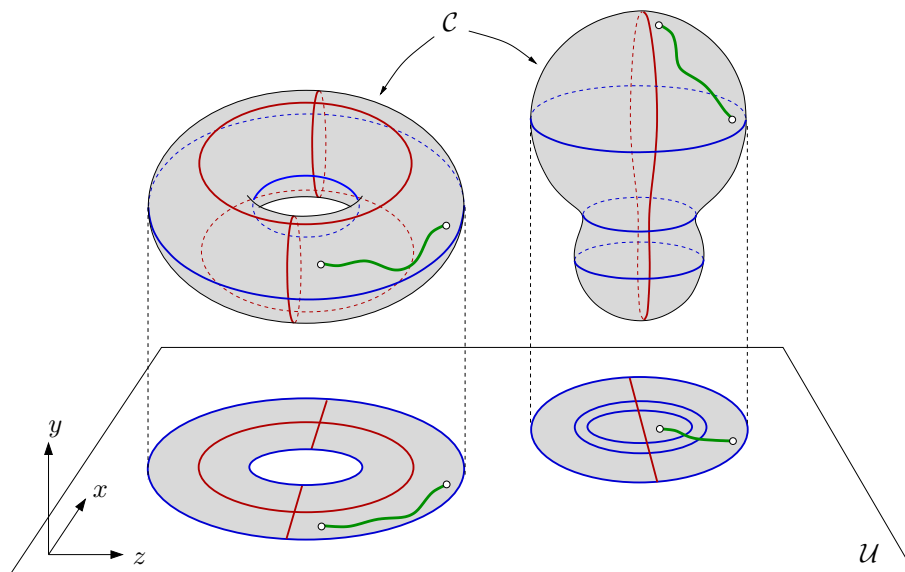


Figure 3.5: A portrait of a synthetic  $\mathcal{C}$ -space with two connected components. The  $\mathcal{V}$  and  $\mathcal{U}$  spaces are assumed to be the  $xy$ - and  $xz$ -planes in this case, so that the forward and inverse singularity loci are the red and blue curves, respectively. Only the portrait on the  $\mathcal{U}$  space is shown for simplicity. The portrait, as in this case, may reveal the existence of several connected components in  $\mathcal{C}$ . Also, it can be used as a safe navigation map, because paths in the portrait not crossing a projected singularity correspond to singularity-free paths on  $\mathcal{C}$  (left path). However, the converse is not necessarily true (right path).

It must be added that the connectivity of the singularity-free regions of  $\mathcal{C}$  is only partially reflected in the portraits. It is easy to see on the right component of Fig. 3.5, for example, that distinct points of  $\mathcal{C}$  may seem to be separated by singularities when looking at the portrait, while they are actually connected by singularity-free paths on  $\mathcal{C}$ . Robust numerical tools to determine the existence of such paths, and to provide the whole singularity-free region of  $\mathcal{C}$  that is reachable from a given configuration, are provided in Chapters 5 and 6.

## 3.5 Examples

The application of the method is next illustrated on a number of examples. In Sections 3.5.1 and 3.5.2 the method is applied to the computation of the forward and inverse singularity sets of planar and spatial manipulators. The computation of Zlatanov's lower-level singularity sets and the complex partition they induce on the C-space is then illustrated in Section 3.5.3 by means of a 2-DOF manipulator.

All computations have been carried out using the parallelized version of the method outlined in Section 3.3.3, implemented in C using the libraries of the CUIK Suite [65], and executed on a grid computer of Xeon processors able to run 160 threads in parallel. A table is given at the end of the section, summarizing the size of the solved systems of and the main performance data on the reported problems.

In all plots that follow, the same color code adopted in Fig. 3.5 has been used to distinguish the forward and inverse singularity loci, and to identify the regions of  $\mathcal{U}$  and  $\mathcal{V}$  attainable by the manipulator.

### 3.5.1 The 3-RRR Manipulator

The 3-RRR manipulator consists of a moving platform linked to the ground by means of three legs, where each leg is a three-revolute chain (Fig. 3.6). Initially, we consider the 3-RRR version of the manipulator, in which the three intermediate joints placed at  $P_4$ ,  $P_5$ , and  $P_6$  are actuated, allowing to control the three degrees of freedom of the platform, and the remaining joints are passive. The inputs of the manipulator are thus the angular velocities at joints  $J_4$ ,  $J_5$ , and  $J_6$ , whereas the outputs are given by the twist of the moving platform  $P_7 - P_8 - P_9$ , which acts as the end-effector.

Several tools can be used to study the singularity set  $\mathcal{S}$  of this manipulator [67–69], which is known to be two-dimensional in general. A good reference summarizing them is [19], where it is shown that the forward singularities can be derived from those of the 3-RPR manipulator [67], whereas the inverse singularities can be generated geometrically, from the so-called



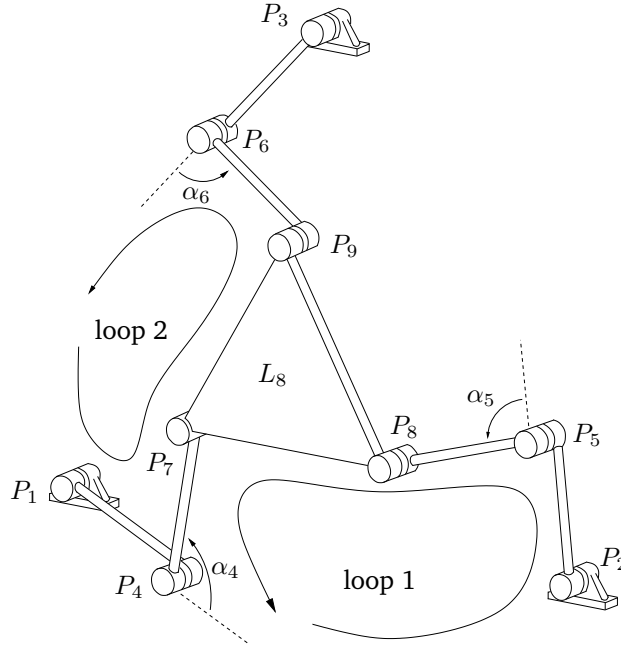


Figure 3.6: A planar 3-RRR manipulator. Points  $P_1$ ,  $P_2$ , and  $P_3$  are fixed to the ground. The end-effector is link  $L_8$ , and is given by  $P_7$ ,  $P_8$ , and  $P_9$ . The relative angles at the intermediate joints of each leg are labelled  $\alpha_i$ . Figure adapted from [66].

vertex-spaces of the legs. These methods are useful, but concentrate on deriving the constant-orientation slices of  $\mathcal{S}$  only, so that a reconstruction of the whole singularity surface involves a discretization on the orientation angle of the platform, which necessarily leaves points of  $\mathcal{S}$  out of the representation. Moreover, only projections of the slices on the plane of two output variables are derived, so that the visualization of the singularity surface on the input space, for example, is not straightforward. The method presented here, in contrast, allows to compute the whole singularity surface directly on  $\mathcal{C}$ , and to project it easily onto any required space, including  $\mathcal{V}$  or  $\mathcal{U}$ , without incurring in any loss of information.

To compute  $\mathcal{S}$ , the proposed method requires formulating the assembly constraints as explained in Section 3.2. For this purpose, we label the links of the manipulator as  $L_1, \dots, L_8$ , where  $L_1$  is the ground link defined by  $P_1$ ,  $P_2$ , and  $P_3$ ,  $L_8$  is the moving platform  $P_7 - P_8 - P_9$ , and the other links  $L_i$  are those defined by points  $P_{i-1}$  and  $P_{i+2}$ . We also label the joints as  $J_1, \dots, J_8$ , where  $J_i$  is the joint at point  $P_i$ . For the particular manipulators under study, the coordinates of points  $P_i$  in the local frames of each link are given in Table 3.2, which correspond to the geometries analyzed in [19, Section 2.4].

Since this is a planar manipulator, the position vectors  $\mathbf{r}_i$  of the links provide points in  $\mathbb{R}^2$ ,

| Link  | $L_1$  | $L_2$   | $L_3$   | $L_4$  |
|-------|--|---|---|--|
| 3-RRR | $\mathbf{p}_1^{\mathcal{F}_1} = [0, 0]^\top$           | $\mathbf{p}_1^{\mathcal{F}_2} = [0, 0]^\top$    | $\mathbf{p}_2^{\mathcal{F}_3} = [0, 0]^\top$    | $\mathbf{p}_3^{\mathcal{F}_4} = [0, 0]^\top$             |
|       | $\mathbf{p}_2^{\mathcal{F}_1} = [-2.386, 0]^\top$      | $\mathbf{p}_4^{\mathcal{F}_2} = [4, 0]^\top$    | $\mathbf{p}_5^{\mathcal{F}_3} = [4, 0]^\top$    | $\mathbf{p}_6^{\mathcal{F}_4} = [4, 0]^\top$             |
|       | $\mathbf{p}_3^{\mathcal{F}_1} = [-1.193, -2.067]^\top$ |   |   |  |
| 3-RRR | $\mathbf{p}_1^{\mathcal{F}_1} = [0, 0]^\top$           | $\mathbf{p}_1^{\mathcal{F}_2} = [0, 0]^\top$    | $\mathbf{p}_2^{\mathcal{F}_3} = [0, 0]^\top$    | $\mathbf{p}_3^{\mathcal{F}_4} = [0, 0]^\top$             |
|       | $\mathbf{p}_2^{\mathcal{F}_1} = [2.35, 0]^\top$        | $\mathbf{p}_4^{\mathcal{F}_2} = [1, 0]^\top$    | $\mathbf{p}_5^{\mathcal{F}_3} = [1, 0]^\top$    | $\mathbf{p}_6^{\mathcal{F}_4} = [1, 0]^\top$             |
|       | $\mathbf{p}_3^{\mathcal{F}_1} = [1.175, 2.035]^\top$   |   |   |  |
| Link  | $L_5$  | $L_6$   | $L_7$   | $L_8$  |
| 3-RRR | $\mathbf{p}_4^{\mathcal{F}_5} = [0, 0]^\top$           | $\mathbf{p}_5^{\mathcal{F}_6} = [0, 0]^\top$    | $\mathbf{p}_6^{\mathcal{F}_7} = [0, 0]^\top$    | $\mathbf{p}_7^{\mathcal{F}_8} = [0, 0]^\top$             |
|       | $\mathbf{p}_7^{\mathcal{F}_5} = [3, 0]^\top$           | $\mathbf{p}_8^{\mathcal{F}_6} = [3, 0]^\top$    | $\mathbf{p}_9^{\mathcal{F}_7} = [3, 0]^\top$    | $\mathbf{p}_8^{\mathcal{F}_8} = [-0.276, 0.276]^\top$    |
|       |  |   |   | $\mathbf{p}_9^{\mathcal{F}_8} = [-0.919, 0.184]^\top$    |
| 3-RRR | $\mathbf{p}_4^{\mathcal{F}_5} = [0, 0]^\top$           | $\mathbf{p}_5^{\mathcal{F}_6} = [0, 0]^\top$    | $\mathbf{p}_6^{\mathcal{F}_7} = [0, 0]^\top$    | $\mathbf{p}_7^{\mathcal{F}_8} = [0, 0]^\top$             |
|       | $\mathbf{p}_7^{\mathcal{F}_5} = [1.35, 0]^\top$        | $\mathbf{p}_8^{\mathcal{F}_6} = [1.35, 0]^\top$ | $\mathbf{p}_9^{\mathcal{F}_7} = [1.35, 0]^\top$ | $\mathbf{p}_8^{\mathcal{F}_8} = [1.2, 0]^\top$           |
|       |  |   |   | $\mathbf{p}_9^{\mathcal{F}_8} = [0.6, 0.6\sqrt{3}]^\top$ |

Table 3.2: Joint coordinates of the considered 3-RRR manipulators.

and the matrices  $\mathbf{R}_i$  can be written in simplified form as

$$\mathbf{R}_i = \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix},$$

where  $c_i$  and  $s_i$  are the cosine and sine of the angle  $\theta_i$  between  $\mathcal{F}_1$  and  $\mathcal{F}_i$ , and thus verify

$$c_i^2 + s_i^2 = 1. \quad (3.21)$$

Note that the assembly constraint of a revolute joint in the plane is equivalent to imposing Eq. (3.1) only. Therefore, Eq. (2.1) is in this case the system formed by Eq. (3.1) for all joints and Eq. (3.21) for all links. However, as shown in [31], it is possible to eliminate the  $r_i$  variables by replacing all Eqs. (3.1) by their sum along a set of independent loops of the mechanism. In such case, Eq. (2.1) can also be formulated by gathering Eq. (3.21) for all links, together with the equations

$$\mathbf{R}_2\mathbf{p}_4^{\mathcal{F}_2} + \mathbf{R}_5\mathbf{p}_7^{\mathcal{F}_5} + \mathbf{R}_8\mathbf{p}_8^{\mathcal{F}_8} - \mathbf{R}_6\mathbf{p}_8^{\mathcal{F}_6} - \mathbf{R}_3\mathbf{p}_5^{\mathcal{F}_3} = \mathbf{0}, \quad (3.22)$$

$$\mathbf{R}_2\mathbf{p}_4^{\mathcal{F}_2} + \mathbf{R}_5\mathbf{p}_7^{\mathcal{F}_5} + \mathbf{R}_8\mathbf{p}_8^{\mathcal{F}_8} - \mathbf{R}_7\mathbf{p}_9^{\mathcal{F}_7} - \mathbf{R}_4\mathbf{p}_6^{\mathcal{F}_4} = \mathbf{0}, \quad (3.23)$$

relative to the closure of loops 1 and 2 indicated in Fig. 3.6. Whereas the original system has 25 equations and 28 variables, this reduced system has only 11 equations and 14 variables, and thus it is easier to solve.

If  $\omega_i$  is the joint velocity of  $J_i$ , and  $\hat{S}_i$  is the associated twist, the twist loop equations relative to loops 1 and 2 can be formulated as

$$\omega_1 \hat{S}_1 + \omega_4 \hat{S}_4 + \omega_7 \hat{S}_7 - \omega_8 \hat{S}_8 - \omega_5 \hat{S}_5 - \omega_2 \hat{S}_2 = \mathbf{0}, \quad (3.24)$$

$$\omega_1 \hat{S}_1 + \omega_4 \hat{S}_4 + \omega_7 \hat{S}_7 - \omega_9 \hat{S}_9 - \omega_6 \hat{S}_6 - \omega_3 \hat{S}_3 = \mathbf{0}, \quad (3.25)$$

and the twist  $\hat{T} = [v_x, v_y, \omega]^\top$  of the end-effector can be introduced using the equation

$$\hat{T} = \omega_1 \hat{S}_1 + \omega_4 \hat{S}_4 + \omega_7 \hat{S}_7. \quad (3.26)$$

Hence, the velocity equation  $L\mathbf{m} = \mathbf{0}$  is in this case defined by Eqs. (3.24)–(3.26), being  $\mathbf{m} = [v_x, v_y, \omega, \omega_1, \dots, \omega_9]^\top$  and

$$L = \begin{bmatrix} I_{3 \times 3} & \hat{S}_1 & \mathbf{0} & \mathbf{0} & \hat{S}_4 & \mathbf{0} & \mathbf{0} & \hat{S}_7 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{S}_1 & -\hat{S}_2 & \mathbf{0} & \hat{S}_4 & -\hat{S}_5 & \mathbf{0} & \hat{S}_7 & -\hat{S}_8 & \mathbf{0} \\ \mathbf{0} & \hat{S}_1 & \mathbf{0} & -\hat{S}_3 & \hat{S}_4 & \mathbf{0} & -\hat{S}_6 & \hat{S}_7 & \mathbf{0} & -\hat{S}_9 \end{bmatrix}. \quad (3.27)$$

Taking into account that  $\mathbf{m}_v = [\omega_4, \omega_5, \omega_6]^\top$  and that  $\mathbf{m}_u = [v_x, v_y, \omega]^\top$ , the matrices  $L_y$  and  $L_z$  needed to formulate Eqs. (2.7) and (2.8) are

$$L_y = \begin{bmatrix} I_{3 \times 3} & \hat{S}_1 & \mathbf{0} & \mathbf{0} & \hat{S}_7 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{S}_1 & -\hat{S}_2 & \mathbf{0} & \hat{S}_7 & -\hat{S}_8 & \mathbf{0} \\ \mathbf{0} & \hat{S}_1 & \mathbf{0} & -\hat{S}_3 & \hat{S}_7 & \mathbf{0} & -\hat{S}_9 \end{bmatrix}, \quad (3.28)$$

and

$$L_z = \begin{bmatrix} \hat{S}_1 & \mathbf{0} & \mathbf{0} & \hat{S}_4 & \mathbf{0} & \mathbf{0} & \hat{S}_7 & \mathbf{0} & \mathbf{0} \\ \hat{S}_1 & -\hat{S}_2 & \mathbf{0} & \hat{S}_4 & -\hat{S}_5 & \mathbf{0} & \hat{S}_7 & -\hat{S}_8 & \mathbf{0} \\ \hat{S}_1 & \mathbf{0} & -\hat{S}_3 & \hat{S}_4 & \mathbf{0} & -\hat{S}_6 & \hat{S}_7 & \mathbf{0} & -\hat{S}_9 \end{bmatrix}. \quad (3.29)$$

Finally, it is easy to see that in the planar case the expression of  $\hat{S}_i$  given in Table 3.1 reduces to

$$\hat{S}_i = \begin{bmatrix} y_i \\ -x_i \\ 1 \end{bmatrix}, \quad (3.30)$$

where  $\mathbf{p}_i = (x_i, y_i)$  are the absolute coordinates of  $P_i$  [59]. These coordinates can be expressed in terms of the configuration of the manipulator using Eq. (3.17), if the  $\mathbf{r}_i$  variables have not

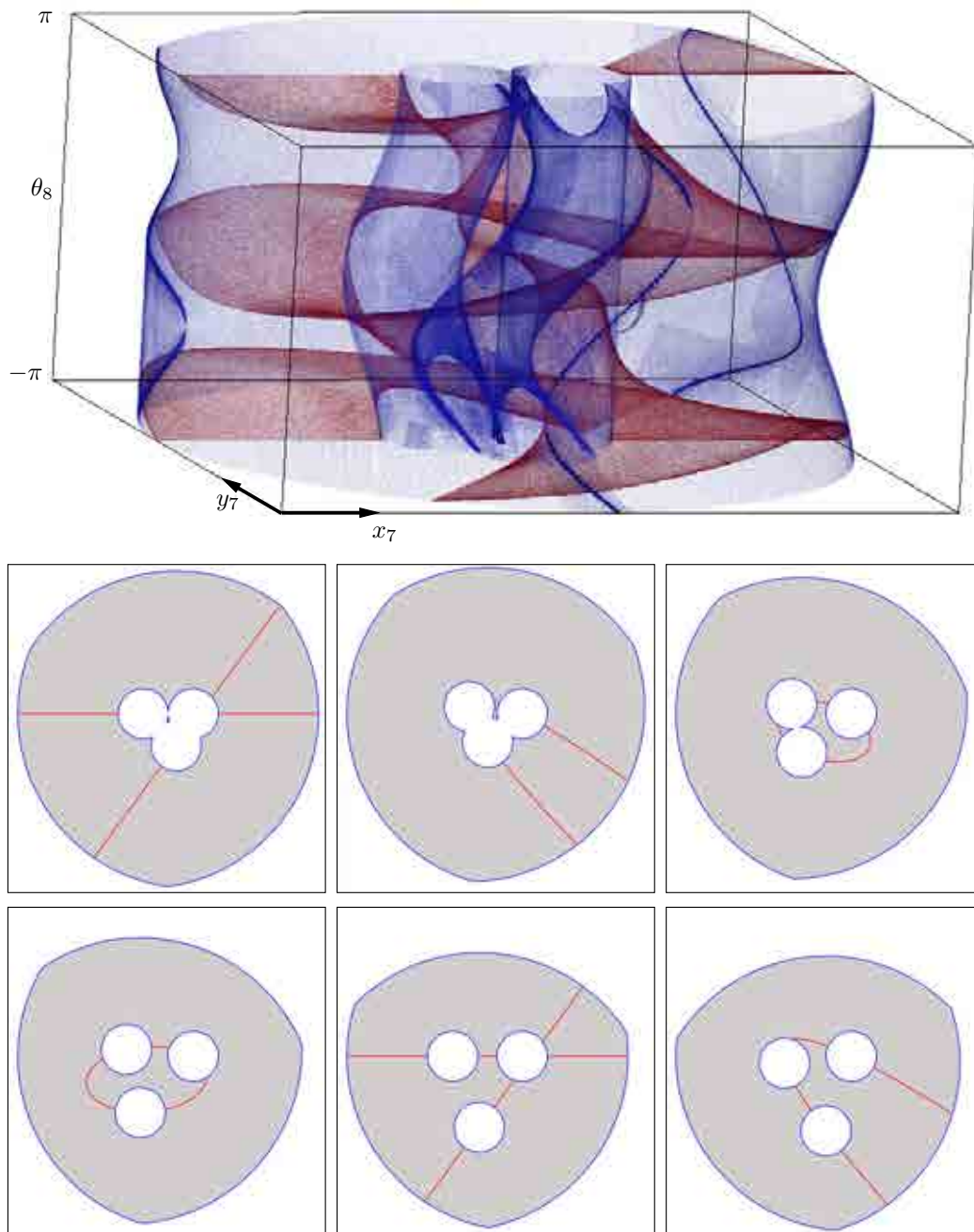


Figure 3.7: Output portrait obtained for the 3-RRR manipulator. Top: Forward (red) and inverse (blue) singularity surfaces in the space defined by  $x_7$ ,  $y_7$ , and  $\theta_8$ . The boxes computed are drawn with translucent faces to better appreciate the shape of the surfaces. Bottom: Slices of the output portrait at a constant value of  $\theta_8$ . From left to right, and from top to bottom, the values of  $\theta_8$  assumed are  $-\pi$ ,  $-\frac{3\pi}{4}$ ,  $-\frac{\pi}{2}$ ,  $-\frac{\pi}{4}$ ,  $0$ , and  $\frac{\pi}{4}$ .

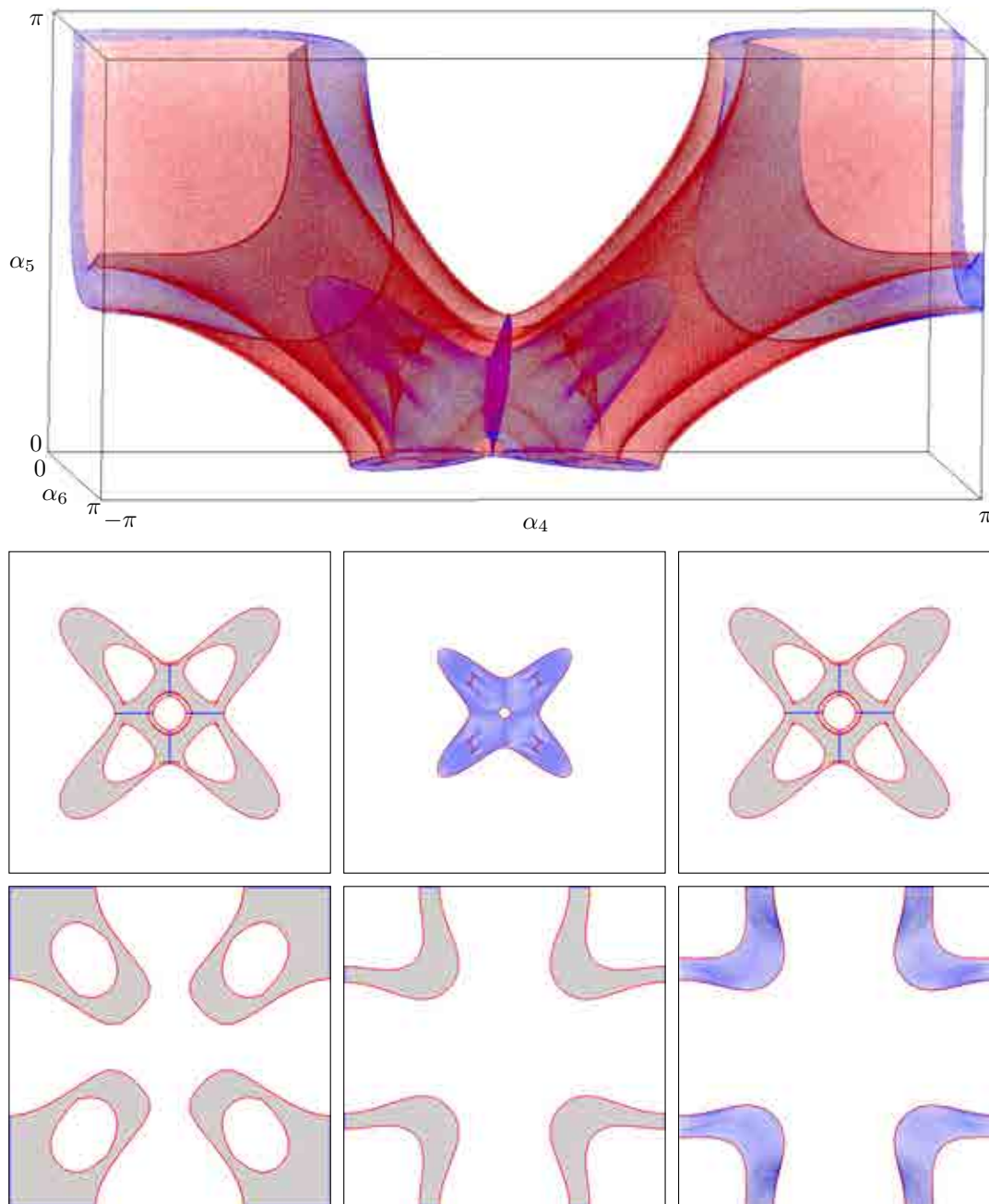


Figure 3.8: Input portrait of the 3-RRR manipulator. Top: Forward (red) and inverse (blue) singularity surfaces in the space defined by  $\alpha_4$ ,  $\alpha_5$ , and  $\alpha_6$ . Only two octants of the space are shown for simplicity, the other octants can be obtained by symmetry. Bottom: Slices of the input portrait at different values of  $\alpha_6$ . From left to right, and from top to bottom, the values of  $\alpha_6$  assumed are  $-\frac{\pi}{4}$ ,  $0$ ,  $\frac{\pi}{4}$ ,  $\frac{\pi}{2}$ ,  $\frac{3\pi}{4}$ , and  $\pi$ .

been eliminated, or by resorting to the expressions

$$\begin{aligned}\mathbf{p}_4 &= \mathbf{p}_1 + \mathbf{R}_2 \mathbf{p}_4^{\mathcal{F}_2}, \\ \mathbf{p}_5 &= \mathbf{p}_2 + \mathbf{R}_3 \mathbf{p}_5^{\mathcal{F}_3}, \\ \mathbf{p}_6 &= \mathbf{p}_3 + \mathbf{R}_4 \mathbf{p}_6^{\mathcal{F}_4}, \\ \mathbf{p}_7 &= \mathbf{p}_4 + \mathbf{R}_5 \mathbf{p}_7^{\mathcal{F}_5}, \\ \mathbf{p}_8 &= \mathbf{p}_5 + \mathbf{R}_6 \mathbf{p}_8^{\mathcal{F}_6}, \\ \mathbf{p}_9 &= \mathbf{p}_6 + \mathbf{R}_7 \mathbf{p}_9^{\mathcal{F}_7},\end{aligned}$$

otherwise. Note that  $\mathbf{p}_1 = \mathbf{p}_1^{\mathcal{F}_1}$ ,  $\mathbf{p}_2 = \mathbf{p}_2^{\mathcal{F}_2}$ , and  $\mathbf{p}_3 = \mathbf{p}_3^{\mathcal{F}_3}$ , so these vectors take the constant values indicated in Table 3.2.

We now know how to formulate Eqs. (2.7) and (2.8) in the form required by the numerical method but, in order to generate the portraits mentioned in Section 3.4, it is convenient to introduce the input and output variables in the systems. The output variables are given by  $\mathbf{u} = [x_7, y_7, \theta_8]^\top$ , and they are already explicit in the formulation. The input variables are given by the relative angles  $\mathbf{v} = [\alpha_4, \alpha_5, \alpha_6]^\top$  indicated in Fig. 3.6, whose sines and cosines can be obtained by completing the system with the equations

$$\begin{aligned}s_{\alpha_i} &= s_{i+1}c_{i-2} - c_{i+1}s_{i-2}, \\ c_{\alpha_i} &= c_{i+1}c_{i-2} + s_{i+1}s_{i-2},\end{aligned}$$

which are nothing but the algebraic version of  $\alpha_i = \theta_{i+1} - \theta_{i-2}$ . By introducing these equations, the numerical method will be able to provide ranges for the sines and cosines of all angles involved,  $\alpha_4$ ,  $\alpha_5$ , and  $\alpha_6$ , from which it is straightforward to obtain the corresponding angular ranges.

The singularity surfaces obtained by the method are shown in Fig. 3.7 (top), projected onto the output space. The blue surface corresponds to the inverse singularity locus, which provides the boundaries of the workspace. The red surface corresponds to the forward singularity locus, i.e., to configurations where the motion control is compromised, due to the specific choice of actuated degrees-of-freedom. Even though these singularity surfaces appear to be quite complex, it can be shown that the constant-orientation slices of the forward singularity locus can be described by conic sections in the plane of the  $x_7$  and  $y_7$  variables [19, 67]. Any of these slices can be readily obtained by the proposed method by simply fixing the value of  $\theta_8$  in the equations, obtaining the red curves shown in Fig. 3.7 (bottom), where only pairs of lines, parabolas, or ellipses appear as expected. The inverse singularity curves in such plots do also coincide with those obtained through the intersection of vertex spaces [19, 68].

By simply changing the projection coordinates we can easily represent  $\mathcal{S}$  in the input space as well, obtaining the results shown in Fig. 3.8. Here, the forward singularities delimit the motion range of the actuators, and it can be seen how the inverse singularities only appear on planes where one of the  $\alpha_i$  angles is either 0 or  $\pi$ , in agreement with the fact that the platform only loses instantaneous mobility when at least one of the legs is fully extended or folded back [19]. To better understand the structure of the singularity surface on the input space, some slices are also shown for constant values of  $\alpha_6$ . Observe how the whole region attainable by the inputs is singular for  $\alpha_6 = 0$  or  $\alpha_6 = \pi$ . On these slices, the inverse singularities are no longer one-dimensional, as one would expect. Whereas this circumstance poses no problem to the proposed method, it may indeed hinder the application of other methods relying on discretization of  $\alpha_6$ .

It must be noted that the structure of the singularity set can become quite complex even on simple manipulators. For example, if on the 3-RRR mechanism we mount the actuators in the ground joints  $J_1$ ,  $J_2$ , and  $J_3$  instead of the intermediate ones, we obtain the 3-RRR architecture, and the constant-orientation slices of the forward singularity locus are then described by polynomials in  $x_7$  and  $y_7$  of minimal degree 42 [19]. Polynomials of such kind constitute valuable tools for the analysis of the singularity set, but their derivation often requires quite involved manipulations guided by intuition [20, 21, 61, 64], which difficults the application of such strategy to every new manipulator that has to be analyzed. The proposed method can compute the mentioned slices just as easily as in the case of the 3-RRR manipulator (Fig. 3.9). The full potential of the method is even more apparent on mechanisms of much higher complexity, where the approach based on descriptive polynomials would be rather difficult to apply [57].

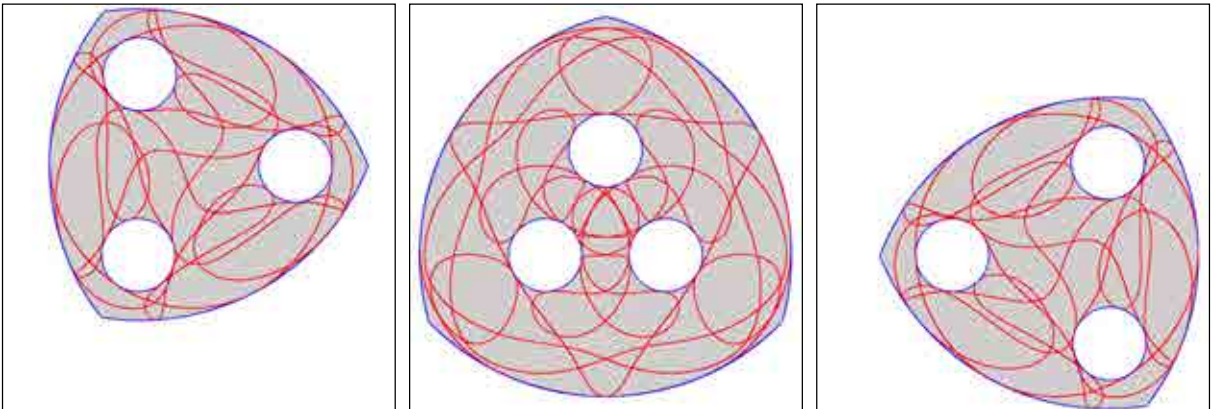


Figure 3.9: Slices of the output portrait of the 3-RRR manipulator computed by the method at fixed orientations of the platform, assuming the geometric parameters in Table 3.2. From left to right, the values assumed for  $\theta_8$  are  $-\frac{\pi}{4}$ , 0 and  $\frac{\pi}{4}$ . The plot of the  $\theta_8 = 0$  slice agrees with the one published in [19, 69].

### 3.5.2 The Stewart Platform

The Stewart platform consists of a moving plate, or platform, connected to a fixed base by means of six legs, where each leg is a universal-prismatic-spherical chain. The most general version of this manipulator follows the so-called 6-6 design, where the leg anchor points are all different (Fig. 3.10), though not necessarily coplanar [70]. The six prismatic joints are actuated, allowing to control the six degrees of freedom of the platform, and the remaining joints are passive.

The assembly constraints can be formulated as explained in Section 3.2. In this case we can define two links, the ground base and the moving platform, labelled  $L_1$  and  $L_2$ , and the points  $P_{1,i}$  and  $P_{2,i}$  at the center of the universal and spherical joints, respectively, for each leg from  $i = 1$  to 6. Let also  $O$  be the origin of the absolute frame,  $\mathcal{F}_1$ , and  $P$  that of the frame attached to the platform,  $\mathcal{F}_2$  (Fig. 3.10). By taking into account the constraints imposed by the universal, prismatic, and spherical joints of the  $i$ -th leg, i.e., Eqs. (3.3)–(3.7), the constraint imposed on the platform by such leg can be reduced to

$$\mathbf{r}_P = \mathbf{p}_{1,i} + d_i \mathbf{d}_i - \mathbf{R}_2 \mathbf{p}_{2,i}^{\mathcal{F}_2}, \quad (3.31)$$

$$\|\mathbf{d}_i\|^2 = 1, \quad (3.32)$$

where  $\mathbf{r}_P$ ,  $\mathbf{p}_{1,i}$ , and  $\mathbf{p}_{2,i}$  are the position vectors of points  $P$ ,  $P_{1,i}$ , and  $P_{2,i}$ , respectively, and  $\mathbf{d}_i$  is a unit vector along the  $i$ -th leg. Also,  $d_i$  is the length of the leg, representing the displacement of the prismatic joint. Thus, Eq. (2.1) can be written in this case by gathering Eq. (3.31) and (3.32) for all legs, and Eqs (3.8)–(3.11) for  $\mathbf{R}_2$ , obtaining a system of 30 equations in 36 variables where

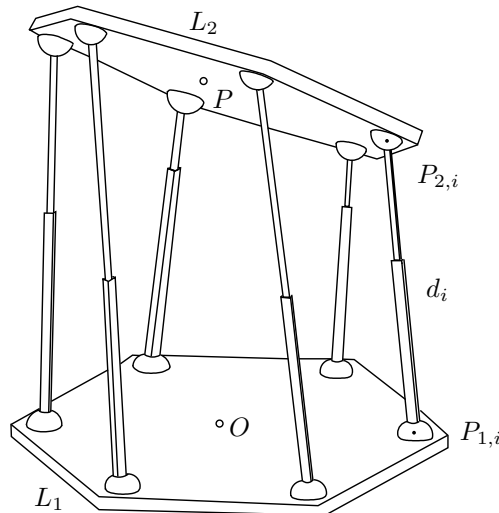


Figure 3.10: The Stewart platform.



| Leg | Base $L_1$  | Platform $L_2$   |
|-----|---|--|
| 1   | $\mathbf{p}_{1,1}^{\mathcal{F}_1} = [92.58, 99.64, 20.10]^\top$   | $\mathbf{p}_{2,1}^{\mathcal{F}_2} = [30.0073.00 - 35.10]^\top$     |
| 2   | $\mathbf{p}_{1,2}^{\mathcal{F}_1} = [132.58, 30.36, 28.45]^\top$  | $\mathbf{p}_{2,2}^{\mathcal{F}_2} = [78.22, -10.52, -23.00]^\top$  |
| 3   | $\mathbf{p}_{1,3}^{\mathcal{F}_1} = [40.00, -120.00, 31.18]^\top$ | $\mathbf{p}_{2,3}^{\mathcal{F}_2} = [48.22, -62.48, -33.60]^\top$  |
| 4   | $\mathbf{p}_{1,4}^{\mathcal{F}_1} = [-46.00, -130.00, 3.10]^\top$ | $\mathbf{p}_{2,4}^{\mathcal{F}_2} = [-44.22, -56.48, -25.50]^\top$ |
| 5   | $\mathbf{p}_{1,5}^{\mathcal{F}_1} = [-130.00, 23.36, 13.48]^\top$ | $\mathbf{p}_{2,5}^{\mathcal{F}_2} = [-70.22, -20.52, -34.10]^\top$ |
| 6   | $\mathbf{p}_{1,6}^{\mathcal{F}_1} = [-82.58, 89.77, 8.76]^\top$   | $\mathbf{p}_{2,6}^{\mathcal{F}_2} = [-34.00, 45.00, -39.00]^\top$  |

Table 3.3: Parameters of the analyzed Stewart platform, in millimeters.

the pose of the end-effector is represented by  $(\mathbf{r}_P, \mathbf{R}_2)$ . The geometric parameters used here are given in Table 3.3 and correspond to the manipulator studied in [21].

It can be seen that each pair of legs forms a closed-loop of the manipulator, but that only five of them are independent. Therefore, the velocity equation can be formulated by gathering Eq. (3.12) for five distinct pairs of legs, and Eq. (3.13) written for any leg. However, on hybrid-chain manipulators like the Stewart platform, the velocity equation can also be obtained by writing Eq. (3.13) for each leg [12]. In such case, the matrix  $L$  adopts the form

$$L = \begin{bmatrix} -I & \hat{S}_1^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_1^p & 0 & 0 & 0 & 0 & 0 \\ -I & 0 & \hat{S}_2^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_2^p & 0 & 0 & 0 & 0 \\ -I & 0 & 0 & \hat{S}_3^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_3^p & 0 & 0 & 0 \\ -I & 0 & 0 & 0 & \hat{S}_4^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_4^p & 0 & 0 \\ -I & 0 & 0 & 0 & 0 & \hat{S}_5^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_5^p & 0 \\ -I & 0 & 0 & 0 & 0 & 0 & \hat{S}_6^a & 0 & 0 & 0 & 0 & 0 & \hat{S}_6^p \end{bmatrix}, \quad (3.33)$$

where  $\hat{S}_i^a$  is the unit twist of the prismatic joint on the  $i$ -th leg, and  $\hat{S}_i^p$  is a  $6 \times 5$  matrix whose columns are the unit twists of the universal and spherical joints of such leg. Then, the velocity vector  $\mathbf{m}$  contains the six components of the output twist, the six active velocities of the prismatic joints, and the 30 passive joint velocities of the universal and spherical joints. Altogether, this is a  $36 \times 42$  matrix, but a method explained in [12] allows to reduce its size by eliminating the passive joint velocities of the equation. The method consists in multiplying each side of Eq. (3.13) of each leg by a unit screw reciprocal to all passive joints of the leg. Note that such a screw is any one that passes through the centers of the universal and spherical joints [71, Section

5.2.2], for instance<sup>1</sup>

$$\hat{\mathbf{s}}_i^r = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{d}_i \times \mathbf{p}_{1,i} \end{bmatrix}. \quad (3.34)$$

By doing the multiplication, Eq. (3.13) reduces to

$$\hat{\mathbf{s}}_i^{r\top} \hat{\mathbf{T}} = \omega_i^a, \quad (3.35)$$

where  $\omega_i^a$  is the velocity of the prismatic joint of leg  $i$ , because  $\hat{\mathbf{s}}_i^{r\top} \hat{\mathbf{S}}_i^p = \mathbf{0}$  and  $\hat{\mathbf{s}}_i^{r\top} \hat{\mathbf{S}}_i^a = 1$ . Collecting Eq. (3.35) for all legs we obtain the velocity equation as

$$\mathbf{J}^\top \hat{\mathbf{T}} = \mathbf{m}_v, \quad (3.36)$$

where  $\mathbf{J}$  is a matrix that has the unit screws  $\hat{\mathbf{s}}_i^r$  as columns, and  $\mathbf{m}_v$  contains the six  $\omega_i^a$ .

For some configurations, the space of reciprocal screws of a given leg may have dimension larger than one. In such a case, the corresponding Eq. (3.13) of the leg should be multiplied by all vectors of a basis of the space of reciprocal screws, obtaining at the end the velocity equation as

$$\mathbf{J}^\top \hat{\mathbf{T}} = \mathbf{H} \mathbf{m}_v, \quad (3.37)$$

where  $\mathbf{J}^\top$  and  $\mathbf{H}$  are rectangular matrices with more rows than columns. This can only happen on the Stewart Platform when the twists of the universal joint become dependent of those of the spherical joint, i.e., when the two joints are coincident and the leg has zero length. This situation corresponds to an RPM type singularity, but it rarely occurs in practice, because the  $d_i$  variables are typically constrained to a positive range of values. In such case,  $\mathbf{J}$  remains square,  $\mathbf{H}$  is the identity matrix because of the choice of  $\hat{\mathbf{s}}_i^r$ , and Eq. (3.36) coincides with the classical simplified version of the velocity equation [72].

Based on Eq. (3.36), it is easy to see that the solution to the FIKP involves the inverse matrix of  $\mathbf{J}$ , so that the forward singularities can be characterized by those configurations where  $\mathbf{J}$  is not full rank. Thus, in order to compute the forward singularities,  $\mathbf{J}$  plays the role of  $\mathbf{L}_y$  in Eq. (2.7), obtaining a much compact system of equations than using  $\mathbf{L}_y$  directly from (3.33). A similar reasoning applies for the inverse singularities, so they can be computed as the solution of Eq. (2.8) replacing  $\mathbf{L}_z$  by  $\mathbf{H}$ . Since  $\mathbf{H}$  is the identity matrix whenever  $d_i \neq 0$ , it is straightforward that the inverse singularities could only occur when some  $d_i$  is zero, which never happens in our case because the  $d_i$  variables are always kept strictly positive. Therefore, we will concentrate on computing the forward singularities only.

<sup>1</sup>The notation  $\hat{\mathbf{s}}_i$  indicates that the twist is written in ray coordinates [59], to allow the reciprocal product.

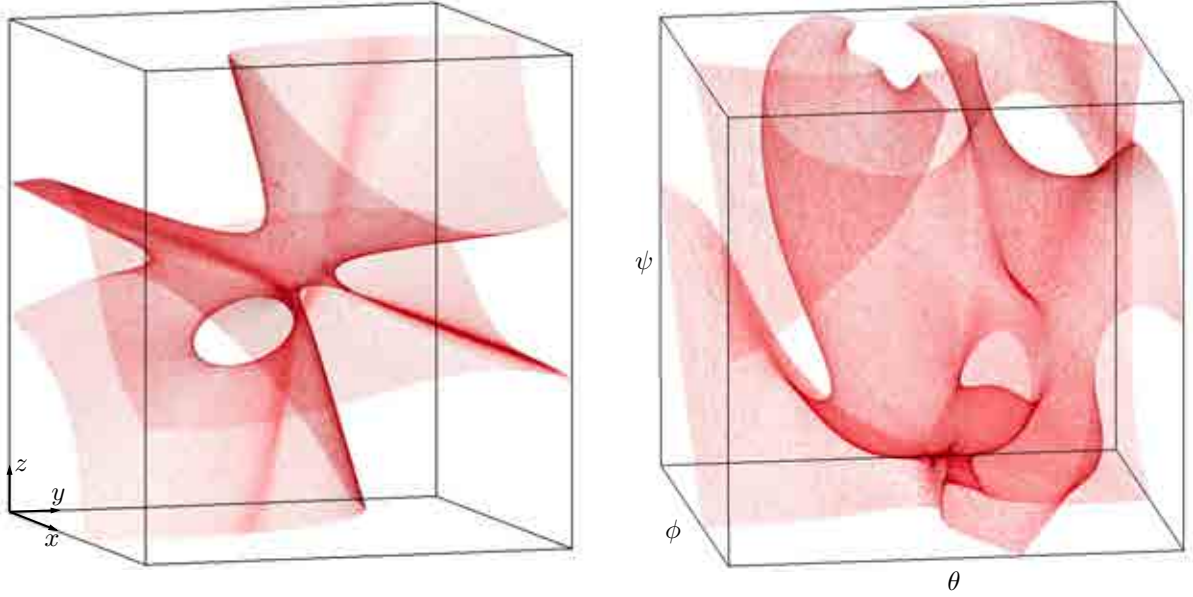


Figure 3.11: Forward singularity locus of the INRIA left hand with the orientation of the platform fixed at  $\phi = -2^\circ$ ,  $\theta = 30^\circ$ , and  $\psi = -87^\circ$  (left), and with point  $P$  fixed at  $\mathbf{p} = [10, 10, 10]^\top$  (right). For ease of visualization the position variables on the left, and the orientation angles on the right, have been limited to the ranges  $[-100, 100]$  and  $[-90^\circ, 90^\circ]$ , respectively.

It is clear that the singularity locus can not be directly visualized in this case due to its high dimension. However, insight can be gained through the projection of three-dimensional slices. In order to plot these slices when they involve the orientation of the platform, it is convenient to introduce three-parameter expressions for  $\mathbf{R}_2$  in the system. It is worth noting that no consensus has been reached as to how should platform orientations be parameterized, but the method herein proposed could equally solve the systems under any possible parameterization, including those based on conventional Euler angles, tilt-and-torsion angles [73], or Euler-Rodrigues parameters. For ease of comparison with [21], we will here adopt the parameterization provided by roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) angles, for which  $\mathbf{R}_2 = \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi)$ , or

$$\mathbf{R}_2 = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (3.38)$$

Two slices of the singularity locus are shown in Fig. 3.11, one computed at a constant orientation of the platform, corresponding to  $\phi = -2^\circ$ ,  $\theta = 30^\circ$ ,  $\psi = -87^\circ$ , and one at a constant position of the point  $P$  in the platform, being  $\mathbf{r}_P = [10, 10, 10]^\top$ . Note however that

any other slice obtained by keeping constant three or more of the pose variables of the platform could also be computed with the method.

### 3.5.3 A Double-loop Manipulator

The 2-DOF manipulator shown in Fig. 3.12 is used here to illustrate the computation of each one of the six lower-level singularity sets. The inputs of the manipulator are the joint velocities at  $A$  and  $E$ , and the output is the velocity of point  $G$ . By gathering the loop-closure equations of the mechanism, and introducing two further equations to include the position of  $G$ , Eq. (2.1) can be formulated as follows

$$\left. \begin{aligned} \cos \theta_A + \cos \theta_B - 2 \cos \theta_D - 1 &= 0 \\ \sin \theta_A + \sin \theta_B - 2 \sin \theta_D &= 0 \\ 2 \cos \theta_D + \frac{3}{2} \cos \theta_C + 2 \cos \theta_G - 3 \cos \theta_E - 1 &= 0 \\ 2 \sin \theta_D + \frac{3}{2} \sin \theta_C + 2 \sin \theta_G - 3 \sin \theta_E &= 0 \\ -x + 2 \cos \theta_D + \frac{3}{2} \cos \theta_C &= 0 \\ -y + 2 \sin \theta_D + \frac{3}{2} \sin \theta_C &= 0 \end{aligned} \right\} \quad (3.39)$$

where  $\theta_A$ ,  $\theta_B$ ,  $\theta_C$ ,  $\theta_D$ ,  $\theta_E$  and  $\theta_G$  are the counterclockwise angles of links  $AB$ ,  $BC$ ,  $CG$ ,  $DC$ ,  $EF$ , and  $GF$ , respectively, relative to the ground, and  $x$  and  $y$  are the coordinates of point  $G$  relative to a fixed frame centered in  $D$ . In this case, the velocity equation of the manipulator is easily obtained by differentiating (3.39) with respect to all variables, although the twist loop equations could also be used. In any case, the systems of equations encoding the six singularity

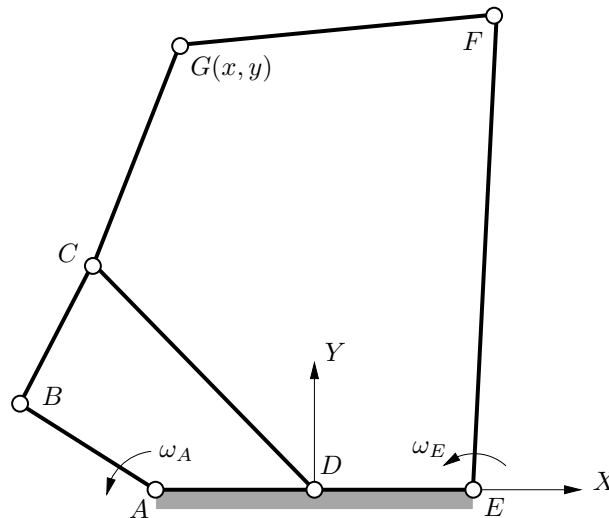


Figure 3.12: A 2-DOF planar manipulator. The link dimensions are  $AB = AD = BC = DE = 1$ ,  $CD = FG = 2$ ,  $CG = 1.5$  and  $EF = 3$ .

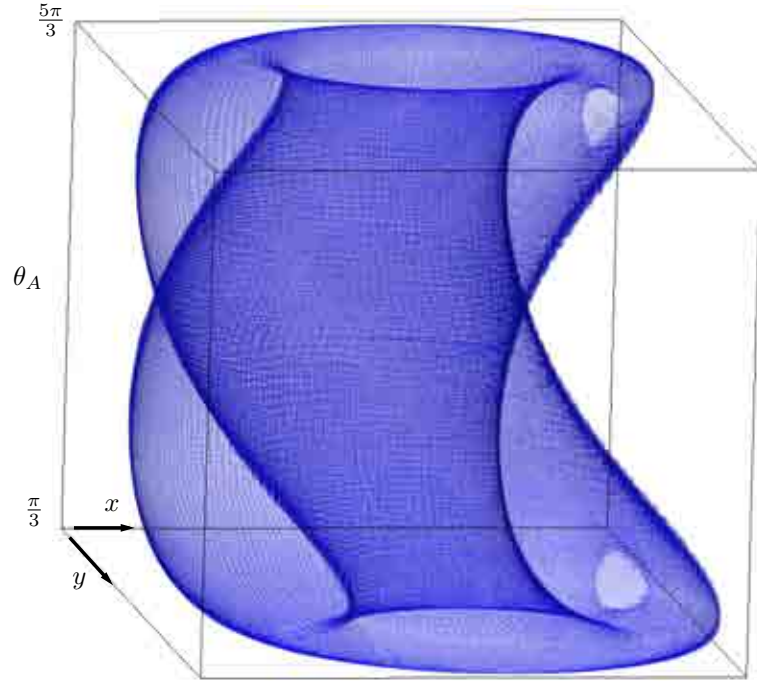


Figure 3.13: Two-dimensional configuration space of the manipulator in Fig. 3.12 computed at  $\sigma = 0.1$ . Two holes can be seen, whose boundary corresponds to configurations where  $E$ ,  $F$ , and  $G$  are aligned.

types, Eq. (2.9)–(2.14), can be written and expanded to the desired quadratic form of Eq. (3.20) by introducing the changes of variables

$$\begin{aligned} c_\tau &= \cos \theta_\tau \\ s_\tau &= \sin \theta_\tau \end{aligned}$$

together with the equation

$$c_\tau^2 + s_\tau^2 = 1,$$

for all angles with  $\tau = \{A, B, C, D, E, G\}$ .

Given that the manipulator has two degrees of freedom, its configuration space is a surface, which is shown projected onto the  $x$ ,  $y$ , and  $\theta_A$  variables in Fig. 3.13. This surface is obtained from the computation of all solutions of Eq. (2.1) using the same numerical technique presented in Section 3.3. Note that by fixing  $x$ ,  $y$ , and  $\theta_A$ , there are still two possible positions of point  $F$ , so that most of the points in this projection correspond, in fact, to two different configurations of the manipulator. Only the points where  $E$ ,  $F$ , and  $G$  are aligned represent a single configuration, and these are exactly the boundaries of the two “holes” that the surface presents.

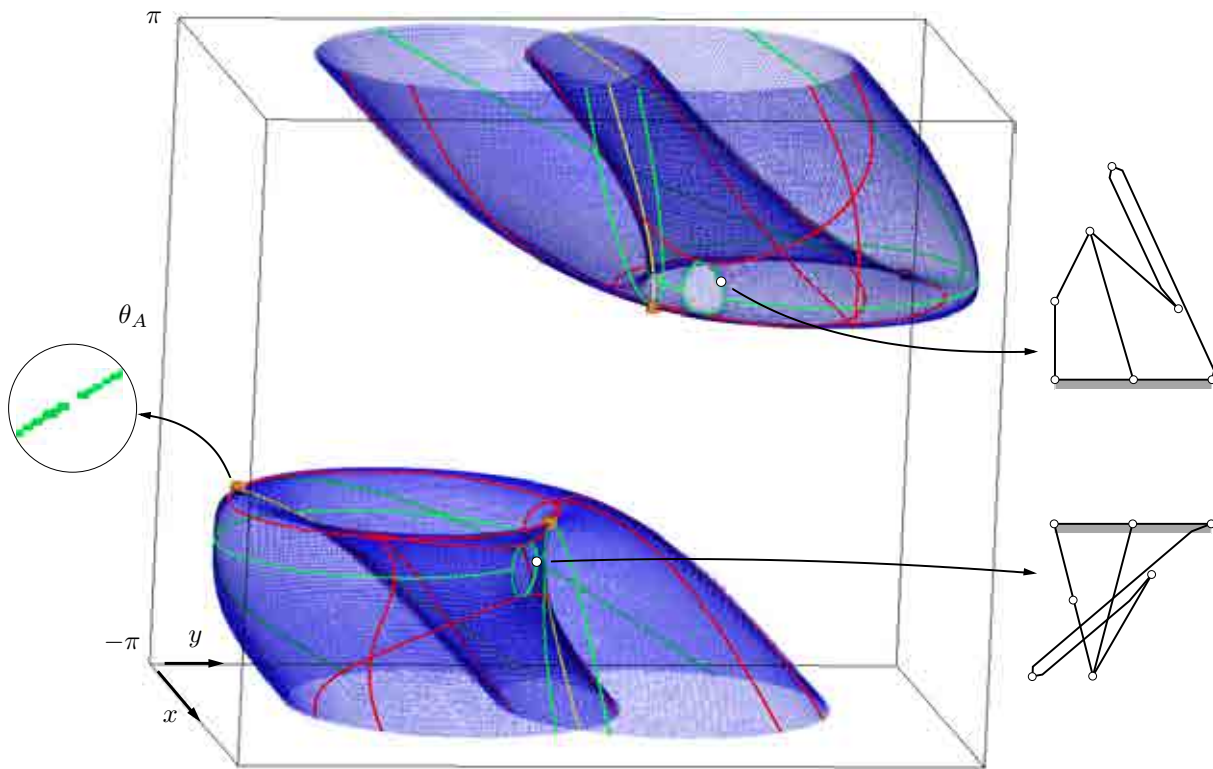


Figure 3.14: The singular configurations of the mechanism in Fig. 3.12 shown overlaid on a projection of its configuration space onto the  $x$ ,  $y$  and  $\theta_A$  variables. Different colors are used to identify the several singularity types encountered: green for the RI and IO types, red for the RO and II types, and orange for the RPM type.

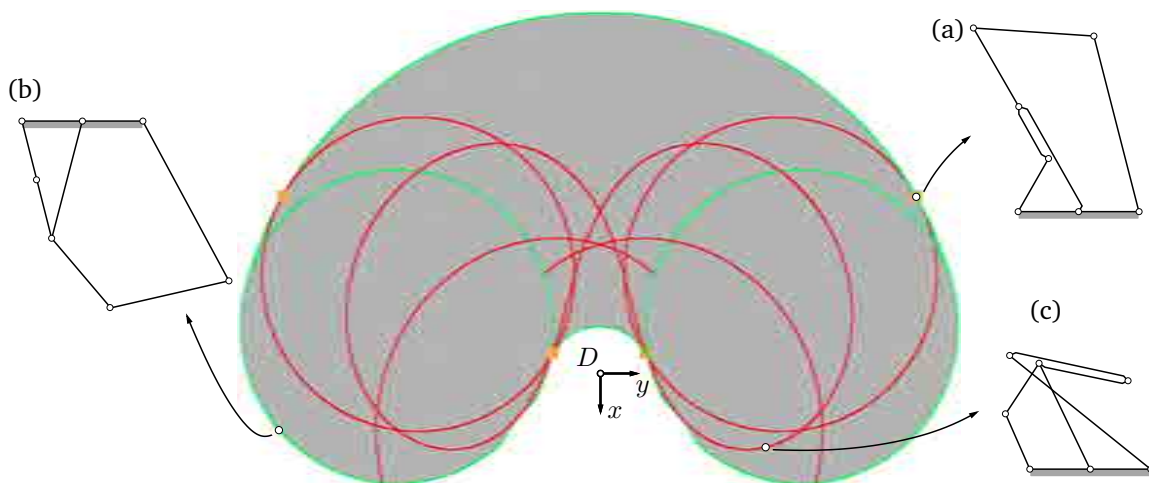


Figure 3.15: A projection of the plot in Fig. 3.14 to the  $(x, y)$  plane. (a) A singularity of RPM, IO, and II type. (b) A singularity of RI and IO type. (c) A singularity of RO and II type.

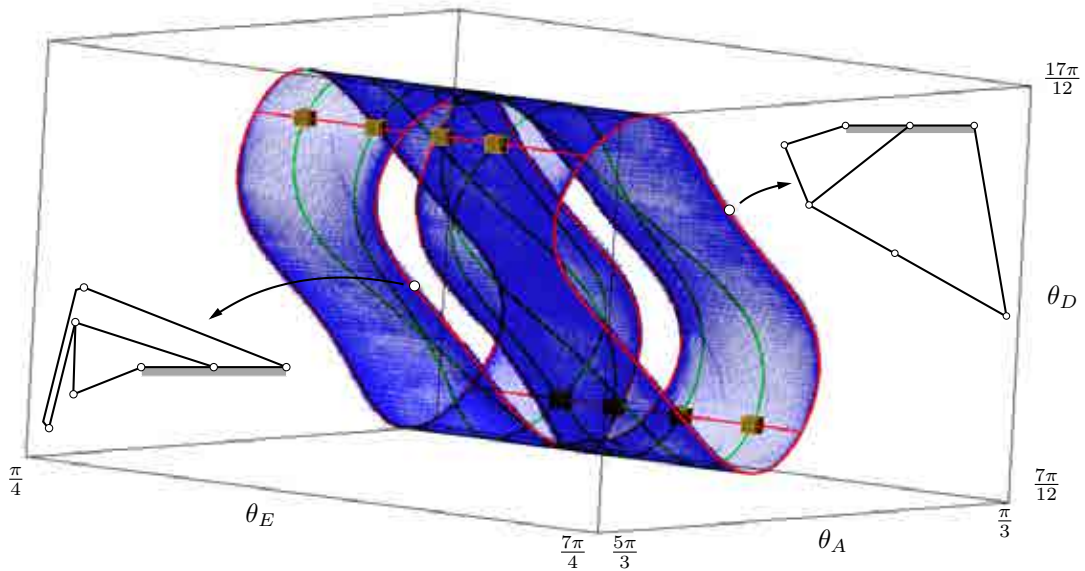


Figure 3.16: A projection of the configuration space and the computed singularities to the  $(\theta_A, \theta_E, \theta_D)$  space, together with two configurations where  $C$ ,  $G$ , and  $F$  are aligned. Green corresponds to the RI and IO types, red to the RO and II types, and orange to the RPM type. There are no singularities of IIM type.

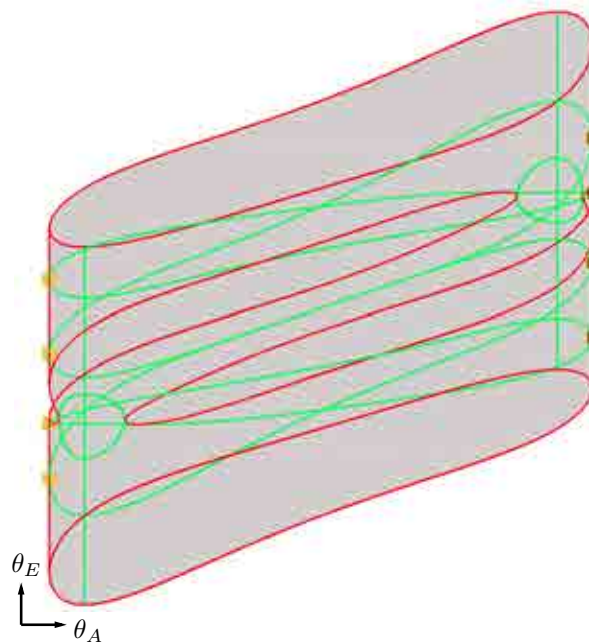


Figure 3.17: A projection of the plot in Fig. 3.16 to the  $(\theta_A, \theta_E)$  space.

The singularity set is generally of lower dimension than the configuration space, so that only curves or points are to be expected in the solution set of all systems of equations. The result of the computation of each singularity type defined in Section 2.3 is shown in Figs. 3.14 and 3.15, projected onto the output and one input  $(x, y, \theta_A)$ , and onto the output only, respectively. In Fig. 3.14, the configuration space is shown in blue, separated in two parts so that a cross-section can be seen, but both parts are actually connected through  $\pi$  and  $-\pi$  as in Fig. 3.13. The grey area in Fig. 3.15 represents all attainable positions of point  $G$ , i.e. the workspace of the manipulator.

As it turns out, this manipulator contains no IIM configurations, and the computation of this type of singularity gives no box as output. On the contrary, there are eight distinct RPM singularities, which in these projections appear coincident in pairs as four orange boxes, corresponding to the two possible locations of  $F$ . Using a different projection, for instance onto  $(\theta_A, \theta_D, \theta_E)$ , the eight boxes appear separated (Fig. 3.16).

The green curves correspond to singularities that are both of RI and IO type. These configurations can be seen to contour the two “holes” of the configuration space in the projection of Fig. 3.14. The red curves correspond to configurations simultaneously belonging to the RO and II type. Even if the curves for RI and IO seem to coincide everywhere, there are some IO configurations that are not of RI type, and the same happens for II and RO singularities, respectively. This is illustrated in Fig. 3.14 with a magnifying bubble that shows only the output of computing RI type singularities. These gaps on the curves of RI and RO, which can be found by properly adjusting the  $\epsilon$  parameter, coincide with the location of the RPM singularities and, hence, the RPM singularities are also of II and IO type but not of RI or RO type. Fig. 3.15(a) shows an example of an (RPM, II, IO) singularity, while Fig. 3.15(b) and Fig. 3.15(c) show examples of (RI, IO) and (RO, II) singularities, respectively.

Yellow curves corresponding to configurations where points  $D$ ,  $B$  and  $G$  are aligned are also shown in Figure 3.14. In these configurations there is only one possible location of point  $C$ . Hence, these configurations, together with those where  $E$ ,  $F$ , and  $G$  are aligned, allow the transition between different assembly modes. In fact, these configurations coincide with the points of self-intersection of the projection of the configuration space on the  $(x, y, \theta_A)$  space. The configuration space itself has no self-intersections as there are no C-space, or IIM type, singularities, and the yellow points are only singularities of the projection map.

Using the same color code, Figs. 3.16 and 3.17 show the projection of the results onto the the 3-dimensional space of the two input angles and one passive joint angle  $(\theta_A, \theta_E, \theta_D)$  and onto the 2-dimensional input space only. The eight RPM singularities appear separated. Similarly as before, for fixed values of  $\theta_A$ ,  $\theta_E$ , and  $\theta_D$ , there are still two possible locations of point  $C$  in general, and almost all points in this projection correspond to two distinct configurations



---

of the manipulator. It can be seen that the configuration space presents four “holes” in these projections. These four contours are made of those configurations where  $G$ ,  $C$ , and  $F$  are aligned and there is only one possibility for  $C$ . Note that none of these “holes” coincides with one in the previous projection, but, once again, crossing each curve allows the transition between two different assembly modes. One can imagine the two assembly modes as the two “sides” of the surface of the configuration-space projection. To “get to the opposite side”, i.e., to change assembly mode, the motion curve must “go through a hole”.

| Figure                      | Manipulator | Computed set  | Slice                       | $dim$ | $N_{eq}-N_{var}$ | $\sigma$                  | Time (s) | $N_{boxes}$ |      |      |        |
|-----------------------------|-------------|---------------|-----------------------------|-------|------------------|---------------------------|----------|-------------|------|------|--------|
| 3.7-3.8                     | 3-RRR       | Forward sing. | —                           | 2     | 29-31            | 0.1                       | 2168     | 150538      |      |      |        |
|                             |             |               | $\theta = -\pi$             | 1     | 28-29            | 0.01                      | 18       | 2692        |      |      |        |
|                             |             |               | $\theta = -\frac{3\pi}{4}$  | 1     | 28-29            | 0.01                      | 14       | 1372        |      |      |        |
|                             |             |               | $\theta = -\frac{\pi}{2}$   | 1     | 28-29            | 0.01                      | 12       | 894         |      |      |        |
|                             |             |               | $\theta = -\frac{\pi}{4}$   | 1     | 28-29            | 0.01                      | 13       | 1113        |      |      |        |
|                             |             |               | $\theta = 0$                | 1     | 28-29            | 0.01                      | 17       | 2612        |      |      |        |
|                             |             |               | $\theta = \frac{\pi}{4}$    | 1     | 28-29            | 0.01                      | 14       | 1658        |      |      |        |
|                             |             |               | $\alpha_3 = -\frac{\pi}{4}$ | 1     | 28-29            | 0.01                      | 186      | 22195       |      |      |        |
|                             |             |               | $\alpha_3 = 0$              | 1     | 28-29            | 0.01                      | 216      | 10158       |      |      |        |
|                             |             |               | $\alpha_3 = \frac{\pi}{4}$  | 1     | 28-29            | 0.01                      | 198      | 22151       |      |      |        |
|                             |             |               | $\alpha_3 = \frac{\pi}{2}$  | 1     | 28-29            | 0.01                      | 118      | 23654       |      |      |        |
|                             |             |               | $\alpha_3 = \frac{3\pi}{4}$ | 1     | 28-29            | 0.01                      | 55       | 13578       |      |      |        |
|                             |             |               | $\alpha_3 = \pi$            | 1     | 28-29            | 0.01                      | 53       | 11950       |      |      |        |
|                             |             |               | 3.9                         | 3-RRR | Inverse sing.    | —                         | 2        | 29-31       | 0.1  | 1182 | 242185 |
|                             |             |               |                             |       |                  | $\theta = -\pi$           | 1        | 28-29       | 0.01 | 65   | 9652   |
| $\theta = -\frac{3\pi}{4}$  | 1           | 28-29         |                             |       |                  | 0.01                      | 61       | 8828        |      |      |        |
| $\theta = -\frac{\pi}{2}$   | 1           | 28-29         |                             |       |                  | 0.01                      | 63       | 8725        |      |      |        |
| $\theta = -\frac{\pi}{4}$   | 1           | 28-29         |                             |       |                  | 0.01                      | 51       | 7748        |      |      |        |
| $\theta = 0$                | 1           | 28-29         |                             |       |                  | 0.01                      | 49       | 7419        |      |      |        |
| $\theta = \frac{\pi}{4}$    | 1           | 28-29         |                             |       |                  | 0.01                      | 46       | 7579        |      |      |        |
| $\alpha_3 = -\frac{\pi}{4}$ | 1           | 28-29         |                             |       |                  | 0.01                      | 15       | 6655        |      |      |        |
| $\alpha_3 = 0$              | 2           | 28-29         |                             |       |                  | 0.1                       | 489      | 106792      |      |      |        |
| $\alpha_3 = \frac{\pi}{4}$  | 1           | 28-29         |                             |       |                  | 0.01                      | 15       | 6653        |      |      |        |
| $\alpha_3 = \frac{\pi}{2}$  | 1           | 28-29         |                             |       |                  | 0.01                      | 18       | 9851        |      |      |        |
| $\alpha_3 = \frac{3\pi}{4}$ | 1           | 28-29         |                             |       |                  | 0.01                      | 12       | 5885        |      |      |        |
| $\alpha_3 = \pi$            | 2           | 28-29         |                             |       |                  | 0.1                       | 447      | 170170      |      |      |        |
| 3.11                        | Stewart     | Forward sing. |                             |       |                  | $\theta = -\frac{\pi}{4}$ | 1        | 22-23       | 0.01 | 9    | 9276   |
|                             |             |               |                             |       |                  | $\theta = 0$              | 1        | 22-23       | 0.01 | 15   | 14548  |
|                             |             |               | $\theta = \frac{\pi}{4}$    | 1     | 22-23            | 0.01                      | 10       | 9335        |      |      |        |
|                             |             | Inverse sing. | $\theta = -\frac{\pi}{4}$   | 1     | 22-23            | 0.01                      | 59       | 19906       |      |      |        |
|                             |             |               | $\theta = 0$                | 1     | 22-23            | 0.01                      | 66       | 18917       |      |      |        |
|                             |             |               | $\theta = \frac{\pi}{4}$    | 1     | 22-23            | 0.01                      | 51       | 19998       |      |      |        |
| 3.13-3.17                   | Double-loop | Forward sing. | const. orientation          | 2     | 25-27            | 0.02                      | 79       | 146420      |      |      |        |
|                             |             |               | const. position             | 2     | 37-39            | 0.25                      | 2554     | 195982      |      |      |        |
|                             |             | C-space       | —                           | 2     | 12-14            | 0.1                       | 31       | 85238       |      |      |        |
|                             |             | RI            | —                           | 1     | 19-20            | 0.01                      | 12       | 14903       |      |      |        |
|                             |             | RO            | —                           | 1     | 19-20            | 0.01                      | 12       | 12773       |      |      |        |
|                             |             | IO            | —                           | 1     | 19-20            | 0.01                      | 14       | 14906       |      |      |        |
|                             |             | II            | —                           | 1     | 19-20            | 0.01                      | 13       | 13062       |      |      |        |
|                             |             | RPM           | —                           | 0     | 19-18            | 0.01                      | 4        | 8           |      |      |        |
| IIM                         | —           | 0             | 21-20                       | 0.01  | 2                | 0                         |          |             |      |      |        |

Table 3.4: Performance data on the reported examples. For each set we indicate if the data refers to the computation of a slice or the whole set, the dimension of the computed set ( $dim$ ), the number of equations ( $N_{eq}$ ) and variables ( $N_{var}$ ) in its defining system, the accuracy threshold assumed ( $\sigma$ ), the time employed in computing the set in seconds, and the number of solution boxes returned by the method ( $N_{boxes}$ ).

# 4

## Workspace Determination

This chapter shows how the developments in Chapters 2 and 3 can be extended to compute the various workspaces of a manipulator. For a given workspace, we will show how its boundary can be determined by computing a set of generalised singularities, and then classifying the points of such set according to whether they constitute traversable or barrier singularities. A detailed map of the workspace is obtained as a result, where the interior and exterior regions, together with the motion impediments that separate them, get clearly identified.

The method is very general. In fact, it is equally applicable to both redundant and non-redundant manipulators, because the shape of the workspace depends only on the underlying mechanism, and not on the specific actuation pattern adopted.

After reviewing the state of the art on the topic (Section 4.1), we describe the notion of workspace mathematically, and then derive the conditions allowing to characterize the boundary of such workspace and the various motion impediments in its interior (Section 4.2). We then explain the main limitations of a continuation approach of a similar generality (Section 4.3), and the alternative method we propose, which overcomes such limitations (Section 4.4). We finally apply the proposed method to several manipulators, in cases that would be difficult to analyze using the continuation approach (Section 4.5).

### 4.1 Related Work

Since the early studies on the problem [74–78], efficient workspace determination methods have been given, but most of them are tailored to a particular robot architecture, or class of architectures. An important group of such methods adopt a constructive geometric approach. Representative of them is [79], which computes the constant-orientation workspace of a spatial parallel manipulator, [80], which extends the approach to deal with other physical constraints, or [68], which provides methods for various planar parallel platforms. Other

significant approaches include interval analysis or discretization techniques for Stewart-type manipulators [81, 82], optimization-based algorithms for fully serial or parallel robots [83], analytic methods for symmetrical spherical mechanisms [61], and analytic, topologic, or algebraic-geometric procedures for serial manipulators [84–88].

The literature on the topic is rich, and elaborate surveys can be found in [83, 89–91], but it should be noted that most previous methods, including [61, 68, 74–82, 84–88], are difficult to apply to manipulators outside the class they consider, because they exploit specificities of the class in some way or another, like the possibility to parametrize the end-effector pose [84, 92], simplifications introduced by dimension symmetries [61], or the feasibility of an algebraic [84–88] or geometric [68, 79, 80] treatment. Even discretization methods, which resort to configuration sampling only [78, 82, 93], rely on the assumption that either the forward or the inverse kinematic problems have a simple solution, which needs not be the case in general.

While ad-hoc methods are desirable because they tend to yield faster or simpler algorithms, methods for general structure manipulators are required too, to be able to analyze robots for which no specific solution exists. Up to the author’s knowledge, only one approach of a similar generality is available in the literature, due to Haug and collaborators [60, and references therein]. In their work, Haug et al. apply the idea that the workspace boundaries can be extracted from a set of generalised singularities, and focus their effort on tracing such singularities numerically using a continuation method. The procedure is elegant and works well in favorable situations, but important weaknesses of the method were identified in [94], including (1) the need to manually guide the method with a priori knowledge of the workspace shape, (2) the fact that only cross-sectional curves of the boundary can be traced in higher-dimensional cases, or (3) the possibility to miss some boundary segments in the presence of voids within the workspace. Unfortunately, the performance of the procedure can further degrade, since one can encounter workspaces with several connected components, hidden barriers, or degenerate barriers where the method will produce incomplete or misleading maps of the workspace (Section 4.3).

## 4.2 The Workspace and its Boundaries

The motion capabilities of a manipulator are typically assessed by analyzing the set of feasible poses of the end-effector, which is usually known as the workspace or accessible output set of the manipulator. However, the notion of workspace is more general. The regions swept by other configuration coordinates could equally be useful to the robot designer, including those of the input coordinates, which identify the motion ranges of the actuators, or those of a point on any link, which could reveal possible collisions with the environment. In this chapter, thus,

we will be interested in computing the set of attainable values of a generic vector  $\mathbf{u}$  of  $n_u$  coordinates, as the manipulator runs through all possible configurations. Note that such a vector will only coincide with the homonym vector of Chapter 2 when computing the workspace of the output coordinates. Moreover, since mechanical joint limits can greatly modify the shape of a workspace, we will take such limits into account in the formulation of Eq. (2.1), as explained in Section 4.4.1, and the configuration space  $\mathcal{C}$  of the manipulator will only contain points  $\mathbf{q}$  fulfilling such limits.

Depending on the choice, the  $\mathbf{u}$  coordinates might or might not be explicit in  $\mathbf{q}$ . However, we will show that it is always possible to derive a system of the form of Eq. (2.1) containing the  $\mathbf{u}$  variables of interest, so that  $\mathbf{q}$  will be assumed to include such variables hereafter. By adopting the partition  $\mathbf{q} = [\mathbf{z}^\top, \mathbf{u}^\top]^\top$ , where  $\mathbf{z}$  accumulates all coordinates in  $\mathbf{q}$  except those in  $\mathbf{u}$ , Eq. (2.1) can be written as

$$\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}, \quad (4.1)$$

and the workspace of the system relative to the  $\mathbf{u}$  coordinates can be defined as the set  $\mathcal{A}$  of points  $\mathbf{u} \in \mathcal{U}$  that satisfy Eq. (4.1) for some  $\mathbf{z}$ , where  $\mathcal{U}$  is here the whole space of the  $\mathbf{u}$  coordinates. We shall also assume that  $n_u \leq n$ , and that  $\mathcal{A}$  is a subset of  $\mathcal{U}$  of dimension  $n_u$ , which is the common situation in general. In particular, if  $n_z$  is the size of  $\mathbf{z}$ , and  $n_e$  is the number of equations in Eq. (2.1),  $n_u \leq n$  implies that  $n_z \geq n_e$ , so that the system in Eq. (4.1) is in general not overconstrained for a fixed value of  $\mathbf{u}$ .

Although a direct computation of  $\mathcal{A}$  could in principle be attempted, it is often more effective to delimit  $\mathcal{A}$  by computing its boundary, because such boundary is a set of lower dimension. A point  $\mathbf{u}$  lies on the boundary of  $\mathcal{A}$ , denoted  $\partial\mathcal{A}$ , if every open set of  $\mathcal{U}$  containing  $\mathbf{u}$  intersects the interior and the exterior of  $\mathcal{A}$ . We next derive the conditions allowing to identify the points  $\mathbf{u} \in \partial\mathcal{A}$ .

### 4.2.1 Jacobian Rank Deficiency Conditions

Observe first that the workspace  $\mathcal{A}$  is exactly the image of  $\mathcal{C}$  through  $\pi_{\mathbf{u}}$ , the projection map from  $\mathbf{q}$  onto the  $\mathbf{u}$  coordinates, i.e.,  $\pi_{\mathbf{u}}(\mathbf{z}, \mathbf{u}) = \mathbf{u}$ . Moreover, it can be seen that the Jacobian matrix  $\Phi_{\mathbf{z}} = [\partial\Phi_i/\partial z_j]$  must be rank deficient at the points  $\mathbf{q} \in \mathcal{C}$  that project onto some  $\mathbf{u} \in \partial\mathcal{A}$ . The reasoning is similar to the one used in Section 2.2.2 to geometrically interpret the forward and inverse singularities.

Note that if  $\Phi_{\mathbf{z}}$  is full rank at  $\mathbf{q} = [\mathbf{z}^\top, \mathbf{u}^\top]^\top \in \mathcal{C}$ , then there exists a non-vanishing  $n_e \times n_e$  minor of  $\Phi_{\mathbf{z}}$ , say relative to the variables  $\mathbf{z}'$ , and by the Implicit Function Theorem it is possible to find a function  $\mathbf{z}' = \mathbf{F}(\mathbf{u}')$  relating  $\mathbf{z}'$  with the remaining  $\mathbf{u}'$  variables, and satisfying  $\Phi(\mathbf{F}(\mathbf{u}'), \mathbf{u}') = \mathbf{0}$  [47]. Thus, the  $\mathbf{u}'$  variables, which include the  $\mathbf{u}$  ones, can be used as a local

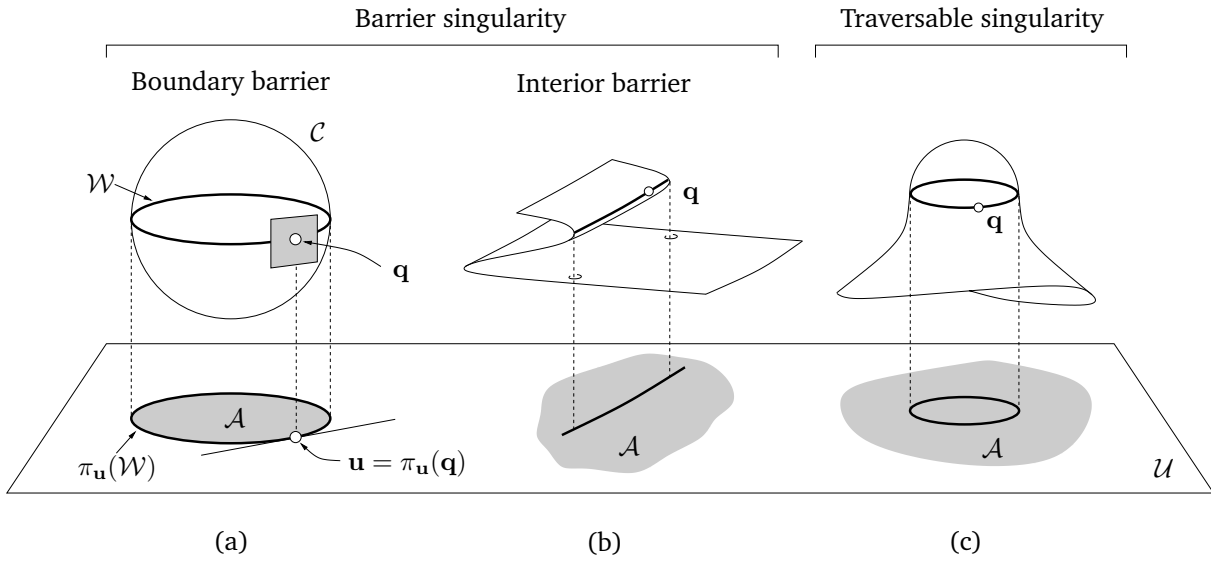


Figure 4.1: (a) Sets  $\mathcal{W}$  and  $\pi_{\mathbf{u}}(\mathcal{W})$  when  $\mathcal{Q} = \mathbb{R}^3$ ,  $\mathcal{U} = \mathbb{R}^2$ ,  $\mathcal{C}$  is the sphere  $x^2 + y^2 + z^2 = 1$ , and  $\pi_{\mathbf{u}}$  is the projection map  $f(x, y, z) = (x, y)$ . The workspace relative to the  $(x, y)$  coordinates is the projection of the sphere onto the  $(x, y)$  plane, and the boundaries of such projection necessarily correspond to points on the sphere where the tangent plane projects onto a line of  $\mathbb{R}^2$ . (b) and (c):  $\pi_{\mathbf{u}}(\mathcal{W})$  can also lie in the interior of  $\mathcal{A}$ .

parameterization of  $\mathcal{C}$  around  $(\mathbf{z}', \mathbf{u}')$ , implying that arbitrary values in a neighborhood of  $\mathbf{u}$  have a corresponding  $\mathbf{z}$  satisfying  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$ , so that  $\mathbf{u}$  must lie in the interior of  $\mathcal{A}$ . Thus,  $\Phi_{\mathbf{z}}$  must be rank-deficient for  $\mathbf{u}$  to belong to  $\partial\mathcal{A}$ .

Geometrically, the points  $\mathbf{q} \in \mathcal{C}$  where  $\Phi_{\mathbf{z}}$  is rank deficient are the critical points of the projection of  $\mathcal{C}$  onto  $\mathcal{U}$ , i.e., those in which the projection of the tangent space to  $\mathcal{C}$  does not span the tangent space of  $\mathcal{U}$  at  $\mathbf{u} = \pi_{\mathbf{u}}(\mathbf{q})$ , so that the manipulator loses instantaneous mobility relative to the  $\mathbf{u}$  variables. Since this result is quite similar to the one obtained for the forward or inverse singularities in Section 2.2.2, the set  $\mathcal{W}$  of all critical points of the projection of  $\mathcal{C}$  onto  $\mathcal{U}$  will be called the *singularity set* in this chapter. When  $\mathbf{u}$  coincides with the input or output coordinates, the set  $\mathcal{W}$  will include the forward or inverse singularities, respectively, but also other points corresponding to the loss of mobility introduced by the mechanical joint limits (Section 4.4.1).

A preliminary idea of how the workspace boundary would look like, thus, can be gained by computing  $\mathcal{W}$  and projecting it to the  $\mathbf{u}$  variables to obtain  $\pi_{\mathbf{u}}(\mathcal{W})$ . Fig. 4.1(a) illustrates the process on a simple example. However, note from the examples in Figs. 4.1(b) and 4.1(c) that the rank deficiency of  $\Phi_{\mathbf{z}}$  is a necessary but not sufficient condition for  $\pi_{\mathbf{u}}(\mathbf{q})$  to lie in  $\partial\mathcal{A}$ , as there can be critical points projecting on the interior of  $\mathcal{A}$  too. We next provide a way to

classify the points of the singularity set according to whether they actually correspond to motion impediments in the workspace.

### 4.2.2 Barrier Analysis

As illustrated in Fig. 4.1, points  $\mathbf{q}$  where  $\Phi_{\mathbf{z}}$  is rank deficient can be classified into two broad categories. They can be *traversable* or *barrier* singularities depending on whether there exists a trajectory in  $\mathcal{C}$  through  $\mathbf{q}$  whose projection on  $\mathcal{U}$  traverses  $\pi_{\mathbf{u}}(\mathcal{W})$  for each neighborhood of  $\mathbf{q}$  in  $\mathcal{C}$  [90,95]. Points corresponding to barrier singularities can in turn be classified as *boundary* or *interior* barriers, according to whether they occur over  $\partial\mathcal{A}$  or over the interior of  $\mathcal{A}$ , respectively. An example of each one of these singularities is depicted in Fig. 4.2 for a planar 3R manipulator.

As noted in [60,94,95], a workspace determination method should ideally detect *all* barriers in the workspace, either interior or on the boundary, as such barriers constitute true motion impediments with respect to the  $\mathbf{u}$  variables. A criterion to determine whether a point  $\mathbf{q}_0 = [\mathbf{z}_0^T, \mathbf{u}_0^T] \in \mathcal{W}$  corresponds to a barrier or a traversable singularity was given in [60], based on the following idea.

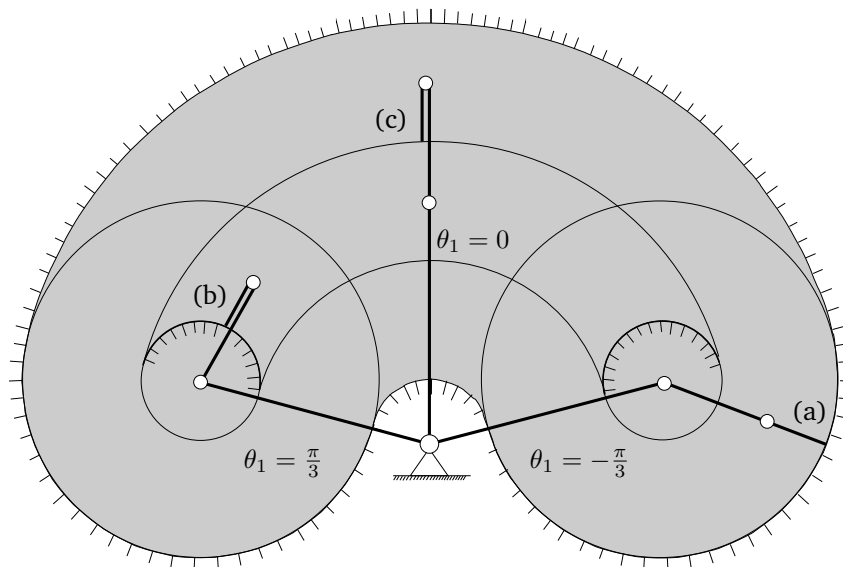


Figure 4.2: Workspace of a planar 3R manipulator relative to the  $(x, y)$  coordinates of the tip point of the last link, assuming that the angle  $\theta_1$  of the first revolute joint is restricted to the  $[-\pi/3, \pi/3]$  range. Points corresponding to singularities are indicated in solid lines, and those relative to boundary and interior barriers are indicated with normal vectors on the forbidden side. Configurations (a), (b), and (c) correspond to a boundary barrier, an interior barrier, and a traversable singularity, respectively.

Let  $\mathbf{q} = \mathbf{q}(\mathbf{v})$  be a smooth parameterization of  $\mathcal{C}$  in a neighborhood of  $\mathbf{q}_0 \in \mathcal{W}$ , where  $\Phi_{\mathbf{q}}(\mathbf{q}_0)$  is full rank,  $\mathbf{v}$  is a vector of  $n$  parameters, and  $\mathbf{q}_0 = \mathbf{q}(\mathbf{v}_0)$  for some  $\mathbf{v}_0 \in \mathbb{R}^n$ . Since  $\Phi_{\mathbf{q}}(\mathbf{q}_0)$  is full rank, the Implicit Function Theorem guarantees that such a parametrization exists [47]. Let also  $\mathbf{n}_0$  be the normal to  $\pi_{\mathbf{u}}(\mathcal{W})$  at  $\mathbf{u}_0$ , which can be computed as follows.

Assume that  $\mathbf{q}(t) = (\mathbf{z}(t), \mathbf{u}(t))$  is an arbitrary trajectory in  $\mathcal{C}$ , parametrized in  $t$ , traversing  $\mathbf{q}_0 = [\mathbf{z}_0^T, \mathbf{u}_0^T]^T$  for  $t = t_0$ . Making the substitution  $\mathbf{q} = \mathbf{q}(t)$  in  $\Phi(\mathbf{q}) = \mathbf{0}$ , and evaluating the time derivatives of this equation at  $\mathbf{q} = \mathbf{q}_0$  yields

$$\Phi_{\mathbf{u}}(\mathbf{q}_0)\dot{\mathbf{u}}(t_0) + \Phi_{\mathbf{z}}(\mathbf{q}_0)\dot{\mathbf{z}}(t_0) = \mathbf{0}. \quad (4.2)$$

Since  $\mathbf{q}_0 \in \mathcal{W}$ ,  $\Phi_{\mathbf{z}}$  must be rank deficient at  $\mathbf{q}_0$ , and there must exist a non-null vector  $\xi_0$  such that

$$\Phi_{\mathbf{z}}^T \xi_0 = \mathbf{0}. \quad (4.3)$$

Multiplying both sides of Eq. (4.2) by  $\xi_0^T$ , and taking into account Eq. (4.3), we get

$$\xi_0^T \Phi_{\mathbf{u}}(\mathbf{q}_0)\dot{\mathbf{u}}(t_0) = 0. \quad (4.4)$$

Because this holds for all vectors  $\dot{\mathbf{u}}(t_0)$  in the tangent space of  $\pi_{\mathbf{u}}(\mathcal{W})$  at  $\mathbf{u}_0$ , the normal vector  $\mathbf{n}_0$  can be computed as

$$\mathbf{n}_0 = \Phi_{\mathbf{u}}(\mathbf{q}_0)^T \xi_0. \quad (4.5)$$

In addition, it can be shown that the component of  $\dot{\mathbf{u}}(t_0)$  along  $\mathbf{n}_0$  must vanish, independently of the chosen trajectory  $\mathbf{q}(t)$ , so that  $\dot{\mathbf{u}}(t_0)$  must necessarily lie in the tangent space to  $\pi_{\mathbf{u}}(\mathcal{W})$  at  $\mathbf{u}_0$ . The component of  $\dot{\mathbf{u}}(t)$  along  $\mathbf{n}_0$  is given by

$$\dot{\mathbf{u}}_{\mathbf{n}_0} = \mathbf{n}_0^T \dot{\mathbf{u}}(t_0),$$

but using Eqs. (4.4) and (4.5) we realise that

$$\dot{\mathbf{u}}_{\mathbf{n}_0} = \xi_0^T \Phi_{\mathbf{u}}(\mathbf{q}_0)\dot{\mathbf{u}}(t_0) = 0,$$

on all possible trajectories  $\mathbf{q}(t)$  in  $\mathcal{C}$  through  $\mathbf{q}_0$ . In particular, this proves that there is a loss of mobility with respect to the  $\mathbf{u}$  variables at all points of  $\mathcal{W}$ .

Now, we can determine whether  $\mathbf{q}_0$  corresponds to a barrier singularity by examining the sign of

$$\chi(\mathbf{v}) = \mathbf{n}_0^T (\mathbf{u}(\mathbf{v}) - \mathbf{u}_0) \quad (4.6)$$

for all local trajectories parameterized in time  $\mathbf{v} = \mathbf{v}(t)$  crossing  $\mathbf{v}_0$  for some  $t = t_0$ , whose



corresponding path  $\mathbf{u} = \mathbf{u}(t)$  is orthogonal to  $\pi_{\mathbf{u}}(\mathcal{W})$  at  $\mathbf{u}_0$ . This can be done by resorting to the second-order Taylor expansion of  $\chi(\mathbf{v})$  around  $\mathbf{v}_0$

$$\chi(\mathbf{v}) \simeq \chi(\mathbf{v}_0) + \delta\mathbf{v}^\top \chi_{\mathbf{v}}(\mathbf{v}_0) + \frac{1}{2} \delta\mathbf{v}^\top \chi_{\mathbf{v}\mathbf{v}}(\mathbf{v}_0) \delta\mathbf{v},$$

where  $\chi_{\mathbf{v}}$  and  $\chi_{\mathbf{v}\mathbf{v}}$  are the gradient and Hessian of  $\chi(\mathbf{v})$ , and  $\delta\mathbf{v} = (\mathbf{v} - \mathbf{v}_0)$  is a small displacement whose corresponding  $\delta\mathbf{u} = (\mathbf{u} - \mathbf{u}_0)$  is orthogonal to  $\pi_{\mathbf{u}}(\mathcal{W})$ .

Note that the first term of the previous expansion vanishes because  $\chi(\mathbf{v}_0) = \mathbf{n}_0^\top (\mathbf{u}_0 - \mathbf{u}_0) = 0$ . Moreover, the time derivative of Eq. (4.6) for  $\mathbf{v} = \mathbf{v}(t)$  is

$$\dot{\chi}(t) = \mathbf{n}_0^\top \dot{\mathbf{u}}(t),$$

which is the component of  $\dot{\mathbf{u}}(t)$  along  $\mathbf{n}_0$  and, as shown above, must vanish irrespectively of the chosen trajectory  $\mathbf{v} = \mathbf{v}(t)$ . Thus, since for  $t = t_0$  it is  $\dot{\chi} = \chi_{\mathbf{v}} \dot{\mathbf{v}} = 0$  for all  $\dot{\mathbf{v}}$ , we conclude that  $\chi_{\mathbf{v}}(\mathbf{v}_0) = 0$  as well, meaning that the second term of the Taylor expansion also vanishes.

In conclusion, the sign of  $\chi(\mathbf{v})$  is mostly determined by the definiteness properties of the quadratic form  $\delta\mathbf{v}^\top \chi_{\mathbf{v}\mathbf{v}}(\mathbf{v}_0) \delta\mathbf{v}$ . If this form is positive- or negative-definite, then all trajectories orthogonal to  $\pi_{\mathbf{u}}(\mathcal{W})$  lie on one side of  $\pi_{\mathbf{u}}(\mathcal{W})$  and  $\mathbf{q}_0$  is a barrier singularity. If this form is indefinite, there are trajectories in  $\mathcal{A}$  that cross  $\pi_{\mathbf{u}}(\mathcal{W})$  and  $\mathbf{q}_0$  is a traversable singularity. Lastly, if this form is semi-definite, we cannot deduce the singularity type unless we examine higher-order terms of the Taylor expansion. However, the latter case only occurs on zero-measure subsets of  $\mathcal{W}$  generally.

The definiteness test just outlined can be implemented by checking the eigenvalues of the matrix form of  $\delta\mathbf{v}^\top \chi_{\mathbf{v}\mathbf{v}}(\mathbf{v}_0) \delta\mathbf{v}$ , as explained in [60].

## 4.3 Issues of Continuation Methods

In order to see the advantages of our approach in comparison to the continuation method in [60], this method is next reviewed briefly, identifying a number of cases where it fails to produce complete maps of the workspace.

We note first that the method in [60] relies on one-dimensional path tracking procedures, and hence it can only trace  $\partial\mathcal{A}$  explicitly on one-dimensional boundaries, i.e., when  $n_u = 2$ . Fig. 4.3(a) explains the method on a simple setting in which  $\mathcal{Q} = \mathbb{R}^3$ ,  $\mathcal{U} = \mathbb{R}^2$ , and  $\mathcal{C}$  contains two connected components,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , which project onto a workspace  $\mathcal{A}$  with two regions. The method begins with a manually-provided configuration of the manipulator  $\mathbf{q}_i = [\mathbf{z}_i^\top, \mathbf{u}_i^\top]^\top$ , with  $\mathbf{u}_i \in \mathcal{A}$ , shoots a ray through  $\mathbf{u}_i$  in  $\mathcal{U}$  on an arbitrary direction, and traces this ray until a point

on a barrier singularity is found. The movement along this ray is achieved by continuation of the corresponding trajectory on  $\mathcal{C}$ , i.e., by iteratively solving  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$  using a Newton method for  $\mathbf{u}$  fixed to discrete values along the ray. This process is repeated until a value of  $\mathbf{u}$  outside  $\mathcal{A}$  is found, which is detected because the Newton method fails to converge. A dichotomy-search process is then performed locally to find a point  $\mathbf{q}_b \in \mathcal{C}$  lying on  $\mathcal{W}$ . A second continuation process is then launched from  $\mathbf{q}_b$  to find the connected component of  $\mathcal{W}$  that is reachable from such point, by solving a system of equations that expresses the rank deficiency of  $\Phi_{\mathbf{z}}$ . Once  $\mathcal{W}$  has been found, the points of  $\pi_{\mathbf{u}}(\mathcal{W})$  are computed by projection, and those corresponding to barrier or traversable singularities are finally detected through the method outlined in the previous section.

Because path-tracking methods based on continuation are fast and robust to bifurcations [96], this approach will rapidly determine  $\mathcal{W}$  in favorable cases. However, one encounters the following situations in which the method will fail to identify  $\partial\mathcal{A}$  completely.

**Multi-component workspaces** Difficulties arise, for example, when computing  $\partial\mathcal{A}$  on workspaces with several connected components, like the one in Fig. 4.3(a). Independently of the chosen direction for the ray, note that the previous process will certainly hit  $\partial\mathcal{A}_1$ , but not  $\partial\mathcal{A}_2$ , because the tracking of the ray cannot continue beyond  $\partial\mathcal{A}_1$  using continuation. To converge to all boundary curves the previous strategy should at least be fed with one point on each connected component of  $\mathcal{C}$ , but no satisfactory method has been given yet to compute such points in general, to the best of our knowledge.

**Hidden barriers** The continuation method may seem to be able to compute, at least, all barriers of the workspace component to which  $\mathbf{q}_i$  belongs, but this is not always the case. Note that  $\mathcal{W}$  may itself have several connected components, and some of such components could be missed depending on the position of  $\mathbf{u}_i$ , even if rays on all possible directions were shot. In Fig. 4.3(b), left, for example, the continuation method may be able to find  $\partial\mathcal{A}_1$  and  $\partial\mathcal{A}_3$  from  $\mathbf{u}_i$ , but not  $\partial\mathcal{A}_2$ , because  $\partial\mathcal{A}_2$  is hidden behind  $\partial\mathcal{A}_1$ . The problem also arises in workspaces with interior barriers, as seen in Fig. 4.3(b), right. Starting the continuation from  $\mathbf{q}_i$  allows hitting the boundary barrier corresponding to  $\mathcal{W}_3$ , but not the interior barriers corresponding to  $\mathcal{W}_1$  and  $\mathcal{W}_2$ , thus ignoring true obstacles lying inside the workspace.

**Degenerate barriers** When  $\partial\mathcal{A}$  has dimension two or higher the continuation method slices  $\partial\mathcal{A}$  through hyperplanes  $\mathcal{H}_i$  in order to obtain one-dimensional curves  $\partial\mathcal{A}_i$  trackable by the method [Fig. 4.3(c), left]. On manipulators of a special geometry, however, portions of  $\partial\mathcal{A}$  can degenerate into lower-dimensional barriers  $\mathcal{D}$ , thus making the slices  $\mathcal{H}_i$  contain isolated

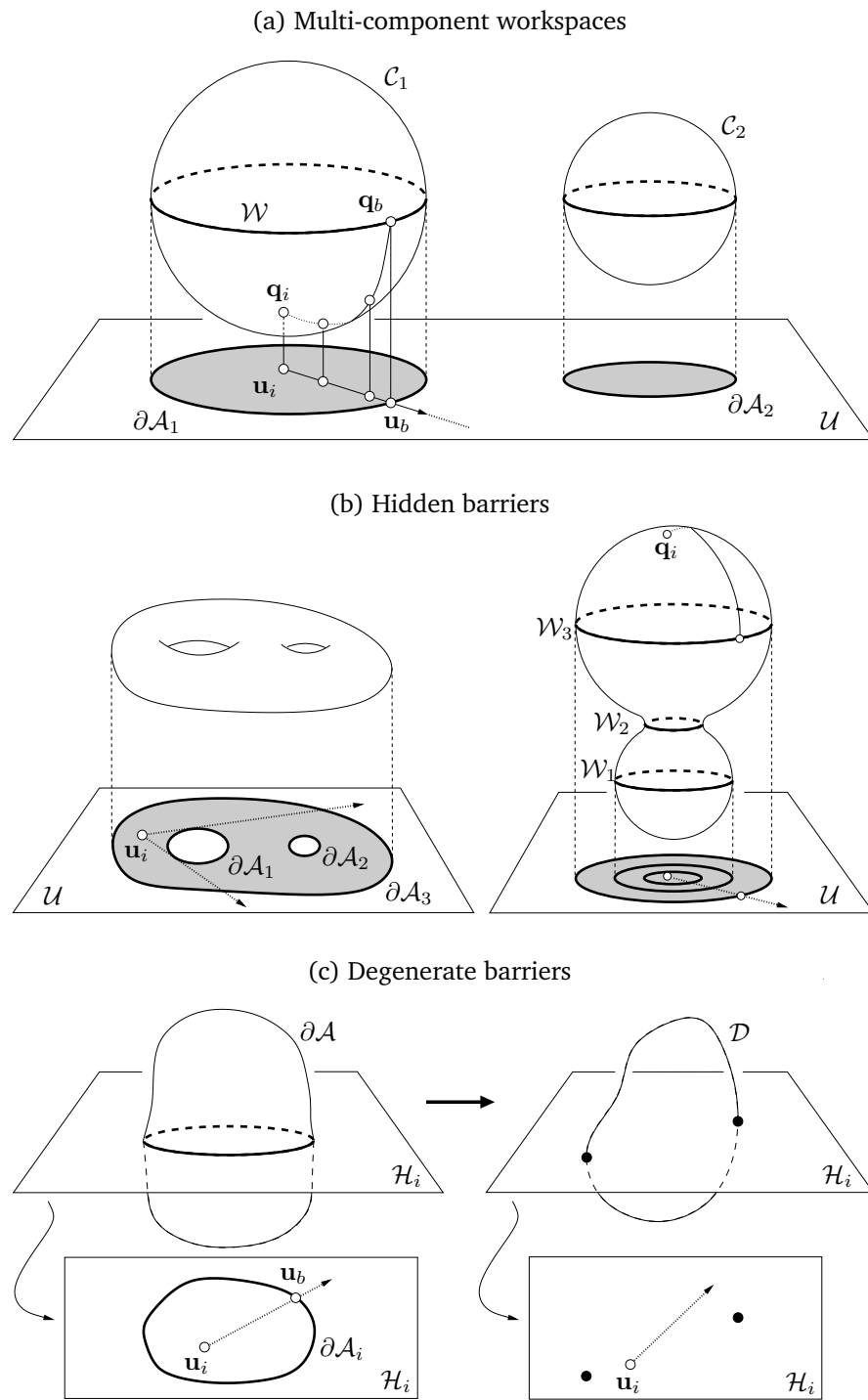


Figure 4.3: Performance of the continuation method in multi-component workspaces (a), hidden regions (b), and degenerate barriers (c).

points only [Fig. 4.3(c), right]. The method will clearly miss the lower-dimensional barriers in such a situation because the ray shooting technique will fail to meet them with probability one, independently of the location of  $\mathbf{u}_i$ .

As it will be shown in Section 4.5, examples of multi-component workspaces, hidden barriers, and degenerate barriers occur easily on real manipulators and, thus, there is a clear need to develop alternative methods that are robust to such situations. Next section provides one using the results of Section 4.2.

## 4.4 An Alternative Approach

The proposed method exploits the same machinery introduced in Chapter 3 to isolate the different singularity sets. It entails modelling the joint limits as equality constraints (Section 4.4.1), then deriving a system of quadratic equations characterizing the set  $\mathcal{W}$  under consideration (Section 3.3), and finally computing a box approximation of  $\mathcal{W}$  of sufficient resolution, in order to classify the points of such set into boundary barriers, interior barriers, or traversable singularities (Section 4.4.3).

### 4.4.1 Joint Limit Constraints

Although mechanical limits on the joints are usually described by inequalities, in our case it is easier to accommodate them by using equality constraints. Two types of limits need to be treated: those referring to angular constraints and those referring to distance constraints. Typically, these arise on revolute and prismatic joints, respectively, but combinations of both limits may be encountered on other types of joints.

Following the formulation of Section 3.2, let  $L_j$  and  $L_k$  be two links connected through a revolute joint  $J_i$ . The relative angle between  $L_j$  and  $L_k$ ,  $\phi_i$ , is defined as the angle between two unit vectors  $\mathbf{m}_i$  and  $\mathbf{n}_i$  orthogonal to  $J_i$ 's axis, moving rigidly with  $L_j$  and  $L_k$  respectively. Suppose that we want to limit  $\phi_i$  to lie in the interval  $[-\phi_i^l, \phi_i^l]$ . We will take these bounds into account by limiting the range of the cosine of  $\phi_i$ , since the previous constraint is equivalent to  $\cos \phi_i \geq \cos \phi_i^l$ . Note for this that, if  $c_i = \cos(\phi_i)$ , then  $c_i$  is related to  $\mathbf{m}_i$  and  $\mathbf{n}_i$  through

$$c_i = \mathbf{m}_i^\top \mathbf{n}_i. \quad (4.7)$$

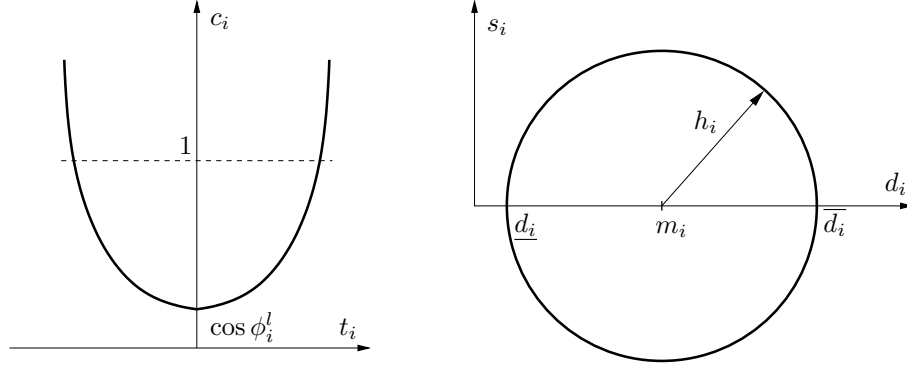


Figure 4.4: Joint limit constraints for angular and distance variables.

Moreover, the fact that  $\mathbf{m}_i$  and  $\mathbf{n}_i$  are fixed in  $\mathcal{F}_j$  and  $\mathcal{F}_k$ , respectively, implies that

$$\mathbf{m}_i = \mathbf{R}_j \mathbf{m}_i^{\mathcal{F}_j}, \quad (4.8)$$

$$\mathbf{n}_i = \mathbf{R}_k \mathbf{n}_i^{\mathcal{F}_k}. \quad (4.9)$$

Thus, to limit  $\phi_i$  we can simply add Eqs. (4.7)–(4.9) to the system to be solved, together with

$$c_i = t_i^2 + \cos \phi_i^l, \quad (4.10)$$

where  $t_i$  is a new auxiliary variable that can take any value [Fig. 4.4(left)]. The constraint  $\phi_i \in [-\phi_i^l, \phi_i^l]$  is satisfied if, and only if, Eq. (4.10) is satisfied for some  $t_i$ . The same equations can be used to limit the angle on universal and spherical joints, where the vectors  $\mathbf{m}_i$  and  $\mathbf{n}_i$  can be chosen as shown in Fig. 4.5.

If  $L_j$  and  $L_k$  are connected through a prismatic joint  $J_i$ , the displacement  $d_i$  of the joint is already present in the system as a variable. Suppose that we want to constrain  $d_i$  to lie within the interval  $[\underline{d}_i, \bar{d}_i]$ . In this case we can already impose this constraint by setting

$$(d_i - m_i)^2 + s_i^2 = h_i^2, \quad (4.11)$$

where  $m_i = \frac{\bar{d}_i + \underline{d}_i}{2}$ ,  $h_i = \frac{\bar{d}_i - \underline{d}_i}{2}$ , and  $s_i$  is a newly-defined auxiliary variable. The values  $m_i$  and  $h_i$  are called the mid-point and half-range of the interval  $[\underline{d}_i, \bar{d}_i]$ , and Eq. (4.11) simply constrains the pairs  $(d_i, s_i)$  to take values on a circle of radius  $h_i$  centered at  $(m_i, 0)$  in the  $(d_i, s_i)$  plane [Fig. 4.4(right)]. As a consequence,  $d_i \in [\underline{d}_i, \bar{d}_i]$  if, and only if, Eq. (4.11) is satisfied for some  $s_i$ .

When the necessary equations modelling the joint limits are introduced, the newly-defined variables  $t_i$  and  $s_i$  become part of  $\mathbf{q}$ , and Eq. (2.1) characterizes the C-space attainable by the manipulator taking into account joint limits.

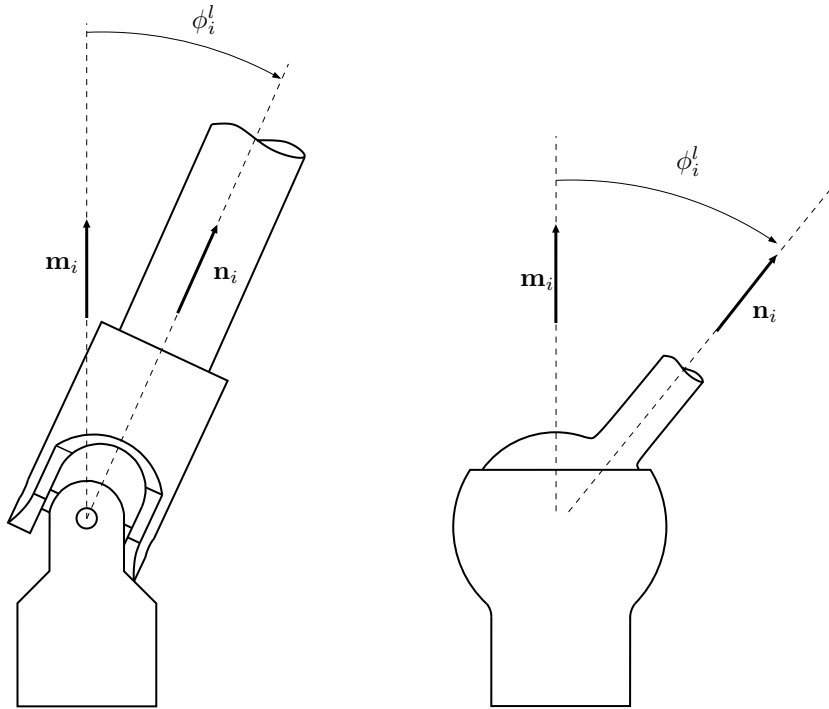


Figure 4.5: Vectors delimiting the mechanical limits on universal and spherical joints.

#### 4.4.2 Equations of the Singularity Set

To explain how a system of equations characterizing  $\mathcal{W}$  can be derived, we shall distinguish two situations, depending on whether the  $\mathbf{u}$  variables appear explicitly in  $\mathbf{q}$ , or are only determined implicitly by some of the variables in  $\mathbf{q}$ , both illustrated in Section 4.5 on particular examples. In all cases, it will be assumed that the assembly constraints in Eq. (2.1) have been formulated following Section 3.2, where the pose of the links appear as  $(\mathbf{r}_i, \mathbf{R}_i)$ , where  $\mathbf{r}_i$  and  $\mathbf{R}_i$  respectively encompass the Cartesian coordinates of a point on the link, and the components of a rotation matrix giving the orientation of the link. Note that the equations added to take the joint limits into account are quadratic, so the whole formulation of Eq. (2.1) remains quadratic too.

#### Dealing with Explicit Workspace Variables

Assume initially that  $\mathbf{q}$  explicitly contains  $\mathbf{u}$ . This occurs usually in positional workspaces, i.e., when  $\mathbf{u}$  includes part or all of the variables of some  $\mathbf{r}_i$ . By adopting the partition  $\mathbf{q} = [\mathbf{z}^T, \mathbf{u}^T]^T$ ,  $\mathcal{W}$  is the set of points satisfying  $\Phi(\mathbf{z}, \mathbf{u}) = \mathbf{0}$  for which  $\Phi_{\mathbf{z}}$  is rank deficient, i.e. the set of points

$\mathbf{q}$  that satisfy

$$\left. \begin{aligned} \Phi(\mathbf{z}, \mathbf{u}) &= \mathbf{0} \\ \Phi_{\mathbf{z}}^T \xi &= \mathbf{0} \\ \xi^T \xi &= 1 \end{aligned} \right\}, \quad (4.12)$$

for some  $\xi$ , where  $\xi$  is an  $n_e$ -dimensional vector of unknowns, and  $\Phi_{\mathbf{z}}$  is transposed since, if  $n_u < n$ , it is a rectangular matrix with more columns than rows. Clearly, the first equation in (4.12) constrains  $\mathbf{q}$  to be a valid configuration respecting the joint limits, and the second and third equations impose the rank deficiency of  $\Phi_{\mathbf{z}}$ . Since (2.1) is quadratic, (4.12) will be quadratic too, because all entries in  $\Phi_{\mathbf{z}}$  will be linear terms, and  $\xi^T \xi$  is directly a quadratic expression. Thus, a system characterizing  $\mathcal{W}$  adopts the form of (4.12) in this case.

### Dealing with Implicit Workspace Variables

There are situations in which the chosen  $\mathbf{u}$  variables do not all intervene in  $\mathbf{q}$  but, instead, they can be related to a subset  $\tilde{\mathbf{u}}$  of  $n_{\tilde{\mathbf{u}}}$  variables in  $\mathbf{q}$ , through a smooth function of the form

$$\tilde{\mathbf{u}} = \mu(\mathbf{u}). \quad (4.13)$$

This occurs whenever  $\mathbf{u}$  contains orientation angles of some link, for instance. Since the formulation of the assembly constraints represents orientations by rotation matrices, orientation angles are only related implicitly to the components of  $\mathbf{R}_i$  through a parametrization of the Special Orthogonal Group under consideration [ $SO(2)$  or  $SO(3)$ , depending on whether the manipulator is planar, or spatial]. In order to transform (2.1) into the form of (4.1) it will be possible, in such situations, to consider the partition

$$\mathbf{q} = [\tilde{\mathbf{z}}^T, \tilde{\mathbf{u}}^T]^T,$$

and subdivide (2.1) into two subsystems as follows

$$\left. \begin{aligned} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) &= \mathbf{0} \\ \eta(\tilde{\mathbf{u}}) &= \mathbf{0} \end{aligned} \right\}, \quad (4.14)$$

where  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  is a subsystem of equations whose solution set can be globally parametrized by (4.13), and  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0}$  collects the rest of equations. Since  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  parametrizes the solution set of  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$ ,  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  can be replaced by  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  in (4.14), obtaining the

equivalent system

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \tilde{\mathbf{u}} = \mu(\mathbf{u}) \end{array} \right\}, \quad (4.15)$$

which now contains  $\mathbf{u}$  explicitly. Therefore, (4.1) adopts the form of (4.15) in this case, with  $\mathbf{z} = [\tilde{\mathbf{z}}^\top, \tilde{\mathbf{u}}^\top]^\top$ , and

$$\Phi(\mathbf{z}, \mathbf{u}) = \begin{bmatrix} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) \\ \tilde{\mathbf{u}} - \mu(\mathbf{u}) \end{bmatrix}, \quad (4.16)$$

so that  $\mathcal{W}$  will be the set of points  $\mathbf{q} = [\mathbf{z}^\top, \mathbf{u}^\top]^\top$  satisfying (4.15) for which  $\Phi_{\mathbf{z}}$  is rank deficient.

Note however that, because of the form of (4.16),  $\Phi_{\mathbf{z}}$  has the block structure

$$\Phi_{\mathbf{z}} = \begin{bmatrix} \Psi_{\tilde{\mathbf{z}}} & \Psi_{\tilde{\mathbf{u}}} \\ \mathbf{0} & \mathbf{I}_{n_{\tilde{\mathbf{u}}}} \end{bmatrix}$$

in this case, where  $\mathbf{I}_{n_{\tilde{\mathbf{u}}}}$  is the  $n_{\tilde{\mathbf{u}}} \times n_{\tilde{\mathbf{u}}}$  identity matrix, so that  $\Phi_{\mathbf{z}}$  will be rank deficient if, and only if, its upper-left block  $\Psi_{\tilde{\mathbf{z}}}$  is rank deficient. Hence,  $\mathcal{W}$  can be characterized as the set of points  $\mathbf{q} = [\tilde{\mathbf{z}}^\top, \tilde{\mathbf{u}}^\top, \mathbf{u}^\top]^\top$  that satisfy

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \tilde{\mathbf{u}} = \mu(\mathbf{u}) \\ \Psi_{\tilde{\mathbf{z}}}^\top \tilde{\xi} = \mathbf{0} \\ \tilde{\xi}^\top \tilde{\xi} = 1 \end{array} \right\} \quad (4.17)$$

for some  $\tilde{\xi}$ , where  $\tilde{\xi}$  is a new vector with the appropriate size.

While we could now proceed to isolate  $\mathcal{W}$  by solving (4.17),  $\mu(\mathbf{u})$  usually introduces trigonometric terms that complicate the solution. Fortunately, since  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  parametrizes the solution set of  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$ , and the  $\mathbf{u}$  variables only intervene in the second equation of (4.17), we can substitute  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  by  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  in (4.17), arriving at the equivalent system

$$\left. \begin{array}{l} \Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0} \\ \eta(\tilde{\mathbf{u}}) = \mathbf{0} \\ \Psi_{\tilde{\mathbf{z}}}^\top \tilde{\xi} = \mathbf{0} \\ \tilde{\xi}^\top \tilde{\xi} = 1 \end{array} \right\}, \quad (4.18)$$

which is quadratic, because  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}})$  and  $\eta(\tilde{\mathbf{u}})$  are quadratic under the adopted formulation. Thus,  $\mathcal{W}$  is characterized by Eq. (4.18) in this case.



### 4.4.3 Equation Solution and Boundary Identification

As we have seen, Eqs. (4.12) and (4.18) can be both written as a quadratic system of equations and, therefore, they can be solved using the same numerical method described in Section 3.3. To construct the initial box bounding the location of their solutions we only have to take into account that:

- The variables  $c_i$  only take values in  $[-1, 1]$  because they represent the cosine of some angle.
- The range of the variables  $t_i$  can be set to  $[0, \sqrt{1 - \cos \phi_i^l}]$  [Fig. 4.4(left)].
- The range of the variables  $s_i$  can be set to  $[0, h_i]$  [Fig. 4.4(right)].
- Being unit vectors, the components of  $\mathbf{m}_i$  and  $\mathbf{n}_i$  will lie in  $[-1, 1]$ .

As a result, the method will deliver a collection  $B$  of boxes tightly enclosing the solution set of the system. Using  $B$ , it is straightforward to obtain a box approximation  $B^{\mathcal{W}}$  of the singular set  $\mathcal{W}$ . If  $B$  is obtained by solving Eq. (4.12), then  $\mathbf{z}$  and  $\mathbf{u}$  explicitly intervene as part of the variables in the system, and each box in  $B$  already has ranges along the  $\mathbf{q} = [\mathbf{z}^T, \mathbf{u}^T]^T$  dimensions. Such ranges define a box in  $\mathbf{q}$ -space enclosing points of  $\mathcal{W}$ , and the collection of all of such boxes directly provides  $B^{\mathcal{W}}$ . If  $B$  is instead obtained from Eq. (4.18), the  $\mathbf{u}$  variables do not intervene in the variables of the system. However, for each box in  $B$  we can consider the ranges along the  $\tilde{\mathbf{u}}$  variables, and derive corresponding ranges for the  $\mathbf{u}$  variables by solving  $\tilde{\mathbf{u}} = \mu(\mathbf{u})$  using interval or linear relaxation techniques [31, 97].

Once  $B^{\mathcal{W}}$  has been obtained, it remains to check whether the points of  $\mathcal{W}$  enclosed in  $B^{\mathcal{W}}$  correspond to boundary barriers, interior barriers, or traversable singularities. This classification is performed in two stages, illustrated in Fig. 4.6.

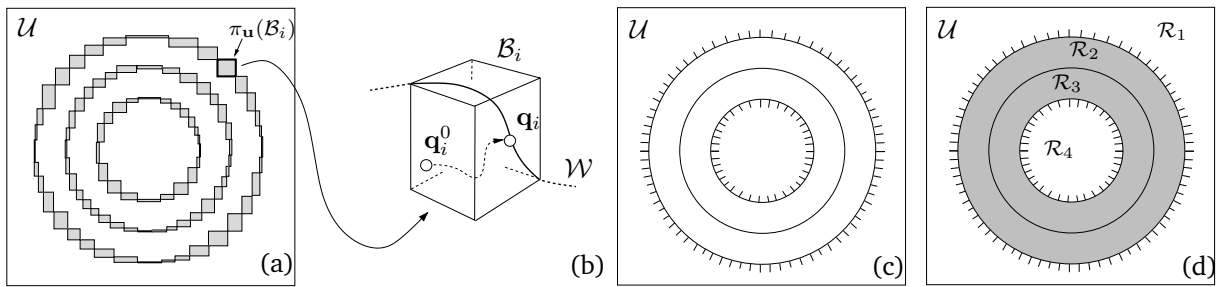


Figure 4.6: Boundary identification process. (a) Box approximation of  $\mathcal{W}$  projected onto the output space  $\mathcal{U}$ . (b) Computation of  $\mathbf{q}_i \in \mathcal{W}$  for each box  $B_i$ . (c) Classification of the points of  $\pi_{\mathbf{u}}(\mathcal{W})$  into barrier or traversable singularities. (d) Regions into which  $\pi_{\mathbf{u}}(\mathcal{W})$  subdivides  $\mathcal{U}$  and their classification into interior (grey) or exterior (white) regions.

In a first stage, we classify the boxes of  $B^{\mathcal{W}}$  according to whether they enclose barrier or traversable singularities. For each box  $\mathcal{B}_i \in B^{\mathcal{W}}$  [Fig. 4.6(a)], a point  $\mathbf{q}_i \in \mathcal{W}$  is computed [Fig. 4.6(b)], and the barrier determination method mentioned in Section 4.2.2 is applied to this point. The computation of  $\mathbf{q}_i$  is done by solving (4.12) or (4.18), depending on the situation, using a Newton-Raphson method starting from a point  $\mathbf{q}_i^0$  taken from an arbitrary point  $\mathbf{x}_i^0$  inside  $\mathcal{B}_i$ . This procedure will quadratically converge to some point  $\mathbf{q}_i \in \mathcal{W}$  provided that the points within  $\mathcal{B}_i$  are close enough to  $\mathcal{W}$ , which can be guaranteed by computing  $B^{\mathcal{W}}$  using a small-enough  $\sigma$  threshold (Section 3.3.3). The singularity type obtained for  $\mathbf{q}_i$  (either barrier or traversable singularity) is taken as an estimation of the singularity type of all points in  $\mathcal{B}_i \cap \mathcal{W}$ , so that, after repeating this process for all boxes in  $B^{\mathcal{W}}$ , it is possible to subdivide  $\mathcal{W}$  into subsets of constant singularity type. If  $\mathbf{q}_i$  is a barrier singularity, a normal vector  $\mathbf{b}_i$  pointing towards the forbidden side of the barrier is drawn at  $\mathbf{u}_i = \pi_{\mathbf{u}}(\mathbf{q}_i)$  [Fig. 4.6(c)].

In a second stage, we determine which of the barrier points  $\mathbf{q}_i$  computed in the previous stage correspond to boundary or interior barriers. To this end, notice that  $\pi_{\mathbf{u}}(\mathcal{W})$  subdivides  $\mathcal{U}$  into several regions  $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$ , where each region fully lies in the interior or in the exterior of  $\mathcal{A}$ , and a barrier point  $\mathbf{u}_i$  will lie on  $\partial\mathcal{A}$  if, and only if, one of its two neighboring regions is exterior to  $\mathcal{A}$ . Thus, determining which of the barrier points  $\mathbf{u}_i$  correspond to boundary barriers boils down to checking whether the regions  $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$  are interior or exterior to  $\mathcal{A}$ . The type of a region  $\mathcal{R}_j$  can be determined by selecting a point  $\mathbf{u}_j$  in the region, and solving  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$ , which is here done by resorting to the same numerical technique proposed. If  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$  has at least one solution, then  $\mathcal{R}_j$  is an interior region, otherwise it is exterior.

While solving  $\Phi(\mathbf{z}, \mathbf{u}_j) = \mathbf{0}$  can be costly, note that it is not necessary to apply this test to most regions because the type of a region can often be decided by noting that:

- If  $\mathbf{u}$  only contains position coordinates of the end-effector, then the outer region will necessarily be exterior to  $\mathcal{A}$ , because the effector can only reach a bounded set of positions in practice.
- A region  $\mathcal{R}_j$  whose boundary contains a traversable singularity can be marked as interior, because  $\mathcal{R}_j$  contains trajectories that enter  $\mathcal{R}_j$  through that singularity.
- A region  $\mathcal{R}_j$  whose boundary contains a barrier point  $\mathbf{u}_i$  with  $\mathbf{b}_i$  pointing outwards from  $\mathcal{R}_j$  can be marked as interior as well, because such barrier indicates that there are feasible trajectories in  $\mathcal{C}$  projecting inside  $\mathcal{R}_j$ .

In Fig. 4.6(d), for example, these observations allow identifying  $\mathcal{R}_1$  as an exterior region if  $\mathbf{u}$  only contains position coordinates, and  $\mathcal{R}_2$  and  $\mathcal{R}_3$  as interior regions. Only the type of  $\mathcal{R}_4$  needs to be disambiguated by checking a point in the region.

| Manipulator    | Workspace      | $d$ | $n_v$ | $n_c$ | $n_s$   | $t_s$ |
|----------------|----------------|-----|-------|-------|---------|-------|
| D. Butterfly   | reachable      | 1   | 37    | 36    | 145369  | 218   |
| Stewart platf. | const.-orient. | 2   | 27    | 25    | 1677906 | 45    |
| 3-UPS/S platf. | orientation    | 2   | 21    | 19    | 156699  | 30    |
| Agile Eye      | orientation    | 1   | 21    | 19    | 100636  | 17    |

Table 4.1: Performance data on computing the considered workspaces assuming a value of  $\sigma = 0.1$ .

## 4.5 Examples

We next illustrate the performance of the method on various situations. Several representative workspaces on planar and spatial manipulators are computed to emphasize the generality of the approach, and to encounter cases of multi-component workspaces, hidden barriers, and degenerate barriers like those described in Section 4.3.

As in Chapter 3, all computations have been executed using a parallelized version of the method implemented in C using the libraries of the CUIK Suite [65] on a grid computer of Xeon processors able to run 160 threads in parallel. For each experiment, Table 4.1 provides the dimension of  $\partial\mathcal{A}$  ( $d$ ), the number of variables ( $n_v$ ) and equations ( $n_c$ ) intervening in (4.12) or (4.18), the amount of CPU time required to solve it ( $t_s$ , in minutes), and the number of solution boxes returned ( $n_s$ ), assuming  $\sigma = 0.1$  in all cases.

### 4.5.1 Multi-component Workspaces

To illustrate the performance of our approach on complex multi-loop linkages, and to give one example of a multi-component workspace, we apply our method to compute the reachable workspace of a planar mobility-three double-butterfly linkage (Fig. 4.7). A version of this mechanism has been used to compare the performance of general position analysis methods [30, 98, 99], but no complete method has been given to compute the boundaries and interior barriers of its reachable workspace yet, as far as we know.

For this example it will be assumed that the end-effector is the upper left body in Fig. 4.7, whose pose is determined by the position vector of point  $P$  and the  $2 \times 2$  rotation matrix of angle  $\theta_1$ , and that two slider joints are mounted to let lengths  $l_5$  and  $l_7$  vary within the ranges  $[11, 13]$  and  $[10, 12]$ , respectively. As for the parameters in Fig. 4.7, we adopt the same values considered in [30, 98, 99]. Namely,  $a_0 = 7$ ,  $a_1 = 7$ ,  $a_2 = 5$ ,  $b_0 = 13$ ,  $b_1 = 6$ ,  $b_2 = 3$ ,  $l_3 = 7$ ,  $l_4 = 9$ ,  $a_6 = 3$ ,  $b_6 = 2$ , and  $\gamma_0 = 36.87^\circ$ ,  $\gamma_1 = 22.62^\circ$ ,  $\gamma_2 = 53.13^\circ$ ,  $\gamma_6 = 36.87^\circ$ , where  $\gamma_i$  is the acute angle between segments  $a_i$  and  $b_i$ . Then, for this manipulator, Eq. (4.1) can be formulated using the following equations:

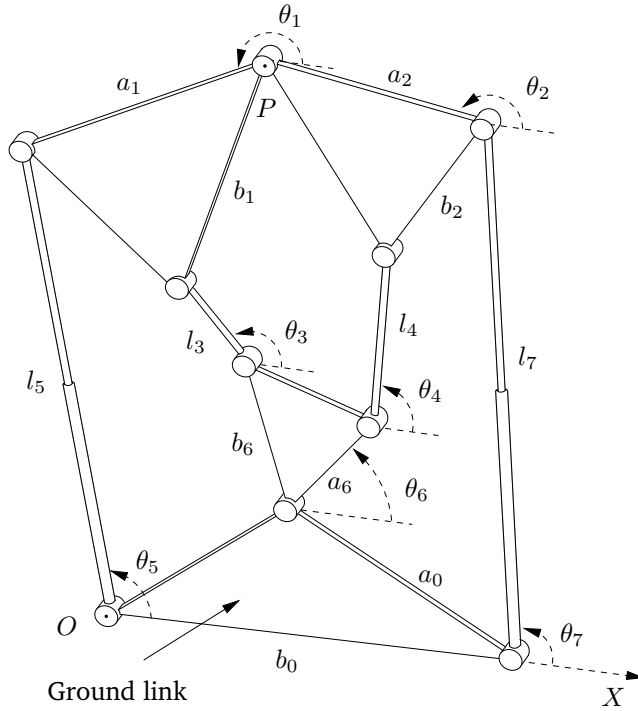


Figure 4.7: The planar double-butterfly linkage with variable lengths  $l_5$  and  $l_7$ . The fixed frame is centered at  $O$  with the  $X$ -axis aligned as indicated, and all angles  $\theta_i$  are measured relative to such axis.

1. The loop equations enforcing the closure of the three loops that leave the ground link via  $l_7$ , and return via  $l_4$ ,  $l_3$  and  $l_5$  [30]:

$$0 = l_7 c_7 + b_2 c_2 c_{\gamma_2} - b_2 s_2 s_{\gamma_2} - l_4 c_4 - a_6 c_6 + a_0 c_{\gamma_0},$$

$$0 = l_7 s_7 + b_2 s_2 c_{\gamma_2} + b_2 c_2 s_{\gamma_2} - l_4 s_4 - a_6 s_6 - a_0 s_{\gamma_0},$$

$$0 = l_7 c_7 + a_2 c_2 + a_1 c_1 - l_5 c_5 + b_0,$$

$$0 = l_7 s_7 + a_2 s_2 + a_1 s_1 - l_5 s_5,$$

$$0 = l_7 c_7 + a_2 c_2 + b_1 c_1 c_{\gamma_1} - b_1 s_1 s_{\gamma_1} - l_3 c_3 - b_6 c_6 c_{\gamma_6} + b_6 s_6 s_{\gamma_6} + a_0 c_{\gamma_0},$$

$$0 = l_7 s_7 + a_2 s_2 + b_1 s_1 c_{\gamma_1} + b_1 c_1 s_{\gamma_1} - l_3 s_3 - b_6 s_6 c_{\gamma_6} - b_6 c_6 s_{\gamma_6} - a_0 s_{\gamma_0},$$

where  $c_{\gamma_i}$  and  $s_{\gamma_i}$  stand for the cosine and sine of  $\gamma_i$ , and  $c_i$  and  $s_i$  for those of  $\theta_i$ .

2. The equations providing the  $x$  and  $y$  coordinates of  $P$  relative to the fixed  $OXY$  frame:

$$\begin{aligned}x &= b_0 + l_7 c_7 + a_2 c_2, \\y &= l_7 s_7 + a_2 s_2.\end{aligned}$$

3. The circle equations constraining  $c_i$  and  $s_i$ :

$$c_i^2 + s_i^2 = 1.$$

4. The joint limit constraints for  $l_5$  and  $l_7$

$$(l_i - m_i)^2 + t_i^2 = h_i^2,$$

where  $m_i$  and  $h_i$  are the mid-point and half-range of the intervals for  $l_5$  and  $l_7$ .

The reachable workspace is defined as the set of attainable locations for a point on the end-effector, e.g.  $P$  in our case. Thus  $\mathbf{u} = [x, y]^T$  for this workspace, and since  $x$  and  $y$  are explicit in the previous equations, we are in the situation of Eq. (4.12). Moreover, since  $n_u = 2$ , the boundary of the reachable workspace will be one-dimensional in general. The proposed method computes the box approximation of  $\pi_{\mathbf{u}}(\mathcal{W})$  shown in Fig. 4.8 in this case, which delimits three workspace areas corresponding to different assembly modes of the mechanism. The result of the boundary identification process on one of such areas is shown in Fig. 4.9. Note that, having several connected components, this workspace would be difficult to map out entirely using the method in [60].

#### 4.5.2 Hidden Barriers

To show the performance of the method on spatial mechanisms, and to encounter workspaces with hidden barriers, we next apply the method to the Stewart platform (Fig. 3.10). This is a challenging test case for any workspace determination method. Its full workspace is six-dimensional, and its boundary five-dimensional, which hinders any attempt of computing it exhaustively, due to the curse of dimensionality. For this reason, and because six-dimensional spaces are impossible to visualize directly in three dimensions, comprehension on this workspace can be gained by obtaining lower-dimensional workspaces like (1) the *constant orientation* workspace, or set of attainable locations by a point  $P$  on the platform for a fixed platform orientation [6, 100], (2) the *constant position* workspace, or set of platform orientations for a fixed position of  $P$  [25, 82, 101, 102], and (3) the *reachable* workspace, or set of locations that  $P$  can attain with at least one platform orientation [103, 104]. All of these workspaces can be



Figure 4.8: Box approximation of the set  $\pi_u(\mathcal{S})$  corresponding to the reachable workspace of the linkage in Fig. 4.7.

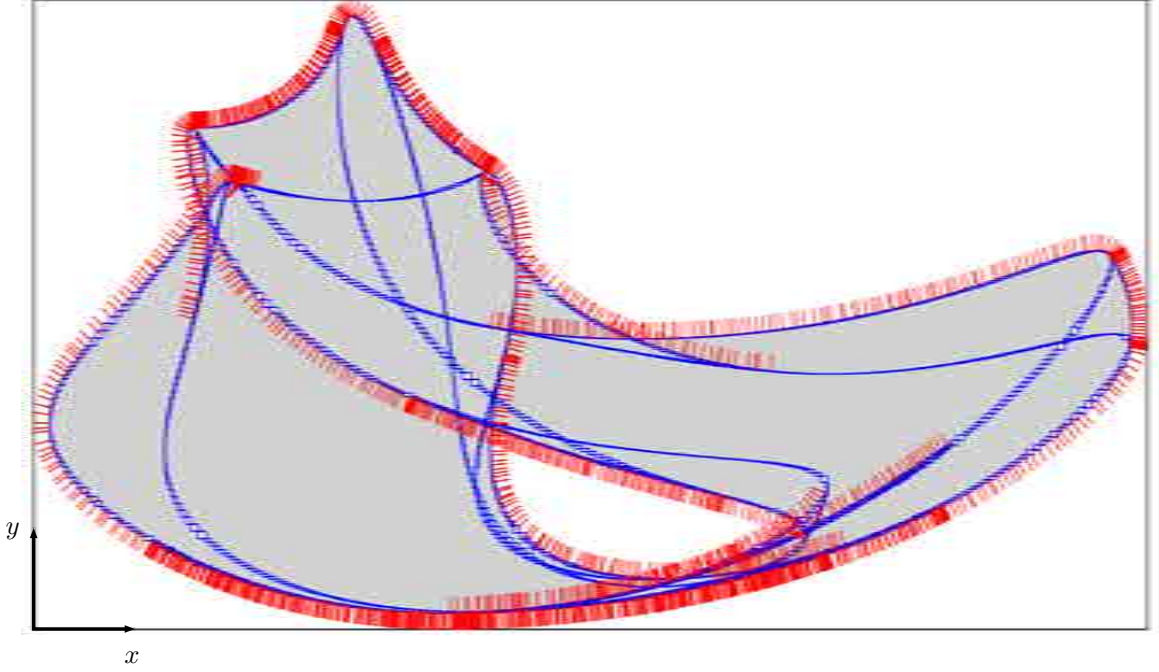


Figure 4.9: Results of applying the boundary identification process from Section 4.4.3 to one of the curve components in Fig. 4.8. The same conventions as in Figs. 4.2 and 4.6(d) are used.

computed by the proposed technique using a proper choice of the  $\mathbf{u}$  variables and fixing others to given values. To provide one example where hidden regions arise, we next compute a constant orientation workspace studied in [6].

Equation (4.1) can be formulated as in the example of the Stewart platform of Section 3.5.2. Here, the rotation matrix of the platform,  $\mathbf{R}_2$ , will be constant and known, since we are computing the constant orientation workspace. Also, since in this case the lengths  $d_i$  will be limited within ranges  $[\underline{d}_i, \overline{d}_i]$ , we must add the equations

$$(d_i - m_i)^2 + t_i^2 = h_i^2 \quad (4.19)$$

to the system, for  $i = 1, \dots, 6$ , where  $m_i$  and  $h_i$  are the mid-point and half-range of  $[\underline{d}_i, \overline{d}_i]$ . Clearly, if  $\mathbf{r}_P = [x, y, z]^T$ , we have  $\mathbf{u} = [x, y, z]^T$ , and we are in the situation in which  $\mathbf{u}$  intervenes explicitly in  $\mathbf{q}$ , so the system characterizing  $\mathcal{W}$  adopts the form of Eq. (4.12).

Figure 4.10 shows 3D views of the box approximation obtained for  $\pi_{\mathbf{u}}(\mathcal{W})$ , which describes an umbrella-like surface overall. The computation was done assuming the parameters in Table 4.2,  $\mathbf{R}_2$  fixed to the identity matrix, and  $[\underline{d}_i, \overline{d}_i] = [454.5, 504.5]$ . In fact, this workspace has an additional connected component symmetric to the one of Fig. 4.10, which corresponds to

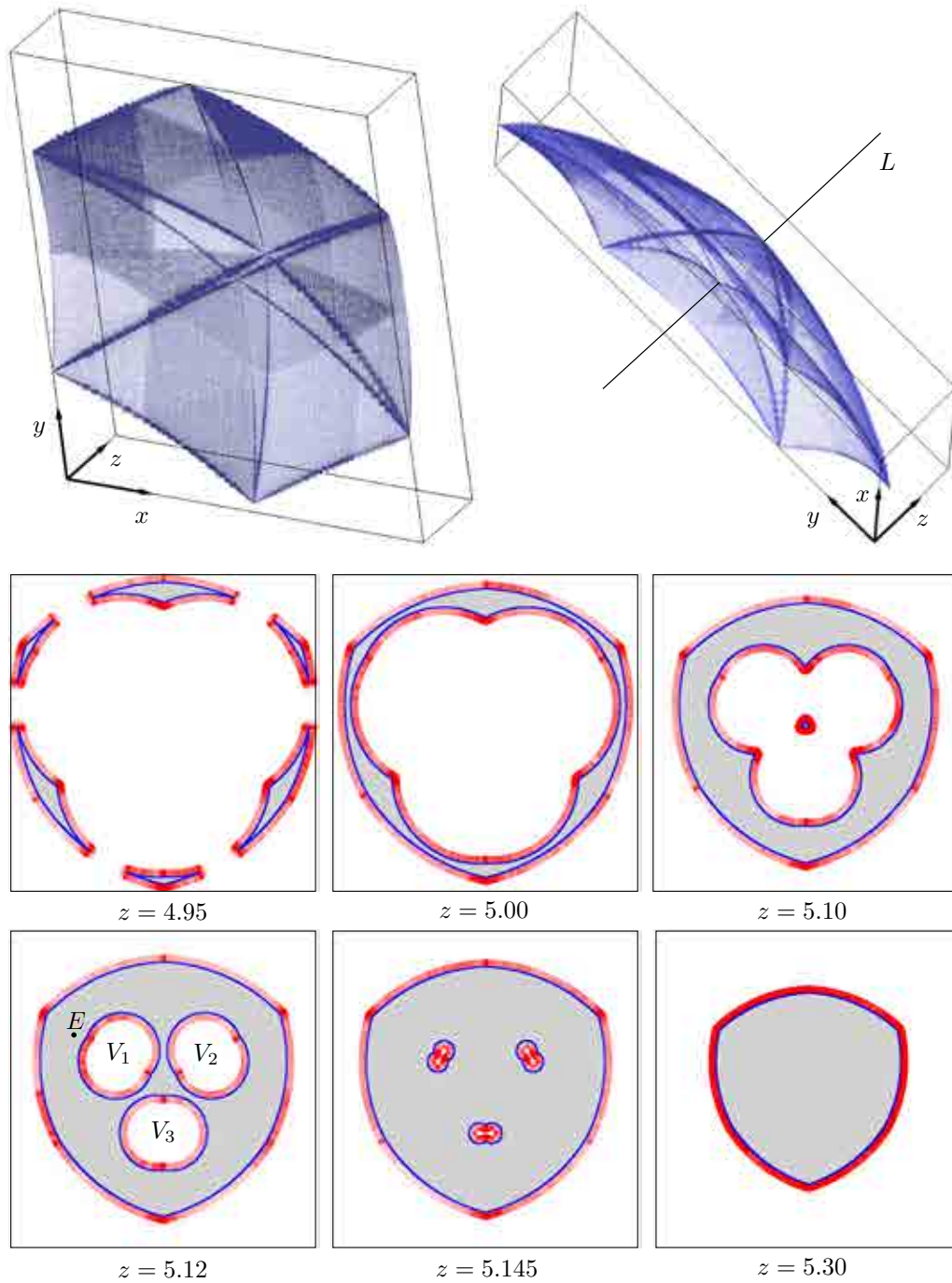


Figure 4.10: Top row: Two views of the boundary of the constant orientation workspace of a Stewart platform. Boxes are translucent to better appreciate the shape. Remaining rows: Slices of the workspace for different values of  $z$ . All points of  $\mathcal{W}$  are classified as barrier singularities in this case.



| Leg | Base $L_1$  | Platform $L_2$  |
|-----|---|---|
| 1   | $\mathbf{p}_{1,1}^{\mathcal{F}_1} = [92.58, 99.64, 23.10]^T$    | $\mathbf{p}_{2,1}^{\mathcal{F}_2} = [30.00, 73.00 - 37.10]^T$   |
| 2   | $\mathbf{p}_{1,2}^{\mathcal{F}_1} = [132.58, 30.36, 23.10]^T$   | $\mathbf{p}_{2,2}^{\mathcal{F}_2} = [78.22, -10.52, -37.10]^T$  |
| 3   | $\mathbf{p}_{1,3}^{\mathcal{F}_1} = [40.00, -130.00, 23.10]^T$  | $\mathbf{p}_{2,3}^{\mathcal{F}_2} = [48.22, -62.48, -37.10]^T$  |
| 4   | $\mathbf{p}_{1,4}^{\mathcal{F}_1} = [-40.00, -130.00, 23.10]^T$ | $\mathbf{p}_{2,4}^{\mathcal{F}_2} = [-48.22, -62.48, -37.10]^T$ |
| 5   | $\mathbf{p}_{1,5}^{\mathcal{F}_1} = [-132.58, 30.36, 23.10]^T$  | $\mathbf{p}_{2,5}^{\mathcal{F}_2} = [-78.22, -10.52, -37.10]^T$ |
| 6   | $\mathbf{p}_{1,6}^{\mathcal{F}_1} = [-92.58, 99.64, 23.10]^T$   | $\mathbf{p}_{2,6}^{\mathcal{F}_2} = [-30.00, 73.00, -37.10]^T$  |

Table 4.2: Parameters of the Stewart platform studied in [6].

the assembly mode of this manipulator where  $P$  sweeps a similar volume for  $z < 0$ . All results obtained are consistent with those in [6].

To better appreciate the shape of the enclosed volume, Fig. 4.10 plots constant- $z$  slices of  $\pi_{\mathbf{u}}(\mathcal{W})$  indicating the results of the boundary identification process. Note from the plots that it would be difficult to compute such slices by continuation [60], because many of them present multi-component boundaries and hidden barriers that difficult the application of the ray-shooting technique described in Section 4.3. If, for example, the ray is shot from point  $E$  on the  $z = 5.12$  slice, it will not hit the boundary of voids  $V_2$  and  $V_3$  on that slice. While it is true that in [100] the authors were able to compute  $\pi_{\mathbf{u}}(\mathcal{W})$  using the method in [60], they did so by defining particular slices of this set obtained by cutting the umbrella with planes through line  $L$  shown in Fig. 4.10. This solution avoids the appearance of internal voids within each slice, but obviously relies on a-priori knowledge of the result.

### 4.5.3 Degenerate Barriers

Because of the complexity of their defining equations, orientation workspaces are considered among the most difficult ones to compute and represent [25, 82, 102, 105]. Their derivation could be illustrated on Stewart platforms, but we shall do so on spherical parallel manipulators (SPM) because this will lead to one example of the degenerate barriers mentioned in Section 4.3, which the method in [60] is unable to identify. The examples are taken from [61] and correspond to popular architectures of three degree-of-freedom SPMs: the 3-UPS/S and 3-RRR designs, depicted in Fig. 4.11(a) and 4.11(b). These are orientational manipulators where the moving platform can be rotated with respect to the base about a fixed point  $O$  by actuating some of the leg joints. We next compute their orientation workspace and verify the results with those of the analytic method in [61]. Note that whereas the method in [61] is only applicable when certain symmetries hold, the method we propose here remains general.

To derive (4.1), note that each leg imposes the same constraint on the moving platform

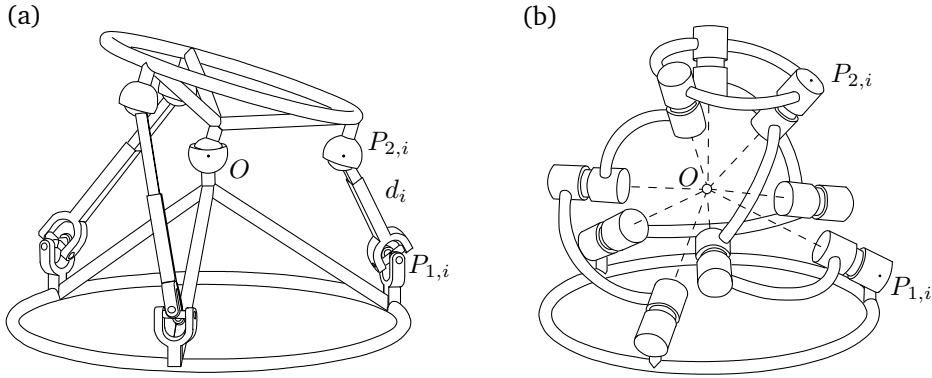


Figure 4.11: The 3-UPS/S and 3-RRR spherical platforms. Figures adapted from [61].

irrespectively of the chosen architecture. In a 3-UPS/S platform, for example,  $d_i$  is constrained to take values within some interval  $[d_i, \bar{d}_i]$  by design, which limits the angle between  $OA_i$  and  $OB_i$  to some range  $[\phi_i, \bar{\phi}_i]$ . In a 3-RRR platform, the angle between  $OA_i$  and  $OB_i$  is also limited to the range  $[\phi_i, \bar{\phi}_i]$  due to mechanical limits on the joints, or to the angles encompassed by the leg links. Both designs are thus kinematically equivalent. Moreover, the 3-UPS/S design can be obtained as a special case of the Stewart platform by making three anchor points of such platform coincident, and locking the corresponding legs. Hence, for both the 3-UPS/S and 3-RRR designs Eq. (4.1) can be formulated as in the previous Section with  $\mathbf{r}_P = [0, 0, 0]^T$  assuming that the fixed and moving frames are centered at  $O$ .

In general, the orientation workspace is defined as the set of possible values for three orientation angles of the platform. Although any set of Euler angles could be used for such matter, we shall here adopt the azimuth ( $\phi$ ), tilt ( $\theta$ ), and torsion ( $\sigma$ ) angles assumed in [61] to ease the comparison of results. Using such angles,  $\mathbf{R}_2 = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\gamma)$ , where  $\gamma = \sigma - \phi$ , and thus the columns of  $\mathbf{R}_2$  are

$$\mathbf{r}_1 = \begin{bmatrix} \cos \phi \cos \theta \cos \gamma - \sin \phi \sin \gamma \\ \sin \phi \cos \theta \cos \gamma + \cos \phi \sin \gamma \\ -\sin \theta \cos \gamma \end{bmatrix}, \quad (4.20)$$

$$\mathbf{r}_2 = \begin{bmatrix} -\cos \phi \cos \theta \sin \gamma - \sin \phi \cos \gamma \\ -\sin \phi \cos \theta \sin \gamma + \cos \phi \cos \gamma \\ \sin \theta \sin \gamma \end{bmatrix}, \quad (4.21)$$

$$\mathbf{r}_3 = \begin{bmatrix} \cos \phi \sin \theta \\ \sin \phi \sin \theta \\ \cos \theta \end{bmatrix}. \quad (4.22)$$

Under the previous convention, Bonev and Gosselin define the orientation workspace as the set of possible values that  $\mathbf{u} = [\phi, \theta, \sigma]^\top$  can attain [61], restricting  $\phi \in (-\pi, \pi)$ ,  $\theta \in [0, \pi)$ , and  $\sigma \in (-\pi, \pi]$  to guarantee a one-to-one relationship between the orientations and the corresponding triples  $\{\phi, \theta, \sigma\}$ . We are thus in the situation in which  $\mathbf{u}$  does not intervene explicitly in  $\mathbf{q}$ , but it can be related to  $\tilde{\mathbf{u}} = [\mathbf{r}_1^\top, \mathbf{r}_2^\top, \mathbf{r}_3^\top]^\top$  using (4.20)–(4.22). Therefore, equation  $\eta(\tilde{\mathbf{u}}) = \mathbf{0}$  is given by Eqs. (3.8)–(3.11) in this case, and  $\Psi(\tilde{\mathbf{z}}, \tilde{\mathbf{u}}) = \mathbf{0}$  consists of (3.31), (3.32) and (4.19), with  $\tilde{\mathbf{z}} = [d_1, d_2, d_3, t_1, t_2, t_3]$ . Overall, Eq. (4.18) contains 19 equations in 21 variables and the boundaries of the orientation workspace are thus expected to be 2-dimensional.

For the particular examples shown next we shall assume the same symmetry conditions as in [61]. Namely,  $A_i$  and  $B_i$  will lie on a unit sphere centered at  $O$  with position vectors

$$\mathbf{a}_i = \begin{bmatrix} \cos((i-1)\frac{2\pi}{3}) \sin \beta_1 \\ \sin((i-1)\frac{2\pi}{3}) \sin \beta_1 \\ -\cos \beta_1 \end{bmatrix}, \mathbf{b}_i = \begin{bmatrix} \cos((i-1)\frac{2\pi}{3}) \sin \beta_2 \\ \sin((i-1)\frac{2\pi}{3}) \sin \beta_2 \\ -\cos \beta_2 \end{bmatrix},$$

and we will set  $[\phi_i, \bar{\phi}_i] = [\phi, \bar{\phi}]$  for all  $i$ .

Figure 4.12 shows the resulting box approximation of  $\pi_{\mathbf{u}}(\mathcal{W})$  for  $\beta_1 = 0^\circ$ ,  $\beta_2 = 35^\circ$ ,  $\phi = 20^\circ$ ,  $\bar{\phi} = 130^\circ$ . These parameters correspond to one of the cases analyzed in [61], where constant-torsion slices of  $\pi_{\mathbf{u}}(\mathcal{W})$  are provided for these manipulators. As expected,  $\pi_{\mathbf{u}}(\mathcal{W})$  is a surface in the  $\{\theta, \phi, \sigma\}$ -space and, by analyzing the neighboring relationships of the returned boxes, this surface can be shown to contain just one connected component. Figs. 4.13 top and middle respectively show a  $\sigma = -30^\circ$  slice of the surface shown in Fig. 4.12 and the barriers identified on such slice. The resulting curve and the interior regions detected match those in [61] when plotted in polar coordinates (Fig. 4.13, bottom).

While the orientation workspace will generally have a two-dimensional boundary, such boundary may degenerate into lower-dimensional barriers for particular choices of the geometric parameters, thus posing serious difficulties to the continuation method in [60]. This is what occurs on the Agile Eye for example, a well-known instance of the 3-RRR design in Fig. 4.11(b), where  $\beta_1 = \beta_2 = \arccos(1/\sqrt{3})$ ,  $\phi = 0$ , and  $\bar{\phi} = \pi$  [106]. As mentioned in Section 4.3, computing such barriers using [60] is almost impossible because the ray shooting technique will fail to converge to the barriers almost always. On the contrary, the presented technique is immune to such situations. If the same equations considered for obtaining the plot in Fig. 4.12 are now used for determining the workspace boundaries of the Agile Eye relative

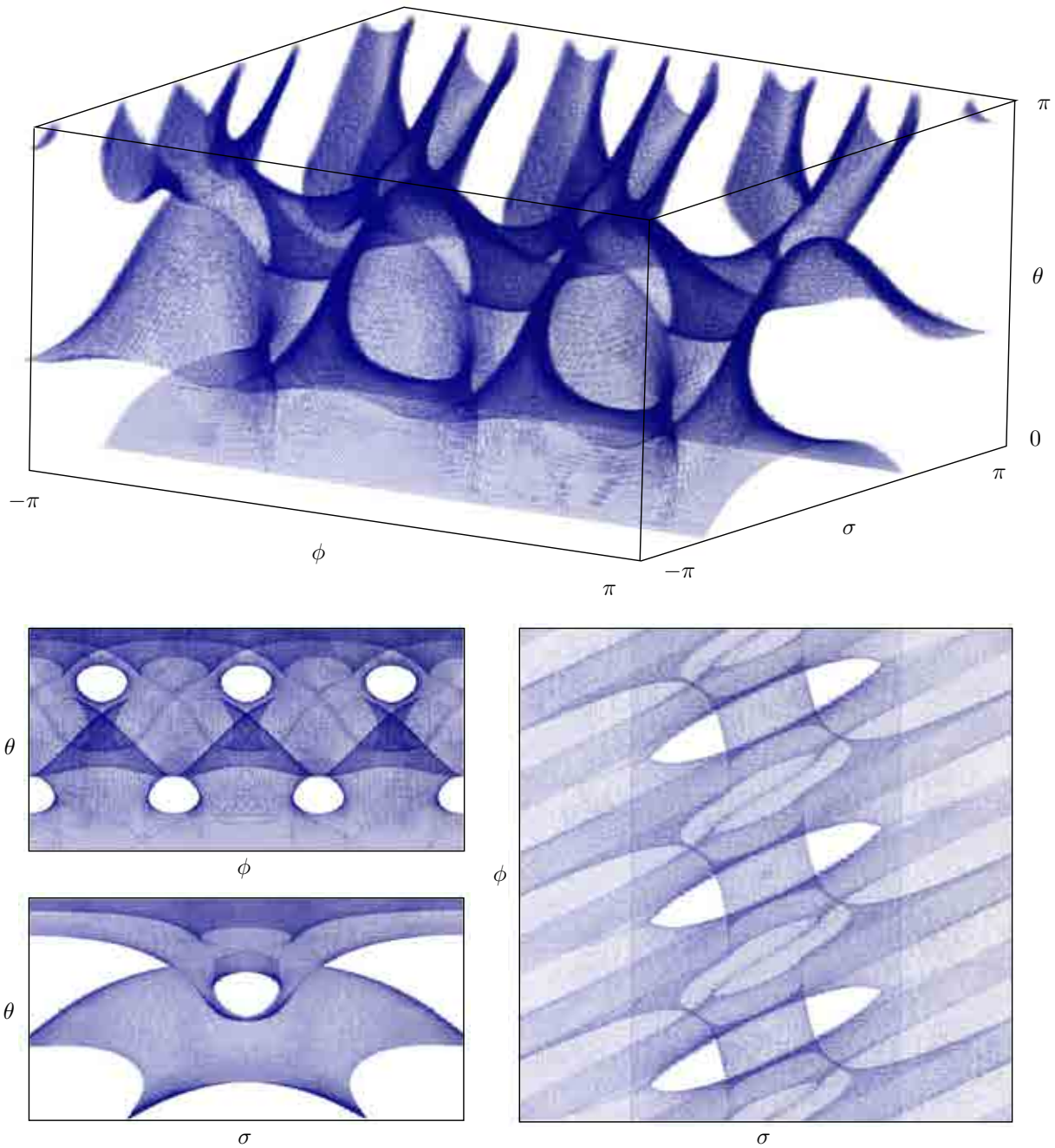


Figure 4.12: Top: 3D view of the boundary of the orientation workspace of the 3-UPS/S manipulator of Fig. 4.11(b). Bottom: orthogonal projections of such boundary onto the coordinate planes.

to  $\mathbf{u} = [\phi, \theta, \sigma]^T$ , we readily obtain the curves depicted in Fig. 4.14, which agree with those described in [61]. When analyzed, these curves are seen to be barrier singularities interior to the

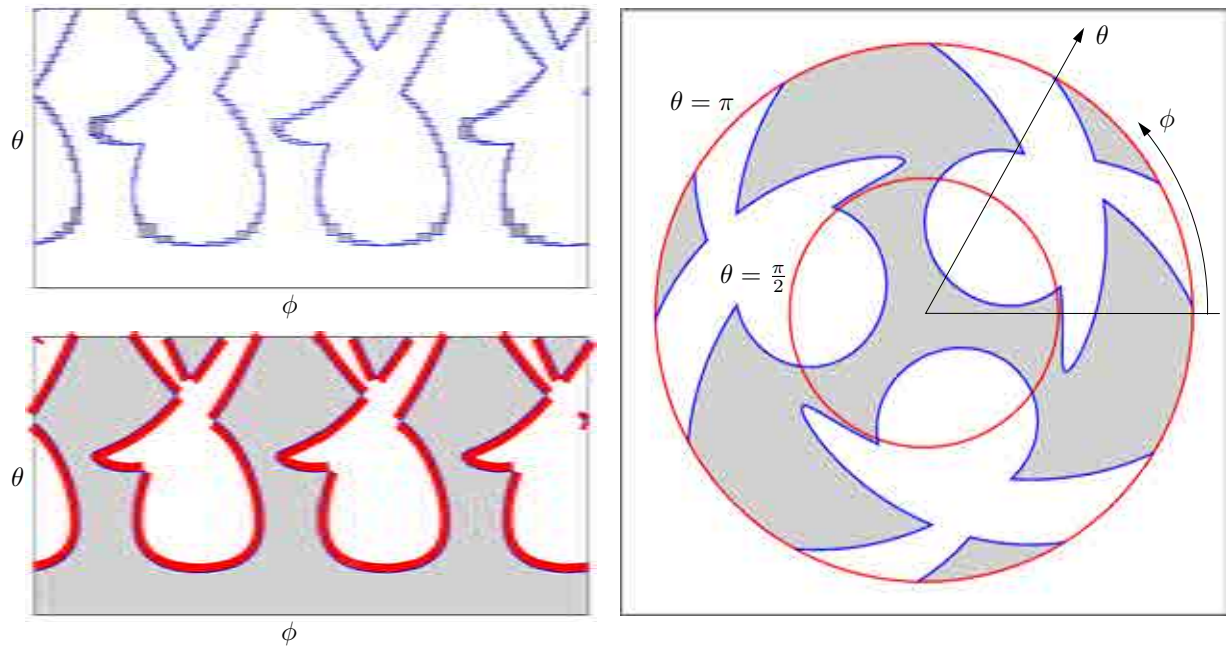


Figure 4.13: Constant torsion ( $\sigma$ ) workspace of the spherical parallel manipulator of Fig. 4.11(b), with parameters  $\beta_1 = 45^\circ$ ,  $\beta_2 = 35^\circ$ ,  $\underline{\alpha} = 20^\circ$ ,  $\bar{\alpha} = 130^\circ$ ,  $\sigma = -30^\circ$ .

workspace. In other words, the manipulator will be able to reach any possible orientation, but it will find a motion impediment when trying to traverse across the curves. The phenomenon is better understood if we view the curves as the limit case of a one-parameter family of workspace boundaries, corresponding to setting  $\bar{\alpha} = \pi - k$ ,  $\underline{\alpha} = k$ , and varying the parameter  $k$  from  $\frac{\pi}{6}$  to 0, while keeping fixed the remaining set of geometric parameters. Tubular boundary barriers are present in all members of the family (first plots of Fig. 4.15), which degenerate into the barrier filaments of the Agile Eye (last plot).

#### 4.5.4 Very Complex Manipulators

To illustrate the method on a highly complex situation, we next apply it to compute the position workspace of a point of the 15-link mechanism in Fig. 4.16(a). The mechanism consists of five quadrilateral links interconnected through bar links and revolute joints, forming a decagonal ring. If we fix one of the quadrilaterals to the ground, the mechanism has mobility two, so that  $\mathcal{C}$  will have dimension  $n = 2$  in general, and  $\mathcal{W}$  will be formed by one or several curves in such space. The mechanism is controlled by actuating the  $\theta_1$  and  $\theta_2$  angles indicated, and the coordinates of interest,  $\mathbf{u} = [x, y]^T$ , are the position variables of a point  $P$  on link  $L$ .

The complexity of this mechanism comes from the fact that it involves many links, and all

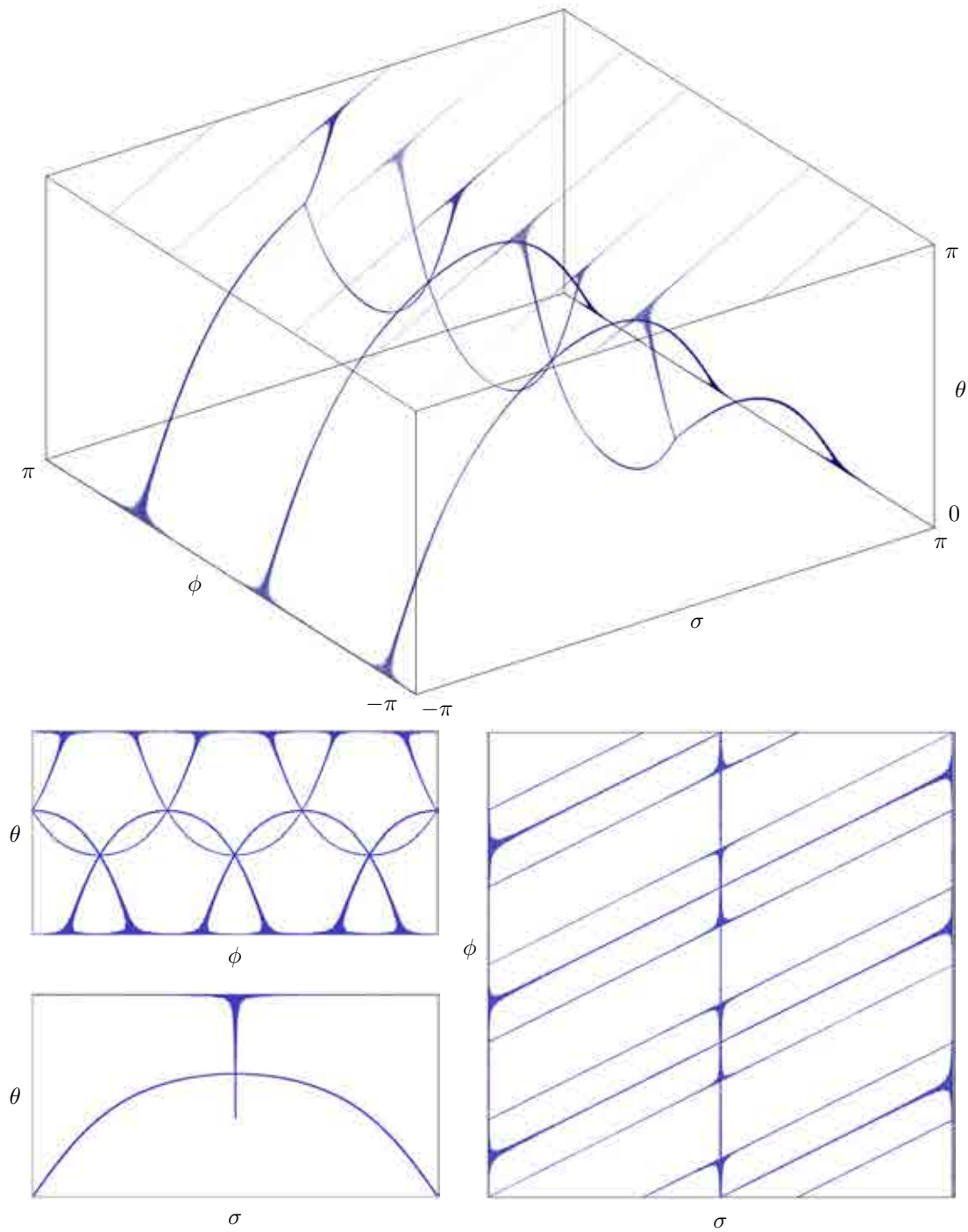


Figure 4.14: The workspace of the Agile Eye has degenerate barriers. Each line in the  $\theta = \pi$  plane corresponds to a unique pose of the platform.

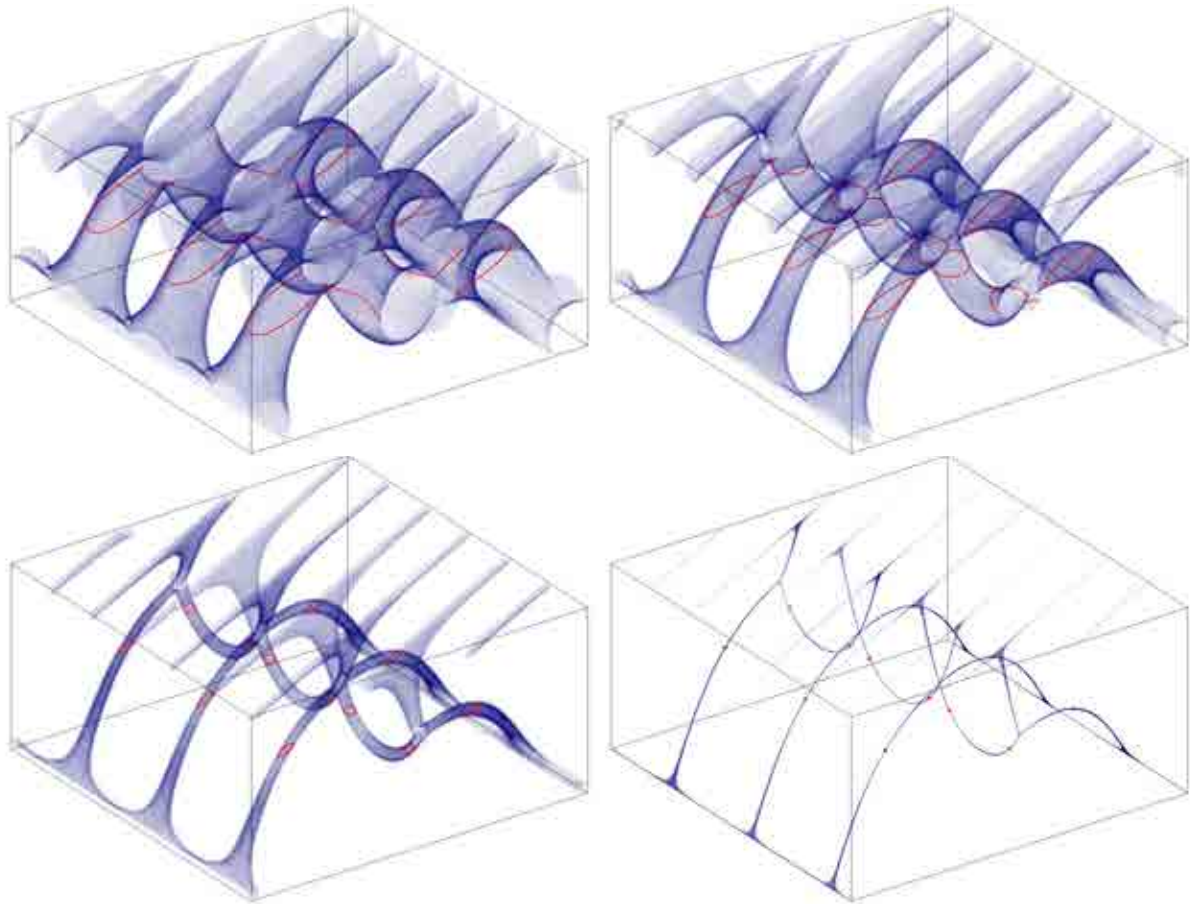


Figure 4.15: The workspace of the Agile Eye is a limit case of a one-parameter family of workspaces. Slices at constant  $\theta$  are also shown, to see how they degenerate into single points.

of them move in a highly-coupled manner. This behaviour is apparent from the structure of the mechanism already, but it can be proved through the application of recent Assur Graph Theory tools [107, 108]. On the basis of these observations, thus, we conjecture that the derivation of minimal-degree polynomials describing the singularity sets of this manipulator is an extremely difficult task. The computation of such sets is even too hard through discretization techniques [109, 110], which define a grid of points in the  $\mathcal{U}$  space, solve the inverse kinematics problem for each point, and finally analyze the resulting configurations one-by-one, identifying those that are close to the singularity sets. Note that this process boils down to discretizing the  $xy$ -plane on this mechanism, and that solving the inverse kinematics problem for each position  $(x, y)$  is equivalent to finding all configurations of a seven-loop truss [Fig. 4.16(b)], which is beyond the capabilities of even the most advanced techniques for position analysis based on characteristic polynomials [111, 112].

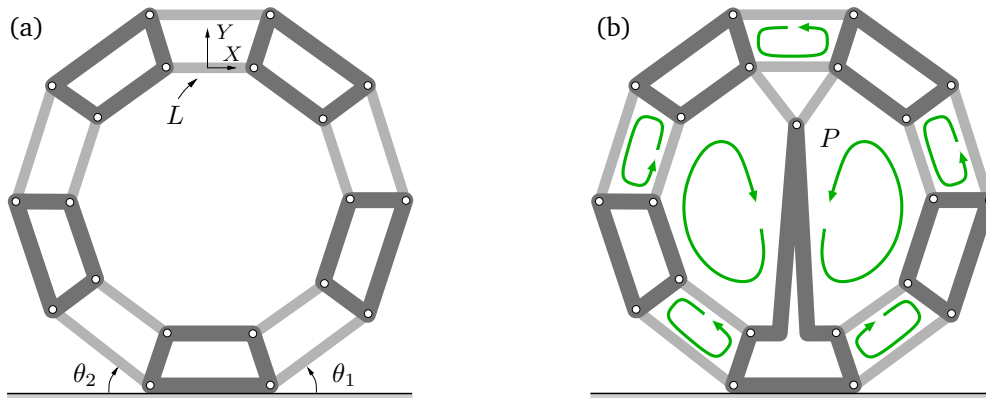


Figure 4.16: (a) A 15-link mechanism. (b) Its inverse kinematics problem is equivalent to solving the position analysis of a seven-loop truss.

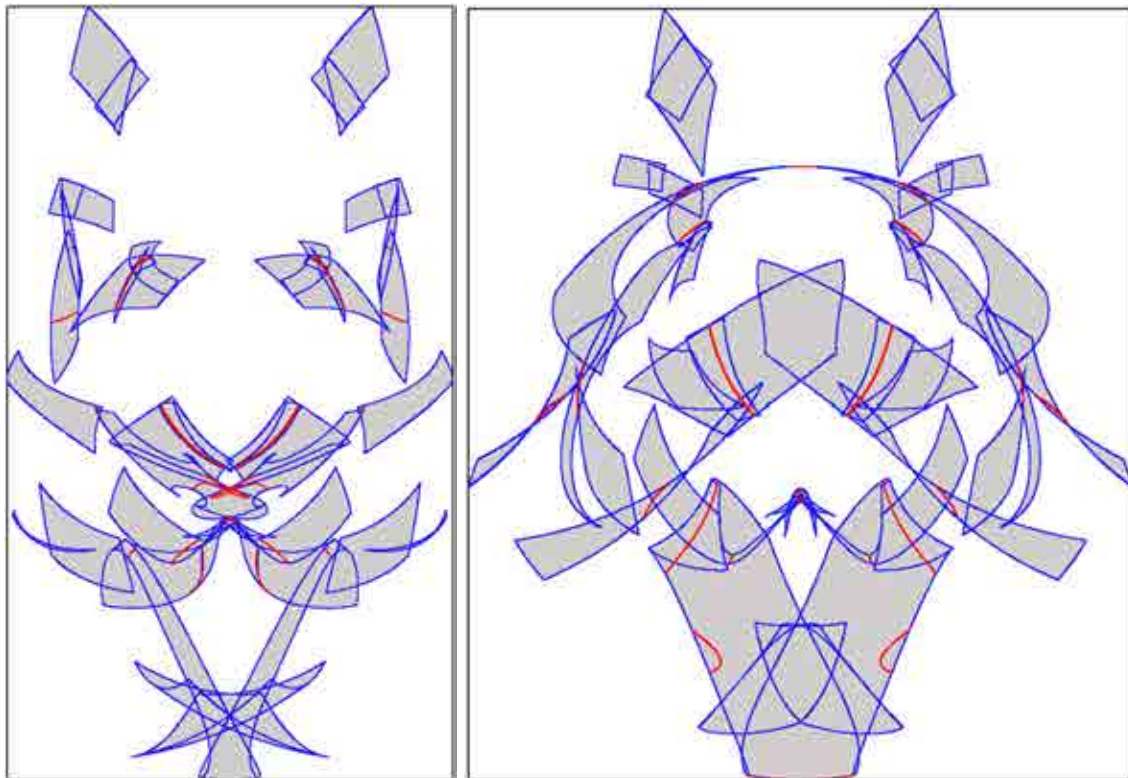


Figure 4.17: Position workspaces of the manipulator in Fig. 4.16 assuming the geometric parameters mentioned in the text. The angles  $\theta_1$  and  $\theta_2$  are limited to the ranges  $[45.57^\circ, 60^\circ]$  (left plot) and  $[36.87^\circ, 53.13^\circ]$  (right plot). The red curves correspond to the forward singularities of the manipulator, while the inverse singularities are included among the blue curves.



Assuming that  $P$  is located in position  $(0, -1)$  of the frame indicated in Fig. 4.16(a), that all quadrilateral links of the mechanism are squares of side 1, and that all bars are of length 2, except  $L$ , which is of length  $\sqrt{2}$ , the method determines the singularity sets shown in Fig. 4.17. The two plots correspond to two variants of the mechanism that differ on the limits imposed on  $\theta_1$  and  $\theta_2$  only, using the equations of Section 4.4.1.



## **Part II**

# **Avoidance of Singularities**



# 5

## A General Singularity-free Path Planner

The methods developed in Part I allow to compute and interpret the different singularity sets of a manipulator. The visualization of the sets is not an easy task, but we have shown how their projections to the input and output spaces generate very rich diagrams, called portraits, summarizing the global motion capabilities of the manipulator. These diagrams can be used as safe navigation maps if desired, because any path not crossing a projected singularity necessarily corresponds to a singularity-free path in the C-space. However, they do not always reveal all possible singularity-free paths between given configurations, and additional tools to determine whether such paths exist become necessary for a complete understanding of the overall motion capabilities.

This chapter provides an algorithm to this end. Given two non-singular configurations of the manipulator, the method attempts to connect them through a path that maintains a minimum clearance with respect to the singularity locus at all points. In comparison to previous approaches (Section 5.1), the method can be applied to non-redundant manipulators of general structure, and it is resolution-complete, in the sense that it always returns a path whenever one exists at a given resolution, or determines path non-existence otherwise. The strategy relies on defining a smooth manifold that maintains a one-to-one correspondence with the singularity-free C-space of the manipulator (Section 5.2), and on using a higher-dimensional continuation technique to explore this manifold systematically from one configuration, until the second configuration is found (Section 5.3). If desired, the method can also be used to compute an exhaustive atlas of the whole singularity-free component that is reachable from a configuration, which is useful to rapidly resolve subsequent planning queries within such component, or to visualize the singularity-free workspace of any set of manipulator coordinates. Examples are included to demonstrate the performance of the method on illustrative situations (Section 5.4).

## 5.1 Previous Approaches

Several works confront the problem of on-line singularity avoidance [113], but only a few tackle the general problem of planning paths between distant configurations. They include an algorithm based on deforming a parametrized path between the query configurations [23], a variational approach that reduces the problem to a boundary value problem [24], and a numerical technique based on treating the singularity locus as a collection of obstacles [114]. All of these methods work well in favorable situations, but [23] and [24] mention limitations relative to proving path existence in some cases, and the one in [114] is computationally intensive, as it requires constructing polytope approximations of the entire singularity set before searching for a feasible path. In some way or another, also, the methods in [23, 24, 114] exploit the fact that the considered C-spaces have closed-form parameterizations, so that it is difficult to extend them to tackle manipulators with a more complex architecture. The method we provide in this chapter, in contrast, makes no recourse to closed-form parameterizations, and can be applied to any non-redundant closed-chain manipulator, with the sole limitations imposed by the curse of dimensionality. As opposed to [114], it treats singularities implicitly, not explicitly as obstacles, resulting in a computationally less intensive approach.

## 5.2 Equations of the Singularity-free C-space

Given two configurations of  $\mathcal{C} \setminus \mathcal{S}$ ,  $\mathbf{q}_s$  and  $\mathbf{q}_g$ , our goal is to generate a singularity-free *path* connecting them, i.e., a continuous map  $\tau : [0, 1] \rightarrow \mathcal{C} \setminus \mathcal{S}$ , such that  $\tau(0) = \mathbf{q}_s$  and  $\tau(1) = \mathbf{q}_g$ . For simplicity, we will focus on avoiding the forward singularities only, because they are those potentially harmful to the manipulator, but the extension to deal with inverse singularities is straightforward in our approach. In the rest of the chapter, thus, we will be content in generating paths  $\tau : [0, 1] \rightarrow \mathcal{C}_{sfree} = \mathcal{C} \setminus \mathcal{S}_f$ , where  $\mathcal{S}_f$  is the forward singularity locus of the manipulator.

Note that  $\mathcal{C}_{sfree}$  is the set of points  $\mathbf{q} \in \mathcal{C}$  for which  $\det(\mathbf{L}_y) \neq 0$ . However, to be able to apply a continuation method, it is necessary to turn the latter condition into equality form. To this end, we introduce an auxiliary variable  $b$  and note that  $\det(\mathbf{L}_y) \neq 0$  if, and only if,  $\det(\mathbf{L}_y) \cdot b = 1$  for some value of  $b$ . Thus, a system of equations characterizing  $\mathcal{C}_{sfree}$  is given by

$$\left. \begin{array}{l} \Phi(\mathbf{q}) = \mathbf{0} \\ \det(\mathbf{L}_y) \cdot b = 1 \end{array} \right\}. \quad (5.1)$$

If other singularities need to be avoided, the determinant of the matrix that loses rank on them can be added as an additional factor in the second equation of this system.

For convenience, system (5.1) will be written as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \quad (5.2)$$

hereafter, where

$$\mathbf{x} = [\mathbf{q}^\top, b]^\top,$$

and

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \Phi(\mathbf{q}) \\ \det(\mathbf{L}_y) \cdot b - 1 \end{bmatrix}. \quad (5.3)$$

Let  $\mathcal{M}$  be the set of points  $\mathbf{x}$  that satisfy (5.2), and define the function

$$b(\mathbf{q}) = \frac{1}{\det(\mathbf{L}_y(\mathbf{q}))}. \quad (5.4)$$

Note that the points  $\mathbf{x} \in \mathcal{M}$  are in one-to-one correspondence with the points  $\mathbf{q} \in \mathcal{C}_{\text{free}}$ , because  $\mathbf{q} \in \mathcal{C}_{\text{free}}$  if, and only if,  $\mathbf{x} = [\mathbf{q}, b(\mathbf{q})]^\top$  satisfies Eq. (5.1). Accordingly, all paths in  $\mathcal{C}_{\text{free}}$  are uniquely represented in  $\mathcal{M}$ , and *vice versa*. Thus, the original problem of computing a singularity-free path on  $\mathcal{C}$  from  $\mathbf{q}_s$  to  $\mathbf{q}_g$  can be reduced to that of connecting  $\mathbf{x}_s = [\mathbf{q}_s^\top, b(\mathbf{q}_s)]^\top$  and  $\mathbf{x}_g = [\mathbf{q}_g^\top, b(\mathbf{q}_g)]^\top$  through some path in  $\mathcal{M}$ . This reduction is advantageous because, by letting the path planner operate in  $\mathcal{M}$ , instead of in  $\mathcal{C}$  directly, guarantees that any computed path in  $\mathcal{M}$  will have a corresponding path in  $\mathcal{C}$  lying entirely in  $\mathcal{C}_{\text{free}}$ . This eliminates the need of checking singularity crossings by the planner, which, as noted in [23], may be difficult in many situations, due to the intricate structure of the singularity locus.

The correspondence of the two problems is schematically illustrated in Fig. 5.1. The horizontal plane in the bottom represents  $\mathcal{C}$ , which in this example coincides with the ambient space  $\mathcal{Q}$  for simplicity, and the singularity locus  $\mathcal{S}_f$  is represented by two red parabolas in this plane. To construct  $\mathcal{M}$ , we add a new dimension  $b$  to  $\mathcal{Q}$  (the vertical axis in the figure), and we lift every point  $\mathbf{q} \in \mathcal{C}$  to the point  $\mathbf{x} = [\mathbf{q}^\top, b(\mathbf{q})]^\top$ . Then,  $\mathcal{M}$  can be thought of as a new manifold extending infinitely in the direction  $b$ , as the projection  $\mathbf{q}$  of a point  $\mathbf{x} \in \mathcal{M}$  approaches  $\mathcal{S}_f$ .

Two important observations regarding the search for a path could be made. On the one hand, note that the Jacobian  $\mathbf{F}_x$  has the block structure

$$\mathbf{F}_x = \begin{bmatrix} \Phi_v & \Phi_y & \mathbf{0} \\ * & * & \det(\mathbf{L}_y) \end{bmatrix},$$

from which we see that  $\mathbf{F}_x$  is full rank at all points  $\mathbf{x} \in \mathcal{M}$ . Note that since  $\mathbf{L}_y$  is full rank over all  $\mathcal{M}$ ,  $\Phi_y$  will be full rank too, because both matrices lose rank at the same configurations

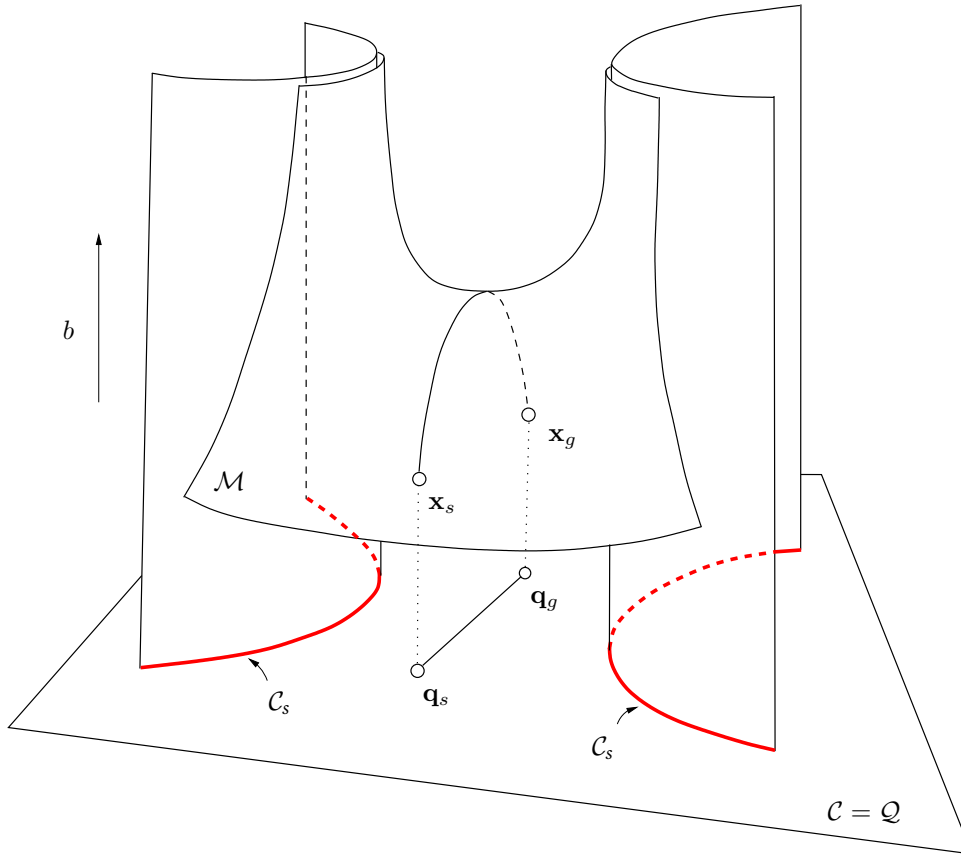


Figure 5.1: The original problem of computing a singularity-free path on  $\mathcal{C}$  connecting  $\mathbf{q}_s$  and  $\mathbf{q}_g$  is transformed into one of finding an arbitrary path on  $\mathcal{M}$  connecting  $\mathbf{x}_s$  and  $\mathbf{x}_g$ .

$\mathbf{q}$  and, hence,  $\Phi_{\mathbf{q}}$  never loses rank in  $\mathcal{M}$ . By the Implicit Function Theorem, this implies that  $\mathcal{M}$  has the structure of a smooth manifold everywhere [47], which is beneficial from the point of view of applying a continuation method to explore  $\mathcal{M}$  [33], because no bifurcations, ridges, or dimension changes are to be found during such exploration, and no recourse to branch-switching operations will be necessary [34]. On the other hand, remember that all of the  $\mathbf{q}$  coordinates have known bounds in practice, and that those bounds derived from mechanical limits on the joints can be taken into account, if necessary, by adding extra equations to the system (Chapters 3 and 4). Moreover,  $|b|$  should be maintained below a given threshold  $b_{max}$  to guarantee some clearance from  $S_f$ , given by the task to accomplish and by the characteristics of the underlying mechanical and control systems. As a result, the search for a path on  $\mathcal{M}$  must be restricted to a given domain  $\mathcal{D}$  of the  $\mathbf{x}$ -space, usually defined as the Cartesian product of a number of intervals derived from the coordinate bounds.



## 5.3 Exploring the Singularity-free C-space for a Path

To determine a singularity-free path connecting  $\mathbf{x}_s$  and  $\mathbf{x}_g$  we can gradually construct an atlas of  $\mathcal{M} \cap \mathcal{D}$ . An atlas is a collection of charts mapping  $\mathcal{M} \cap \mathcal{D}$  entirely, where each chart defines a local map from  $\mathbb{R}^n$  to a neighborhood in  $\mathcal{M}$  of a point  $\mathbf{x}_i$ . The atlas will be computed using the higher-dimensional continuation method proposed in [33], which generalizes the one-dimensional pseudo-arclength procedure [115]. These tools generate the atlas for one component of the  $n$ -dimensional variety implicitly defined by a system of equations,

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},$$

with  $\mathbf{F} : \mathbb{R}^m \rightarrow \mathbb{R}^{m-n}$ . In our case,  $\mathbf{F}$  is a smooth function defined as in Eq. (5.3),  $m$  is  $n_q + 1$ , and  $n$  is the dimension of  $\mathcal{M}$  which, as we saw, is actually a smooth manifold.

We will first describe the method to generate an atlas covering the full component of the manifold attainable from  $\mathbf{x}_s$ , and then we will explain how the expansion of the atlas can be guided in order to avoid computing the full atlas if only  $\mathbf{x}_g$  has to be reached.

### 5.3.1 Defining a Chart

Given a point on the manifold,  $\mathbf{x}_i$ , a chart  $C_i$  defines a mapping  $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}^m$  between a domain  $\mathcal{P}_i \in \mathbb{R}^n$  and an open set containing  $\mathbf{x}_i$ , with  $\psi_i(\mathbf{0}) = \mathbf{x}_i$ . The mapping is defined using the matrix  $\Psi_i$ , whose columns define an orthonormal basis of  $T_{\mathbf{x}_i}\mathcal{M}$ , the  $n$ -dimensional tangent space of  $\mathcal{M}$  at  $\mathbf{x}_i$ . This basis is the  $m \times n$  matrix satisfying

$$\begin{bmatrix} \mathbf{F}_{\mathbf{x}_i} \\ \Psi_i^\top \end{bmatrix} \Psi_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{n \times n} \end{bmatrix}.$$

The mapping  $\mathbf{x}_j = \psi_i(\mathbf{s}_j^i)$  is determined by first selecting an  $n$ -dimensional vector  $\mathbf{s}_j^i$  of parameters in  $\mathcal{T}_{\text{vr}\mathbf{x}_i}\mathcal{M}$  [Fig. 5.2(a)], and then generating a point  $\mathbf{x}_j^i \in \mathbb{R}^m$  in the neighborhood of  $\mathbf{x}_i$  as follows:

$$\mathbf{x}_j^i = \mathbf{x}_i + \Psi_i \mathbf{s}_j^i. \quad (5.5)$$

A point  $\mathbf{x}_j$  on  $\mathcal{M}$  is finally computed by orthogonally projecting  $\mathbf{x}_j^i$ . This projection is obtained by solving the system

$$\left. \begin{array}{l} \mathbf{F}(\mathbf{x}_j) = \mathbf{0} \\ \Psi_i^\top (\mathbf{x}_j - \mathbf{x}_j^i) = \mathbf{0} \end{array} \right\}$$

using a Newton method initialized at  $\mathbf{x}_j^i$  [116], and iteratively updated with the increment  $\Delta \mathbf{x}_j$

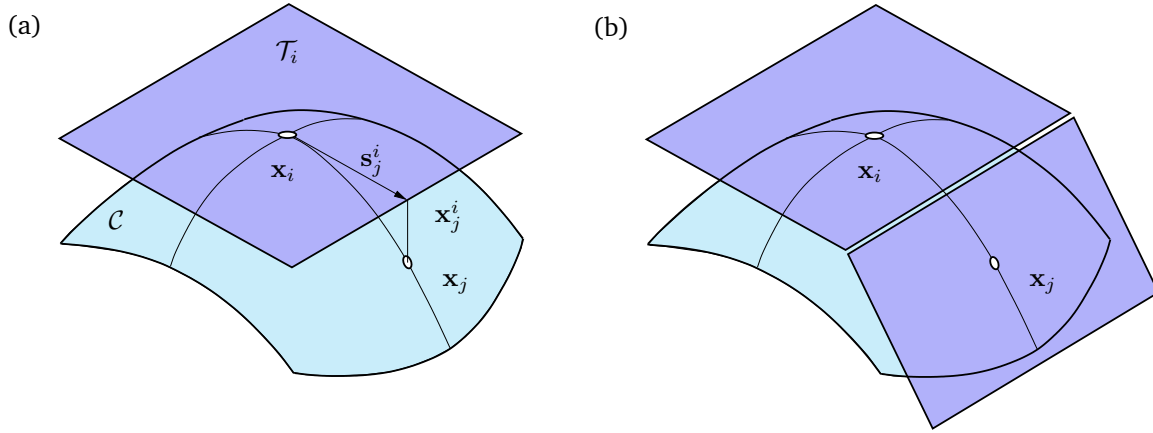


Figure 5.2: The higher-dimensional continuation method applied to a two-dimensional manifold in  $\mathbb{R}^3$ . Left: A point  $\mathbf{x}_j$  on  $\mathcal{M}$  can be obtained by orthogonally projecting a point  $\mathbf{x}_j^i$  on  $T_{\mathbf{x}_i}\mathcal{M}$ . Right: If a new chart is defined at  $\mathbf{x}_j$ , it must be properly coordinated with the chart at  $\mathbf{x}_i$  so that their projections smoothly cover the manifold.

fulfilling

$$\begin{bmatrix} \mathbf{F}_{\mathbf{x}_i} \\ \mathbf{\Psi}_i^\top \end{bmatrix} \Delta \mathbf{x}_j = - \begin{bmatrix} \mathbf{F}(\mathbf{x}_i) \\ \mathbf{\Psi}_i^\top (\mathbf{x}_j - \mathbf{x}_j^i) \end{bmatrix}. \quad (5.6)$$

The update is applied until either the norm of the right-hand side of (5.6) becomes negligible, or a maximum number of iterations is attained. If the process does not converge, the input parameter  $\mathbf{s}_j^i$  is out of  $\mathcal{P}_i$ , the applicability area for the chart  $\mathcal{C}_i$ . The areas out of the scope of the chart are to be parameterized by other charts.

The inverse of the mapping  $\psi_i$  can be computed as

$$\mathbf{s}_j^i = \psi_i^{-1}(\mathbf{x}_j) = \mathbf{\Psi}_i^\top (\mathbf{x}_j - \mathbf{x}_i),$$

and can be applied for any point  $\mathbf{x}_j$  in  $\mathbb{R}^m$ , regardless of whether or not it is on the manifold or if it is actually projected onto  $\mathcal{P}_i$ .

### 5.3.2 Constructing an Atlas

Since the applicability area for each chart is limited, a full parameterization of the manifold requires the construction of a whole atlas. Each point on the manifold is the potential center of a new chart [Fig. 5.2(b)], and a method due to Henderson can be used to decide where to place the chart centers and how to bound their associated applicability areas so as to ensure a

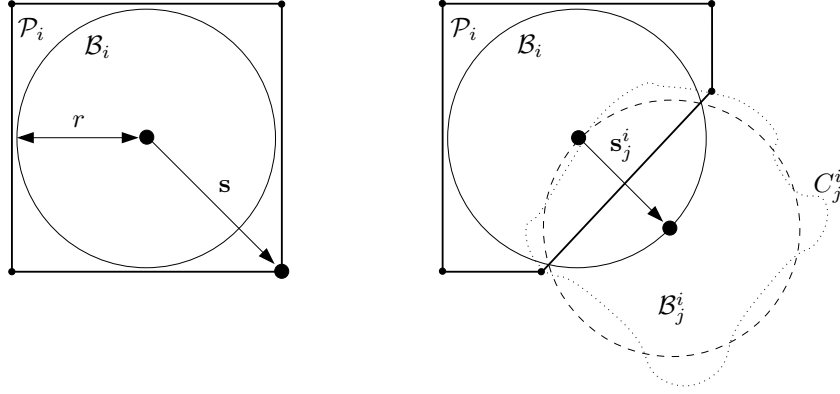


Figure 5.3: Polytope-based chart construction. Left: The domain for chart  $C_i$ ,  $\mathcal{P}_i$ , is a box including a ball of radius  $r$  around  $x_i$ , both defined in the tangent space associated with the chart. Right:  $\mathcal{P}_i$  is refined using a ball  $\mathcal{B}_j^i$  that approximates  $C_j^i$ , the projection on  $C_i$  of the part of the manifold covered by  $C_j$ .

good coverage of the manifold [33]. In his approach, the domain  $\mathcal{P}_i$  of chart  $C_i$  is represented as a polytope that is initialized as an  $n$ -dimensional hypercube enclosing a ball  $\mathcal{B}_i$  of radius  $r$ , both defined in  $T_{x_i}\mathcal{M}$ . Then,  $\mathcal{P}_i$  is progressively refined as new charts are added to the atlas, as illustrated in Fig. 5.3.

A vector  $\mathbf{s}$  to a vertex of  $\mathcal{P}_i$  external to  $\mathcal{B}_i$  is used to generate a new chart. Using  $\mathbf{s}$ , a vector of local parameters giving a point on  $\mathcal{B}_i$  is computed as

$$\mathbf{s}_j^i = \alpha \frac{\mathbf{s}}{\|\mathbf{s}\|}, \quad (5.7)$$

where  $\alpha$  is initialized to  $r$ . A new neighboring chart,  $C_j$ , is then defined on the point  $\mathbf{x}_j$  resulting from applying the mapping  $\psi_i(\mathbf{s}_j^i)$ , i.e., from projecting the point  $\mathbf{x}_j^i$  computed using (5.5). If the projection from  $\mathbf{x}_j^i$  to  $\mathcal{M}$  does not converge, or if the new chart  $C_j$  at  $\mathbf{x}_j$  is too far or too different from  $C_i$ , i.e., if

$$\|\mathbf{x}_j - \mathbf{x}_j^i\| > \epsilon, \quad (5.8)$$

or

$$\det(\Psi_i^T \Psi_j) < 1 - \epsilon, \quad (5.9)$$

for a given threshold  $\epsilon$ , the new chart is discarded and a new attempt of chart generation is performed with a smaller  $\alpha$  in Eq. (5.7), which gives a set of parameters  $\mathbf{s}_j^i$  closer to  $\mathbf{x}_i$ . This procedure adapts the area covered by each chart to the local curvature of the manifold.

Each new chart  $C_j$  added to the atlas has to be properly coordinated with the applicability

areas of those charts already in the atlas. In the example of Fig. 5.3,  $C_j$  is used to refine  $\mathcal{P}_i$  from the intersection between  $\mathcal{B}_i$  and  $\mathcal{B}_j^i$ , a ball of radius  $r$  centered at the point given by  $\mathbf{s}_j^i$  that approximates  $C_j^i$ , the projection on  $T_{\mathbf{x}_i}\mathcal{M}$  of the part of the manifold covered by  $C_j$ , as shown in Fig. 5.3(right).

The hyperplane defined by the intersection of  $\mathcal{B}_i$  and  $\mathcal{B}_j^i$  can be computed by subtracting the equations for the two balls. As shown in Fig. 5.3(right), this plane defines a new face of  $\mathcal{P}_i$  that eliminates some of its vertices (in particular the one given by  $\mathbf{s}$ ) and generates new ones. Symmetrically, the polytope  $\mathcal{P}_j$  associated with  $C_j$  is cropped using an approximation of the projection of  $C_i$  into  $C_j$ . The applicability areas of the two neighboring charts are not necessarily continuous in the ambient space  $\mathbb{R}^m$ , but under mild conditions their projection to the manifold slightly overlaps, smoothly covering it. The transition from one local parameterization to the other can be done by using the direct and inverse mappings for the two neighboring charts. For a given vector of parameters  $\mathbf{s}_i$  at the border of  $\mathcal{P}_i$ , the corresponding vector in  $C_j$  is

$$\mathbf{s}_j = \psi_j^{-1}(\psi_i(\mathbf{s}_i)).$$

When a chart is fully surrounded by other charts, all vertices of its polytope are inside the associated ball and the chart is considered to be *closed*, meaning that its applicability area is bounded and no further expansion needs to be attempted from that chart. Charts whose center is out of the domain  $\mathcal{D}$  are also considered closed. This process of chart expansion continues as far as there are *open* (non-closed) charts in the atlas. At the end of the process, the connected component of  $\mathcal{M}$  containing the initial point  $\mathbf{x}_s$  gets fully covered.

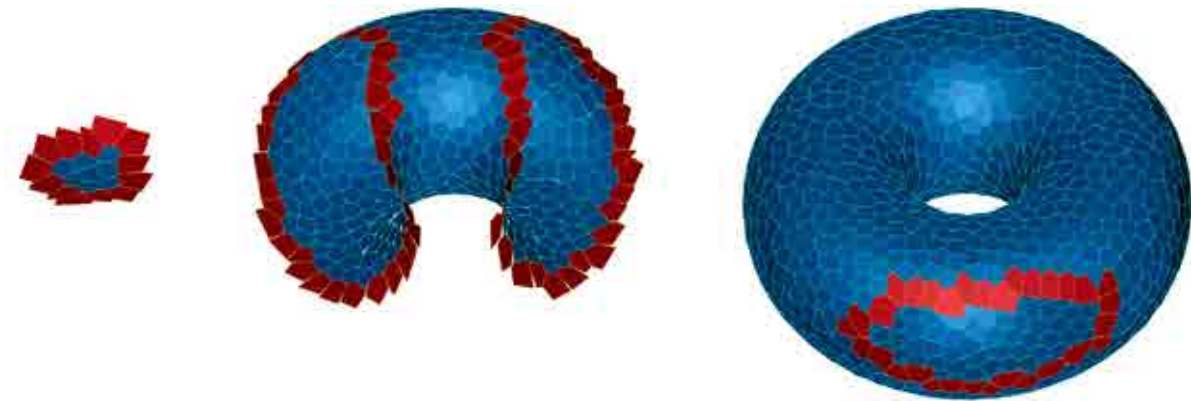


Figure 5.4: Progression of the algorithm on the surface of the torus defined by  $(\sqrt{x^2 + y^2} - R_1)^2 + z^2 - R_2^2 = 0$ , with  $R_1 = 0.5$ ,  $R_2 = 0.8$ , and using the continuation parameters  $r = 0.1$  and  $\epsilon = 0.25$ . Red and blue polygons correspond to open and closed charts.

As an example, Fig. 5.4 illustrates the progression of the algorithm on tracing the surface of a torus from a given point, where open and closed charts are colored in red and blue, respectively. Since a torus is a 2-dimensional manifold, the balls  $\mathcal{B}_i$  are circles in this case, and the polytopes  $\mathcal{P}_i$  are polygons.

### 5.3.3 Focusing on the Path to the Goal

Once a whole atlas for the explored component is available, a graph can be built whose nodes are the chart centers and whose edges represent the neighboring relations between charts. If a path exists from  $\mathbf{x}_s$  to  $\mathbf{x}_g$ , the chart centered at  $\mathbf{x}_g^*$  must be connected to the atlas built from  $\mathbf{x}_s$  and, thus, a solution path can be determined by searching the graph using standard graph search methods. If the chart centered at  $\mathbf{x}_g$  is not reached, path non-existence can be established at the considered value of  $r$ . Note, however, that this process requires the exploration of the whole component reachable from  $\mathbf{x}_s$  and does not capitalize the fact that in many cases we are only concerned with computing a path between two given configurations. This fact can be taken into account by properly guiding the expansion of the atlas.

To guide the search towards  $\mathbf{x}_g$ , we may use several strategies. A basic approach is to use a Greedy Best-First strategy where the chart to be expanded is the one yielding a minimum expected cost to reach the goal. Another one is using an A\* strategy [117], which finds a least-cost path by expanding the chart with the lowest known heuristic cost, while keeping a sorted priority queue of alternative path segments along the way. In contrast with A\*, the Greedy Best-First strategy does not necessarily generate all of the neighbors of a chart under expansion. The generation of children charts proceeds only while such charts have higher cost than the parent, which strongly reduces the final number of charts produced. However, the path obtained by the Greedy Best-First method is not necessarily optimal, and the generation of an optimal, resolution-complete path requires an A\* strategy, which, usually, explores a larger portion of the manifold due to its optimality, but provides good results in reasonable times.

In any case, the search is stopped as soon as the goal is connected to the rest of the atlas, or when the full manifold has been covered, meaning that there is no possible path connecting the start and goal configurations. The algorithm implementing the A\* strategy is outlined next.

### 5.3.4 The Planner Algorithm

Algorithm 5.1 gives the pseudo-code of the singularity-free path planner herein proposed. It implements an A\* search strategy, that takes into account given cost and heuristic functions. The first of these functions evaluates the cost of the transition between configurations and the second provides a lower bound of this cost. In our implementation, we use the Euclidean distance in  $\mathbb{R}^m$

for both functions and, thus, the planner will compute shortest paths on  $\mathcal{M}$ , up to the resolution used to define the atlas. More sophisticated cost functions could be used though, considering for instance the travel time or avoiding collisions. In the latter case, the cost function only has to assign an infinite cost to the transitions between charts that cause collisions [118]. At the beginning, the algorithm defines charts on the start and goal configurations and uses them to initialize the atlas (lines 1 to 3). Additionally, it defines the set of charts from where the search can be expanded  $\mathcal{H}$  (line 4) and the set of charts already processed along the search,  $\mathcal{V}$  (line 5). It then initializes the pointers to the best parent for each chart (line 6), the cost to reach the initial chart (line 7), and the heuristic estimating the cost from this chart to the goal (line 8). After that, the algorithm iterates while there are charts in  $\mathcal{H}$  and the goal chart has not been reached yet (lines 10 to 31). In this iteration, the chart  $C_i$  with minimum expected cost to the goal is extracted from  $\mathcal{H}$  (line 11). If  $C_i$  is not the goal chart (line 12), and while it is not a closed chart (line 13), all its neighbors are generated (lines 14 to 20). Note that charts whose center is out of the domain  $\mathcal{D}$  are considered closed and, thus, their neighbors are not generated. If necessary, the neighbors are generated by selecting a vertex of  $\mathcal{P}_i$  not in  $\mathcal{B}_i$  and using this vertex to define the parameters as in (5.7). The process of generating a neighboring chart is repeated until the conditions given in (5.8) and (5.9) hold (line 19). When this happens, the new chart is added to the atlas, coordinating it with the charts already in it (line 20). When all the neighbors for  $C_i$  are eventually generated, the chart is added to  $\mathcal{V}$  (line 21) and the search is expanded from it. For each of the non-processed neighbors, the tentative cost to the neighbor via  $C_i$  is computed (line 26). Charts not yet in  $\mathcal{H}$ , or charts where the tentative cost is lower than the best cost computed up to the moment, are added to  $\mathcal{H}$  (line 28), changing their parent chart (line 29), setting their new cost (line 30), and, finally, computing the heuristic estimation of the cost to the goal (line 31). At the end of the search, if the goal was found, the path connecting  $\mathbf{x}_s$  and  $\mathbf{x}_g$  is derived using the pointers to the parent chart stored in  $p$  (line 33). Otherwise, an empty path is returned (line 35) indicating that it is not possible to determine a singularity-free path at the used resolution.

The cost of the algorithm at each step is dominated by the cost of two searches among the set of charts: one to find the potential neighbors of a new chart when adding it to the atlas (line 20), and another one to find an open chart from which to continue the search. The performance of the first search can be increased using a  $k$ -d tree storing the centers of the charts. If  $\mathcal{H}$  is implemented using a heap, both the extraction of the next chart to be expanded (line 11) and the insertion of a new chart in  $\mathcal{H}$  (line 28) are logarithmic in the number of expandable charts.

**Algorithm 5.1:** Singularity-free path planner.

---

**PathComputation**( $\mathbf{F}, \mathbf{x}_s, \mathbf{x}_g, r, \epsilon$ )  
**input** : The functions defining  $\mathcal{M}$ ,  $\mathbf{F}$ , the start and goal configurations,  $\mathbf{x}_s$  and  $\mathbf{x}_g$ , a function, CST, giving the cost of the transition between configurations, a heuristic function, HST, giving a lower bound of the cost of the transition between configurations, and the parameters  $r, \epsilon$  used to build the atlas.  
**output**: A singularity-free path connecting  $\mathbf{x}_s$  and  $\mathbf{x}_g$ .

- 1  $C_s \leftarrow \text{NEWCHART}(\mathbf{F}, \mathbf{x}_s, r)$
- 2  $C_g \leftarrow \text{NEWCHART}(\mathbf{F}, \mathbf{x}_g, r)$
- 3  $\mathcal{A} \leftarrow \{C_s, C_g\}$
- 4  $\mathcal{H} \leftarrow \{C_s\}$
- 5  $\mathcal{V} \leftarrow \emptyset$
- 6  $p(s) \leftarrow 0$
- 7  $c(s) \leftarrow 0$
- 8  $h(s) \leftarrow \text{HST}(\mathbf{x}_s, \mathbf{x}_g)$
- 9  $C_i \leftarrow C_s$
- 10 **while**  $\mathcal{H} \neq \emptyset$  **and**  $C_i \neq C_g$  **do**
- 11      $C_i \leftarrow \text{EXTRACTMIN}(\mathcal{H}, h)$
- 12     **if**  $C_i \neq C_g$  **and**  $c(i) < \infty$  **then**
- 13         **while**  $\text{OPEN}(C_i)$  **do**
- 14              $\alpha \leftarrow r$
- 15              $\mathbf{s} \leftarrow \text{VERTEX}(\mathcal{P}_i)$  s.t.  $\mathbf{s} \notin \mathcal{B}_i$
- 16             **repeat**
- 17                  $C_j \leftarrow \text{CREATENEIGHBORCHART}(C_i, \alpha, \mathbf{s})$
- 18                  $\alpha \leftarrow \alpha \cdot 0.9$
- 19             **until not**  $\text{SIMILARCHARTS}(C_i, C_j, \epsilon)$
- 20              $\mathcal{A} \leftarrow \mathcal{A} \cup \{C_j\}$
- 21          $\mathcal{V} \leftarrow \mathcal{V} \cup \{C_i\}$
- 22          $\mathbf{x}_i \leftarrow \text{CENTER}(C_i)$
- 23         **forall**  $C_j \in \text{NEIGHBOUR}(C_i)$  **do**
- 24             **if**  $C_j \notin \mathcal{V}$  **then**
- 25                  $\mathbf{x}_j \leftarrow \text{CENTER}(C_j)$
- 26                  $t \leftarrow c(i) + \text{CST}(\mathbf{x}_i, \mathbf{x}_j)$
- 27                 **if**  $C_i \notin \mathcal{H}$  **or**  $t < c(j)$  **then**
- 28                      $\mathcal{H} \leftarrow \mathcal{H} \cup \{C_j\}$
- 29                      $p(j) \leftarrow i$
- 30                      $c(j) \leftarrow t$
- 31                      $h(j) \leftarrow t + \text{HST}(\mathbf{x}_j, \mathbf{x}_g)$
- 32 **if**  $C_i = C_g$  **then**
- 33      $\text{RETURN}(\text{RECONSTRUCTPATH}(s, g, p))$
- 34 **else**
- 35      $\text{RETURN}(\emptyset)$

---

## 5.4 Examples

The performance of the method in computing singularity-free paths is next illustrated in two situations: first on a fictitious three-dimensional C-space, and then on a 3-RRR manipulator. The former case is chosen for its simplicity, to illustrate and visualize the method in detail, and the latter shows the method's performance on a real application of considerable complexity. Note that, in all cases, no use is made of closed-form parameterizations of  $\mathcal{C}$ . Also, despite the singularity locus and the workspace are shown for reference in the figures (computed using the methods in Chapters 3 and 4), explicit knowledge of such sets is not used by the planner in its computations. All results reported have been obtained from an implementation in C of the method available in [65], executed on a MacBook Pro equipped with a 2.66 GHz Intel Core i7 processor.

### 5.4.1 A Three-dimensional Example

Consider the fictitious C-space defined implicitly by

$$\Phi(q_1, q_2, q_3) = q_1 - \sigma \cos(\omega (q_2^2 + q_3^2)) = 0,$$

with  $\sigma = 0.5$  and  $\omega = 0.25$ . It is not difficult to see that this equation defines a sinusoidal surface in the space of  $\mathbf{q} = [q_1, q_2, q_3]^T$ , shown in Fig. 5.5 for  $\mathbf{q} \in [-1, 1] \times [-20, 20] \times [-20, 20]$ .

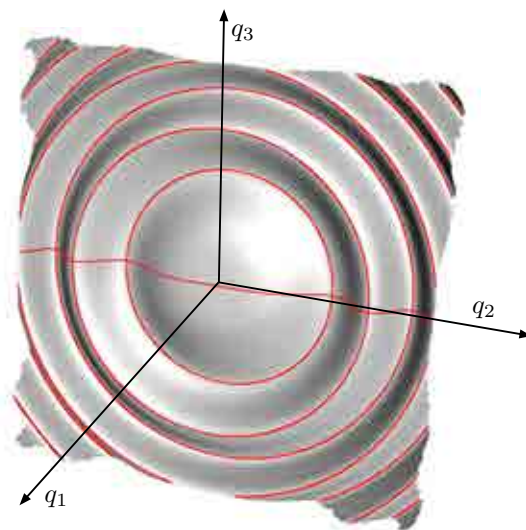


Figure 5.5: A fictitious three-dimensional C-space with its singularities highlighted in red.



Let us assume for this example that the vector of actuated degrees of freedom is  $\mathbf{v} = [q_1, q_2]^T$ , so that  $\mathbf{y} = [q_3]$ . Then, the forward singularities corresponding to such choice are given, using differentiation, by the equation

$$\det(\Phi_{\mathbf{y}}) = \frac{\partial \Phi}{\partial q_3} = 2 \sigma \omega q_3 \sin(\omega (q_2^2 + q_3^2)) = 0,$$

which, for non-null parameters  $\omega$  and  $\sigma$ , holds whenever  $\omega (q_2^2 + q_3^2) = n \pi$  with  $n \in \mathcal{Z}$ , or when  $q_3 = 0$ . Thus the singularity locus is formed by a family of concentric circles and a sinusoidal line, which are the red curves shown in Fig. 5.5. Note that, in accordance to the geometric interpretation of the singularities in Section 2.2, the points of such locus are those where the tangent plane to the C-space projects vertically on a line, in the space defined by  $\mathbf{v} = [q_1, q_2]^T$ .

Figure 5.6 shows the results obtained by the planner, when trying to connect the configurations  $\mathbf{q}_s = [0, 4.33, -0.38]^T$  and  $\mathbf{q}_g = [0, -4.33, -0.38]^T$ . To compare the results, the figure shows the computed path, in green, when navigating  $\mathcal{C}$  (left figure) and  $\mathcal{M}$  (right figure), so that the crossing of singularities is permitted and avoided, respectively. In both cases, the planner

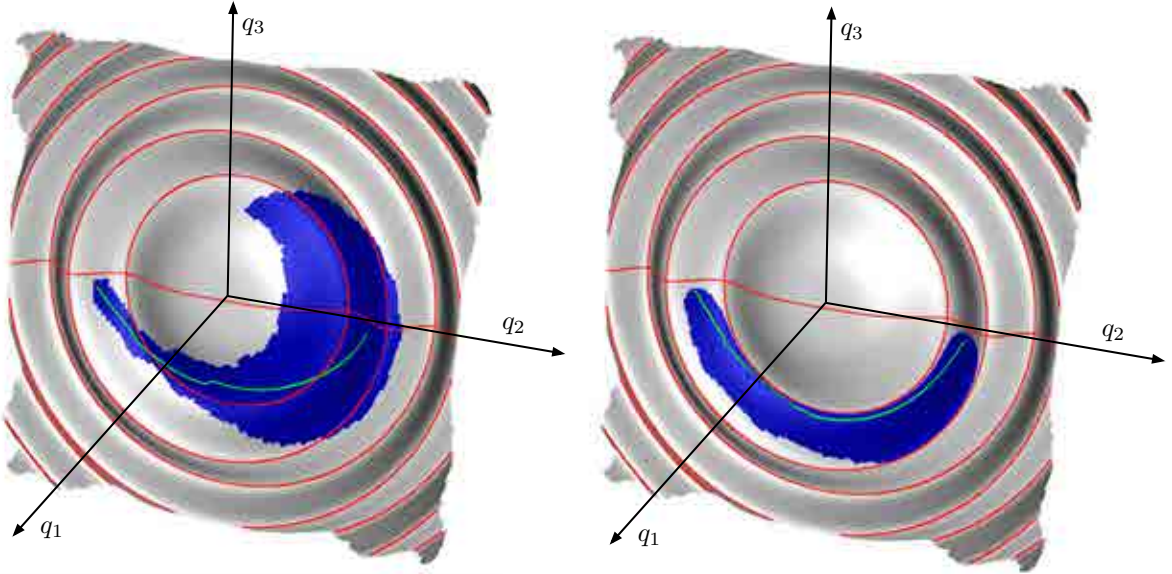


Figure 5.6: The path computed by the proposed algorithm when neglecting and considering singularity avoidance (left and right, respectively) on the manifold of Fig. 5.5. In the plots, the singularity locus is highlighted in red, the charts explored to connect the two query configurations are shown as blue polygons, and the final returned path is shown in green. While the path on the left figure crosses the singularity set twice, the path on the right figure is singularity-free.

returns the shortest path up to the resolution of the generated atlas. The charts of these atlases are shown in blue in both figures.

Note that the path on the left figure crosses the considered singularity locus twice, while the path on the right figure, although longer, avoids crossing it. While the latter path approaches the singularity locus, we note that a certain clearance is always guaranteed, because the value of  $|b|$  is always kept below a given threshold  $b_{max}$ . In this case, the value  $b_{max} = 12$  was used, resulting in the shown path, but alternative paths with a larger clearance can be obtained if desired, by simply reducing  $b_{max}$ . Nevertheless,  $b_{max}$  should always be chosen larger than the maximum of  $|b(\mathbf{q}_s)|$  and  $|b(\mathbf{q}_g)|$ , so as to guarantee that the domain  $\mathcal{D}$  within which the search is restricted (Section 5.2) includes both  $\mathbf{x}_s$  and  $\mathbf{x}_g$ . The computation of the singularity-free path took 0.04 seconds in this example, using the continuation parameters  $r = 0.25$  and  $\epsilon = 0.25$ .

### 5.4.2 The 3-RRR Manipulator

Now, let us consider the same planar 3-RRR manipulator of Section 3.5.1, shown in Fig. 5.7, and let the three joints attached to the ground be the actuated ones, so that the geometric parameters of the 3-RRR indicated in Table 3.2 are also assumed here. Due to its relatively large workspace, and the fact that mounting the actuators on the base reduces weight in the mobile equipment, this is the most common architecture for a 3-DOF planar parallel manipulator [19]. Although the planning of singularity-free paths has been addressed for particular instances of this manipulator [119], no general path planner has been given yet to avoid its singularities, to the best of our knowledge.

In the considered example, the geometry of the manipulator is set so that the two links of each leg are in a different plane and, thus, they can not collide. In this way we can illustrate the fact that, although legs might change their working mode along the computed path, the controllability of the manipulator will always be maintained during the movement, despite initial doubts about this fact expressed in the literature. Springs on intermediate leg joints or the exploitation of link inertias are said to be necessary to force the flipping of the legs [19, page 72], but we show that such flips can be performed without crossing forward singularities. In fact, the possibility of switching working modes has been also recently demonstrated on real prototypes through simple active-joint control [120].

The same formulation of Section 3.5.1 can be used here to write Eq. (5.1). The singularity locus of this manipulator is two-dimensional, and its shape is known to be rather complex. Some of its slices for a fixed value of  $\theta_8$  have been shown in Fig. 3.9, and the slice corresponding to  $\theta_8 = 0$  is reproduced in Fig. 5.7(top). From the figure, it may seem that the safe motion range of the manipulator is severely reduced by the presence of singularities. For instance, it appears that  $P_7$  cannot be moved from  $P_s$  to  $P_g$  while keeping  $\theta_8 = 0$ , but note that we are actually

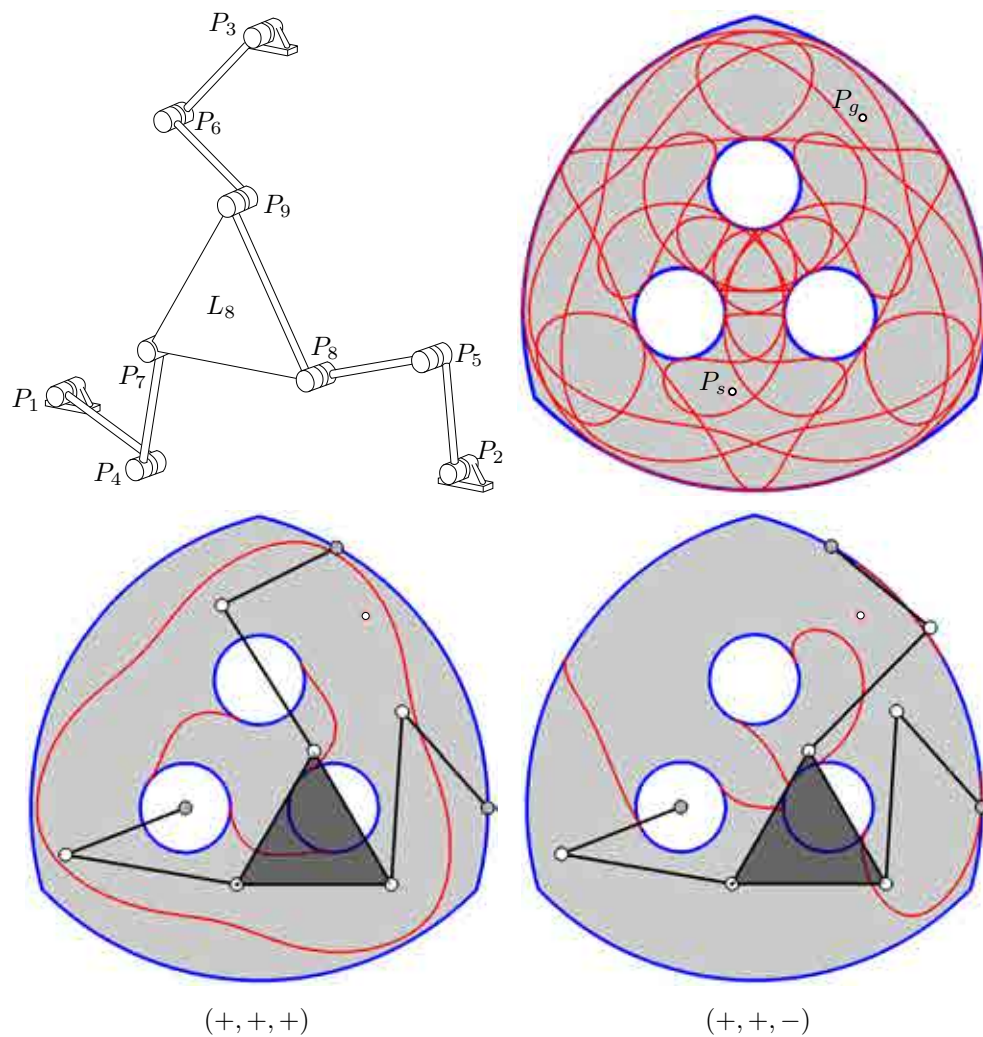


Figure 5.7: Top: 3-RRR manipulator and constant orientation workspace (in gray) with boundaries (in blue) and interior singularities (in red) for the geometry of Table 3.2, assuming  $\theta_8 = 0$ . Bottom: Layers corresponding to two working modes of the manipulator, differing on the sign of the third leg triangle only, and their corresponding singular curves.

observing a projection of the C-space onto the  $(x_7, y_7)$ -plane, and that for each pose  $[x_7, y_7, \theta_8]^T$  of the platform there are up to eight possible inverse kinematic solutions of the manipulator. Each such solution corresponds to a different working mode of the mechanism, identified by the sign triple  $(\kappa_1, \kappa_2, \kappa_3)$ , where  $\kappa_i$  gives the orientation of the triangle  $P_i P_{i+3} P_{i+6}$  [19]. The C-space, thus, has more structure than it looks. It is formed by several “layers”, each one of them corresponding to a different working mode, and if we project such layers separately, larger singularity-free regions get revealed.

To illustrate this, the layers corresponding to the  $(+, +, +)$  and  $(+, +, -)$  working modes

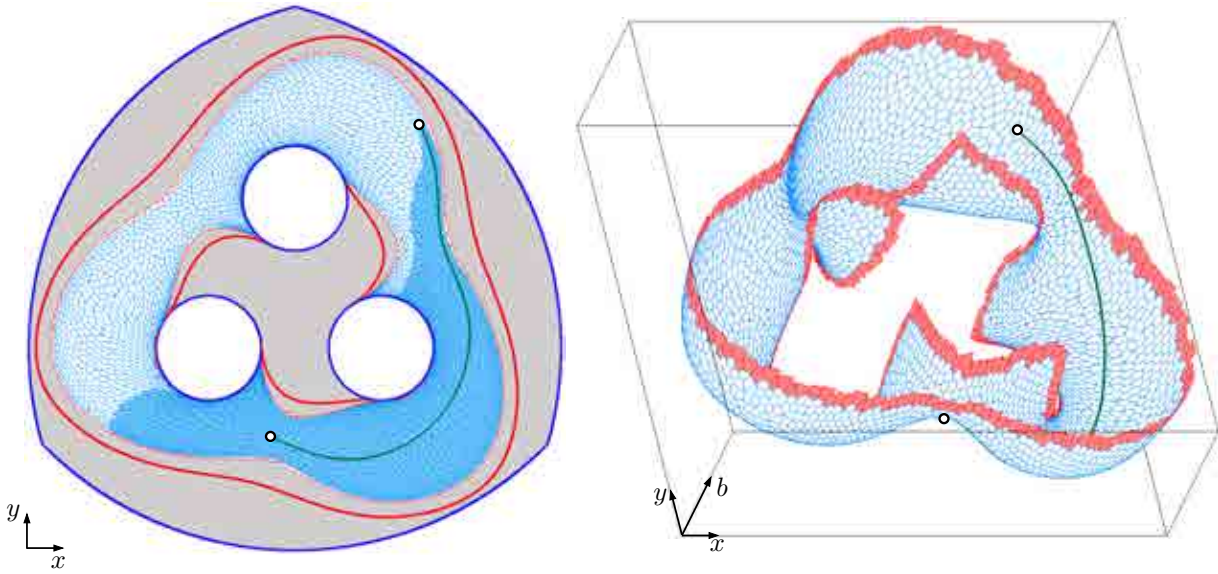


Figure 5.8: Results of the method on computing a singularity-free path to connect points  $P_s$  and  $P_g$  indicated in Fig. 5.7, assuming the platform orientation fixed to  $\theta_8 = 0$ . All results are shown projected onto the  $(x_7, y_7)$ -plane (left) and onto the  $(x_7, y_7, b)$ -space (right). The obtained path is shown in green overlaid onto the atlas of the singularity-free component of  $\mathcal{C}$  attainable from the start configuration. The atlas charts are shown colored in white, with blue or red edges depending on whether they lie inside or outside of the domain  $\mathcal{D}$ . The part actually explored by the algorithm to connect the two configurations is shown shadowed in blue on the left figure.

for  $\theta_8 = 0$  are shown in Fig. 5.7(bottom), together with the portion of the singularity curve lying on them, and a representative configuration in each case, with  $P_7$  coinciding on  $P_s$ . As we see,  $P_s$  and  $P_g$  are actually connectable through a singularity-free path lying entirely on the  $(+, +, +)$  layer. The proposed planner is able to compute such a path in only 0.05 seconds, using  $b_{max} = 3.333$ ,  $r = 0.2$ , and  $\epsilon = 0.25$ , obtaining the results shown in Fig. 5.8, where the position vectors  $\mathbf{p}_s = [0.4, -0.6]^T$  and  $\mathbf{p}_g = [1.4, 1.5]^T$  have been assumed for  $P_s$  and  $P_g$ . Note from the right plot how the manifold  $\mathcal{M}$  reaches higher values of  $b$  as it approaches the singularities, but the found path avoids these zones and the algorithm returns the shortest possible path on  $\mathcal{M}$ . The partial atlas generated to resolve the planning query is shadowed in blue on the left plot.

Constant-orientation paths in other layers can be computed if desired, but the full potential of the method comes out when solving complex planning queries in which the platform is allowed to rotate and the manipulator has to change its working mode along the path, to avoid passing through singular configurations (Fig. 5.9). Due to the difficulty of illustrating a 3-dimensional C-space clearly, the obtained path is shown as a series of pictures in this case, showing several snapshots of the movement of the manipulator when following the path. The

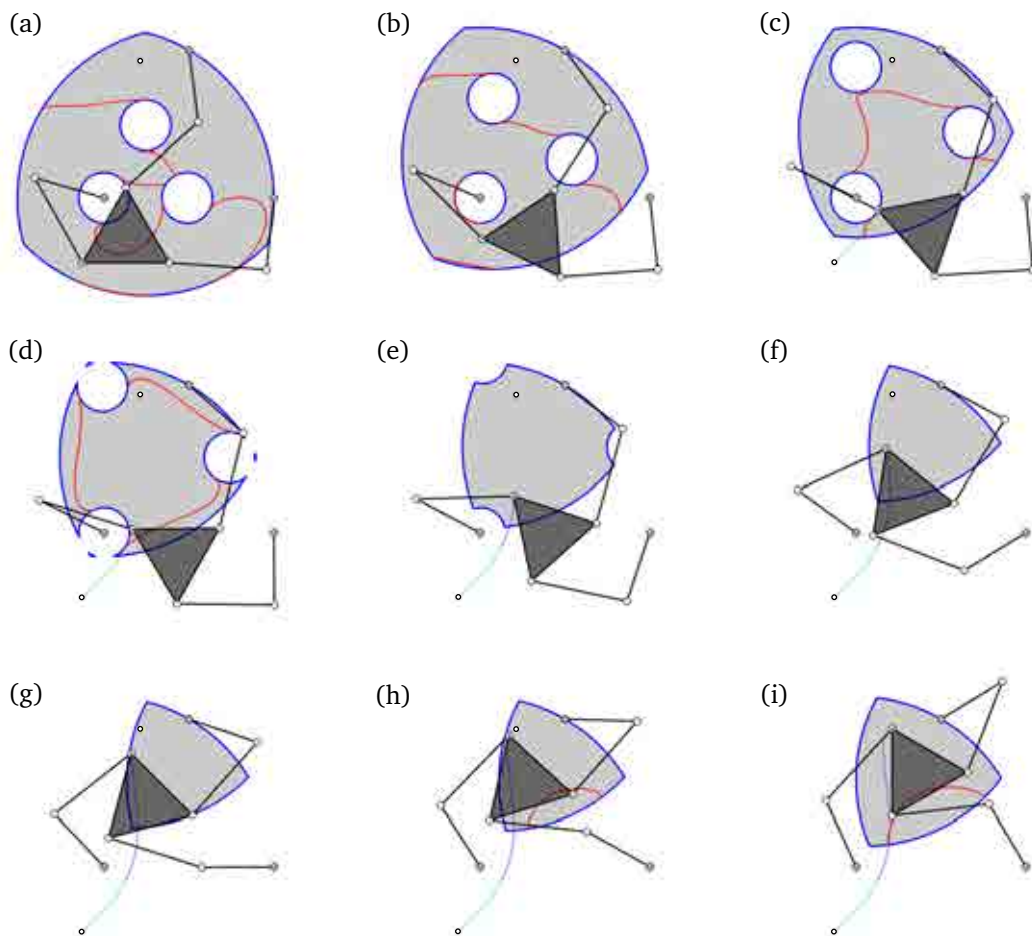


Figure 5.9: Several steps of the movement of the manipulator while performing the computed path between configurations (a) and (i). The manipulator is shown overlaid on the constant orientation workspace (in gray with its boundaries in blue) and the singular curves at the corresponding working mode (in red). Changes of working mode occur between steps (c) and (d) for the first leg, and between (g) and (h) for the second leg. An animation of this figure is available in the supplementary downloadable material associated with this document.

first and last snapshots, (a) and (i), show the start and goal configurations to be connected, which are given by  $\mathbf{p}_s = [-0.3, -0.9]^T$  and  $\theta_s = 0$ , and  $\mathbf{p}_g = [0.5, 1.9]^T$  and  $\theta_g = -\frac{\pi}{2}$ , assuming the working modes  $(+, -, -)$  and  $(-, +, -)$  respectively. Note that at least two legs should change their working mode along the path, and this can be seen to happen between snapshots (c) and (d) for the first leg, and between (g) and (h) for the second leg. The workspace and the singularities shown in each picture correspond to the orientation of the platform at that

moment only, and can be seen to vary with the orientation. Indeed, the path is obtained by exploring a 3-dimensional C-space, with a 2-dimensional singularity locus, defined both in a higher-dimensional ambient space, and each picture in Fig. 5.9 shows a constant- $\theta_8$  slice of such C-space, projected onto the  $x_7$  and  $y_7$  coordinates. The computation of the singularity-free path took 1 second in this example, using  $r = 0.3$  and  $\epsilon = 0.25$ . An animated version of Fig. 5.9 can be seen in a video associated to this document, as part of a larger movement planned through three waypoints.

# 6

## Adding Further Constraints

The path planner of Chapter 5 measures the clearance of the path relative to the singularity locus in terms of the determinant of a Jacobian matrix. It is often argued that such a measure lacks physical significance [121], but in the early design stages of the manipulator only the kinematic architecture has been decided, and further information leading to more meaningful measures, such as the actuators' positioning accuracy, their maximum force/velocity deliverable, or the link inertias, is often not yet available. The planner of Chapter 5 already covers the needs of a robot designer in such situations, because it can compute a reasonably accurate roadmap of the feasible movements, but for subsequent design stages, or for motion planning during normal operation, further physical limitations should also be taken into account. This chapter presents a way to do so, exemplified in the particular case of classical or cable-driven hexapods. Following the spirit of [122, 123] the clearance of the path relative to the singularity locus is defined as the one that results from allowing wrench-feasible configurations only, i.e., those in which the leg forces remain within given limits, for any platform wrench subject to six-dimensional uncertainty (Section 6.1). The method relies on defining a system of equations whose solution manifold corresponds to the wrench-feasible C-space of the manipulator, so that maneuvering through such manifold guarantees singularity avoidance at all times, while maintaining leg forces and lengths within their allowable bounds (Section 6.2). This manifold, as well as any of its submanifolds obtained by coupling some of the pose parameters under mild conditions, is shown to be smooth everywhere, which allows exploiting the same higher-dimensional continuation strategy of Chapter 5 to derive connection paths between given configurations. The approach is illustrated on two specific hexapods, the Stewart platform and the cable-driven NIST Robocrane (Section 6.3), but hints on how to extend the developments to account for alternative constraints or manipulator architectures are given in the end (Section 6.4).

## 6.1 Problem Statement

For the purpose of this chapter, a hexapod consists of a moving platform connected to a fixed base by means of six legs, which can be universal-prismatic-spherical chains, as in the Stewart platform (Section 3.5.2), or cables winding around independent winches, as in cable-driven robots (Fig. 6.1). The leg lengths  $d_i$  can be varied within prescribed limits  $[\underline{d}_i, \overline{d}_i]$  by actuating the prismatic joints or the winch drives, which allows to control the six degrees of freedom of the platform within a given workspace.

The C-space of a hexapod can be implicitly defined in a way similar to that followed in Section 3.5.2. We attach fixed and moving reference frames  $\mathcal{F}_1$  and  $\mathcal{F}_2$  to the base and platform links, centered in  $O$  and  $P$ , respectively (Fig. 6.1), and represent the platform pose by the pair  $\mathbf{q} = (\mathbf{p}, \mathbf{R}) \in SE(3)$  subject to the constraints

$$\mathbf{p} = \mathbf{a}_i + \mathbf{d}_i - \mathbf{R} \mathbf{b}_i, \quad (6.1)$$

$$\mathbf{d}_i^\top \mathbf{d}_i = d_i^2, \quad (6.2)$$

$$d_i \in [\underline{d}_i, \overline{d}_i],$$

for  $i = 1, \dots, 6$ , where  $\mathbf{p}$ ,  $\mathbf{a}_i$ , and  $\mathbf{b}_i$  are the position vectors of points  $P$ ,  $A_i$ , and  $B_i$ ,  $\mathbf{R}$  is the rotation matrix providing the orientation of the platform, and  $\mathbf{d}_i$  is the vector from  $A_i$  to  $B_i$ . It has been assumed that  $\mathbf{p}$ ,  $\mathbf{a}_i$ , and  $\mathbf{d}_i$  are expressed in frame  $\mathcal{F}_1$ , while  $\mathbf{b}_i$  is expressed in  $\mathcal{F}_2$ .

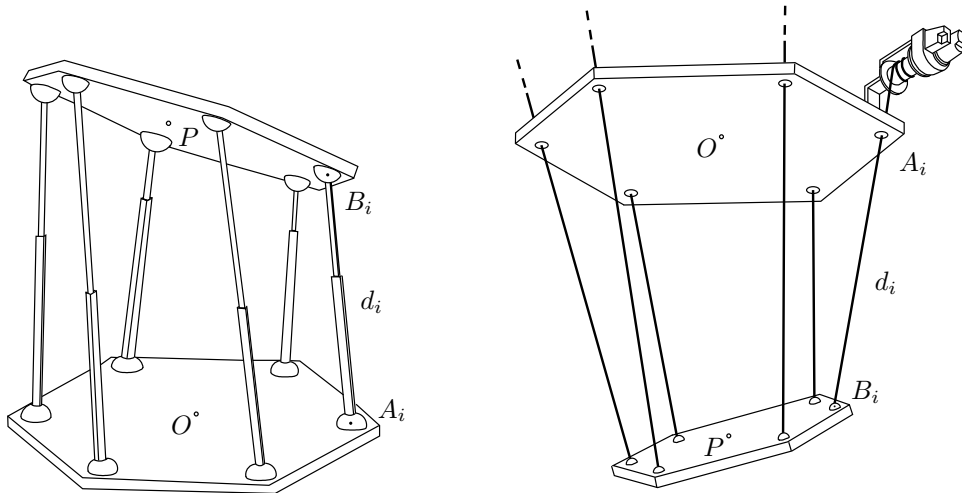


Figure 6.1: Two hexapods with rigid (left) and cable (right) legs. The cable-driven manipulator is maintained under a stable position due to the action of gravity.



As shown in Chapter 3, the orthonormality of  $\mathbf{R}$  can be imposed in a variety of ways, but we here resort to the parameterization provided by the tilt and torsion angles,  $\tau = \{\phi, \theta, \sigma\}$ , for which

$$\mathbf{R} = \mathbf{R}_z(\phi) \mathbf{R}_y(\theta) \mathbf{R}_z(\sigma - \phi), \quad (6.3)$$

because of its intuitive interpretation and attractive properties [73].

In practice, not all configurations satisfying Eqs. (6.1)–(6.3) are allowed, because the leg forces must be kept within their admissible ranges. This is specially important in cable-driven manipulators, where all cables must be kept in tension to control the platform, but also in any other hexapod to avoid excessive stress on the actuators or eventual breakage of the structure. For this reason, we require any configuration to be *wrench-feasible*, in the sense that it must allow the platform to equilibrate any external wrench  $\hat{\mathbf{w}}$  acting on it, subject to lie inside a prescribed six-dimensional region  $\mathcal{K} \subset \mathbb{R}^6$ .

The significance of  $\mathcal{K}$  depends on the particular context of application. In payload transportation, for instance,  $\mathcal{K}$  may be given by the gravitational wrench acting on the platform and slight perturbations introduced by inertia forces or external agents like the wind. In contact situations,  $\mathcal{K}$  might be given by the wrench that should be suppliable to the environment, which is in general subject to six-dimensional uncertainty. Specifically, the wrench-feasibility requirement on a given configuration  $\mathbf{q}$  implies that for each wrench  $\hat{\mathbf{w}} \in \mathcal{K}$  there must be a vector

$$\mathbf{f} = [f_1, \dots, f_6]^\top \in \mathcal{D} = [\underline{f}_1, \overline{f}_1] \times \dots \times [\underline{f}_6, \overline{f}_6]$$

of leg forces satisfying

$$\mathbf{J} \mathbf{f} = \hat{\mathbf{w}},$$

where  $\mathbf{J}$  is the  $6 \times 6$  screw Jacobian of the manipulator, which depends on the configuration  $\mathbf{q}$ , and  $[\underline{f}_i, \overline{f}_i]$  is the range of force magnitudes that can be resisted by the  $i$ th leg. For cable-driven manipulators it must be  $\underline{f}_i > 0$ .

Here,  $\mathcal{K}$  will be assumed to be a six-dimensional ellipsoid centered at  $\hat{\mathbf{w}}_0$ , defined as

$$(\hat{\mathbf{w}} - \hat{\mathbf{w}}_0)^\top \mathbf{E} (\hat{\mathbf{w}} - \hat{\mathbf{w}}_0) \leq 1,$$

where  $\mathbf{E}$  is a  $6 \times 6$  positive-definite symmetric matrix. In practice, this ellipsoid can be constructed by propagating known bounds on other variables related to  $\hat{\mathbf{w}}$ , using the tools of an ellipsoidal calculus for example [127], and in a most general setting it can be considered to be dependent on the configuration  $\mathbf{q}$ , because both  $\hat{\mathbf{w}}_0$  and  $\mathbf{E}$  may be functions of  $\mathbf{q}$  given by the specific application of the robot, i.e.,  $\hat{\mathbf{w}}_0 = \hat{\mathbf{w}}_0(\mathbf{q})$  and  $\mathbf{E} = \mathbf{E}(\mathbf{q})$ . To have an idea, note that hexapod robots are currently used in a variety of contexts (Fig. 6.2), including machine tool



Figure 6.2: From left to right and top to bottom: The Okuma PM-600 machining center (top), and a close-up of the hexapod platform holding the tools (bottom) [124]; a worker programming and monitoring the TETRA RoboCrane from the NIST [125], used to pick up and place a steel beam in a stanchion; a cable-driven hexapod for aircraft building and maintenance [125]; and a swell simulator [126].

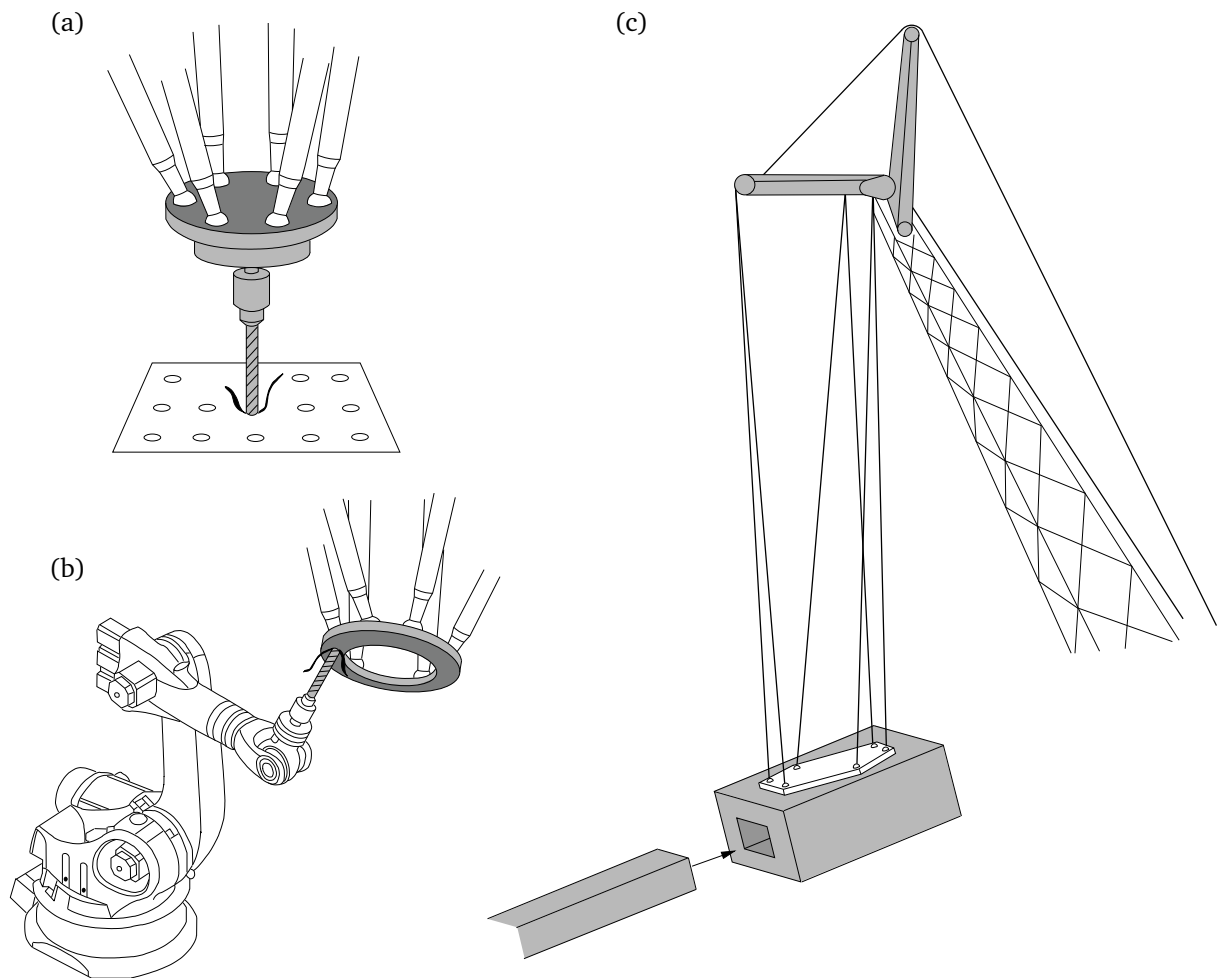


Figure 6.3: Some schematic examples of typical situations encountered in hexapod applications.

technology, flight and automotive simulation, payload positioning, inspection, aircraft building and maintenance, swell simulation, oil-well fire fighting, or pipe assembling, to name a few, and each particular case demands a specific definition of  $\mathcal{K}$ . A typical situation encountered in machine tools like those fabricated by the Okuma [124], Ingersoll [128], Symétrie [126], Toyota [129], or Mikrolar companies [130], for example, is depicted in Fig. 6.3(a). In this case, the legs must counteract the efforts given by the drilling operation, which can be modelled as a six-dimensional ellipsoid, with the matrix  $\mathbf{E}$  accounting for the uncertainty of the operation, centered at  $\hat{\mathbf{w}}_0 = \hat{\mathbf{w}}_{drill}$ . Note that if the weight of the platform-tool can be neglected, both  $\mathbf{E}$  and  $\hat{\mathbf{w}}_0$  are constant in the fixed frame, but that  $\hat{\mathbf{w}}_0$  must be shifted by  $\hat{\mathbf{w}}_{weight}$  otherwise, which varies with  $\mathbf{q}$  if such a frame is used. A more complex drilling operation could be performed if the workpiece moves attached to the hexapod and, at the same time, is being drilled in a

coordinate manner by a tool driven by another manipulator [Fig. 6.3(b)]. In such situation it would be simpler to express  $\mathcal{K}$  in the platform frame, where  $\mathbf{E}$  is constant and  $\hat{\mathbf{w}}_0$  is either constant or shifted by  $\hat{\mathbf{w}}_{weight}$ , as before. In applications involving a high payload that must be manipulated, like the one depicted in Fig. 6.3(c), it might be interesting to consider slight perturbations of the weight introduced by inertia forces or external agents like the wind. Since the weight always points in the same direction,  $\mathcal{K}$  can be defined in a frame that translates with the platform with the origin at the center of mass, where  $\mathbf{E}$  and  $\hat{\mathbf{w}}_0 = \hat{\mathbf{w}}_{weight}$  are constant. However, if the platform frame is used for some reason, both  $\mathbf{E}$  and  $\hat{\mathbf{w}}_0$  have expressions that are dependent on the configuration  $\mathbf{q}$ .

Let us define the *wrench-feasible C-space* of the manipulator,  $\mathcal{C}_w$ , as the set of configurations  $\mathbf{q}$  for which the platform is able to counteract all  $\hat{\mathbf{w}} \in \mathcal{K}$  with  $\mathbf{f} \in \mathcal{D}$ . Given two configurations in  $\mathcal{C}_w$ ,  $\mathbf{q}_s$  and  $\mathbf{q}_g$ , the goal is to compute a path on  $\mathcal{C}_w$  connecting them, if one exists, or to determine path non-existence otherwise. To this end, we next define a system of equations characterizing  $\mathcal{C}_w$  which allows using the same continuation approach of Chapter 5 to compute such a path.

## 6.2 Equations of the Wrench-feasible C-space

### 6.2.1 Formulating the Wrench-feasible C-space

For a given configuration  $\mathbf{q}$ , let  $\mathbf{f}_0$  be the vector of leg forces, or cable tensions, corresponding to  $\hat{\mathbf{w}}_0 \in \mathcal{K}$ , i.e.,

$$\mathbf{J} \mathbf{f}_0 = \hat{\mathbf{w}}_0. \quad (6.4)$$

By noting that  $\mathbf{J}(\mathbf{f} - \mathbf{f}_0) = \hat{\mathbf{w}} - \hat{\mathbf{w}}_0$ , it is easy to see that the set  $\mathcal{F}$  of leg forces  $\mathbf{f}$  corresponding to all  $\hat{\mathbf{w}} \in \mathcal{K}$  is the ellipsoid given by

$$(\mathbf{f} - \mathbf{f}_0)^\top \mathbf{B} (\mathbf{f} - \mathbf{f}_0) \leq 1,$$

where  $\mathbf{B} = \mathbf{J}^\top \mathbf{E} \mathbf{J}$ . This ellipsoid will be bounded in all directions or unbounded in some, depending on whether  $\det(\mathbf{J}) \neq 0$  or not. However, it is not difficult to see that  $\mathbf{J}$  is non-singular for all  $\mathbf{q} \in \mathcal{C}_w$  (Section 6.2.2), so that  $\mathcal{F}$  will always be a bounded ellipsoid in our case.

Now, for  $\mathbf{q}$  to be wrench-feasible, we must have  $\mathcal{F} \subseteq \mathcal{D}$ , which can be checked as follows. For each  $i = 1, \dots, 6$ , let  $\mathbf{v}_i \in \mathbb{R}^6$  be a vector satisfying

$$\left. \begin{array}{l} \mathbf{v}_i^\top \mathbf{B} \mathbf{v}_i = 1 \\ \mathbf{B}^i \mathbf{v}_i = \mathbf{0} \end{array} \right\}, \quad (6.5)$$

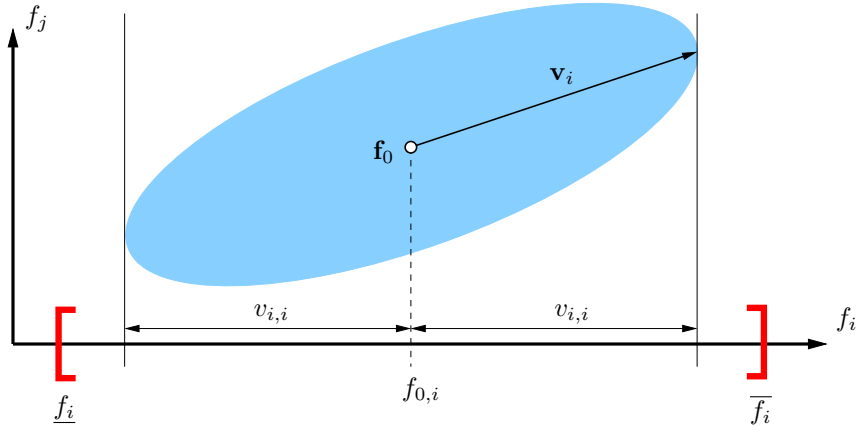


Figure 6.4: The vector  $\mathbf{v}_i$  provides the maximum and minimum values of  $f_i$ .

where  $B^i$  stands for the matrix  $B$  with its  $i$ th row removed. Observe that if  $\mathbf{J}$  is non-singular, then both  $B$  and  $B^i$  are full row rank, and if the  $i$ th component of  $\mathbf{v}_i$ ,  $v_{i,i}$ , is chosen non-negative, then there is exactly one vector  $\mathbf{v}_i$  satisfying Eq. (6.5). Using Lagrange multipliers, it can be shown that, for the solutions  $\mathbf{v}_i$  of Eq. (6.5) with  $v_{i,i} \geq 0$ ,  $\mathbf{f}_0 - \mathbf{v}_i$  and  $\mathbf{f}_0 + \mathbf{v}_i$  are the vectors in  $\mathcal{F}$  attaining the smallest and largest value along the  $i$ th coordinate (Fig. 6.4). Hence, when  $\det(\mathbf{J}) \neq 0$ ,  $\mathcal{F} \subseteq \mathcal{D}$  if, and only if,

$$f_{0,i} - v_{i,i} > \underline{f}_i,$$

and

$$f_{0,i} + v_{i,i} < \overline{f}_i,$$

for  $i = 1, \dots, 6$ . In order to use the continuation method, these two conditions have to be converted into equalities. Instead of using the circle equations of Section 4.4.1, we prefer here to use hyperbola equations because it simplifies the demonstrations of Section 6.2.2 and, also, naturally penalizes the paths closer to the limits, since they are longer. Therefore, we impose

$$(f_{0,i} - v_{i,i} - \underline{f}_i) \cdot s_i = 1, \quad (6.6)$$

$$(\overline{f}_i - f_{0,i} - v_{i,i}) \cdot t_i = 1, \quad (6.7)$$

with  $s_i \geq 0$  and  $t_i \geq 0$ , where  $s_i$  and  $t_i$  are newly-defined auxiliary variables. Looking at the first equation, for instance, it is clear that neither  $f_{0,i} - v_{i,i} - \underline{f}_i$  nor  $s_i$  can be zero, so that for any  $s_i > 0$ , it will always be  $f_{0,i} - v_{i,i} > \underline{f}_i$  as desired. In a similar way, the leg length constraints

$d_i \in [d_i, \bar{d}_i]$  are equivalent to imposing

$$(d_i - \underline{d}_i) \cdot (\bar{d}_i - d_i) \cdot g_i = 1, \quad (6.8)$$

in conjunction with  $g_i \geq 0$  acting on the new variable  $g_i$ .

At this point, let us consider the system formed by Eqs. (6.1)–(6.8), which we will write compactly as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad (6.9)$$

where  $\mathbf{x}$  refers to an  $n_x$ -vector encompassing all of its variables, and let us distinguish between the solution set of Eq. (6.9),

$$\mathcal{M} = \{\mathbf{x} : \mathbf{F}(\mathbf{x}) = \mathbf{0}\},$$

and the set

$$\mathcal{M}^+ = \{\mathbf{x} \in \mathcal{M} : v_{i,i} \geq 0, s_i \geq 0, t_i \geq 0, g_i \geq 0, \text{ for } i = 1, \dots, 6\}.$$

Note that every configuration  $\mathbf{q} \in \mathcal{C}_w$  has a corresponding point  $\mathbf{x} \in \mathcal{M}^+$  and, conversely, each point in  $\mathcal{M}^+$  projects down to one  $\mathbf{q} \in \mathcal{C}_w$ . Moreover, any continuous path in  $\mathcal{C}_w$  will also be represented by a continuous path in  $\mathcal{M}^+$  and viceversa, so that, similarly as done in Chapter 5, the original problem of computing a wrench-feasible path in  $\mathcal{C}_w$  from  $\mathbf{q}_s$  to  $\mathbf{q}_g$  can be reduced to that of computing a path in  $\mathcal{M}^+$  connecting the points  $\mathbf{x}_s$  and  $\mathbf{x}_g$  corresponding to  $\mathbf{q}_s$  and  $\mathbf{q}_g$ . In this case, the variables  $s_i$ ,  $t_i$ , and  $g_i$  play a role equivalent to that of the variable  $b$  in Chapter 5.

## 6.2.2 Properties of $\mathcal{M}$

Two properties regarding the structure of  $\mathcal{M}$  and  $\mathcal{M}^+$  can be exploited, which are essential in order to apply the continuation strategy of Chapter 5.

It is easy to see, first, that  $v_{i,i}$ ,  $s_i$ ,  $t_i$ , and  $g_i$  never vanish on  $\mathcal{M}$ , so that there does not exist any path traversing from  $\mathcal{M}^+$  to its complement  $\mathcal{M} \setminus \mathcal{M}^+$ . Indeed, if  $\mathbf{x}_s$  and  $\mathbf{x}_g$  are chosen with positive values for  $v_{i,i}$ ,  $s_i$ ,  $t_i$ , and  $g_i$ , then any continuous path on  $\mathcal{M}$  connecting  $\mathbf{x}_s$  and  $\mathbf{x}_g$  will entirely lie on  $\mathcal{M}^+$ , and it will correspond to a path on  $\mathcal{C}_w$  therefore. From Eqs. (6.6), (6.7), and (6.8) it follows directly that  $s_i$ ,  $t_i$  and  $g_i$  can never be zero on  $\mathcal{M}$ . The same property for  $v_{i,i}$  can be proved by contradiction. Let us assume that  $v_{i,i} = 0$  for some  $i$ . If we consider Eq. (6.5), then, by replacing  $\mathbf{B}^i \mathbf{v}_i = \mathbf{0}$  into  $\mathbf{v}_i^T \mathbf{B} \mathbf{v}_i = 1$ , we obtain the dot product of two vectors:  $\mathbf{v}_i$ , with  $v_{i,i} = 0$ , and the vector  $\mathbf{B} \mathbf{v}_i$ , whose components are all zero except that in position  $i$ . The result of this dot product is 0, which contradicts the first equation in (6.5), as it should be 1. As a result, the set  $\mathcal{M}^+$  and its complement  $\mathcal{M} \setminus \mathcal{M}^+$  are disconnected. In other words, when

trying to connect  $\mathbf{x}_s$  and  $\mathbf{x}_g$  by continuation on  $\mathcal{M}$ , the positivity constraints on  $v_{i,i}$ ,  $s_i$ ,  $t_i$ , and  $g_i$  will be implicitly fulfilled, and they can be safely neglected.

Second, it can be shown that  $\mathcal{M}$ , and in particular  $\mathcal{M}^+$ , is a six-dimensional smooth manifold everywhere, so that every point  $\mathbf{x}$  has a well-defined tangent space  $T_{\mathbf{x}}\mathcal{M}$ , which facilitates the application of the mentioned continuation strategy because no bifurcations, sharpnesses, or dimension changes will be found when exploring  $\mathcal{M}$ . If we can verify that  $\mathbf{F}(\mathbf{x})$  is a differentiable function with full rank Jacobian  $\mathbf{F}_{\mathbf{x}}$  everywhere, then the smoothness of  $\mathcal{M}$  will follow from the Implicit Function Theorem. By construction, all functions intervening in  $\mathbf{F}(\mathbf{x})$  are differentiable all over  $\mathcal{M}$ , and the Jacobian matrix  $\mathbf{F}_{\mathbf{x}}$  can be expressed in the following block-triangular form after re-organizing some equations and variables

$$\mathbf{F}_{\mathbf{x}} = \begin{pmatrix} \Phi_{\mathbf{y}} & & & & & \\ * & \mathbf{J} & & & & \\ & & \begin{matrix} 2\mathbf{v}_1^{\top} \mathbf{B} \\ \mathbf{B}^1 \end{matrix} & & & \\ * & & & \ddots & & \\ & & & & \begin{matrix} 2\mathbf{v}_6^{\top} \mathbf{B} \\ \mathbf{B}^6 \end{matrix} & \\ & * & & * & & \mathbf{S} \\ & * & & * & & \mathbf{T} \end{pmatrix},$$

where empty blocks represent zero-matrices and asterisks indicate non-zero blocks.

Due to the triangular structure of  $\mathbf{F}_{\mathbf{x}}$  it suffices to verify that the five blocks in the diagonal are full-rank in order to prove the smoothness of  $\mathcal{M}$ . The first block is the Jacobian matrix of the system  $\Phi(\mathbf{y}) = \mathbf{0}$  formed by Eqs. (6.1)–(6.3) and (6.8), which, by rearranging equations and differentiating with respect to  $\mathbf{y} = (\tau, \mathbf{p}, \mathbf{R}, \mathbf{d}_i, d_i, g_i)$ , takes the form

$$\Phi_{\mathbf{y}} = \begin{pmatrix} * & & \mathbf{I}_9 & & & \\ & * & * & -\mathbf{I}_{18} & & \\ & & & * & -2\mathbf{L} & \\ & & & & * & \mathbf{G} \end{pmatrix}. \quad (6.10)$$

Here  $\mathbf{L}$  and  $\mathbf{G}$  are  $6 \times 6$  diagonal matrices with diagonal elements  $-2d_i$  and  $(d_i - \underline{d}_i) \cdot (\overline{d}_i - d_i)$ , respectively. To see that  $\Phi_{\mathbf{y}}$  is full rank, observe that its last four block-columns comprise a non-singular square submatrix of maximum size, since its diagonal elements do not vanish over

$\mathcal{M}$  by virtue of Eq. (6.8) if we assume  $\underline{d}_i > 0$ .

The remaining four diagonal blocks of  $\mathbf{F}_x$  are the Jacobian matrices of Eqs. (6.4)–(6.7), differentiated with respect to the variables  $\mathbf{f}_0$ ,  $\mathbf{v}_i$ ,  $s_i$  and  $t_i$ , respectively, where the blocks  $\mathbf{S}$  and  $\mathbf{T}$  are  $6 \times 6$  diagonal matrices with elements  $f_{0,i} - v_{i,i} - \underline{f}_i$  and  $\bar{f}_i - f_{0,i} - v_{i,i}$ , respectively. The screw Jacobian  $\mathbf{J}$  can be shown to be full rank over  $\mathcal{M}$  by contradiction. Indeed, if  $\mathbf{J}(\mathbf{q})$  were rank deficient for some  $\mathbf{q}$ , then so would be  $\mathbf{B}$ , and therefore the kernel of  $\mathbf{B}$  would contain non-zero vectors. In such case, for some  $i$  all solutions of  $\mathbf{B}^i \mathbf{v}_i = 0$  would satisfy  $\mathbf{v}_i \in \ker(\mathbf{B})$  and, thus, it would be  $\mathbf{v}_i^T \mathbf{B} \mathbf{v}_i = 0$ , which contradicts Eq. (6.5) and, hence,  $\mathbf{J}$  cannot be rank deficient over  $\mathcal{M}$ . The  $6 \times 6$  block matrices involving  $\mathbf{B}$  and  $\mathbf{B}^i$  can only be rank deficient if  $v_{i,i} = 0$ , but this can never happen as we have already seen. All these blocks are therefore full rank over  $\mathcal{M}$ . Finally, it is clear that  $\mathbf{S}$  and  $\mathbf{T}$  are also full rank over  $\mathcal{M}$ , since their diagonal elements never vanish due to Eqs. (6.6) and (6.7), and this completes the proof of the smoothness of  $\mathcal{M}$  and, in particular, that of  $\mathcal{M}^+$ .

It is worth mentioning that not only  $\mathcal{M}$  and  $\mathcal{M}^+$  are smooth, but also any submanifold obtained by coupling any of the position or orientation parameters,  $\mathbf{u} = [\mathbf{p}, \tau]^T$ , either by adding equations of the form  $\mathbf{C}(\mathbf{u}) = \mathbf{0}$ , with a full-rank Jacobian  $\mathbf{C}_u$ , or by introducing parametric expressions of the form  $\mathbf{u} = \mathbf{u}(\lambda)$ , where both  $\mathbf{C}(\mathbf{u})$  and  $\mathbf{u}(\lambda)$  are assumed to be smooth functions. Indeed, it can be seen that the addition of  $\mathbf{C}(\mathbf{u}) = \mathbf{0}$  or  $\mathbf{u} - \mathbf{u}(\lambda) = \mathbf{0}$  to  $\mathbf{F}(\mathbf{x})$  results in a new full-rank block matrix over the first two block-columns of Eq. (6.10), which does not change the global rank of the differential  $\mathbf{F}_x$ . This allows to model situations in which the moving platform is not allowed to maneuver in six-dimensional space, such as contact situations, or constant position/orientation movements. For instance, if we need the point  $P$  of the platform to lie on a unit sphere, we could use the function  $\mathbf{C}(\mathbf{p}) = |\mathbf{p}|^2 - 1$ . If, instead, we would like the point to follow a certain Bézier patch, we could use the parameterization of such patch,  $\mathbf{p} = \mathbf{p}(\lambda)$ , to restrict the possible values of  $\mathbf{p}$ .

With this, the higher-dimensional continuation strategy described in Chapter 5 can be safely applied to connect  $\mathbf{x}_s$  and  $\mathbf{x}_g$  using Eq. (6.9), as the following examples illustrate.

### 6.3 Examples

The performance of the method is shown here by means of two hexapods, the Stewart platform and the NIST Robocrane, with rigid and cable legs, respectively. Note that the same implementation of the method of Chapter 5 can be used, taking as input the system of equations (6.9) defined in this chapter. All experiments have also been run on a MacBook Pro equipped with a 2.66 GHz Intel Core i7 processor.



### 6.3.1 The Stewart Platform

The geometric parameters of the INRIA left hand have been used to test the method on a Stewart Platform [70]. All legs of this manipulator admit forces in the range  $[\underline{f}_i, \overline{f}_i] = [-300, 300]$ , and for the experiments we have set  $\hat{\boldsymbol{w}}_0 = [0, 0, 150, 0, 0, 0]^\top$  and  $\boldsymbol{E} = \boldsymbol{I}_{6 \times 6}$ , both expressed in a frame that translates with the platform, and assuming SI units throughout. In this case, the wrench-feasible C-space of this manipulator is close to the workspace defined by its allowable leg lengths [70], and the algorithm solves typical planning queries in a few seconds.

To graphically illustrate the performance of the method, however, it is better to adopt the geometric parameters of the more academic manipulator used in Section 3.5.2, corresponding to the one studied in [21], where large variations of the leg lengths are allowed, leading to a very large workspace with interesting singularity surfaces, as shown in Fig. 3.11. Two experiments are shown for this manipulator, assuming  $[\underline{f}_i, \overline{f}_i] = [-300, 300]$  and  $\boldsymbol{E} = \boldsymbol{I}_{6 \times 6}$  as before, but using  $\hat{\boldsymbol{w}}_0 = [0, 0, 1, 0, 0, 0]^\top$ . In a first experiment we compute a wrench-feasible path for the

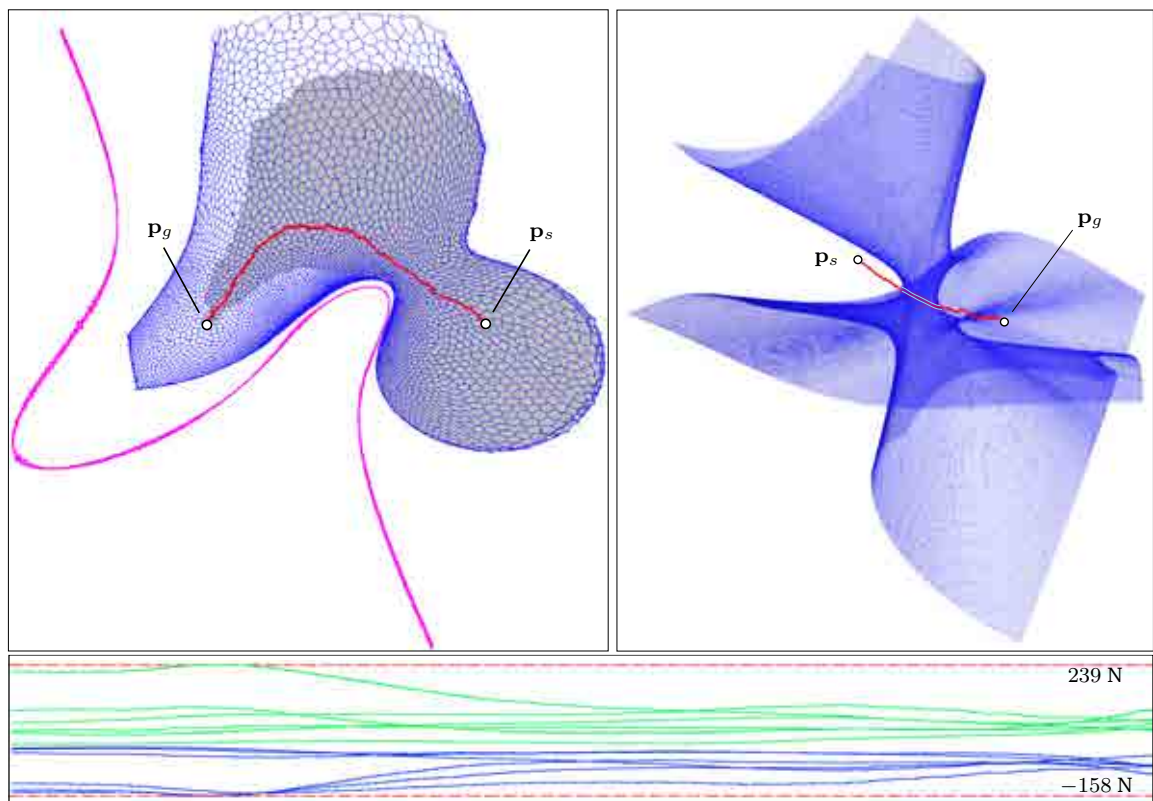


Figure 6.5: Top: Two paths computed for the manipulator in [21]. Bottom: Plot of the maximum and minimum forces supported by each leg along the left path.

platform moving at the same constant orientation of Section 3.5.2, which corresponds to the tilt-and-torsion angles  $\tau = \{-1.449, 0.525, -1.509\}rad$ , and at a constant value of  $z$ . Using the start and goal configurations defined by the positions  $\mathbf{p}_s = [0.4, 0, 0.1]^T$  and  $\mathbf{p}_g = [-0.3, 0, 0.1]^T$  for  $P$ , the resulting path in the XY plane is computed in 578 seconds. Fig. 6.5, left shows this path, together with the singularity curve to be avoided, the atlas corresponding to the whole wrench-feasible connected component accessible from  $\mathbf{p}_s$  (shown as a mesh), and the region explored by the  $A^*$  algorithm (shaded in grey). It can be seen that the interpolated path between  $\mathbf{p}_s$  and  $\mathbf{p}_g$  would go through singularities, but the computed path correctly avoids them while keeping the leg forces within the specified ranges (Fig. 6.5, bottom). In a second experiment, we solve the same planning query but keeping constant the orientation of the platform only, obtaining the path in XYZ shown in Fig. 6.5, right in 90 minutes. As shown, the singularity surface in blue is correctly avoided by the computed path. The singularity surface, computed using the method of Chapter 3, is here shown for illustration purposes only, but there is no need for its actual computation to obtain the desired path. It must be noted that these are hard planning queries, since the workspace of the manipulator considered is enormous when compared to typical workspaces arising in usual platforms. Moreover, once a partial atlas is computed, all planning queries between configurations covered by such atlas can be solved in a few milliseconds.

### 6.3.2 Cable-driven Hexapods

In recent years, cable-driven parallel mechanisms have been increasingly studied and applied to more and more relevant tasks, such as manipulation of heavy loads [131, 132], high-precision positioning [133], monitoring of aquatic environments [134, 135], automated construction of civil structures [136], rescue systems [137], or motion simulators [138], among others. Their generally simple design results in manipulators with low weight but high load capacity. These advantages, together with the fact that they can usually achieve larger workspaces than their counterparts with rigid-limb legs, make cable-driven manipulators very energy-efficient and appropriate to maneuver heavy loads. However, additional constraints apply: their cables can pull but are unable to push the platform, which obliges to keep all cable tensions positive during normal operation.

The workspace of a cable-driven manipulator is limited by a number of hypersurfaces where the control of the manipulator is compromised [139], corresponding to configurations where the tension of some cable is either zero, for which the cable goes slack and control of one degree of freedom is lost, or goes to infinity, which indicates that the mechanism is on a singular configuration and the cable can break. Several authors have proposed strategies for the determination of wrench-feasible workspaces [122, 139–143], but the problem of planning

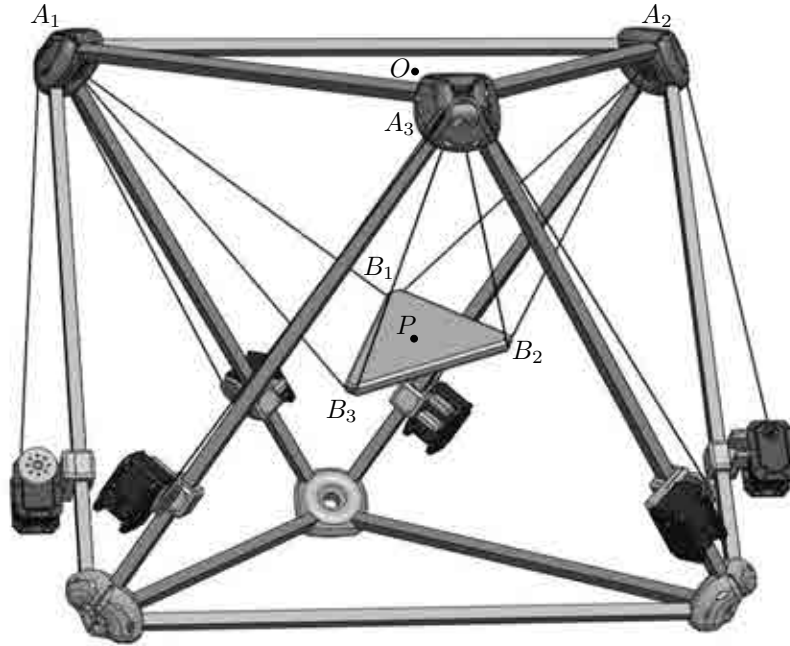


Figure 6.6: CAD model of the cable-driven hexapod under construction at Institut de Robòtica i Informàtica Industrial (IRI), inspired in the NIST Robocrane [147] (courtesy of P. Grosch).

paths between two configurations in such spaces has not been intensively studied. While some approaches indeed exist for cable-driven manipulators [144–146], the path they compute is evaluated for feasibility at discrete points only, and the fulfillment of all constraints along the whole path is not guaranteed.

To verify the performance of the method herein presented on a realistic cable-driven manipulator, we will use here the geometric parameters of the NIST Robocrane manipulator [147] (Fig. 6.6). This manipulator follows an octahedral design where both the base and the platform are equilateral triangles of sides  $2b$  and  $2a$ , respectively. The fixed reference frame  $\mathcal{F}_1$  of the base link is defined with the  $Z$  axis pointing downwards for convenience, and the mobile reference frame of the platform,  $\mathcal{F}_2$ , is centered at a point  $P$ . The coordinates of the vertex points of the base and the platform are given in Table 6.1.

In all experiments, the platform is required to withstand a weight of  $m$  Newtons applied at a point  $P_m$  with position vector  $\mathbf{p}_m^{\mathcal{F}_2} = (\frac{a}{5}, \frac{a}{5}, 0)$ , and we need to set  $f_i > 0$  for all legs to guarantee the control of the manipulator. Since the weight always points downwards, this corresponds to a wrench  $\hat{\mathbf{w}}_0$  that is constant in a reference frame parallel to  $\mathcal{F}_1$  translating with  $P_m$ , i.e.,  $\hat{\mathbf{w}}_0 = [0, 0, m, 0, 0, 0]^T$ . For simplicity, the small variations that may be introduced by inertia forces or external agents are represented by the ellipsoid  $\mathcal{K}$  centered in  $\hat{\mathbf{w}}_0$  defined by

| Leg | Base $L_1$  | Platform $L_2$  |
|-----|---|---|
| 1   | $\mathbf{p}_{1,1}^{\mathcal{F}_1} = [-b, -b\sqrt{3}/3, 0]^\top$       | $\mathbf{p}_{2,1}^{\mathcal{F}_2} = [0, -2a\sqrt{3}/3, 0]^\top$       |
| 2   | $\mathbf{p}_{1,2}^{\mathcal{F}_1} = [b, -b\sqrt{3}/3, 0]^\top$        | $\mathbf{p}_{2,2}^{\mathcal{F}_2} = \mathbf{p}_{2,1}^{\mathcal{F}_2}$ |
| 3   | $\mathbf{p}_{1,3}^{\mathcal{F}_1} = \mathbf{p}_{1,2}^{\mathcal{F}_1}$ | $\mathbf{p}_{2,3}^{\mathcal{F}_2} = [a, a\sqrt{3}/3, 0]^\top$         |
| 4   | $\mathbf{p}_{1,4}^{\mathcal{F}_1} = [0, 2b\sqrt{3}/3, 0]^\top$        | $\mathbf{p}_{2,4}^{\mathcal{F}_2} = \mathbf{p}_{2,3}^{\mathcal{F}_2}$ |
| 5   | $\mathbf{p}_{1,5}^{\mathcal{F}_1} = \mathbf{p}_{1,4}^{\mathcal{F}_1}$ | $\mathbf{p}_{2,5}^{\mathcal{F}_2} = [-a, a\sqrt{3}/3, 0]^\top$        |
| 6   | $\mathbf{p}_{1,6}^{\mathcal{F}_1} = \mathbf{p}_{1,1}^{\mathcal{F}_1}$ | $\mathbf{p}_{2,6}^{\mathcal{F}_2} = \mathbf{p}_{2,5}^{\mathcal{F}_2}$ |

Table 6.1: Parameters of the NIST Robocrane manipulator.

$\mathbf{E} = 10^4 \cdot \mathbf{I}_6$ , which is a six-dimensional sphere of radius  $10^{-2}$ , and the tensions and lengths for all cables are set to remain positive but lower than  $\bar{f}_i = m N$  and  $\bar{\rho}_i = 5a$ , respectively.

To illustrate the complexity of the path planning problem, Fig. 6.7 shows several slices of the wrench-feasible C-space of the manipulator, computed in Matlab using discretization with  $a = 1$ ,  $b = 2$ , and  $m = 1$ . The configurations that cannot be reached due to either cable lengths or tensions out of range are represented by the red and blue areas, respectively, while those corresponding to  $\mathcal{C}_w$  are indicated in green. The figure also shows the singularity curves where  $\det(\mathbf{J}(\mathbf{q})) = 0$ , in red, computed with the method of Chapter 3 under no constraints on the cable tensions or lengths. It can be observed that the wrench-feasible C-space naturally avoids crossing singularities, although in some zones this point may not be clear due to the resolution of the discretization, and to the small size of the figures. The top row of Fig. 6.7 corresponds to slices where the point  $P$  and the torsion  $\sigma$  are held fixed. In the bottom row, the whole orientation  $\tau$  of the platform and one of the coordinates of  $P$  are held fixed. As it can be seen from the figures, the navigation between two configurations of  $\mathcal{C}$  is not a trivial task. In particular, evaluating the wrench-feasibility conditions on discrete points along a path could result in erroneous paths that, for example, could join points belonging to different connected components of  $\mathcal{C}$ . The method herein presented is able to solve such hard planning queries, even when permitting the variation of all pose parameters. However, due to illustration limitations, the performance of the method is demonstrated here by means of two experiments where four and three pose parameters are held fixed.

In the first experiment, we compute two different paths on the top-center slice of  $\mathcal{C}_w$  shown in Fig. 6.7, where  $\mathbf{p} = [0, 0, 2]^\top$  and  $\sigma = 35^\circ$ . Using the start  $\mathbf{q}_1$  and goal  $\mathbf{q}_2$  configurations defined by  $\tau_1 = \{-0.5, 0.9, \frac{7}{36}\pi\} \text{ rad}$  and  $\tau_2 = \{2, 0.9, \frac{7}{36}\pi\} \text{ rad}$ , respectively, the resulting path is computed in 20 seconds. Fig. 6.8(top) shows this path in red together with the atlas corresponding to the whole connected component of  $\mathcal{C}_w$  accessible from the start configuration (shown as a green mesh), and the region explored by the algorithm (shaded in grey). In order to evaluate the performance of the method on a more challenging situation, a second

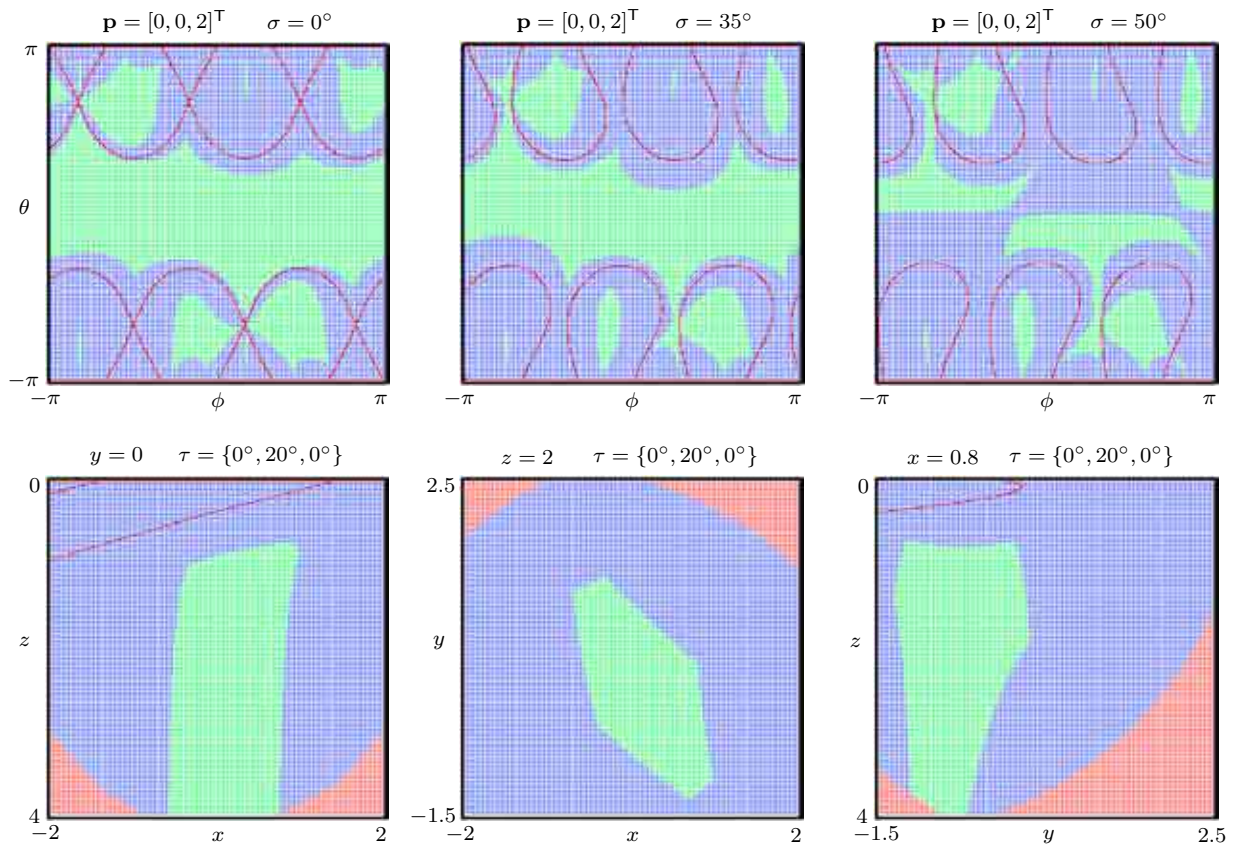


Figure 6.7: Slices of the wrench-feasible C-space obtained by discretization. Blue, red and green zones correspond to configurations where some force is out of range, configurations not attainable due to cables length, and configurations belonging to the wrench-feasible C-space, respectively.

path is computed between configurations  $\mathbf{q}_3$  and  $\mathbf{q}_4$  given by  $\tau_3 = \{0.8, -2.4, \frac{7}{36}\pi\}$  rad and  $\tau_4 = \{-2.4, 2.4, \frac{7}{36}\pi\}$  rad. In this case the path is computed in 280 seconds and the region explored is shaded in green in Fig. 6.8(top). On both planning queries, note that the interpolated path between the start and goal configurations would violate some of the constraints of  $\mathcal{C}_w$ , giving rise to uncontrollable motions of the platform, or to breakage of some of the cables, but the computed paths correctly avoid these situations. Indeed, an advantageous property of the employed continuation strategy, which cannot be ensured by methods relying on discretization, is that the computed path will not jump between distinct connected components of  $\mathcal{C}_w$ , even when such components are close to each other, thus ensuring that the tensions on all cables remain within the prescribed ranges along the whole path. As an example, these tensions can be seen in Fig. 6.8(bottom) for the path from  $\mathbf{q}_3$  to  $\mathbf{q}_4$ . The evolution along this path of the maximum and minimum tensions on each cable are plotted in green and blue, respectively.

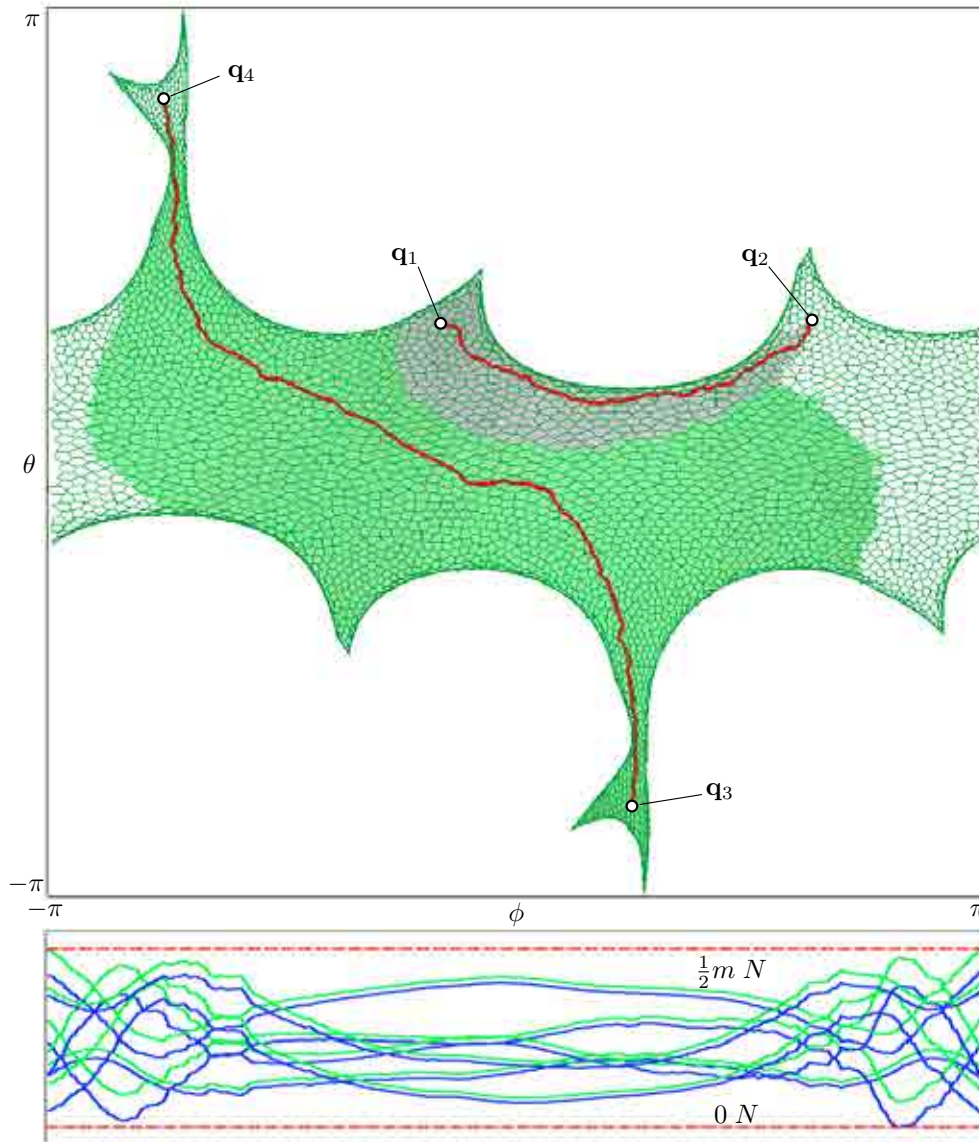


Figure 6.8: Top: Two paths and the areas of  $\mathcal{C}$  explored to compute them, on the slice of the wrench-feasible C-space corresponding to  $\mathbf{p} = [0, 0, 2]^T$  and  $\sigma = 35^\circ$ . The hexagon-like polygons shown in dark green in this figure correspond to the polytopes  $\mathcal{P}_i$  of the full atlas, included for illustration means. Bottom: evolution of the maximum (green) and minimum (blue) possible tensions for each cable, along the path from  $\mathbf{q}_3$  to  $\mathbf{q}_4$ .

Towards the end of the path, some cable may attain a near-zero tension, but this can easily be avoided, if desired, by simply setting a higher value of  $\underline{f}_i$ .

On the second experiment, only three pose variables are held fixed, namely  $x = 0$ ,  $y = 0$ , and  $\sigma = 35^\circ$ , giving rise to a three-dimensional slice of the wrench-feasible C-space. Here, the

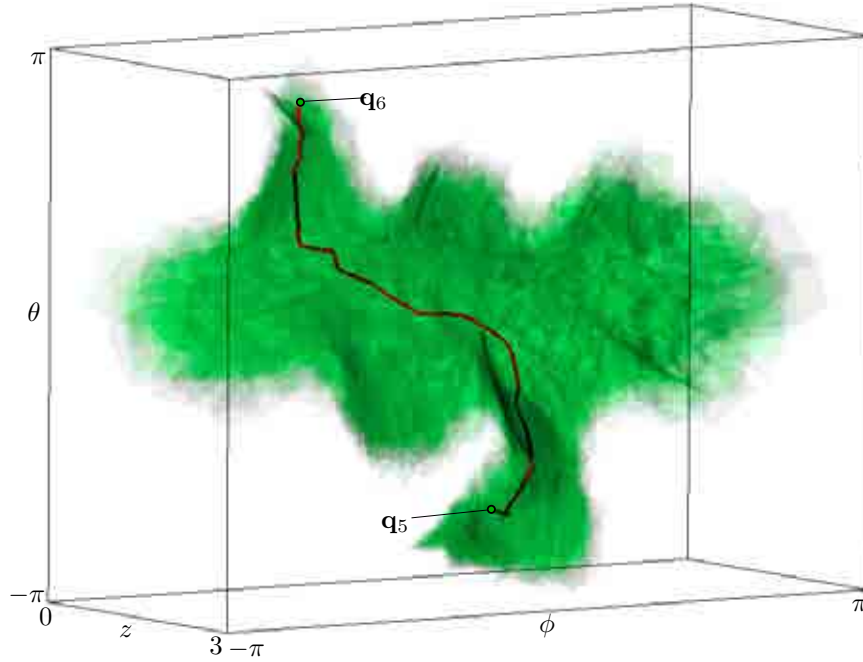


Figure 6.9: A path on a three-dimensional slice of the wrench-feasible C-space with  $x = 0$ ,  $y = 0$  and  $\sigma = 35^\circ$ . The green volume corresponds to the portion of  $\mathcal{C}_w$  explored by the algorithm, which is here visualized by projecting the chart polytopes  $\mathcal{P}_i$  to the space of  $z$ ,  $\phi$ , and  $\theta$ , and representing their faces as semi-transparent walls, in order to visualize the computed path in the interior.

start and goal configurations,  $\mathbf{q}_5$  and  $\mathbf{q}_6$ , are given by  $\mathbf{p}_5 = [0, 0, 1.2]^\top$ ,  $\tau_5 = \{0.8, -2.6, \frac{7}{36}\pi\}$  rad and  $\mathbf{q}_6 = \mathbf{q}_4$ . The resulting path can be seen in Fig. 6.9 in red, together with the atlas generated by the algorithm in green. This case represents a hard planning query and, therefore, the computation time increases significantly. However, once a partial atlas is computed, all planning queries between configurations covered by such atlas can be solved in a few milliseconds.

## 6.4 Additional Constraints and Robot Architectures

In this chapter we have shown how to account for additional constraints in the problem of singularity-free path planning. For the sake of concreteness, we have described how to treat wrench-feasibility constraints in the context of Stewart platforms and cable-driven hexapods, but the methodology could be extended in a number of ways.

For example, it could be generalised to encompass other robot architectures, because it only relies on the availability of a Jacobian relating the actuator and end-effector forces. Using the reciprocal product method outlined in Section 3.5.2, Zlatanov shows how to obtain a simplified

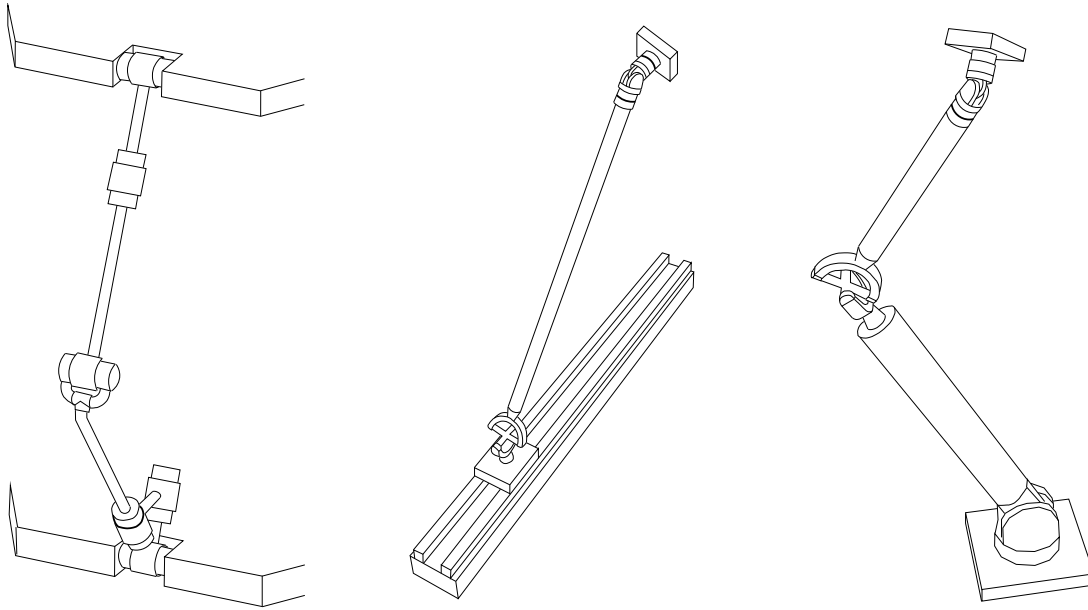


Figure 6.10: Examples of serial kinematic chains used in parallel robots. From left to right: a 6R leg, a PUS leg used in hexaglide platforms, and the RUS chain adopted for the HEXA design or the MicARH rotary hexapod. Figures adapted from [66, 151].

velocity equation for the broad class of hybrid-chain manipulators [12], from which such a Jacobian can be easily deduced, and more complex manipulators could also be dealt with using the extensions presented in [148]. In particular, manipulators with other common legs like those in Fig. 6.10, or adaptations of them, could potentially be addressed, including the Hexaglide [3], manipulators with the HEXA design [149, 150], or newer platforms like the MicARH rotary hexapod [43].

It would also be interesting to accommodate other kinds of constraints. For instance, one could constrain the movements of the robot so that the positioning error of the end-effector were always below a threshold, giving rise to a theory reciprocal to the one that has been presented. To do so, we could use Eq. (3.36),

$$\mathbf{J}^T \hat{\mathbf{T}} = \mathbf{m}_v,$$

which provides a relation between the twist of the end-effector,  $\hat{\mathbf{T}}$ , and the velocities of the actuators,  $\mathbf{m}_v = \dot{\mathbf{v}}$ . Note that if the pose is represented by  $\mathbf{u} = [\mathbf{p}, \boldsymbol{\tau}]^T$ , then  $\hat{\mathbf{T}} = \mathbf{A}\dot{\mathbf{u}}$ , where  $\mathbf{A}$  is a matrix that depends on the chosen orientation parameters  $\boldsymbol{\tau}$  [70, 151]. Thus, by defining  $\mathbf{J}_d = \mathbf{J}^T \mathbf{A}$ , we obtain the relation

$$\mathbf{J}_d \dot{\mathbf{u}} = \dot{\mathbf{v}},$$



or its infinitesimal version,

$$\mathbf{J}_d \delta \mathbf{u} = \delta \mathbf{v},$$

relating the small displacement errors in the actuated joints,  $\delta \mathbf{v}$ , to the corresponding errors in the pose coordinates,  $\delta \mathbf{u}$ . Given an uncertainty ellipsoid for  $\delta \mathbf{v}$  and similarly as we did in Section 6.2, we could now obtain the uncertainty ellipsoid for  $\delta \mathbf{u}$  and force it to remain inside a rectangular domain. We anticipate that the resulting C-space would be smooth too, which would allow to apply the proposed continuation method to compute paths in it.



## **Part III**

# **Closing Remarks**



# 7

## Conclusions

### 7.1 Summary of Contributions

This thesis contributes with new and general approaches for manipulator singularity computation and avoidance. Unlike in previous works, the methods are not limited to specific types of manipulators. Instead, the problems have been addressed assuming a general kinematic architecture and geometry of the underlying mechanisms. By design, the methods are applicable to any non-redundant manipulator, the only limitation coming from the computational power available. A special emphasis has been put on manipulators with complex architectures, since they are those typically arising in today's robotics, where the development of new machines with increasingly complex motions is a common trend. The obtained results aim at contributing to the general understanding of the key role played by singularities in the motion analysis and programming of such manipulators. Specifically, the following results have been obtained in this thesis:

1. **A general method for singularity set computation** (Chapters 2 and 3). Systems of equations are obtained following from the general definition of the diverse singularity types, taking advantage of the sudden change in the rank of the matrices that govern the kinematics of the manipulator that occur at singular configurations. The method is based on formulating such systems in a suitable way for the application of a numerical procedure that relies on a branch-and-prune strategy whereby an initial box bounding the singularity set is recursively reduced and bisected, producing finer and finer approximations of the set successively, until the accuracy of the result is below a given threshold. The method can isolate the whole singularity set independently of its dimension, even in the presence of self-intersections or dimension-changes, with the sole limitations imposed by the course of dimensionality. An effort has also been made to provide guidelines on how to represent the singularity set once computed, in order to produce suitable diagrams for the robot

designer. On this regard, it has been shown that the set can be easily projected to the input and output spaces to provide global information on the motion capabilities of the manipulator, including the reachable input/output areas, the locations where control or dexterity losses can arise, and a delimitation of regions where manipulator motions can safely be planned.

2. **A general method for workspace determination** (Chapter 4). The proposed method delivers a detailed map of the workspace indicating all motion barriers that may be encountered by the manipulator, which is a substantially richer output than that produced by several previous methods for particular manipulators. In comparison to general methods based on continuation, the method is advantageous in that it is uninformed, since it does not need to be fed with a-priori knowledge of the workspace, such as pre-computed assembly configurations or suitable slicing directions, and complete, as it returns the full workspace map even in the presence of several connected components, hidden regions, or degenerate barriers. The method is able to compute workspaces of any dimension but the curse of dimensionality restricts its applicability to up to three-dimensional workspaces typically. While workspaces of larger dimension arise frequently, as in spatial manipulators, it must be noted that such workspaces are impossible to visualize directly in three dimensions. The common practice is then to obtain three-dimensional subsets that are meaningful to the robot designer, like the reachable, constant-orientation, and constant-position workspaces, which, as shown, can all be computed by the method. Furthermore, while many previous methods are limited to obtain cross-sectional curves of the boundary only, the proposed method will directly isolate all points of such boundary.
3. **A general singularity-free path planner** (Chapter 5). Due to the complexity of the involved C-spaces, and of their singularity loci, previous planners have only considered C-spaces with explicit parametrizations. In contrast, the approach presented here makes no recourse to such parametrizations, and can thus be applied to any non-redundant manipulator. The problem has been tackled by defining a system of equations implicitly characterizing the singularity-free C-space of the manipulator, which avoids the need of representing the singularity locus explicitly as an obstacle. The solution manifold of this system can be freely navigated without fear of crossing any singularity of the manipulator. Higher-dimensional continuation techniques are then used to progressively construct an atlas of the component of this manifold that contains the start configuration, until the goal configuration is reached, or path non-existence is proved at the resolution of the atlas. Note that, if desired, the method can also be used to detect non-singular transitions between assembly modes, and that it can also generate an exhaustive atlas of

the singularity-free C-space component that is reachable from one configuration, which is useful to resolve subsequent planning queries rapidly, or to visualize the singularity-free workspace of the manipulator relative to any set of coordinates.

4. **A wrench-feasible path planner for hexapods** (Chapter 6). The approach is based on imposing the resolvability of a six-dimensional set of wrenches at any point on the path. It has been shown how all configurations satisfying this constraint can be encoded in a system of equations whose solution set is a smooth manifold, which can be explored to obtain the desired paths using higher-dimensional continuation techniques without resorting to branch-switching procedures. As a by-product, the method implicitly ensures that the screw Jacobian of the manipulator will be non-singular along the path, thus allowing a full control of the platform motions at all times.

These results have been disseminated so far through the publications listed in Appendix A.

## 7.2 Future Research Directions

Some prospects for further research have been identified and are briefly discussed next:

1. **Non-redundant manipulators.** A natural continuation of this research would be to extend the developments to also account for redundant manipulators. Their treatment was left out of the scope of the present work, but Zlatanov already provides definitions and classification methods for the singularities of such manipulators [12], which would provide the ideal basis for such an extension.
2. **Collision constraints.** The possible collisions between the various bodies of the manipulator or with the environment have been neglected in the whole study. This can definitely be the subject of further investigation, and possible research directions could include the formulation of such constraints with equations of a mild form, to be able to add them to the systems given in the thesis, or the randomization of the path planners of Chapters 5 and 6, following the spirit of [152] or [118], for example.
3. **Detailed singularity analysis of complex spatial manipulators.** The interpretation and visualization of the various singularity sets through multiple C-space projections requires a profound and detailed analysis that can now be tackled using the methods proposed in this thesis. It would be interesting to further illustrate their power by applying them to the analysis of specific challenging manipulators.

4. **Stratification of the singularity set.** As noted in [12], ideally one would like to obtain a stratification of the singularity set, which would decompose the set into non-intersecting manifolds consisting of singularities of the same class, understanding a singularity class as a possible combination of Zlatanov's lower-level singularity types. This would be relevant in the early stages of the design of a manipulator, to identify the exact physical phenomena that occur at singular configurations, but also to study the behaviour of the mechanism at special configurations. Preliminary work in this sense shows that the developments of this thesis can be used to design a hierarchy of tests on the whole singularity set, which yields sets of classes to which an identified singularity cannot belong [153]. The performance of this approach and its application on spatial manipulators actually demands additional work.
5. **Exploitation of symmetries.** Almost always, designers tend to produce symmetric manipulator designs. The presence of such symmetries usually translates into symmetries of configuration space, and of the various singularity sets studied in this thesis. Previous work in [61], and several manipulators herein studied provide particular examples of this situation. It would thus be desirable to exploit such symmetries in order to focus the computations on the strictest necessary portion of configuration space, in order to reduce the overall computational burden as much as possible.
6. **Dexterous workspaces.** A class of workspaces not treated explicitly in this thesis are dexterous workspaces, defined as the set of end-effector positions that can be reached with *any* orientation within a given range. While computing such workspaces seems plausible under the proposed approach, several modifications need to be introduced in order to do so. Mathematical conditions allowing to identify the boundary of such workspaces have already been investigated [154–159], which could be used as a starting point for the development of this extension.
7. **Variants of the path planners.** The resolution completeness of the proposed path planners comes at the expense of a computational cost that scales exponentially with the dimension of the explored manifold. To deal with higher-dimensional problems, however, we could adapt the randomized approach in [160], which trades off resolution completeness by efficiency and probabilistic completeness, or the approach in [161] which, additionally, guarantees asymptotic optimality. The evaluation of these variants in the context of singularity-free or wrench-feasible path planning certainly deserves further attention, but with the current state of the art we foresee the solvability of problems in dimensions six and above in reasonable times.



8. **Clearance relative to the singularities.** A point left open in the case of the singularity-free path planner is how to establish the value of the threshold  $b_{max}$  governing the clearance of the path relative to the singularity locus. As mentioned, this threshold has to be determined case-wise, depending on the particular context of application, but specific versions of the planner on particular manipulators may lead to systematic ways to set the clearance. For example, Chapter 6 has shown how to do so on hexapod manipulators, using the idea that the leg forces should remain within the allowed limits. It should be possible to develop a dual strategy that fixes the clearance in terms of a maximum position/orientation error allowed for the platform, which would be nice to have on manipulators designed for fine-positioning applications for instance [43, 120, 162].
9. **A general wrench-feasible path planner.** The wrench-feasible path planner of Chapter 6 should be extendable to other mechanism topologies. Note that the method depends on the availability of a force Jacobian relating the platform wrench with the joint forces or torques. On hybrid-chain mechanisms, for example, this Jacobian can be obtained using the reciprocal screw method developed in [12, Chapter 4], but larger mechanism classes should be addressable using the results in [148].
10. **Dynamical aspects.** Some manipulators, like those used for fast pick-and-place operations, are designed to be operated at very high speeds, so that accounting for dynamical effects in the proposed path planners would be of great interest. In particular, further research should be conducted to allow the computation of the trajectory that minimizes the travel time between the query configurations.





## List of Publications

The following is a list of publications resulting from the research reported in this thesis:

### Journal papers

1. O. Bohigas, L. Ros, and M. Manubens, "A complete method for workspace boundary determination on general structure manipulators," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 993–1006, 2012.
2. J. M. Porta, L. Jaillet, and O. Bohigas, "Randomized path planning on manifolds based on higher-dimensional continuation," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 201–215, 2012.
3. O. Bohigas, M. Manubens, and L. Ros, "A linear relaxation method for computing workspace slices of the Stewart platform," *ASME Journal of Mechanisms and Robotics*, vol. 5, pp. 0011 005–1–011 005–10, 2012.
4. O. Bohigas, M. Manubens, and L. Ros, "Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case," *Mechanism and Machine Theory*, 2013. In press.
5. O. Bohigas, M. Henderson, L. Ros, M. Manubens, and J. Porta, "Planning singularity-free paths on closed-chain manipulators," *IEEE Transactions on Robotics*, 2013. In press.

### Journal papers under review

6. J. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet, "The CUIK suite: Motion analysis of closed-chain multibody systems," *conditionally accepted in IEEE Robotics and Automation Magazine*.
7. O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, and J. Porta, "A general method for the numerical computation of manipulator singularity sets," *submitted to IEEE Transactions on Robotics*.

### Conference papers

8. O. Bohigas, L. Ros, and M. Manubens, "A complete method for workspace boundary determination," in *Advances in Robot Kinematics*, J. Lenarcic and M. Stanisic, Eds. Springer, 2010, pp. 329–338.

9. O. Bohigas, L. Ros, and M. Manubens, "A unified method for computing position and orientation workspaces of general Stewart platforms," in *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE (Washington DC, USA)*, 2011.
10. O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, and J. Porta, "Numerical computation of manipulator singularities," in *IEEE International Conference on Robotics and Automation, ICRA (St.Paul, USA)*, 2012.
11. O. Bohigas, M. Henderson, L. Ros, and J. Porta, "A singularity-free path planner for closed-chain manipulators," in *IEEE International Conference on Robotics and Automation, ICRA (St.Paul, USA)*, 2012.
12. O. Bohigas, M. Manubens, and L. Ros, "Planning singularity-free force-feasible paths on the Stewart platform," in *Latest Advances in Robot Kinematics*, J. Lenarcic and M. Husty, Eds. Springer, 2012, pp. 245–252.
13. O. Bohigas, D. Zlatanov, M. Manubens, and L. Ros, "On the numerical classification of the singularities of robot manipulators," in *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE*, 2012.
14. O. Bohigas, M. Manubens, and L. Ros, "Navigating the wrench-feasible C-space of cable-driven hexapods," in *Cable-Driven Parallel Robots*, T. Bruckmann and A. Pott, Eds. Springer, 2012, pp. 53–68.
15. L. Ros, J. Porta, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet, "An open-source toolbox for motion analysis of closed-chain mechanisms," *accepted in the 6th International Workshop on Computational Kinematics, CK2013 (Barcelona, Spain)*, 2013.

# References

- [1] “Experiment at the Vorarlberg University of Applied Sciences.” [http://youtu.be/l\\_LmUsBYwWA?t=3m31s](http://youtu.be/l_LmUsBYwWA?t=3m31s).
- [2] A. Rull, “Disseny, implementació i control d’un robot paral·lel de 5 graus de llibertat,” Master’s thesis, Universitat Politècnica de Catalunya, 2011.
- [3] M. Honegger, A. Codourey, and E. Burdet, “Adaptive control of the hexaglide, a 6 dof parallel manipulator,” in *IEEE International Conference on Robotics and Automation, ICRA (Albuquerque, USA)*, pp. 543–548, 1997.
- [4] “Ingeniería Mecánica Aplicada y Computacional, Universidad Pública de Navarra.” <http://www.imac.unavarra.es/~cnc/>.
- [5] J. M. McCarthy, “Kinematics, polynomials, and computers – a brief history,” *Editorial of ASME Journal of Mechanisms and Robotics*, vol. 3, pp. 7–11, 2011.
- [6] C. M. Gosselin and J. Angeles, “Singularity analysis of closed-loop kinematic chains,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281–290, 1990.
- [7] D. Zlatanov, R. G. Fenton, and B. Benhabib, “Singularity analysis of mechanisms and robots via a motion-space model of the instantaneous kinematics,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 1994. 980–985.
- [8] D. Zlatanov, R. G. Fenton, and B. Benhabib, “Singularity analysis of mechanisms and robots via a velocity-equation model of the instantaneous kinematics,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 1994. 986–991.
- [9] D. Zlatanov, R. G. Fenton, and B. Benhabib, “Analysis of the instantaneous kinematics and singular configurations of hybrid-chain manipulators,” in *Proceedings of the ASME 23rd Biennial Mechanisms Conference*, vol. 72, 1994. 467–476.
- [10] F. C. Park and J. W. Kim, “Singularity analysis of closed kinematic chains,” *ASME Journal of Mechanical Design*, vol. 121, no. 1, pp. 32–38, 1999.
- [11] I. A. Bonev and D. Zlatanov, “The mystery of the singular SNU translational parallel robot,” 2001.
- [12] D. Zlatanov, *Generalized Singularity Analysis of Mechanisms*. PhD thesis, University of Toronto, 1998.
- [13] D. Zlatanov, I. A. Bonev, and C. M. Gosselin, *Advances in Robot Kinematics: Theory and Applications*, ch. Constraint Singularities as C-Space Singularities. Kluwer Academic Publishers, 2002. 183-192.
- [14] D. Zlatanov, I. A. Bonev, and C. M. Gosselin, “Constraint singularities of parallel mechanisms,” in *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 1, 2002. 496–502.

- [15] O. Ma and J. Angeles, "Architecture singularities of parallel manipulators.," *International Journal of Robotics & Automation*, vol. 7, no. 1, pp. 23–29, 1992.
- [16] J. Borràs, F. Thomas, and C. Torras, "Architecture singularities in flagged parallel manipulators," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2008. 3844–3850.
- [17] P. S. Donelan, "Singularity-theoretic methods in robot kinematics," *Robotica*, vol. 25, no. 6, pp. 641–659, 2007.
- [18] P. S. Donelan, *Kinematic Singularities of Robot Manipulators*. InTech, 2010.
- [19] I. A. Bonev, *Geometric Analysis of Parallel Mechanisms*. PhD thesis, Faculté des Sciences et de Génie, Université de Laval, 2002.
- [20] B. M. St-Onge and C. M. Gosselin, "Singularity analysis and representation of the general Gough-Stewart platform," *The International Journal of Robotics Research*, vol. 19, no. 3, pp. 271–288, 2000.
- [21] H. Li, C. M. Gosselin, M. J. Richard, and B. M. St-Onge, "Analytic form of the six-dimensional singularity locus of the general Gough-Stewart platform," *ASME Journal of Mechanical Design*, vol. 128, no. 1, pp. 279–287, 2006.
- [22] J.-P. Merlet, "A formal-numerical approach for robust in-workspace singularity detection," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 393–402, 2007.
- [23] B. Dasgupta and T. S. Mruthyunjaya, "Singularity-free path planning for the Stewart platform manipulator," *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 711–725, 1998.
- [24] S. Sen, B. Dasgupta, and A. K. Mallik, "Variational approach for singularity-free path-planning of parallel manipulators," *Mechanism and Machine Theory*, vol. 38, no. 11, pp. 1165–1183, 2003.
- [25] Q. Jiang and C. M. Gosselin, "Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform," *Mechanism and Machine Theory*, vol. 44, no. 6, pp. 1281–1293, 2009.
- [26] J. Borràs, F. Thomas, and C. Torras, "Singularity-invariant families of line-plane 5-SPU platforms," *IEEE Transactions on Robotics*, 2011.
- [27] J. Borràs, F. Thomas, and C. Torras, "On  $\Delta$ -transforms," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1225–1236, 2009.
- [28] J. Borràs, *Singularity-Invariant Leg Rearrangements in Stewart-Gough Platforms*. PhD thesis, Technical University of Catalonia, 2011.
- [29] J. Borràs, F. Thomas, and C. Torras, "Singularity-invariant leg rearrangements in doubly-planar Stewart-Gough platforms," in *Robotics: Science and Systems Conference*, pp. 1–8, 2011.

- [30] J. M. Porta, L. Ros, T. Creemers, and F. Thomas, "Box approximations of planar linkage configuration spaces," *ASME Journal of Mechanical Design*, vol. 129, no. 4, pp. 397–405, 2007.
- [31] J. M. Porta, L. Ros, and F. Thomas, "A linear relaxation technique for the position analysis of multi-loop linkages," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 225–239, 2009.
- [32] J. M. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet, "The CUIK suite: Motion analysis of closed-chain multibody systems," *conditionally accepted in IEEE Robotics and Automation Magazine*.
- [33] M. E. Henderson, "Multiple parameter continuation: Computing implicitly defined k-manifolds," *International Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 451–476, 2002.
- [34] M. E. Henderson, "Multiparameter parallel search branch switching," *International Journal of Bifurcation and Chaos in Applied Science and Engineering*, vol. 15, no. 3, pp. 967–974, 2005.
- [35] M. E. Henderson, *Numerical Continuation Methods for Dynamical Systems: path following and boundary value problems*, ch. Higher-Dimensional Continuation, pp. 77–115. Springer, 2007.
- [36] A. Albu-Schaffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.
- [37] "Leibniz Universität Hannover." <http://www.imes.uni-hannover.de/>.
- [38] "Robotics Lab at Université Laval." <http://robot.gmc.ulaval.ca>.
- [39] "Hydra-Power Systems." <http://www.hydrapowersystems.com/>.
- [40] C. M. Gosselin and E. St.-Pierre, "Development and experimentation of a fast three-degree-of-freedom camera-orienting device," *International Journal of Robotics Research*, vol. 16, no. 5, pp. 619–630, 1997.
- [41] "The Exechon parallel kinematics machine." <http://www.exechon.com/>.
- [42] "Kawasaki Robotics." <http://www.kawasakirobotics.com>.
- [43] J. Coulombe and I. A. Bonev, "A new rotary hexapod for micropositioning," in *IEEE International Conference on Robotics and Automation, ICRA (Karlsruhe, Germany)*, 2013.
- [44] "Kinematics and robot design group at the Institut de Robòtica i Informàtica Industrial." <http://www.iri.upc.edu/research/kinematics>.
- [45] J. G. De Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems*. Springer Verlag, 1993.

- [46] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge, Massachusetts: The MIT Press, 1988.
- [47] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory and Applications*. Boston: Birkhäuser, 2002.
- [48] B. Sturmfels, *Solving systems of polynomial equations*, vol. 97. American Mathematical Society, 2002.
- [49] D. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, vol. 10. Springer, 2007.
- [50] A. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, vol. 57. Society for Industrial Mathematics, 2009.
- [51] T.-Y. Li, “Numerical solution of polynomial systems by homotopy continuation methods,” *Handbook of numerical analysis*, vol. 11, pp. 209–304, 2003.
- [52] E. L. Allgower and K. Georg, “Introduction to numerical continuation methods,” 2003.
- [53] E. C. Sherbrooke and N. M. Patrikalakis, “Computation of the solutions of nonlinear polynomial systems,” *Computer Aided Geometric Design*, vol. 10, no. 5, pp. 379–405, 1993.
- [54] E. Hansen and G. W. Walster, *Global optimization using interval analysis: revised and expanded*, vol. 264 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., 2nd ed., 2003.
- [55] A. J. Sommese and C. W. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific Publishing, 2005.
- [56] J. M. Porta, L. Ros, F. Thomas, F. Corcho, J. Cantó, and J. J. Pérez, “Complete maps of molecular-loop conformational spaces,” *Journal of Computational Chemistry*, vol. 28, no. 13, pp. 2170–2189, 2007.
- [57] O. Bohigas, M. Manubens, and L. Ros, “Singularities of non-redundant manipulators: A short account and a method for their computation in the planar case,” *Mechanism and Machine Theory*, 2013. In press.
- [58] J. K. Davidson and K. H. Hunt, *Robots and Screw Theory: Applications of Kinematics and Statics to Robotics*. Oxford University Press, 2004.
- [59] J. Duffy, *Statics and kinematics with applications to robotics*. Cambridge University Press, 1996.
- [60] E. J. Haug, C.-M. Luh, F. A. Adkins, and J.-Y. Wang, “Numerical algorithms for mapping boundaries of manipulator workspaces,” *ASME Journal of Mechanical Design*, vol. 118, no. 2, pp. 228–234, 1996.
- [61] I. A. Bonev and C. M. Gosselin, “Analytical determination of the workspace of symmetrical spherical parallel mechanisms,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1011–1017, 2006.



- [62] M. Zein, P. Wenger, and D. Chablat, "Singular curves in the joint space and cusp points of 3-RPR parallel manipulators," *Robotica*, vol. 25, no. 6, pp. 717–724, 2007.
- [63] E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez, *Advances in Robot Kinematics: Analysis and Design*, ch. Transitions between multiple solutions of the Direct Kinematic Problem. Springer, 2008. 301-310.
- [64] P. Wenger and D. Chablat, "Kinematic analysis of a class of analytic planar 3-RPR parallel manipulators," in *Proc. of the 5th International Workshop on Computational Kinematics*, 2009. 43–50.
- [65] "The CUIK project home page." <http://www.iri.upc.edu/cuik>.
- [66] J. Angeles, *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Springer, 2006.
- [67] J. Sefrioui and C. M. Gosselin, "On the quadratic nature of the singularity curves of planar three-degree-of-freedom parallel manipulators," *Mechanism and Machine Theory*, vol. 30, no. 4, pp. 533–551, 1995.
- [68] J.-P. Merlet, C. M. Gosselin, and N. Mouly, "Workspaces of planar parallel manipulators," *Mechanism and Machine Theory*, vol. 33, no. 1-2, pp. 7–20, 1998.
- [69] I. A. Bonev and C. M. Gosselin, "Singularity loci of planar parallel manipulators with revolute joints," in *Proc. of the 2nd Workshop on Computational Kinematics*, 2001. 291–299.
- [70] J.-P. Merlet, *Parallel robots*. Springer-Verlag, 2006.
- [71] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. Wiley-Interscience, 1999.
- [72] K. J. Waldron and K. H. Hunt, "Series-parallel dualities in actively coordinated mechanisms," *International Journal of Robotics Research*, vol. 10, no. 5, pp. 473–480, 1991.
- [73] I. A. Bonev, D. Zlatanov, and C. M. Gosselin, "Advantages of the modified Euler angles in the design and control of PKMs," in *Proceeding of the 3rd Chemnitz Parallel Kinematics Seminar/2002 Parallel Kinematic Machines International Conference, Chemnitz, Germany*, pp. 171–188, 2002.
- [74] B. Roth, "Performance evaluation of manipulators from a kinematic viewpoint," *National Bureau of Standards Special Publication*, vol. 459, pp. 39–62, 1976.
- [75] F. Freudenstein and E. J. F. Primrose, "On the analysis and synthesis of the workspace of a three-link, turning-pair connected robot arm," *ASME J. Mech. Trans. Automat. Des.*, vol. 106, no. 3, pp. 365–370, 1984.
- [76] A. H. Soni and Y. C. Tsai, "An algorithm for the workspace of a general  $n$ -R robot," *ASME J. Mech. Trans. Automat. Des.*, vol. 105, pp. 52–57, 1983.

- [77] J. A. Hansen, K. C. Gupta, and S. M. K. Kazerounian, "Generation and evaluation of the workspace of a manipulator," *Int. J. Robot. Res.*, vol. 2, no. 3, pp. 22–31, 1983.
- [78] J. Rastegar and B. Fardanesh, "Manipulation workspace analysis using the Monte Carlo method," *Mech. Mach. Theory*, vol. 25, no. 2, pp. 233–239, 1990.
- [79] C. M. Gosselin, "Determination of the workspace of 6-DOF parallel manipulators," *ASME J. Mech. Des.*, vol. 112, pp. 331–336, 1990.
- [80] J.-P. Merlet, "Geometrical determination of the workspace of a constrained parallel manipulator," in *Advances in Robot Kinematics*, pp. 326–329, 1992.
- [81] J.-P. Merlet, "Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries," *International Journal of Robotics Research*, vol. 18, no. 9, pp. 902–916, 1999.
- [82] I. A. Bonev and J.-C. Ryu, "A new approach to orientation workspace analysis of 6-DOF parallel manipulators," *Mechanism and Machine Theory*, vol. 36, no. 1, pp. 15–28, 2001.
- [83] J. A. Snyman, L. J. du Plessis, and J. Duffy, "An optimization approach to the determination of the boundaries of manipulator workspaces," *ASME Journal of Mechanical Design*, vol. 122, no. 4, pp. 447–456, 2000.
- [84] K. Abdel-Malek and H. J. Yeh, "Analytical boundary of the workspace for general 3-DOF mechanisms," *International Journal of Robotics Research*, vol. 16, no. 2, pp. 198–213, 1997.
- [85] M. Ceccarelli, "A formulation for the workspace boundary of general N-revolute manipulators," *Mech. Mach. Theory*, vol. 31, no. 5, pp. 637–646, 1996.
- [86] K. Abdel-Malek, H. J. Yeh, and N. Khairallah, "Workspace, void, and volume determination of the general 5-DOF manipulator," *J. Structural Mechanics*, vol. 27, no. 1, pp. 89–115, 1999.
- [87] M. Zein, P. Wenger, and D. Chablat, "An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications," *Mechanism and Machine Theory*, vol. 41, no. 8, pp. 971–986, 2006.
- [88] E. Ottaviano, M. Husty, and M. Ceccarelli, "Identification of the workspace boundary of a general 3-R manipulator," *ASME Journal of Mechanical Design*, vol. 128, no. 1, pp. 236–242, 2006.
- [89] D.-Y. Jo, *Numerical Analysis of Workspaces of Multibody Mechanical Systems*. PhD thesis, The University of Iowa, Iowa, 1988.
- [90] F. A. Adkins, *Numerical continuation and bifurcation methods for mechanism workspace and controllability issues*. PhD thesis, The University of Iowa, Iowa, 1996.
- [91] K. Abdel-Malek, J. Yang, D. Blackmore, and K. Joy, "Swept volumes: foundation, perspectives, and applications," *Int. J. Shape Model.*, vol. 12, no. 1, pp. 87–127, 2006.

- [92] Y. Lu, Y. Shi, and B. Hu, "Solving reachable workspace of some parallel manipulators by computer-aided design variation geometry," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 222, no. 9, pp. 1773–1781, 2008.
- [93] G. Castelli, E. Ottaviano, and M. Ceccarelli, "A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators," *Mech. Based Des. Struct. Mach.*, vol. 36, no. 1, pp. 14–33, 2008.
- [94] K. Abdel-Malek, F. A. Adkins, H. J. Yeh, and E. J. Haug, "On the determination of boundaries to manipulator workspaces," *Robotics and Computer Integrated Manufacturing*, vol. 13, no. 1, pp. 63–72, 1997.
- [95] D. Oblak and D. Kohli, "Boundary surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators," *ASME J. Mech. Trans. Automat. Des.*, vol. 110, no. 4, pp. 389–396, 1988.
- [96] E. L. Allgower and K. Georg, *Numerical Continuation Methods*. Springer-Verlag, Berlin, 1990.
- [97] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*. Society for Industrial Mathematics, 2009.
- [98] C. W. Wampler, "Solving the kinematics of planar mechanisms by Dixon's determinant and a complex plane formulation," *ASME J. Mech. Des.*, vol. 123, pp. 382–387, September 2001.
- [99] E. Celaya, T. Creemers, and L. Ros, "Exact interval propagation for the efficient solution of position analysis problems on planar linkages," *Mechanism and Machine Theory*. To appear.
- [100] D. Y. Jo and E. J. Haug, "Workspace analysis of closed-loop mechanisms with unilateral constraints," *Advances in Design Automation*, vol. 19, no. 3, pp. 53–60, 1989.
- [101] J.-P. Merlet, "Determination of the orientation workspace of parallel manipulators," *J. Intell. Rob. Syst.*, vol. 13, no. 2, pp. 143–160, 1995.
- [102] Y. Cao, Z. Huang, H. Zhou, and W. Ji, "Orientation workspace analysis of a special class of Stewart-Gough parallel manipulators," *Robotica*, vol. 28, no. 7, pp. 989–1000, 2010.
- [103] C.-M. Luh, F. A. Adkins, E. J. Haug, and C. C. Qiu, "Working capability analysis of Stewart platforms," *ASME Journal of Mechanical Design*, vol. 118, no. 2, pp. 220–227, 1996.
- [104] F. Pernkopf and M. Husty, "Workspace analysis of Stewart-Gough-type parallel manipulators," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 220, no. 7, pp. 1019–1032, 2006.
- [105] J.-P. Merlet and C. M. Gosselin, *Springer Handbook of Robotics*, ch. Parallel Mechanisms and Robots, pp. 269–285. Berlin, Heidelberg: Springer, 2008.

- [106] I. A. Boney, D. Chablat, and P. Wenger, "Working and assembly modes of the Agile Eye," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2317–2322, 2006.
- [107] B. Servatius, O. Shai, and W. Whiteley, "Combinatorial characterization of the Assur graphs from engineering," *European Journal of Combinatorics*, vol. 31, no. 4, pp. 1091–1104, 2010.
- [108] A. Sljoka, O. Shai, and W. Whiteley, "Checking mobility and decomposition of linkages via pebble game algorithm," in *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE*, 2011.
- [109] O. Altuzarra, C. Pinto, R. Avilés, and A. Hernández, "A practical procedure to analyze singular configurations in closed kinematic chains," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 929–940, 2004.
- [110] E. Macho, O. Altuzarra, E. Amezua, and A. Hernandez, "Obtaining configuration space and singularity maps for parallel manipulators," *Mechanism and Machine Theory*, vol. 44, no. 11, pp. 2110–2125, 2009.
- [111] N. Rojas and F. Thomas, "On closed-form solutions to the position analysis of Baranov trusses," *Mechanism and Machine Theory*, vol. 50, pp. 179–196, 2011.
- [112] N. Rojas and F. Thomas, "Distance-based position analysis of the three seven-link Assur kinematic chains," *Mechanism and Machine Theory*, vol. 46, no. 2, pp. 112–126, 2010.
- [113] S. Bhattacharya, H. Hatwal, and A. Ghosh, "Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators," *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 965–974, 1998.
- [114] A. K. Dash, I. Chen, S. H. Yeo, and G. Yang, "Workspace generation and planning singularity-free path for parallel manipulators," *Mechanism and Machine Theory*, vol. 40, no. 7, pp. 776–805, 2005.
- [115] H. B. Keller, *Applications of Bifurcation Theory*. Academic Press, 1977.
- [116] W. C. Rheinboldt, "MANPACK: A set of algorithms of computations on implicitly defined manifolds," *Computers and Mathematics with Applications*, vol. 32, no. 12, pp. 15–28, 1996.
- [117] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
- [118] J. M. Porta and L. Jaillet, "Path planning on manifolds using randomized higher-dimensional continuation," in *Algorithmic Foundations of Robotics IX* (D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, eds.), vol. 68 of *Springer Tracts in Advanced Robotics*, pp. 337–353, Springer Berlin / Heidelberg, 2011.
- [119] D. Chablat and P. Wenger, "The kinematic analysis of a symmetrical three-degree-of-freedom planar parallel manipulator," in *CISM-IFTOMM Symposium on Robot Design, Dynamics, and Control*, 2004.

- [120] L. Campos, F. Bourbonnais, I. A. Bonev, and P. Bigras, “Development of a five-bar parallel robot with large workspace,” in *Proc. of the ASME Design Engineering Technical Conferences*, 2010.
- [121] P. A. Voglewede and I. Ebert-Uphoff, “Overarching framework for measuring closeness to singularities of parallel manipulators,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1037–1045, 2005.
- [122] P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff, “Wrench-feasible workspace generation for cable-driven robots,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 890–902, 2006.
- [123] J. Hubert, *Manipulateurs parallèles, singularités et analyse statique*. Ph.D. thesis, École Nationale Supérieure des Mines de Paris, September 2010.
- [124] “Okuma corporation.” <http://www.okuma.com/>.
- [125] “National institute of standards and technology.” <http://www.nist.gov>.
- [126] “Symétrie, the hexapod company.” <http://www.symetrie.fr/en/>.
- [127] L. Ros, A. Sabater, and F. Thomas, “An ellipsoidal calculus based on propagation and fusion,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 4, pp. 430–442, 2002.
- [128] “Ingersoll machine tools.” <http://www.camozzimachinery.com/>.
- [129] “Toyoda machinery usa.” <http://www.toyodausa.com/>.
- [130] “Mikrolar, inc..” <http://www.mikrolar.com/>.
- [131] P. Cheng, J. Fink, S. Kim, and V. Kumar, “Cooperative towing with multiple robots,” in *Algorithmic Foundation of Robotics VIII* (G. Chirikjian, H. Choset, M. Morales, and T. Murphey, eds.), vol. 57 of *Springer Tracts in Advanced Robotics*, pp. 101–116, Springer Berlin / Heidelberg, 2009.
- [132] S.-R. Oh, J.-C. Ryu, and S. K. Agrawal, “Dynamics and control of a helicopter carrying a payload using a cable-suspended robot,” *Journal of Mechanical Design*, vol. 128, no. 5, pp. 1113–1121, 2006.
- [133] A. M. Lytle, K. S. Saidi, R. V. Bostelman, W. C. Stone, and N. A. Scott, “Adapting a teleoperated device for autonomous control using three-dimensional positioning sensors: experiences with the NIST RoboCrane,” *Automation in Construction*, vol. 13, pp. 101–118, 2004.
- [134] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, “Nims-pl: A cable-driven robot with self-calibration capabilities,” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1005–1015, 2009.

- [135] T. C. Harmon, R. F. Ambrose, R. M. Gilbert, J. C. Fisher, M. J. Stealey, and W. J. Kaiser, "High-resolution river hydraulic and water quality characterization using rapidly deployable networked infomechanical systems (NIMS RD)," *Environmental Engineering Science*, vol. 24, no. 2, pp. 151–159, 2007.
- [136] P. Bosscher, R. L. Williams II, L. S. Bryson, and D. Castro-Lacouture, "Cable-suspended robotic contour crafting system," *Automation in Construction*, vol. 17, pp. 45–55, 2007.
- [137] J.-P. Merlet and D. Daney, "A portable, modular parallel wire crane for rescue operations," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2834–2839, 2010.
- [138] S. Tadokoro, T. Matsushima, Y. Murao, and H. Kohkawa, "A parallel cable-driven motion base for virtual acceleration," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1700–1705, 2001.
- [139] R. Verhoeven, *Analysis of the workspace of tendon based Stewart platforms*. PhD thesis, Universität Duisburg-Essen, 2004.
- [140] M. Gouttefarde, D. Daney, and J.-P. Merlet, "Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 1–13, 2011.
- [141] J. Pusey, A. Fattah, S. K. Agrawal, and E. Messina, "Design and workspace analysis of a 6-6 cable-suspended parallel robot," *Mechanism and Machine Theory*, vol. 3, pp. 761–778, 2004.
- [142] E. Stump and V. Kumar, "Workspaces of cable-actuated parallel manipulators," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 159–167, 2006.
- [143] R. Verhoeven, M. Hiller, and S. Tadokoro, "Workspace of tendon-driven Stewart platforms: Basics, classification, details on the planar 2-dof class.," in *Proc. of the 4th Int. Conf. on Motion and Vibration Control*, pp. 871–876, 1998.
- [144] S. Fang, D. Franitza, R. Verhoeven, and M. Hiller, "Proceedings of the 11th IFToMM world congress in mechanism and machine science," (Tianjin, China), China Machinery Press, 2003.
- [145] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitza, "Design, analysis and realization of tendon-based parallel manipulators," *Mechanism and Machine Theory*, vol. 40, no. 4, pp. 429–445, 2005.
- [146] S. Lahouar, E. Ottaviano, S. Zeghouel, L. Romdhane, and M. Ceccarelli, "Collision free path-planning for cable-driven parallel robots," *Robotics and Autonomous Systems*, vol. 57, no. 11, pp. 1083–1093, 2009.
- [147] J. Albus, R. V. Bostelman, and N. Dagalakakis, "The NIST Robocrane," *Journal of Robotic Systems*, vol. 10, no. 5, pp. 709–724, 1993.
- [148] M. Zoppi, D. Zlatanov, and R. Molino, "On the velocity analysis of interconnected chains mechanisms," *Mechanism and Machine Theory*, vol. 41, no. 11, pp. 1346–1358, 2006.

- [149] H. Bruyninckx, “The 321-HEXA: A fully-parallel manipulator with closed-form position and velocity kinematics,” in *IEEE International Conference on Robotics and Automation, ICRA (Albuquerque, USA)*, pp. 2657–2662, 1997.
- [150] F. Pierrot, P. Dauchez, and A. Fournier, “HEXA: a fast six-DOF fully-parallel robot,” in *Fifth International Conference on Robots in Unstructured Environments*, vol. 2, pp. 1158–1163, 1991.
- [151] H. Bruyninckx and J. de Shutter, “Introduction to intelligent robotics,” tech. rep., Katholieke Universiteit de Leuven, 2001.
- [152] J. Cortés and T. Simeon, “Probabilistic motion planning for parallel mechanisms,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 4354–4359, 2003.
- [153] O. Bohigas, D. Zlatanov, M. Manubens, and L. Ros, “On the numerical classification of the singularities of robot manipulators,” in *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE (Chicago, USA)*, 2012.
- [154] E. J. Haug, J.-Y. Wang, and J. K. Wu, “Dextrous workspaces of manipulators. I. analytical criteria,” *Journal of Structural Mechanics*, vol. 20, no. 3, pp. 321–361, 1992.
- [155] J.-Y. Wang and J. K. Wu, “Dextrous workspaces of manipulators, part 2: Computational methods,” *Journal of Structural Mechanics*, vol. 21, no. 4, pp. 471–506, 1993.
- [156] C. C. Qiu, C.-M. Luh, and E. J. Haug, “Dextrous workspaces of manipulators, part III: Calculation of continuation curves at bifurcation points,” *Journal of Structural Mechanics*, vol. 23, no. 1, pp. 115–130, 1995.
- [157] L. J. Du Plessis and J. A. Snyman, “A numerical method for the determination of dextrous workspaces of Gough–Stewart platforms,” *International Journal for numerical methods in engineering*, vol. 52, no. 4, pp. 345–369, 2001.
- [158] A. M. Hay and J. A. Snyman, “A multi-level optimization methodology for determining the dextrous workspaces of planar parallel manipulators,” *Structural and Multidisciplinary Optimization*, vol. 30, no. 6, pp. 422–427, 2005.
- [159] F.-C. Yang and E. J. Haug, “Numerical analysis of the kinematic dexterity of mechanisms,” *Journal of mechanical design*, vol. 116, no. 1, pp. 119–126, 1994.
- [160] L. Jaillet and J. M. Porta, “Path planning under kinematic constraints by rapidly exploring manifolds,” *IEEE Transactions on Robotics*. To appear.
- [161] L. Jaillet and J. M. Porta, “Asymptotically-optimal path planning on manifolds,” in *Robotics: Science and Systems*, 2012.
- [162] F. Bourbonnais, “Utilisation optimale de l’espace de travail des robots parallèles en affrontant certains types de singularités,” Master’s thesis, École de Technologie Supérieure, Université du Québec, 2012.





# Author Index

- Abdel-Malek, K. 64, 67  
Adkins, F. A. 41, 64, 67, 69, 81, 85, 87  
Agrawal, S. K. 126  
Albu-Schaffer, A. 6  
Albus, J. 127  
Allgower, E. L. 30, 70  
Altuzarra, O. 41, 91  
Ambrose, R. F. 126  
Amezua, E. 91  
Angeles, J. 3, 4, 12, 45, 81, 83, 85, 132  
Avilés, R. 91
- Batalin, M. A. 126  
Bayo, E. 12, 31  
Benhabib, B. 4  
Bhattacharya, S. 98  
Bigras, P. 110, 141  
Blackmore, D. 64  
Bohigas, O. 5, 34, 51, 140  
Bonev, I. A. 4, 6, 41, 44, 45, 50, 51, 55, 64, 81, 85–88, 110, 111, 117, 132, 140, 141  
Borgstrom, B. J. 126  
Borgstrom, P. H. 126  
Borràs, J. 4  
Bosscher, P. 115, 126  
Bostelman, R. V. 126, 127  
Bourbonnais, F. 110, 141  
Bruyninckx, H. 132  
Bryson, L. S. 126  
Burdet, E. 3, 132
- Campos, L. 110, 141  
Canny, J. 17  
Cantó, J. 31  
Cao, Y. 81, 85  
Castelli, G. 64  
Castro-Lacouture, D. 126  
Ceccarelli, M. 64, 127  
Celaya, E. 79  
Chablat, D. 41, 51, 64, 87, 110  
Chen, I. 98  
Cheng, P. 126  
Cloud, M. J. 77
- Codourey, A. 3, 132  
Corcho, F. 31  
Cortés, J. 139  
Coulombe, J. 6, 132, 141  
Cox, D. 29  
Creemers, T. 5, 79, 80
- Dagalakis, N. 127  
Daney, D. 126  
Dasgupta, B. 4, 98, 99  
Dash, A. K. 98  
Dauchez, P. 132  
Davidson, J. K. 34  
De Jalón, J. G. 12, 31  
de Shutter, J. 132  
Donelan, P. S. 4  
Du Plessis, L. J. 140  
Duffy, J. 34, 47, 54, 64
- Ebert-Uphoff, I. 115, 126
- Fang, S. 127  
Fardanesh, B. 63, 64  
Fattah, A. 126  
Fenton, R. G. 4  
Fink, J. 126  
Fisher, J. C. 126  
Fournier, A. 132  
Franitza, D. 127  
Freudenstein, F. 63, 64
- Georg, K. 30, 70  
Ghosh, A. 98  
Gilbert, R. M. 126  
Gosselin, C. M. 3, 4, 6, 12, 41, 44, 50, 51, 53, 55, 63, 64, 81, 83, 85–88, 117, 125, 140  
Gouttefarde, M. 126  
Gupta, K. C. 63, 64
- Haddadin, S. 6  
Hansen, E. 30  
Hansen, J. A. 63, 64  
Harmon, T. C. 126  
Hatwal, H. 98

- Haug, E. J. 41, 64, 67, 69, 81, 85, 87, 140  
Hay, A. M. 140  
Henderson, M. E. 5, 30, 100, 101, 103  
Hernandez, A. 41, 91  
Hiller, M. 126, 127  
Hirzinger, G. 6  
Honegger, M. 3, 132  
Hu, B. 64  
Huang, Z. 81, 85  
Hubert, J. 115  
Hunt, K. H. 34, 54  
Husty, M. 64, 81
- Jaillet, L. 5, 106, 139, 140  
Ji, W. 81, 85  
Jiang, Q. 4, 81, 85  
Jo, D. Y. 81, 85  
Jordan, B. L. 126  
Joy, K. 64
- Kaiser, W. J. 126  
Kazerounian, S. M. K. 63, 64  
Kearfott, R. B. 77  
Keller, H. B. 101  
Khairallah, N. 64  
Kim, J. W. 4  
Kim, S. 126  
Kohkawa, H. 126  
Kohli, D. 67  
Krantz, S. G. 19, 65, 68, 100  
Kumar, V. 126
- Lahouar, S. 127  
Li, H. 4, 41, 51, 53, 55, 125  
Li, T.-Y. 30  
Little, J. 29  
Lu, Y. 64  
Luh, C.-M. 41, 64, 67, 69, 81, 85, 87, 140  
Lytle, A. M. 126
- Ma, O. 4  
Macho, E. 41, 91  
Mallik, A. K. 4, 98  
Manubens, M. 5, 34, 51, 140  
Matsushima, T. 126  
McCarthy, J. M. 1
- Merlet, J.-P. 4, 44, 50, 52, 63, 64, 81, 85, 125, 126, 132  
Messina, E. 126  
Mielczarek, S. 127  
Molfino, R. 132, 141  
Moore, R. E. 77  
Morgan, A. 30  
Mouly, N. 44, 50, 63, 64  
Mruthyunjaya, T. S. 4, 98, 99  
Murao, Y. 126
- Norvig, P. 105
- Oblak, D. 67  
Oh, S.-R. 126  
O'Shea, D. 29  
Ott, C. 6  
Ottaviano, E. 64, 127
- Park, F. C. 4  
Parks, H. R. 19, 65, 68, 100  
Patrikalakis, N. M. 30  
Pérez, J. J. 31  
Pernkopf, F. 81  
Pierrot, F. 132  
Pinto, C. 41, 91  
Porta, J. M. 5, 30, 31, 37, 41, 46, 77, 79, 80, 106, 139, 140  
Primrose, E. J. F. 63, 64  
Pusey, J. 126
- Qiu, C. C. 81, 140
- Rastegar, J. 63, 64  
Rheinboldt, W. C. 101  
Richard, M. J. 4, 41, 51, 53, 55, 125  
Riechel, A. T. 115, 126  
Rojas, N. 91  
Romdhane, L. 127  
Ros, L. 5, 30, 31, 34, 37, 41, 46, 51, 77, 79, 80, 117, 140  
Rosales, C. 5  
Roth, B. 63, 64  
Rull, Alex 3  
Russell, S. J. 105  
Ryu, J.-C. 64, 81, 85, 126

- Sabater, A. 117  
Saidi, K. S. 126  
Scott, N. A. 126  
Sefrioui, J. 44, 50  
Sen, S. 4, 98  
Servatius, B. 91  
Shai, O. 91  
Sherbrooke, E. C. 30  
Shi, Y. 64  
Simeon, T. 139  
Sljoka, A. 91  
Snyman, J. A. 64, 140  
Sommese, A. J. 30  
Soni, A. H. 63, 64  
St-Onge, B. M. 4, 41, 51, 53, 55, 125  
St.-Pierre, E. 6  
Stealey, M. J. 126  
Stemmer, A. 6  
Stone, W. C. 126  
Stump, E. 126  
Sturmfels, B. 29  
Sukhatme, G. S. 126  
Tadokoro, S. 126  
Thomas, F. 4, 5, 30, 31, 37, 41, 46, 77, 79, 80, 91, 117  
Torras, C. 4  
Tsai, L.-W. 54  
Tsai, Y. C. 63, 64  
Verhoeven, R. 126, 127  
Voglewede, P. A. 115  
Waldron, K. J. 54  
Walster, G. W. 30  
Wampler, C. W. 30, 79  
Wang, J.-Y. 41, 64, 67, 69, 81, 85, 87, 140  
Wenger, P. 41, 51, 64, 87, 110  
Whiteley, W. 91  
Williams II, R. L. 126  
Wimbock, T. 6  
Wu, J. K. 140  
Yang, F.-C. 140  
Yang, G. 98  
Yang, J. 64  
Yeh, H. J. 64, 67  
Yeo, S. H. 98  
Zeghoul, S. 127  
Zein, M. 41, 64  
Zhou, H. 81, 85  
Zlatanov, D. 4, 11–13, 16, 20, 25, 34, 37, 53, 55, 117, 132, 139–141  
Zoppi, M. 132, 141