

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Meccanica e Scienze Avanzate dell'Ingegneria
Curriculum Meccanica Applicata
Ciclo XXVI

Settore Concorsuale di afferenza: 09/A2

Settore Scientifico disciplinare: ING-IND/13

**Displacement Analysis of
Under-Constrained Cable-Driven Parallel Robots**

Presentata da:

Ghasem Abbasnejad Matikolaie

Relatore della Tesi:

Ing. Marco Carricato

Coordinatore della Scuola di Dottorato:

Prof. Vincenzo Parenti Castelli

Esame finale anno 2014

Displacement Analysis of Under-Constrained Cable-Driven Parallel Robots



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Ghasem Abbasnejad

Department of Industrial Engineering

University of Bologna

A thesis submitted for the degree of

Doctor of Philosophy

I would like to dedicate this thesis to:
the memory of my father;
and my mother.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Marco Carricato, for accepting me to work with him and providing me with the opportunity to undertake research on this topic. I am indebted to him for sharing his invaluable expertise in kinematics and robotics with me. I could not have imagined completing this thesis without his guidance and support.

I would also like to extend my appreciation to Prof. Vincenzo Parenti Castelli for warmly accepting me as a member of his research group, for his support and for providing me with all the facilities necessary to complete this thesis.

I would like to thank all of my colleagues in GRAB (Group of Robotics and Articular Biomechanics). It was a great pleasure working with all of them. Their support, not only as colleagues but also as friends, made me feel at home despite being outside my country of origin.

I am also very grateful to Adrian Lutey for his help in editing the use of English in this thesis.

I would like to express gratitude to the University of Bologna for granting me a generous scholarship that has been a great financial support for my study. Without this contribution I am not sure that I would have been able to complete this thesis.

As it was necessary to acquire more insight into the mathematics of robots in order to accomplish my research, I spent five months studying "Computational Algebraic Geometry" under the supervision of Prof. Manfred Husty at the University of Innsbruck. I would like to give a special thanks to him for kindly hosting me and for sharing his enlightening insights into kinematic geometry with me. I would also like to thank all members of his group who friendly supported me during the period I spent there.

Finally, my deepest gratitude goes to my family, for their unconditional support throughout my life and my studies.

Abstract

This dissertation studies the geometric static problem of under-constrained cable-driven parallel robots (CDPRs) supported by n cables, with $n \leq 6$. The task consists of determining the overall robot configuration when a set of n variables is assigned. When variables relating to the platform posture are assigned, an inverse geometric static problem (IGP) must be solved; whereas, when cable lengths are given, a direct geometric static problem (DGP) must be considered. Both problems are challenging, as the robot continues to preserve some degrees of freedom even after n variables are assigned, with the final configuration determined by the applied forces. Hence, kinematics and statics are coupled and must be resolved simultaneously.

In this dissertation, a general methodology is presented for modelling the aforementioned scenario with a set of algebraic equations. An elimination procedure is provided, aimed at solving the governing equations analytically and obtaining a least-degree univariate polynomial in the corresponding ideal for any value of n . Although an analytical procedure based on elimination is important from a mathematical point of view, providing an upper bound on the number of solutions in the complex field, it is not practical to compute these solutions as it would be very time-consuming. Thus, for the efficient computation of the solution set, a numerical procedure based on homotopy continuation is implemented. A continuation algorithm is also applied to find a set of robot parameters with the maximum number of real assembly modes for a given DGP. Finally, the end-effector pose depends on the applied load and may change due to external disturbances. An investigation into equilibrium stability is therefore performed.

The present dissertation is structured as follows. Chapter 2 is devoted to the description of some mathematical techniques concerning the solution of systems of polynomials. In particular, techniques based on computational algebraic geometry and homotopy continuation are discussed. The Dietmaier algorithm is also presented for computation of the upper bound on the number of real solutions of a system of polynomials. In Chapter 3, cable-driven parallel robots are modelled and a strategy is provided for deriving the equations governing the related geometric and static problems. In Chapter 4, the aforementioned equations are obtained for any value of $n \leq 6$ and the corresponding problem-solving elimination procedure are implemented. By obtaining a least-degree univariate polynomial in each case, a proof of the number of solutions in the complex field is provided. Subsequently, an estimation of the upper bound on the number of real solutions is obtained by applying the Dietmaier algorithm. Chapter 5 is devoted to the software DGP – Solver,

composed to solve the direct displacement analysis. The high complexity of this problem justifies the necessity for composing software capable of efficiently obtaining complete solution sets. Finally, Chapter 6 concludes the thesis by summarising the results obtained throughout Chapters 2 to 5.

Contents

Contents	v
List of Figures	viii
Nomenclature	viii
1 Introduction	1
1.1 Serial and parallel manipulators	2
1.2 Cable-driven parallel manipulators	5
1.2.1 Fully-constrained CDPRs	6
1.2.2 Under-constrained CDPRs	10
1.2.2.1 The challenges in studying displacement of under-constrained CD- PRs and the objective of the thesis	14
2 Mathematical development for solving systems of polynomials	16
2.1 Computational algebraic geometry	17
2.1.1 Polynomials and affine varieties	17
2.1.2 Ideals	18
2.1.3 Monomial orders and polynomial division	19
2.1.4 Groebner bases	20
2.1.5 Solving polynomial systems based on Groebner bases	21
2.1.5.1 Elimination ordering	21
2.1.5.2 Eigenproblem method	22
2.1.5.3 Diallytic elimination	23
2.2 Homotopy Continuation	24
2.2.1 Basic polynomial continuation	26
2.2.2 Coefficient-parameter homotopy continuation	28
2.3 Dietmaier’s algorithm	28
2.3.1 Distance functions between solutions	29
2.3.2 Decreasing the distance between a pair of complex conjugate solutions	29
2.3.3 Increasing the distance between a pair of real solutions	32

3	Geometric Static model of under-constrained cable-driven parallel robot	33
3.1	Mathematical geometric static model of CDPRs	34
3.2	Rodrigues' parameterisation	38
3.3	Study's parameterisation	39
3.4	Dietmaier's parameterization	39
4	Problem-solving algorithm for the geometric static problem of under-constrained CDPRs	41
4.1	The elimination approach	42
4.1.1	Inverse geometric static problem of 2-2 CDPRs	42
4.1.2	Inverse geometric static problem of 3-3 CDPRs	45
4.1.2.1	Inverse geometric static problem of 3-3 CDPR with assigned orientation	45
4.1.2.2	Inverse geometric static problem of 3-3 CDPR with assigned position	48
4.1.3	Inverse geometric static problem of 4-4 CDPRs	50
4.1.3.1	Inverse geometric static problem of 4-4 CDPRs with assigned orientation	51
4.1.3.2	Inverse geometric static problem of 4-4 CDPR with assigned position	52
4.1.4	Direct geometric static problem of CDPRs	53
4.1.4.1	Direct geometric static problem of 2-2 CDPRs	54
4.1.4.2	Direct geometric static problem of 3-3 CDPRs	55
4.1.4.3	Direct geometric static problem of 4-4 CDPR	57
4.1.4.4	Direct geometric static problem of 5-5 CDPRs	59
4.2	Numerical computation of the solution set of the DGP	61
4.2.1	Eigenvalue formulation	62
4.2.2	Homotopy continuation	62
4.2.2.1	General homotopy continuation	63
4.2.2.2	Parameter homotopy continuation	64
4.3	Number of real-valued solutions of the DGP	65
4.3.1	Governing equations and parameterisation	66
4.3.2	Application of the Dietmaier algorithm	67
4.3.2.1	Auxiliary parameters of the algorithm	68
4.3.2.2	Case study	70
5	DGP – Solver	72
5.1	Feasible equilibrium configurations	72
5.1.1	The software DGP – Solver	77
5.1.1.1	Input file	77
5.1.1.2	Output files	78
5.2	Case study	81
6	Conclusions	86

Appendix A	88
References	96

List of Figures

1.1	The SCARA robot, manufactured by MITSUBISHI (Mitsubishi [2013])	2
1.2	The Art 72-500 full flight simulator: an example of the Gough-Stewart platform (Baltic Aviation [2013])	3
1.3	The Agile Eye: a 3-DOF spherical parallel mechanism (Laval University [2013])	3
1.4	The IRB 340 FlexPicker: an example of the Delta robot (ABB [2013])	4
1.5	The Falcon-7 (Kawamura et al. [2000])	6
1.6	The CAT4 robot configuration (Kossowski and Notash [2002])	7
1.7	The WARP robot configuration (Tadokoro et al. [2002])	7
1.8	The general structure of the BetaBot (Behzadipour and Khajepour [2005])	8
1.9	The general structure of the LCDR (Alikhani et al. [2009])	9
1.10	Examples of the point mass cable robot	10
1.11	Concept of the portable rescue system proposed by Tadokoro et al. [1999]	11
1.12	The STRING-MAN configuration (Surdilovic et al. [2007])	12
1.13	The NeReBot overall view and structural diagram (Rosati et al. [2007])	12
1.14	The InTensino test rig (built in 2002) for relatively compact rigid bodies up to 400 kg (Gobbi et al. [2011])	13
2.1	Schematic of path tracking	26
2.2	Schematic of Dietmier’s algorithm	30
3.1	A CDPR with n cables: (a) geometric model and (b) static model.	36
4.1	Geometric model of a cable-driven parallel robot with 2 cables: (a) operation mode I; (b) operation mode II (Carricato and Merlet [2013])	43
4.2	A cable-driven parallel robot with 4 cables: (a) model for the IGP with assigned orientation, (b) model for the IGP with assigned position.	51
5.1	Schematic of a robot with one slack cable.	73
5.2	Stability of a pendulum	74
5.3	Schematic configurations of DGP solutions for 2-cable robot reported in Table 5.2	81
5.4	Schematic configurations of DGP solutions for robot with 4 cables reported in Table 5.3	82
5.5	Schematic of the robot samples used for the stochastic investigation.	83

Chapter 1

Introduction

A *manipulator* is a device used to manipulate materials without direct contact (Wikipedia [2013]). This definition may be found as a first result of a simple search for the term "manipulator" in Google. More precisely, as stated by Angeles [2002], manipulators are a subclass of dynamic mechanical system. A dynamic system is a system with three elements: a state, an input and an output. A mechanical system is a dynamic system composed of mechanical elements. Furthermore, a man-made mechanical system can be either controlled or uncontrolled and, in the former case, the classification may be further divided into robotic or non-robotic¹. As with most current industrial robots, a robotic mechanical system may be programmable. Hence, by the term manipulator, a programmable mechanical system that assists in executing a particular manipulation is intended. This component is the subject that will be addressed in this dissertation.

In recent decades, a growing demand has been witnessed for the use and control of manipulators in various industrial applications to optimise productivity in production and to increase reliability, precision and access to environments unreachable by humans. Examples of manipulators are the well-known six-axis industrial manipulators, six-degree-of-freedom flight simulators, walking machines, mechanical hands and rolling robots. For many years, the most common manipulator structure implemented in industry consists of joining several kinematic joints successively to obtain a serial kinematic chain, which has an anthropomorphic character resembling a human arm. These are called *serial manipulators*. Due some drawbacks of these devices, however, another type of manipulator called the *parallel manipulator* has seen some use. A parallel manipulator consists of a base platform, a moving platform and various legs. Each leg, in turn, is a kinematic serial chain whose end links are the two platforms. In general, these two types of robots are the main features of conventional manipulators. A brief overview of these two types of manipulators will be presented.

¹Non-robotic systems are those supplied with primitive controllers, mostly analogue, such as thermostats, servo valves etc. (Angeles [2002])

1.1 Serial and parallel manipulators

A *serial manipulator* consists of a chain of rigid bodies, each being linked to its predecessor and successor by a one-degree-of-freedom joint. Exceptions are the two end links, which are coupled to either a predecessor or successor, but not to both (Angeles [2002]; Merlet [2006]). The *SCARA robot* is a well-known example of a serial manipulator, presented in Fig. 1.1. Although serial manipulators are



Figure 1.1: The SCARA robot, manufactured by MITSUBISHI (Mitsubishi [2013])

the most common type used in robotics, they suffer from two main drawbacks: poor ability to transport load and poor accuracy (Angeles [2002]; Merlet [2006]). These drawbacks are consequences of the serial coupling nature of the links. Each link, in addition to the load imposed on the end-effector, must support that imposed by the subsequent links and must therefore be more robust than its successor. Thus, each link in the chain and, consequently, the robot, becomes heavy. Moreover, as links are joined to each other in series, errors are magnified from the base to the end-effector. Therefore a small measurement error leads to a large error in the position of the end-effector.

Parallel manipulators are instead closed-chain mechanisms in which an end-effector is linked to a fixed base by at least two independent kinematic chains (Merlet [2006]). The load of the end-effector is distributed across the chains so that each must support only part of the end-effector load. For example, as shown in Fig. 1.2, the well-known Gough-Stewart platform consists of an end-effector supported by 6 chains. When the manipulator is in its central position, each actuator carries approximately 1/6 of the total load. As a consequence, the chosen link size may be smaller and the overall weight of the manipulators lower. Bending deformation of the links is reduced and the stiffness of the manipulator increased. The amplitude of error is almost the same for the actuators and end-effector, as actuator errors only slightly affect the position of the end-effector. Due to these properties and some other advantages of parallel manipulators versus their serial counterparts, parallel manipulators have attracted increasing attention over the last few decades. They have been widely used in industrial, medical and mining applications, as well as for walking machines, planetary exploration, high precision machine



Figure 1.2: The Art 72-500 full flight simulator: an example of the Gough-Stewart platform ([Baltic Aviation \[2013\]](#))

tools and the like. In much of the literature, the Gough-Stewart platform is reported as the first parallel manipulator applied in industry. It was first developed by [Gough \[1956-1957\]](#) in 1954 for a universal tyre testing machine and later, in 1965, [Stewart \[June 1965\]](#) published a paper in which he proposed the same manipulator as a flight simulator. Because the manipulator provides all 6 DOF, it has also been implemented in many other applications such as Agile Eye, motion simulations, underground excavation, milling machines etc.

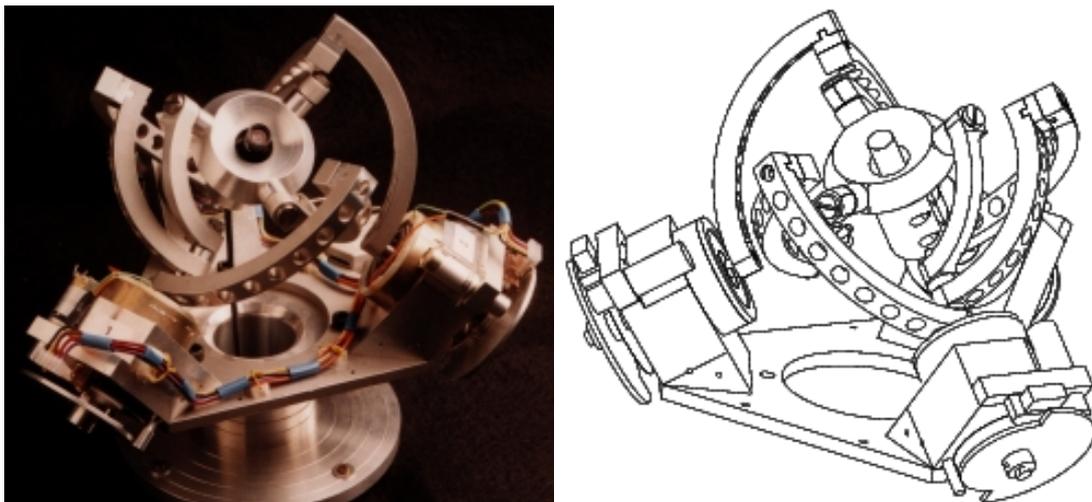


Figure 1.3: The Agile Eye: a 3-DOF spherical parallel mechanism ([Laval University \[2013\]](#))

There are many applications, however, in which motion with less than 6-DOF is needed and the complexity of the 6-DOF Gough-Stewart platform is unnecessary. For these cases, parallel mechanisms

with limited-DOF and simpler structures are preferred.

A 3-DOF spherical parallel mechanism (Fig. 1.3), introduced by [Gosselin and Hamel \[1994\]](#), is such a mechanism that is used for applications like camera orienting devices and wrist motion simulators. The structure of this manipulator is such that the axes of all revolute joints intersect at one common point that is the centre of rotation of the device. The manipulator only produces the three rotational degrees of freedom that are needed for the application.

The well-known Delta robot, with 3 translational degrees of freedom ([Clavel \[1988\]](#)), originally invented by Clavel in 1988, is another parallel robot with limited DOF. As depicted in Fig. 1.4, the robot consists of 3 identical chains, each of which consists of a lever and a parallelogram four-bar linkage. The lever is attached via an actuated revolute joint to the base on one side and via a revolute joint to a parallelogram on the other side. At the end of this parallelogram is a revolute joint that is linked to the end-effector.



Figure 1.4: The IRB 340 FlexPicker: an example of the Delta robot ([ABB \[2013\]](#))

The Delta has attracted much research interest for its unique properties, such as simple inverse and direct kinematic solutions, decoupling of the position and orientation of the moving platform and very high acceleration due to the light weight of moving parts. The manipulator has been used in a large number of applications, particularly in the electronics, food and pharmaceutical industries for which reliable product standards are required. A more detailed survey of the application of parallel manipulators is presented by [Patel and George \[2012\]](#).

Though conventional serial and parallel manipulators have been efficiently implemented in many industrial applications, there are some in which such conventional manipulators are not practical due to problems such as limited workspace, weight or size constraints, vibration, noise, cost etc. In long-reach robotic applications, such as inspection and repair in shipyards and airplane hangars, for example, workspace requirements may be three to four orders of magnitude larger than what conventional robots can provide ([Oh and Agrawal \[2005\]](#)). In applications of modern assembly operations with high speed

robotic positioning systems, use of serial link manipulators presents problems relating to weight, vibration and cost, while parallel manipulators suffer from limited workspace and high motor torque ripple (S.Kawamura et al. [1995]).

To overcome these problems, in recent years, use of cables in the place of rigid links has received increasing attention. The following section presents more details on such cable-driven parallel robots.

1.2 Cable-driven parallel manipulators

Cable-driven parallel robots (CDPRs) employ cables in place of rigid body extendable legs to control the posture of an end-effector. In these manipulators, the pose of the end-effector is controlled appropriately by controlling the length of the cables. Cables are usually rolled on drums attached to a base and are actuated by rotary motor. Cable-driven parallel robots have special advantages such as a larger workspace, reduced manufacturing and maintenance costs, ease of assembly and disassembly, high transportability, superior modularity and ease of reconfiguration. As a consequence of the flexibility and low weight of the cables, implementation of long cables can be easily handled. By controlling cable lengths within broad ranges, a very large workspace can be accessed. This property makes CDPRs appropriate for applications in which force transmission or access over a long distance is needed. One of the main advantages of parallel manipulators is the reduced moving mass and inertial force. Using cables instead of rigid links further decreases the moving mass, as the actuators do not change position and are attached to a fixed base such that the only moving parts are the cables and end-effector. As a consequence, a robot with higher speed and agility is obtained and the payload of the robot may be increased (Behzadipour et al. [2003]).

Manufacturing costs of cable-driven robots are significantly lower than those of conventional manipulators. A cable-driven manipulator is simple to set up with low-cost hardware. As outlined by Merlet and Daney [2010], a portable robot with a load capacity of more than 1 ton can be set up by assembling a number of low cost winches and cables.

The cable characteristic that makes the control of the end-effector challenging is its inability to withstand compression. Due to this fact, if a CDPR is intended to control a total number of end-effector degrees of freedom (DOFs), f , at least $f + 1$ cables are required (Kurtz and Hayward [1995]; Ming and Higuchi [1994]; Roberts et al. [1998]). This redundancy of control actions is usually necessary to guarantee tensile force in all cables and to prevent them from becoming slack when the imposed load on the moving platform changes. This maintains the control by cable of all degrees of freedom of the moving platform. Such systems of cable-driven parallel robots are called *fully-controlled* or *completely-restrained*. If the end-effector instead preserves some level of freedom once the actuators are locked and the cable lengths are fixed, the system is called *under-constrained* or *incompletely-restrained*. This typically occurs when the end-effector is controlled by a number of cables, n , smaller than f . In such a system, the platform may move and deviate from its equilibrium position. Therefore, the posture of the moving platform depends on, in addition to the length of the cables, the imposed load on the moving platform. Fully-constrained robots are attracting increasing interest in the research

community and, as such, a rich literature exists. In the following section, some examples of these robots are presented.

1.2.1 Fully-constrained C DPRs

As one of the first examples of over-constrained cable robots, an ultrahigh speed robot FALCON is introduced by Kawamura et al. [2000], taking advantage of the small mass of cables in comparison with rigid links. As depicted in Fig. 1.5, the robot is fully-constrained and an object or end-effector is suspended by 7 wires (tendons). It achieves peak accelerations of up to 43g and maximum velocities of 13m/s, even though small motors (60W) are used.

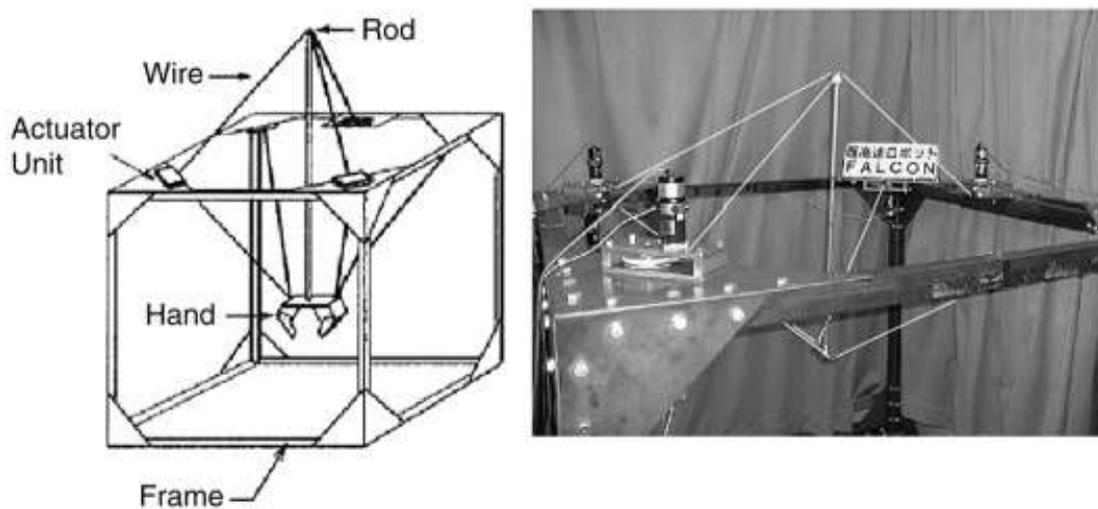


Figure 1.5: The Falcon-7 (Kawamura et al. [2000])

The CAT4 (Cable Actuated Truss 4-DOF) robot (Kossowski and Notash [2002]) is a 4 DOF parallel robot that utilises a passively jointed central linkage and six control cables for actuation. The 4 DOFs of this robot include 3 translational and 1 rotational DOF (pitch angle). The robot, shown in Fig. 1.6, consists of a wishbone-shaped structural base at the top that forms the backbone of the manipulator and may be collapsible for transit. A passively jointed linkage descends from the centre of this structure, comprised of six rigid links (two upper linkage beams, two lower linkage beams, a medial beam and a tie beam), each of which is a truss in order to minimise structural weight. The truss elements that form the central linkage are connected with 18 revolute joints. A subset of these joints is sensed by position encoders or potentiometers to determine the end-effector position and orientation. This jointed central linkage gives the end-effector, which is attached to the lower linkage beams, the required 3 translational DOFs and 1 rotational DOF. Brakes are required on a subset of the central linkage joints in order to ensure single-string fail-safe operation.

A mechanism using eight cables called WARP is introduced by Tadokoro et al. [2002]. The main concept of this design is the location of some actuator units at the same place in order to avoid tangling of cables. The combination of actuator units and their positions have considerable effect, while the

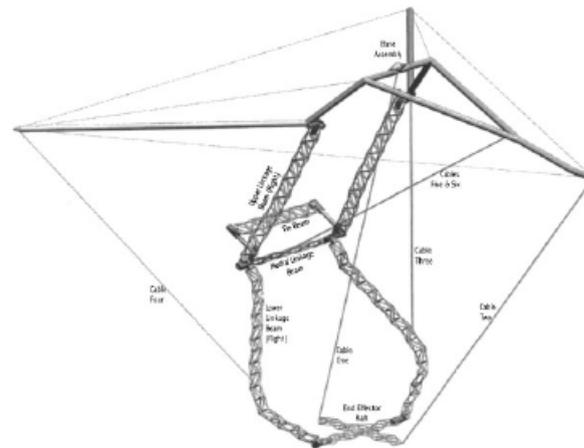


Figure 1.6: The CAT4 robot configuration (Kossowski and Notash [2002])

combination of suspended points influences performance. In order to optimise the configuration, all possible mechanisms are classified and compared. As explained by Tadokoro et al. [2002], there are 17 different combinations of actuator unit groups. Considering symmetric cable configurations in both vertical and horizontal directions while minimising the possibility of cable tangling, maximising the efficiency of moment generation and avoiding cable collisions with the environment, an 8-cable robot, WARP, is chosen as the best combination of actuator units (Fig. 1.7).

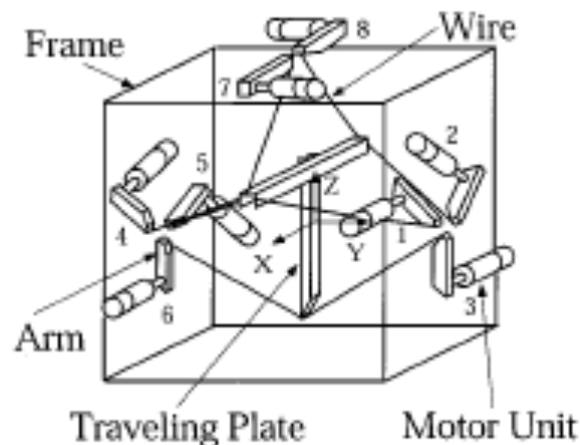


Figure 1.7: The WARP robot configuration (Tadokoro et al. [2002])

This architecture has the following remarkable points:

1. Larger rotational motion range than other cable-driven parallel mechanisms. This advantage allows large virtual acceleration for a long period of time.

2. The motion platform may stay on the ground, as the bottom of the platform does not have any mechanism.
3. All walls may be used for scene projection of computer graphics, like in the CAVE virtual reality system.
4. Redundancy of cables improves safety in the case of failure.

Behzadipour and Khajepour [2005] introduce a new cable-based parallel robot, BetaBot, in which cables are used to apply the necessary kinematic constraints for three pure translational degrees of freedom. This design demonstrates that an over-constrained robot may be obtained not only by implementing additional cables, but also by linking the end-effector to a constraining mechanism. In order to maintain tension in the cables, a collapsible element called a 'spine' is placed between the end-effector and the base of the robot. The kinematic analysis of this robot is similar to that of a rigid link parallel manipulator provided that the cables are in tension. In Fig. 1.8, the general design of the BetaBot is shown. Three pairs of parallel cables are attached to the end-effector and are collected by three spools after passing through guide holes on the frame of each spool. Each spool shaft is connected to a motor, permitting the modification of the respective cable's length. The spools and their frames are attached to the base, making, together with the cables and end-effector, three parallelograms. The spine is a collapsible element used to apply a pushing force between the base and end-effector. This element can be a spring or an air cylinder that is connected to the end-effector and base by universal joints.

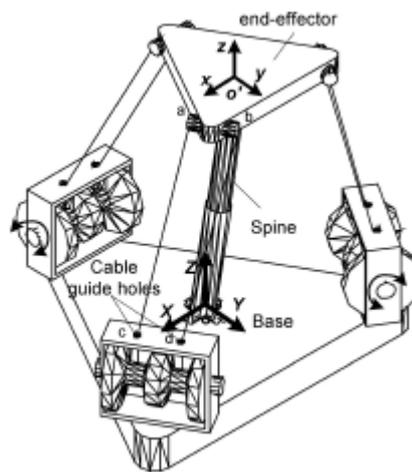


Figure 1.8: The general structure of the BetaBot (**Behzadipour and Khajepour [2005]**)

Alikhani et al. [2009] introduce a cable-driven mechanism based on the concept of the BetaBot (**Behzadipour and Khajepour [2005]**). This mechanism is called the Large Cable Delta Robot (LCDR) and provides motion with three translational degrees of freedom suitable for manipulation on a large scale. Extra cables are utilised to ensure tension in all such elements; differing from the BetaBot where cable pretension is provided by a passive cylinder or 'spine'. Though use of this cylinder simplifies

the design, it puts limits on the size of the workspace and also complicates control of the robot as the mechanism is not fully restrained by the cables.

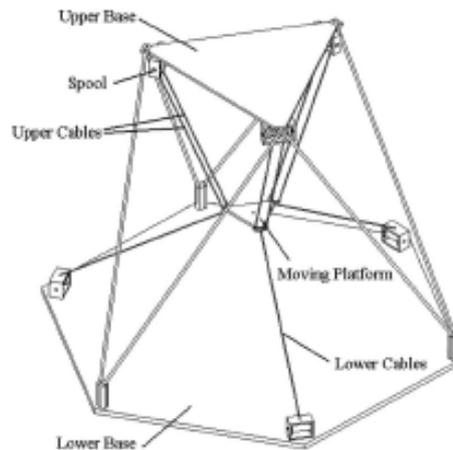


Figure 1.9: The general structure of the LCDR (Alikhani et al. [2009])

These shortcomings are addressed by the extra cables, replacing the spine and making it a fully-restrained cable-driven mechanism. In Fig. 1.9, a schematic design of the LCDR is shown. The middle triangle is the moving platform (end-effector) and the lower and upper triangles are the bases. Three pairs of parallel cables are attached to the moving platform and collected by three spools mounted on the upper base after passing through guide holes on the frame of each spool. Each spool shaft is connected to a motor (not shown in the figure), permitting the modification of the respective cable's length. The spools and their frames are attached to the base, making, together with cables and end-effector, three parallelograms. The lower cables, which consist of three cables and the corresponding motorised spools, are used to maintain tension in the mechanism. The robot therefore needs six rotary actuators: three motors at the top, each driving one pair of parallel cables, and three motors at the bottom, driving the lower cables. In this mechanism, the cable arrangement eliminates rotational motion, leaving the moving platform with three degrees of freedom. The mechanism provides potential for large-scale manipulation and robotics in harsh environments. The mechanism can develop tensile forces in all cables to maintain its rigidity under arbitrary external loading.

Bosscher et al. [2006] study wrench-feasible workspace (WFW) of point-mass cable robots as one of the important classes of fully-constrained cable robots. A method is presented for analytically generating the boundaries of the WFW for these robots. This method uses the available net wrench set, which is the set of all wrenches that a cable robot can apply to its surroundings without violating the tension limits of the cables. The geometric properties of this set permit calculation of the boundaries of the WFW. Complete analytical expressions for the WFW boundaries are detailed for planar cable and spatial point-mass cable robots. In these manipulators, all cables attach to a single point on the end-effector and can change length to control the position of the end-effector. Typically, the end-effector

is modelled as a lumped mass located at the point of intersection of the cables. As an example, the manipulators in Fig. 1.10 can be modelled as point-mass cable robots. Due to the fact that the structure of point-mass cable robots is simple, they are relatively easy to implement and are used in applications such as camera positioning ([Cablecam](#); [SkyCam](#)), haptics ([Bonivento et al. \[1997\]](#)) and cargo handling ([Gorman et al. \[2001\]](#)).

Several studies are available in the literature concerning the WFW. The reader may refer to [Bouchard et al. \[2010\]](#); [Ghasemi et al. \[2009\]](#); [Gouttefarde and Gosselin \[2006\]](#); [Gouttefarde et al. \[2011\]](#); [Lau et al. \[2011\]](#)



Figure 1.10: Examples of the point mass cable robot

1.2.2 Under-constrained CDPRs

As mentioned previously, in contrast to their fully-constrained counterparts, under-constrained CDPRs are equipped with a number of cables, n , that is smaller than f , allowing the control of only n end-effector degrees of freedom. The use of CDPRs with a limited number of cables is justified in several applications such as measurement, rescue, service and rehabilitation operations, in which the task to be performed requires limited control or a limitation of dexterity is acceptable in order to decrease complexity, cost, set-up time, likelihood of cable interference etc. For example, [Tadokoro et al. \[1999\]](#) proposes an under-constrained cable-driven robot for search and rescue following large-scale urban earthquakes. Search and rescue is an important application of robotics in such incidents. Robots must have the potential to efficiently save a number of lives and to reduce the exposure to danger of rescue squads. The key points in the rescue are different from industrial robots and intelligent robots. According to [Tadokoro et al. \[1999\]](#), the essential points in rescue robots following urban earthquakes are as follows:

- streets are narrowed by destroyed buildings;
- rescue robots should provide access to buried people through debris;
- a large number of robots are necessary at one time; and,
- the search activity for buried people is the most critical element.

The above investigation concludes that essential points to robotic systems for search and rescue are: portability, promptness of installation on site and simplicity. An under-constrained cable-driven parallel robot, as shown in Fig. 1.11, is proposed that satisfies most of the requirements of the rescue robot system. It is portable and may be assembled rapidly in destroyed houses.

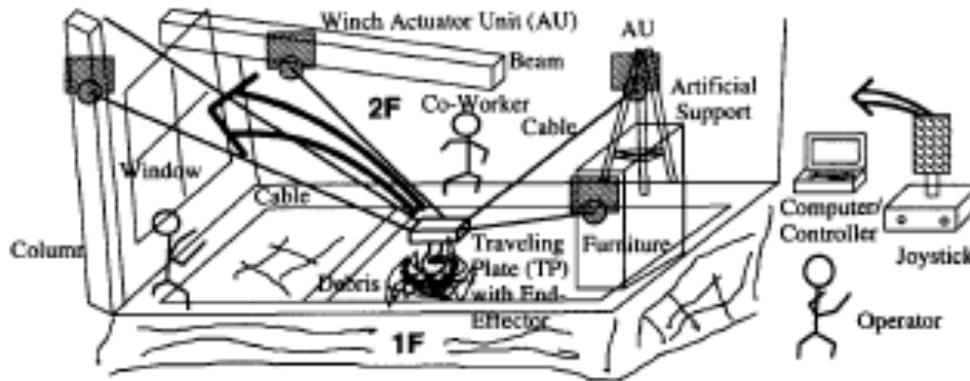


Figure 1.11: Concept of the portable rescue system proposed by [Tadokoro et al. \[1999\]](#)

A winch actuator unit consists of a winch for changing cable lengths, a battery for energy and a local controller of position and force. Multiple cables are connected to a travelling plate with an end-effector that moves with many degrees of freedom. Teleoperation is performed by a multi-degree-of-freedom joystick. This system is carried by rescue staff into a collapsed building. The installation (assembly and identification) may be completed promptly, permitting the transportation of debris immediately.

[Surdilovic et al. \[2007\]](#) address concepts regarding modular, light-weight and interactive gait rehabilitation devices and robots based on wire-robot technology. A prototype active weight-bearing and balancing system (STRING-MAN) is presented, which opens possibilities for assisting rehabilitation of posture, balance and gait motor functions. As shown in Fig. 1.12, the system consists of a wire robot in which the wires are connected via a user interface (harness and corsage) to the human trunk and pelvis.

By closing the kinematic chains in such a way, the person is uniquely integrated into the wire robot system representing a ‘common robot platform’. This robotic structure optimally provides the requirements for controlling the posture in 6-DOFs, as well as for balancing the weight on the legs according to specific gait patterns and training programs. Moreover, by sensing the interaction of forces, this system can quantify the patient’s effort and therefore control the interaction. For example, the system can support the patient’s own initiative by applying force or impedance control. STRING-MAN is a powerful robotic system for supporting gait rehabilitation and restoring motor functions by combining the advantages of partial body-weight bearing (PWB) with a number of industrial and humanoid robot control functions. A safe, reliable and dynamically controlled weight-suspension and posture control supports the patients in autonomously performing gait recovery training from the early stages of rehabilitation onwards.



Figure 1.12: The STRING-MAN configuration (Surdilovic et al. [2007])

Rosati et al. [2007] present the development and clinical tests of NeReBot (NEuroREhabilitation roBOT), a 3-DOF wire-driven robot for post stroke upper-limb rehabilitation. A diagram of the mechanical structure of NeReBot is shown in Fig. 1.13. In basic terms, the robot consists of a set of three wires independently driven by three electric motors. The base of the robot is designed in such a way that the patient can be treated while sitting in a wheelchair (Fig. 1.13) or lying in a hospital bed. The base consists of a C-shaped frame, featuring omni-directional wheels that can fit under any commercial hospital bed. A square-section column is fixed on the central part of the base, holding three horizontal round-section, hollow aluminium arms on top, which support the wires.

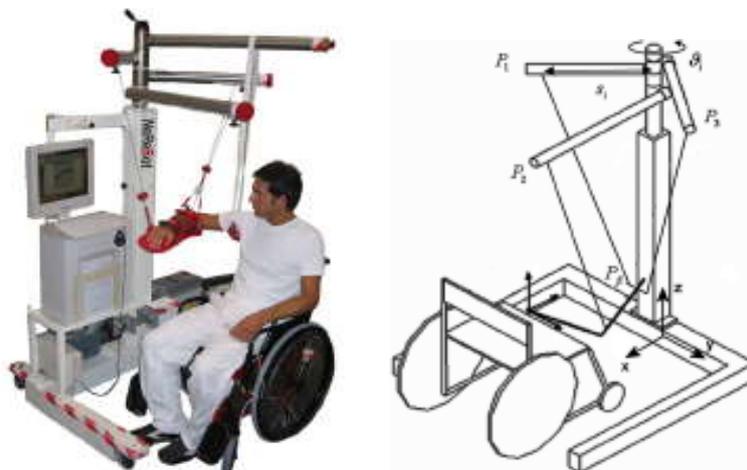


Figure 1.13: The NeReBot overall view and structural diagram (Rosati et al. [2007])

The free ends of each wire are fastened to the patient's arm by means of a special splint. By controlling wire length, rehabilitation treatment (based on the passive or active spatial motion of the limb) can be delivered over a large working space. The arm trajectory is set by the therapist through a very simple teach-by-showing procedure, enabling most common 'hands on' therapy exercises to be reproduced by the robot. Compared to other rehabilitation robots, NeReBot offers advantages due to its low-cost mechanical structure, intrinsically safe treatment thanks to the use of wires, good acceptance by patients who do not feel constrained by an 'industrial-like' robot, transportability as it can easily be placed beside a hospital bed and/or wheelchair and good trade-off between low number of DOF and spatial performance. These features, along with the very encouraging results of the first clinical trials, make the NeReBot a good candidate for adoption in the rehabilitation of subacute stroke survivors.

A method for measuring the inertial properties of rigid bodies by using a parallel cable-driven robot is presented by [Gobbi et al. \[2011\]](#). Common practice for the estimation of rigid body inertial properties is their computation by means of 3D CAD models; however, this estimation is prone to errors generated by the large amount of data that must be entered, geometric tolerances or defects and uncertainty in material properties such as density. Errors of more than 10% (and sometimes much more) in the inertial tensor components are commonly experienced for complex systems composed of thousands of parts. The most reliable way to get an estimation of the inertial properties of a body (or of a set of inter-connected bodies) is instead via experimental tests. A test rig for the measurement of the inertial tensor of a rigid body, proposed by [Gobbi et al. \[2011\]](#), was first developed in 2001 and was constructed the year after to measure the inertial properties of vehicle components such as engines and gearboxes (Fig. 1.14).



Figure 1.14: The InTensino test rig (built in 2002) for relatively compact rigid bodies up to 400 kg ([Gobbi et al. \[2011\]](#))

The proposed test rig was essentially a multi-cable pendulum. The pendulum was made to swing freely with well defined initial conditions. The test rig was composed of a frame for carrying a rigid body and three or four cables connecting this component to another external frame. The cables were

connected at both ends by low-friction spherical joints, each comprising a Hook's joint fitted with roller bearings and an axial bearing. Given a rigid body with a particular mass, the method allows identification and measurement of the centre of gravity and the inertial tensor with a single test. The proposed technique is based on the analysis of the free motion of the multi-cable pendulum to which the body under consideration is connected. The motion of the pendulum and the forces acting on the system are recorded and the inertial properties identified by means of a mathematical procedure based on a least squares estimation. The natural frequencies of the pendulum and the accelerations involved are quite low, making this method suitable for many practical applications.

1.2.2.1 The challenges in studying displacement of under-constrained CDPRs and the objective of the thesis

Compared to fully-constrained manipulators, under-constrained CDPRs have seen little attention in the literature and the study of these types of robots is still an open field. For instance, analysis of displacement as a first step in the study of robots is ongoing for the case of under-constrained CDPRs. The major challenge in the displacement study of under-constrained CDPRs consists of the intrinsic coupling between kinematics and statics (or dynamics). When a fully-constrained CDPR operates in the portion of its workspace in which the required set of output wrenches is guaranteed with purely tensile cable forces, the posture of the end-effector is determined in a purely geometric way by assigning cable lengths. Conversely, for an under-constrained CDPR, when the actuators are locked and the cable lengths are assigned, the end-effector is still able to move, so the actual configuration is determined by the applied forces. As a consequence, the end-effector posture depends on both the cable lengths and equilibrium equations. Moreover, as the end-effector pose depends on the applied load, it may change due to external disturbances. As such, these factors are fundamental to the investigation of equilibrium stability. The necessity of simultaneously dealing with kinematics and statics increases the complexity of position problems aimed at determining the overall robot configuration when a set of n variables is assigned. The solution to these problems is significantly more difficult than analogous tasks concerning rigid-link parallel manipulators. In the literature, some procedures have been presented to solve the following problems.

[Ghasemi et al. \[2010\]](#) suggest the use of neural networks to solve the system of polynomial equations associated with forward displacement analysis of under-constrained cable-driven parallel manipulators. According to this scheme, the neural network is trained by solving the corresponding inverse displacement analysis, which is much easier than the original problem, over a large set of poses. The resulting neural network may provide a good approximation of the forward displacement analysis in many cases; however, it does not guarantee the convergence to an equilibrium pose in general.

[Michael et al. \[2011\]](#) propose a solution to the case of $n = 2$ cables (i.e. the planar case). This solution is obtained by finding the equilibrium points on the coupler curve of the analogous planar four-bar linkage. In the same work, the authors adopt an energetic approach to the case of $n = 3$, using the fact that an equilibrium pose corresponds to a minimum in the potential energy. This leads to a non-convex optimisation problem, whose optima are obtained by varying the initial guess of a local

optimisation procedure. [Fink et al. \[2011\]](#) instead relax the same formulated optimisation problem into a convex optimisation problem. The optimum objective value of the relaxed problem may be regarded as a lower bound on the optimum value of the original problem. Furthermore, the authors provide geometric conditions under which the lower bound is guaranteed as tight; that is, under which the optima of the relaxed and original problem coincide.

In a parallel effort, [Jiang and Kumar \[2010\]](#) were able to compute all stable equilibrium poses for a class of special cases in three-dimensional space. A particular geometry is a member of this class if:

- the cable attachment points on the rigid body are located at the vertices of a regular polygon;
- the rigid-body centre of mass is at the centroid of the said polygon;
- the fixed cable attachment points (i.e. those on the supporting frame) form a regular polygon with the same planes of symmetry as the rigid-body polygon;
- this fixed polygon is perpendicular to gravity (i.e. lies in a horizontal plane).

These constraints allow the decomposition of the spatial problem into several planar problems, which may be solved by computing the stationary points on the coupler curve of the equivalent four-bar linkage.

[Collard and Cardou \[2013\]](#), proceeding very much like [Fink et al. \[2011\]](#), relax the energy minimisation problem. The relaxation, however, is different and the lower bound provided by it is cast in a branch-and-bound algorithm, which allows the computation of a global optimum to the energy-minimisation problem. This global optimum corresponds to the lowest equilibrium pose and provides a tight lower bound on the height of the rigid-body centre of gravity, a piece of information that is useful for guaranteeing no collisions while moving a cable-suspended object above possible obstacles. The proposed method can be applied to problems with large numbers of cables, which sets it apart from the other solutions.

In light of this brief overview, it may be noted that most studies related to under-constrained CD-PRs rely on purely local numerical solution strategies; whereas, no adequate consideration is given to conceiving static geometric models capable of providing broader solution strategies, tailored to obtain the complete solution sets of the nonlinear equations governing the problem.

In the present thesis, a methodology is proposed for the kinematic, static and stability analysis of general under-constrained n - n CDPRs, namely parallel robots in which a fixed base and a mobile platform are inter-connected by n cables, with $n \leq 6$, and the anchor points on the base and the platform are generally distinct. The procedure aims at effectively solving the inverse and direct static geometric problems; that is, at finding the overall robot configuration and cable tensions when a set of n platform posture coordinates or n cable lengths are assigned, under the assumption that a constant force is applied on the platform, that the cables are inextensible and massless, and that interference problems are not present.

Chapter 2

Mathematical development for solving systems of polynomials

Due to the high complexity of systems of polynomials that arise in displacement analysis of under-constrained cable-driven parallel robots, a suitable mathematical framework for their resolution is required. Two approaches are often used in the context of kinematics: elimination methods based on computational algebraic geometry and continuation homotopy. The former is a geometric manifestation of the solutions of systems of polynomial equations (Cox et al. [2005]). It provides a powerful theoretical technique for studying the qualitative and quantitative features of the solution sets. The application of algebraic geometry to kinematic analysis is more natural to a global understanding of the entire solution set, as opposed to finding only some of the solutions (Masouleh [2010]).

On the other hand, the elimination method may be time consuming and, as such, a numerical procedure is required as a robust and fast method for computing the complete set of equations governing the problem for a given set of parameters. To this end, polynomial homotopy continuation may be used. The procedure has the advantage that very little symbolic information must be extracted from a polynomial system to proceed. It often suffices, for example, to simply know the degree of each polynomial, which is easily obtained without a full expansion into terms.

Although the upper bound on the number of solutions of a system of polynomials in the complex field may be known and all solutions computed by the two aforementioned methods, there may not be any available information about the upper bound on the number of real solutions that a family of systems of polynomials may exhibit. In fact, since there may be roots in the solution set that always remain complex, it may result that the number of real solutions is smaller than the number of complex ones. The upper bound on the number of real solutions has significant importance in the kinematic analysis of robots, as it indicates the maximum number of assembly modes that a robot may admit. Thus, finding a set of robot parameters admitting the maximum number of real solutions is a classic question in robot kinematics. For example, although it was known that the direct kinematic problem of the Gough-Stewart platform admitted 40 solutions, a set of the robot parameters with such a number of real solutions was unknown for years until Dietmaier [1998] proposed an algorithm capable of providing them. In this context, the same algorithm is adapted to finding a set of robot parameters with

the maximum number of real assembly modes to the problems emerging in displacement analysis of CDPRs.

Accordingly, in the first part of this chapter, the concept of computational algebraic geometry is introduced and the required algorithms presented for solving the systems of polynomials arising in displacement analysis of under-constrained CDPRs. In the subsequent part, the homotopy continuation method is presented briefly with its various features and some required terminology. In the final section of this chapter, the Dietmaier algorithm is also briefly discussed.

2.1 Computational algebraic geometry

The origin of algebraic geometry dates back to Descartes' introduction of coordinates to describe points in Euclidean space and the idea of describing curves and surfaces by algebraic equations. Over the long history of the subject, both powerful general theories and detailed knowledge of many specific examples have been developed (Cox et al. [2005]). Recently, the advent of computer algebra systems has made it possible to implement many theories of algebraic geometry relating to robotics using computational algorithms (Brunthaler [2006]; Masouleh [2010]; Pfurner [2006]). As will be seen in the following chapters, similar problems emerging in displacement analysis of under-constrained CDPRs are analysed using these concepts. Before discussing computational algebraic geometry, some important terminology must be provided. In particular, an introduction to the ideals of the polynomial ring $\mathbb{K}[x_1, \dots, x_n]$ will be provided in the following section, followed by an explanation of the total order of monomials and an introduction to the Groebner basis as one of the powerful elimination strategies for solving systems of polynomials. There are several software packages that offer the Groebner basis technique for solving systems of polynomials. In the present thesis, Maple (Maplesoft) is mainly used to implement the Groebner base technique.

2.1.1 Polynomials and affine varieties

A **field** is a set where addition, subtraction, multiplication and division are defined with their usual properties. Standard examples are the real numbers and complex numbers; whereas, integers are not a field since division fails. Due to the importance of both real and complex solutions to the problem at hand, most of the computation will be performed in the complex field $\mathbb{K} = \mathbb{C}$. On the other hand, for simplification of computation at many points, the field of rational numbers $\mathbb{K} = \mathbb{Q}$ will be used.

A **monomial** in x_1, \dots, x_n is a product of the form:

$$x_1^{\alpha_1} \cdot x_2^{\alpha_2} \dots x_n^{\alpha_n} \tag{2.1}$$

where all exponents $\alpha_1, \dots, \alpha_n$ are nonnegative integers. The **total degree** of this monomial is the sum $\alpha_1 + \dots + \alpha_n$. The notation for monomials is simplified by setting $\alpha = (\alpha_1, \dots, \alpha_n)$ as a n -tuple of

nonnegative integers and, correspondingly:

$$x^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \dots x_n^{\alpha_n} \quad (2.2)$$

Knowing the definition of a monomial, a polynomial is defined as follows:

A **polynomial**, f in x_1, \dots, x_n with coefficients in \mathbb{K} , is a finite linear combination (with coefficients in \mathbb{K}) of monomials. It can be written in the form:

$$f = \sum_{\alpha} a_{\alpha} x^{\alpha} \quad (2.3)$$

where the sum is over a finite number of n -tuples $\alpha = (\alpha_1, \dots, \alpha_n)$. The set of all polynomials in $\mathbf{X} = x_1, \dots, x_n$ with coefficients in \mathbb{K} is denoted $\mathbb{K}[\mathbf{X}]$. When dealing with a polynomial such as $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$, a_{α} is called the **coefficient** of the monomial x^{α} . If $a_{\alpha} \neq 0$, then $a_{\alpha} x^{\alpha}$ is a **term** of f and the total degree of f , denoted $\deg(f)$, is the maximum $|\alpha|$ where the coefficient a_{α} is nonzero.

According to a special property of polynomials over the field of complex numbers where every non-constant polynomial $f \in \mathbb{K}$ has a root in \mathbb{K} , the concept of affine variety is defined as follows.

Let f_1, \dots, f_s be polynomials in $\mathbb{K}[\mathbf{X}]$, then $V(f_1, \dots, f_s)$, called the **affine variety** defined by f_1, \dots, f_s , is:

$$V(f_1, \dots, f_s) = \{(x_1, \dots, x_n) \in \mathbb{K}^n : f_i(x_1, \dots, x_n) = 0, \quad 1 \leq i \leq s\} \quad (2.4)$$

An affine variety, $V(f_1, \dots, f_s) \subset \mathbb{K}^n$, is the set of all solutions of the system of equations $f_1(x_1, \dots, x_n) = \dots = f_s(x_1, \dots, x_n) = 0$. In the present work, the letters V, W etcetera, where not otherwise stated, are used to denote affine varieties.

2.1.2 Ideals

Ideals define the basic algebraic objects that are used in the present work. A subset $\langle I \rangle \in \mathbb{K}[X]$ is an ideal if it satisfies:

- $0 \in \langle I \rangle$
- If $f, g \in \langle I \rangle$, then $f + g \in \langle I \rangle$
- If $f \in \langle I \rangle$ and $h \in \mathbb{K}[\mathbf{X}]$, then $hf \in \langle I \rangle$

Given a collection of polynomials, $I = \{f_1, \dots, f_s\} \in \mathbb{K}[\mathbf{X}]$, all polynomials that may be built up from these by multiplication or sums of arbitrary polynomials are denoted as:

$$\langle I \rangle = \langle f_1, \dots, f_s \rangle = \{p_1 f_1 + \dots + p_s f_s : p_i \in \mathbb{K}[\mathbf{X}] \quad \text{for } i = 1, \dots, s\} \quad (2.5)$$

From this definition it can be proven that $\langle I \rangle$ is an ideal. $\langle I \rangle$ is called the **ideal generated by** $I = \{f_1, \dots, f_s\}$.

2.1.3 Monomial orders and polynomial division

One of the key elements for all operations relating to algorithms for solving a system of polynomials is ordering of terms. This will be used regularly in the present work. A **monomial order** on $\mathbb{K}[\mathbf{X}]$ is any relation $>$ on the set of monomials x^α in $\mathbb{K}[\mathbf{X}]$ satisfying:

1. $>$ is a total (linear) ordering relation
2. $>$ is compatible with multiplication in $\mathbb{K}[\mathbf{X}]$; that is, if $x^\alpha > x^\beta$ and x^γ is any monomial, then $x^\alpha x^\gamma = x^{\alpha+\gamma} > x^{\beta+\gamma} = x^\beta x^\gamma$
3. $>$ is well-ordering; that is, every nonempty collection of monomials has a smallest element under $>$.

According to this definition, there are many ways to define monomial orders. The most important of these, in the present study, are as follows.

Lexicographic order (lex) first compares exponents of x_1 and, in the case of equality, compares exponents of x_2 and so forth. Therefore, if x^α and x^β are monomials in $\mathbb{K}[\mathbf{X}]$, $x^\alpha >_{lex} x^\beta$ if, in the difference $\alpha - \beta \in \mathbb{Z}^n$, the leftmost nonzero entry is positive.

Graded lexicographic order (grlex) first compares the sum of all exponents and, in the case of equality, applies lexicographic ordering. In this case, $x^\alpha >_{grlex} x^\beta$ if $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$; or if $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$, then $x^\alpha >_{lex} x^\beta$.

Graded reverse lexicographic order (grevlex) first compares the total degree then compares exponents of the last indeterminate x_n , reversing the result. In the case of equality, a similar comparison of x_{n-1} is performed and so forth ending in x_1 . Accordingly, $x^\alpha >_{grevlex} x^\beta$ if $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$ or if $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ and the rightmost nonzero entry is negative in the difference $\alpha - \beta \in \mathbb{Z}^n$.

There are many other monomial orders besides the ones considered here; however, the aforementioned orders will be used at many points in the present work. It will be shown that, while grevlex is almost always much easier for performing some computations, there are many cases in which calculations based on lex or grlex must be performed. It will also be shown that by implementing the computations in certain orders and combinations they may be performed more efficiently.

For a particular monomial order $>$, the terms are considered to be of the form $c_\alpha x^\alpha$. The leading term of f (with respect to $>$) is then the product $c_\alpha x^\alpha$, where x^α is the largest monomial appearing in f in the ordering $>$. The notation $LT_{>}(f)$ is used for the leading term, or simply $LT(f)$ where it is clear which monomial order is being used. Furthermore, if $LT(f) = c_\alpha x^\alpha$, then $LC(f) = c_\alpha$ is the leading coefficient of f and $LM(f) = x^\alpha$ is the leading monomial. One of the first uses of monomial orders is in the **division algorithm**, defined as follows:

Taking any monomial order $>$ in $\mathbb{K}[\mathbf{X}]$ and letting $I = \{f_1, \dots, f_s\}$ be a set of polynomials in $\mathbb{K}[\mathbf{X}]$, then every $f \in \mathbb{K}[\mathbf{X}]$ can be written as:

$$f = a_1 f_1 + \dots + a_s f_s + r \tag{2.6}$$

Term r is a remainder of f on division by I , with either $r = 0$ or r being a linear combination of monomials, none of which is divisible by any of $LT_{>}(f_1), \dots, LT_{>}(f_s)$.

According to this definition of division, reordering I or changing the monomial order can produce different a_i and r in some cases.

2.1.4 Groebner bases

Since a division algorithm in $\mathbb{K}[\mathbf{X}]$ has been obtained, one would expect the possibility to determine if a given $f \in \mathbb{K}[\mathbf{X}]$ is a member of an ideal, $\langle I \rangle = \langle f_1, \dots, f_s \rangle$, by computing its remainder on division. From Eq.(2.6) it follows that, if $r = 0$ on dividing by $I = \{f_1, \dots, f_s\}$, then $f = a_1 f_1 + \dots + a_s f_s$ and $f \in \langle I \rangle = \langle f_1 \dots f_s \rangle$. It can be shown, however, that $r = 0$ is not guaranteed for every $f \in \langle I \rangle$ if an arbitrary basis I is used for $\langle I \rangle$. To produce zero remainders for all elements of I upon division, a Groebner base as a basis of the ideal is defined as follows. The key idea is that once a monomial ordering is chosen, each $f \in \mathbb{K}[\mathbf{X}]$ has a unique leading term, $LT(f)$.

Monomial ideal: An ideal $\langle I \rangle$ in $\mathbb{K}[\mathbf{X}]$ is a monomial ideal if it is generated by a collection (not necessarily finite) of monomials. Let $\langle I \rangle$ be an ideal in $\mathbb{K}[\mathbf{X}]$, other than $\langle 0 \rangle$. Let $LT(I)$ denote the set $\{LT(f_i), f_i \in \langle I \rangle\}$ and $\langle LT(I) \rangle$ denote the ideal generated by the elements of $LT(I)$. $\langle LT(f_1), \dots, LT(f_t) \rangle$ and $\langle LT(I) \rangle$ can be different ideals. In general, $\langle LT(f_1), \dots, LT(f_t) \rangle \in \langle LT(I) \rangle$.

Hilbert basis theorem: Every ideal $\langle I \rangle$ in $\mathbb{K}[\mathbf{X}]$ has a finite generating set. That is, $I = \{g_1, \dots, g_r\}$ for some $g_1, \dots, g_r \in I$.

Groebner bases: For a fixed monomial order $>$ on $\mathbb{K}[\mathbf{X}]$ and $\langle I \rangle \subset \mathbb{K}[\mathbf{X}]$ as an ideal, a Groebner basis $G_{>}[I]$ of $\langle I \rangle$ (with respect to $>$) is a finite collection of polynomials $G = \{g_1, \dots, g_t\} \subset \langle I \rangle$ with the property that for every nonzero $f \in \langle I \rangle$, $LT(f)$ is divisible by $LT(g_i)$ for some i .

It can be proven that the remainder, r , on division of a generic polynomial $f \in \mathbb{K}[\mathbf{X}]$ by a Groebner basis, $G_{>}[I]$ of $\langle I \rangle$, is uniquely dependent on the choice of monomial order only and not on the way the division is performed. The remainder, r , is called the normal form of f . Indeed, uniqueness of remainders is the main characterisation of Groebner bases. In other words, all of the monomials in r are in the normal set of $\langle I \rangle$, which is a collection of all monomials not in $\langle LT(g_i) \rangle$. The normal set, $\mathbf{N}[I]$, contains all monomials that may appear in the remainder of all polynomials on division by $G_{>}[I]$; defined as follows:

Normal set: Let $\langle I \rangle$ be in $\mathbb{K}[\mathbf{X}]$ and $G_{>}[I]$ be its Groebner basis (with respect to $>$); then:

$$\mathbf{N}[I] = \{x^\alpha \mid x^\alpha \notin \langle LT(G_{>}[I]) \rangle\} \tag{2.7}$$

is the normal set (of I with respect to $>$).

As a consequence of the Hilbert basis theorem it can be stated that every ideal $\langle I \rangle$ in $\mathbb{K}[\mathbf{X}]$ other than $\langle 0 \rangle$ has a Groebner basis. Furthermore, any Groebner basis $G_{>}[I]$ of $\langle I \rangle$ is a basis of $\langle I \rangle$.

From the definition of a Groebner basis, stated above, one may conclude that if $G_{>}[I]$ is a Groebner basis of $\langle I \rangle$, then the normal set of $\langle I \rangle$ is just the normal set of the leading monomials of $G_{>}[I]$.

Useful for many purposes relating to Groebner bases is an algorithm developed by Buchberger that

takes an arbitrary generating set $\{f_1, \dots, f_s\} \subset \langle I \rangle$ and produces a Groebner basis $G_{>}[I]$ of $\langle I \rangle$ from it. This algorithm works by forming new elements of I using expressions guaranteed to cancel leading terms and uncover other possible leading terms. It was developed from two basic tools, the reduction or division process and the critical pairs or S -polynomials.

Let f, g in $\mathbb{K}[\mathbf{X}]$ be non-zero polynomials.

1. If $\deg(f) = \alpha$ and $\deg(g) = \beta$, then γ is $(\gamma_1, \dots, \gamma_n)$, where $\gamma_i = \max(\alpha_i, \beta_i)$ for each i . Term x^γ is called the **least common multiple of $LM(f)$ and $LM(g)$** , noting that $x^\gamma = LCM(LM(f), LM(g))$.
2. The **S -polynomial** of f and g is:

$$S(f, g) = \frac{x^\gamma}{LT(f)}f - \frac{x^\gamma}{LT(g)}g \quad (2.8)$$

An S -polynomial, $S(f, g)$, is designed to produce cancellation of leading terms. Using the S -polynomials another characterisation of Groebner bases is obtained, which is, from an algorithmic point of view, more useful than the definition:

Let $\langle I \rangle$ be a polynomial ideal. Then a basis $G = \{g_1, \dots, g_s\}$ for $\langle I \rangle$ is a Groebner basis $G_{>}[I]$ of $\langle I \rangle$ if and only if, for all pairs $i \neq j$, the remainder of the division of $S(g_i, g_j)$ by $G_{>}[I]$ (listed in some order) is zero.

More details on the algorithm and how to compute the Groebner bases can be found in [Cox et al. \[2007\]](#). There are many computer algebra systems in which Buchberger's algorithm is installed; however, as the algorithm may need large amounts of storage and take many steps, the actual Groebner basis computation may fail in many cases.

2.1.5 Solving polynomial systems based on Groebner bases

How to solve a system of equations based on the knowledge of Groebner bases will be outlined in the present section.

2.1.5.1 Elimination ordering

The most straightforward procedure for solving a system of polynomials, I , is based on the elimination properties of Groebner bases computed according to some elimination monomial order such as the Lexicographic order. For this technique, the main tools are the Elimination and Extension Theorems. If $\mathbf{X}_l = [x_1, \dots, x_l]$ is a list of l variables in $\mathbf{X} = [x_1, \dots, x_{l-1}, x_l, x_{l+1}, \dots, x_n]$ and $\mathbf{X} \setminus \mathbf{X}_l$ is the (ordered) relative complement of \mathbf{X}_l in \mathbf{X} , a monomial order $>_l$ on $\mathbb{K}[\mathbf{X}]$ is of l -elimination type provided that any monomial involving a variable in \mathbf{X}_l is greater than any monomial in $\mathbb{K}[\mathbf{X} \setminus \mathbf{X}_l]$. If $G_{>_l}[I]$ is a Groebner basis of $\langle I \rangle$ with respect to $>_l$, then $G_{>_l}[I] \cap \mathbb{K}[\mathbf{X} \setminus \mathbf{X}_l]$ is a basis of the l th elimination ideal, $\langle I_l \rangle := \langle I \rangle \cap \mathbb{K}[\mathbf{X} \setminus \mathbf{X}_l]$ ([Cox et al. \[2007\]](#)). The elements of I_l are actually linear combinations of $I = \{f_1, \dots, f_n\}$, with polynomial coefficients that eliminate x_1, \dots, x_l from the equations $f_1 = \dots = f_n = 0$. As $\langle I \rangle$ comprises n variables, the polynomials of $\langle I_l \rangle$ contain $n - 1$ variables, the polynomials of $\langle I_2 \rangle$

$n - 2$, and so on. $\langle I_{n-1} \rangle$ comprises a single variable and, thus, contains a scalar multiple of the least-degree polynomial of $\langle I \rangle$ in that variable. The implemented l -elimination monomial order induces grevlex orders on both $\mathbb{K}[\mathbf{X}_l]$ and $\mathbb{K}[\mathbf{X} \setminus \mathbf{X}_l]$.

Considering a point $(a_{l+1}, \dots, a_n) \in V(I_l) \subset \mathbb{K}^{n-l}$ as a partial solution, it can be proven that, in \mathbb{K} , this partial solution may be extended to $(a_l, a_{l+1}, \dots, a_n)$ in $V(I_{l-1})$. The Elimination theorem shows that a *lex* Groebner basis, $G_{>}$, successively eliminates more and more variables. Accordingly, to find all solutions of the system one may start with the polynomials in $G[I]$ with the fewest variables, solve them and then try to extend these partial solutions to those of the whole system by applying the extension theorem one variable at a time. A *lex* Groebner basis, however, tends to be very large and thus, even for problems of moderate complexity, they have little chance of actually being computed. Conversely, the graded reverse lexicographic order produces bases that are endowed with no particular structure suitable for elimination purposes; however, it provides more efficient calculations. In this perspective, the FGLM algorithm (Faugère et al. [1993]), which converts a Groebner basis from one monomial order to another, may be called upon to compute elimination ideals of the type $\langle I \rangle \cap \mathbb{K}[\mathbf{X} \setminus \mathbf{X}_l]$, starting from a Groebner basis $G[I]$ computed with respect to a grevlex monomial ordering. For example, one can derive $G_{>l}[I]$ from $G[I]$, for some l . Once $G_{>l}[I]$ is known, one may extract the subset of all polynomials of $G_{>l}[I]$ that comprise variables in $\mathbf{X} \setminus \mathbf{X}_l$ only. These polynomials will form a Groebner basis of $\langle I_l \rangle$ with respect to $\text{grevlex}(\mathbf{X} \setminus \mathbf{X}_l)$. By computing elimination ideals via the FGLM algorithm, a least-degree polynomial in one variable may be obtained.

2.1.5.2 Eigenproblem method

In contrast to the discussion in the previous section, where elimination was necessary to obtain a univariate polynomial, the eigenproblem method only needs some Groebner basis, not necessarily an elimination order. In this section, a simple explanation of the method by Sommese and Wampler [2005] is provided.

Consider an ideal $\langle I \rangle$ in $\mathbb{K}[\mathbf{X}]$ with Groebner basis $G_{>l}[I]$. Let λ be any linear combination:

$$\lambda = c_0 + c_1 x_1 + \dots + c_n x_n \quad (2.9)$$

for given constants c_0, \dots, c_n . Consider the normal set, $\mathbf{N}[I]$:

$$\mathbf{N}[I] = [t_1, \dots, t_n]^T \quad (2.10)$$

Let the polynomial $P_i(\mathbf{X}) = \lambda t_i$ for some i and consider a solution of $\langle I \rangle$ as \mathbf{X}^* .

Since $f(\mathbf{X}^*) = 0$ for any $f \in \langle I \rangle$, any multiple of a polynomial in the ideal $\langle I \rangle$ can be added to P_i without changing the value of $P_i(\mathbf{X}^*)$. This implies that, if r_i is the remainder of P_i on division by $G[I]$, then $r_i(\mathbf{X}^*) = P_i(\mathbf{X}^*) = \lambda t_i$. But $r_i(\mathbf{X})$ is a sum of terms in the normal set, so it can be written as $r_i = [a_{i1} \dots a_{ik}] \mathbf{N}[I]$. The entries a_{ij} are the constant coefficients in the formulae for the remainders $r_i, i = 1, \dots, k$. Since $r_i - \lambda t_i$ belongs to $\langle I \rangle$, it must vanish on $V[I]$. By assembling all equations of this

kind that may be obtained for t_1, \dots, t_n , one has:

$$(\mathbf{A}[I, \lambda] - \lambda \mathbf{I})\mathbf{N}[I] = 0 \quad (2.11)$$

where $\mathbf{A}[I, \lambda] = [a_{ik}]$ is an $n \times n$ numeric matrix called the multiplication matrix for λ and \mathbf{I} is the $n \times n$ identity matrix. Therefore, by computing remainders using the Groebner basis, an eigenvalue problem is derived. For each eigenvector, $[N]$, a unique solution can be obtained, as it is either in $\mathbf{N}[I]$ or is a leading monomial of G . In the latter case, the solution must simply be evaluated using the Groebner basis element for which it is the leading monomial.

2.1.5.3 Diallytic elimination

As discussed in Section 2.1.5.1, Groebner bases with respect to some elimination monomial orders are required in order to eliminate unknowns. In theory, by computing elimination ideals via the FGLM algorithm, a least-degree polynomial in one variable may be calculated. In practice, however, computing $\langle J_l \rangle$ is very demanding in terms of both computation time and memory usage, and the procedure may likely fail. A more efficient alternative that will be implemented in the present work is provided by the Groebner-Sylvester hybrid approach, proposed by [Dhingra et al. \[2000\]](#). Let $\langle I \rangle$ be an ideal of polynomials in $\mathbb{K}[\mathbf{X}]$ and $G_{>_l}[I]$ be the corresponding Groebner base with respect to an ordering $>_l$. With Groebner base G known, it may be possible to construct Sylvester's matrix using all the polynomials in $G[I]$ or a subset, $H[I]$, of $G[I]$. Accordingly, a variable $x_l \in \mathbf{X}$ may exist such that the number of polynomials in $G[I]$ or $H[I]$ equals the number of monomials in the variables of $\mathbf{X} - x_l$ appearing in the polynomials. The subset $H[I]$ of $G[I]$ may even be derived from the Groebner basis of any elimination ideal of $\langle I \rangle$. Using the FGLM algorithm, a subset of the original unknowns is eliminated, thus computing $G[J_l]$ for some l . The elimination process is then completed by applying Diallytic elimination to the polynomials of $G[J_l]$. More specifically, if either the entire set of polynomials in $G[I]$ or a subset, $H[I]$, of $G[I]$ is used to set up the Sylvester's matrix, there may exist q polynomials h_i , $i = 1 \dots q$ in terms of $\{x_1, \dots, x_n\} \subset \mathbf{X}$. If each $h_i \in \mathbb{K}[x_1, \dots, x_n]$ is expressed as:

$$h_i = \sum_j a'_{ij} m'_j, \quad a'_{ij} \in \mathbb{K}[x_l], \quad m'_j \in \mathbb{K}[x_1, \dots, x_{l-1}, x_{l+1}, \dots, x_n] \quad (2.12)$$

the number of monomials, m'_j , in $H = \{h_1, \dots, h_q\}$, including 1, may be equal to q (the number of polynomials in H). Considering H as a square system of homogeneous linear polynomials in the unknown monomials $m'_1 = 1, m'_2, \dots, m'_q$, the matrix form may be derived as:

$$\begin{bmatrix} h_1 \\ \vdots \\ h_q \end{bmatrix} = \underbrace{\begin{bmatrix} q \times q \\ a'_{ij} \in \mathbb{K}[x_l] \end{bmatrix}}_S \underbrace{\begin{bmatrix} m'_q \\ \vdots \\ m'_2 \\ m'_1 = 1 \end{bmatrix}}_E \quad (2.13)$$

where S is a $q \times q$ Sylvester's (coefficient) matrix polynomial in x_l and E is a q -dimensional vector comprising all monomials in $G[I_l]$ having variables in $\mathbf{X} - x_l$. Letting the determinant of Sylvester's matrix, $S(x_l)$, vanish yields a spurious-root-free univariate least-degree polynomial in x_l . In this algorithm, the key factor in computing the univariate polynomial efficiently is reaching a compromise between the time required to compute the l th elimination ideal and the size of the Sylvester matrix that is obtained from the corresponding $G[I_l]$.

2.2 Homotopy Continuation

General homotopy continuation consists of a start system with known solutions, a schedule for transforming the start system into a target system and a method for tracking solution paths as the transformation proceeds. Before discussion of each step, some terminology necessary for understanding polynomial continuation must be introduced.

1. **Total degree of polynomial system:** The total degree of a system of n polynomial equations for n unknowns is the product $\prod_{j=1}^n d_j$, where d_j is the degree of the j th polynomial.

2. **Projective Space:**

N -dimensional complex projective space, denoted \mathbb{P}^N , is the space of complex lines through the origin in \mathbb{C}^{N+1} . Points in \mathbb{P}^N are given by $(N+1)$ -tuples of complex numbers $[z_0, \dots, z_N]$, not all zero, with the equivalence relation given by $[z_0, \dots, z_N] \sim [z_0', \dots, z_N']$ if and only if there is a nonzero complex number λ such that $z_j' = \lambda z_j$ for $j = 0, \dots, N$.

This definition makes sense, because a line through the origin in \mathbb{C}^{N+1} is a set of the form:

$$\{(\lambda z_0, \dots, \lambda z_N) \in \mathbb{C}^{N+1} \mid \lambda \in \mathbb{C}\} \quad (2.14)$$

with not all z_i zero. The z_i occurring within the brackets, $[z_0, \dots, z_N]$, are called homogeneous coordinates, even though they are not coordinates on \mathbb{P}^N , but rather coordinates on \mathbb{C}^{N+1} .

3. **Multi-homogeneous polynomial:**

A polynomial system F of n equations, f_1, f_2, \dots, f_n in the n unknowns, z_1, z_2, \dots, z_n is homogenized by partitioning the variables into m collections, denoted Z_1, \dots, Z_m where $Z_j = \{z_{1j}, \dots, z_{k_j j}\}$. So that Z_j contains k_j variables and $\sum_{j=1}^m k_j = n$. Now choosing homogeneous variables z_{0j} for $j = 1$ to m , and including these in Z_j gives $Z_j = \{z_{0j}, z_{1j}, \dots, z_{k_j j}\}$. Replacing the substitution $z_{ij} \leftarrow z_{ij}/z_{0j}$ for $i = 1$ to k_j and $j = 1$ to m , generates a system F' of n equations in $n+m$ unknowns (after we clear the denominators of powers of the z_{0j}).

Such a system is called m -homogeneous, understood to include the case of 1-homogeneous. It is said that a multi-homogeneous system, F , is compatible with the multi-projective space, X , if the dimensions n_1, \dots, n_m match.

As a further explanation of multi-homogeneous polynomials, the following example by [Wampler et al. \[1990\]](#) is provided. Consider the system:

$$\begin{aligned}x^2 - 1 &= 0 \\xy - 1 &= 0\end{aligned}\tag{2.15}$$

which is the intersection of two vertical lines, $x = \pm 1$, with a hyperbola. The system has a total degree of 4, but only two finite solutions: $(x, y) = (1, 1)$ and $(-1, -1)$. Introducing the homogeneous variable w via the substitutions $x \leftarrow x/w, y \leftarrow y/w$, we obtain:

$$\begin{aligned}x^2 - w^2 &= 0 \\xy - w^2 &= 0\end{aligned}\tag{2.16}$$

which, in addition to the original solutions $(w, x, y) = (1, 1, 1), (1, -1, -1)$, has a solution at infinity $(0, 0, 1)$ of multiplicity two. Now, consider what happens if two homogeneous variables are introduced via the substitutions $x \leftarrow x/w_1, y \leftarrow y/w_2$, creating the system:

$$\begin{aligned}x^2 - w_1^2 &= 0 \\xy - w_1 w_2 &= 0\end{aligned}\tag{2.17}$$

Disallowing any solution where $(w_1, x) = (0, 0)$ or $(w_2, y) = (0, 0)$, one may confirm that the only solutions are the original finite solutions: $(w_1, x; w_2, y) = (1, 1; 1, 1)$ and $(1, -1; 1, -1)$. This is due to the different treatment of infinity with the introduction of more than one homogeneous variable. The system is called “2-homogeneous” because there are two homogeneous variables. Thus, it can be seen that the use of multiple homogeneous variables can sometimes reduce the number of solutions at infinity, which will reduce the computational load when calculating all solutions of the system.

4. Bezout’s Theorem:

In a multi-homogeneous system of polynomials, the multi-homogeneous degree of equation l with respect to group j , d_{jl} , is computed as the sum of the variable exponents of group j in any term from polynomial l . Bezout’s theorem states that the Bezout number of a multi-homogeneous system of polynomial equations in complex projective space is equal to the coefficient of $\prod_{j=1}^m \alpha_j^{k_j}$ in the product

$$\prod_{l=1}^n \left(\sum_{j=1}^m d_{jl} \alpha_j \right)\tag{2.18}$$

Applying this formula to equation Eq. (2.17), it is found that the coefficient of $\alpha_1 \alpha_2$ in $2\alpha_1(\alpha_1 + \alpha_2)$ is 2, as expected. It can be shown that for a 1-homogeneous system, this formula yields the total degree.

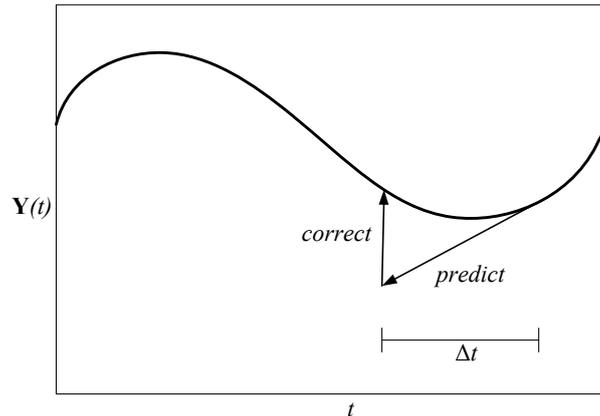


Figure 2.1: Schematic of path tracking

2.2.1 Basic polynomial continuation

Basic polynomial continuation is a path-tracking technique that transforms a start system of polynomial equations with known solutions to a target system whose solutions must be found (Sommese and Wampler [2005]). The method tracks the evolution of a system such as:

$$\mathbf{H}(\mathbf{Y}, t) = \gamma(1-t)\mathbf{F}_0(\mathbf{Y}) + t\mathbf{F}_1(\mathbf{Y}) = \mathbf{0} \quad (2.19)$$

where $\mathbf{F}_0(\mathbf{Y})$ and $\mathbf{F}_1(\mathbf{Y})$ are, respectively, the start and target systems, γ is a randomly selected complex number¹ and t is a real number called a *continuation parameter*. The concept consists of varying t from 0 to 1 while tracking the solutions of the problem from those of $\mathbf{F}_0(\mathbf{Y}) = \mathbf{0}$, known, to those of $\mathbf{F}_1(\mathbf{Y}) = \mathbf{0}$, unknown.

The heart of the continuation method is its path-tracking algorithm. General path trackers must deal with all sorts of difficult issues; for example, a path that bifurcates into several paths or a path that reverses direction. As discussed by Sommese and Wampler [2005], with proper care in forming a homotopy, one can ensure that the paths for solving polynomial systems have none of these troubles; they advance steadily as the homotopy parameter, t , advances and never intersect except possibly at the end target. More precisely, the probability of a singularity occurring on a path is zero. The path-tracking problem can be turned into an initial-value problem for an ordinary differential equation. Subsequently, using a predictor/corrector method based on an explicit homotopy, $\mathbf{H}(\mathbf{Y}, t)$, avoids build-up errors which often accumulate in numerical O.D.E. solvers. Basic prediction and correction, schematically illustrated in Fig.2.1, are both accomplished by considering a local model of the homotopy function via its Taylor series:

$$\mathbf{H}(\mathbf{Y} + \Delta\mathbf{Y}, t + \Delta t) = \mathbf{H}(\mathbf{Y}, t) + \mathbf{H}_{\mathbf{Y}}(\mathbf{Y}, t)\Delta\mathbf{Y} + \mathbf{H}_t(\mathbf{Y}, t)\Delta t + \text{higher-order terms} \quad (2.20)$$

where $\mathbf{H}_{\mathbf{Y}} = \partial\mathbf{H}/\partial\mathbf{Y}$ is the $n \times n$ Jacobian matrix and $\mathbf{H}_t = \partial\mathbf{H}/\partial t$ is of size $n \times 1$. Having a point

¹The parameter γ , usually computed as $e^{i\theta}$ with $\theta \in [-\pi, \pi]$, avoids specially behaved singular paths. More details may be found in Sommese and Wampler [2005].

(\mathbf{Y}_1, t_1) near the path $\mathbf{H}(\mathbf{Y}_1, t_1) \approx 0$, one may predict a new approximate solution at $t_1 + \Delta t$ by setting $\mathbf{H}(\mathbf{Y} + \Delta\mathbf{Y}, t_1 + \Delta t) = 0$ and solving the first-order terms to get:

$$\Delta\mathbf{Y} = -\mathbf{H}_{\mathbf{Y}}^{-1}(\mathbf{Y}_1, t_1)\mathbf{H}_t(\mathbf{Y}_1, t_1)\Delta t \quad (2.21)$$

On the other hand, if $\mathbf{H}(\mathbf{Y}_1, t_1)$ is not as small as desired, one may hold t constant by setting $\Delta t = 0$ and solving the equation to get:

$$\Delta\mathbf{Y} = -\mathbf{H}_{\mathbf{Y}}^{-1}(\mathbf{Y}_1, t_1)\mathbf{H}(\mathbf{Y}_1, t_1) \quad (2.22)$$

These procedures are precisely Euler prediction and Newton correction, respectively. A robust and efficient path tracker may be obtained by halving the time step whenever too many correction iterations are required to stay within a given tolerance of the continuation path and doubling it when the prediction step has been sufficiently accurate several consecutive times.

In general, when no information is known about the roots of the target system, a start system yielding the maximum possible number of solutions must be constructed. Three restrictions on the start system are necessary: all of its solutions must be known, each solution must be nonsingular and the system must have the same multi-homogeneous structure as the target system. First, considering a 1-homogeneous target system, an acceptable start system is:

$$x_j^{d_j} - 1 = 0, \quad (j = 1, \dots, n) \quad (2.23)$$

where d_j is the degree of the j th equation of the target system. Each equation yields d_j distinct solution values for x_j and the entire set of $\prod_{j=1}^n d_j$ is found by taking all possible combinations of these solutions.

Supposing the case of a multi-homogeneous system rather than a 1-homogeneous system, with d_{jl} denoting the degree of equation l with respect to group j . The corresponding start equation is constructed as a product of factors, $\prod_{j=1}^m f_{jl}(x_{j1}, \dots, x_{jk_j})$, where the degree of f_{jl} is d_{jl} . This yields a start system with an identical multi-homogeneous structure as the target system. Choosing sufficiently generic factors (perhaps by choosing random coefficients), the proper number of nonsingular solutions can be assured. Solutions to the system are found by setting one factor from each equation equal to zero. Not all choices of factors set to zero yield solutions. In fact, exactly k_j distinct factors f_{jl} for each group j must be chosen. If the factors are all linear, each solution is found by solving m systems of linear equations; one for each homogeneous group. All solutions to all such choices produce the entire solution set, with exactly the same number of elements as the Bezout number. This number, however, is usually much larger than the number of finite solutions of the target system of equations N_{sol} in the complex field. As a consequence, for $t \rightarrow 1$, many paths diverge to infinity, whereas only a limited number of them, equal to N_{sol} , converge to finite solutions. Tracking of diverging paths causes significant and non-beneficial computational burden.

2.2.2 Coefficient-parameter homotopy continuation

Equations arising in robot kinematics express relationships between various physical parameters. Some of these parameters are the variables, whose unknown values are to be found, while others are known. Accordingly, a system of equations, \mathbf{F} , may be expressed as $\mathbf{F}(\mathbf{Y}, \mathbf{P}) = \mathbf{0}$, where \mathbf{Y} is a vector of unknown values that are to be found and \mathbf{P} are known values that may be considered as a set of geometric parameters. Thus, any one problem can be considered a member of a whole family of problems, defined by letting the parameters range over all admissible values (Sommese and Wampler [2005]). The essence of any continuation method is to track one or more solutions known for one set of parameter values, \mathbf{P}_0 , to get solutions for some new set of parameters, \mathbf{P}_1 . Continuous parameters enter only through the coefficients; that is, the coefficients are functions of the parameters. Assuming that the coefficients are continuous functions of the parameters, a continuous path through the parameter space determines a continuous evolution of the coefficients and is generally also a solution.

When all nonsingular isolated roots of the solutions are known for a general member of a family of equations (for instance, because they are computed by the general strategy discussed in Section 2.2.1), ‘coefficient-parameter’ homotopy continuation efficiently finds solutions for all other robots of the same family. Accordingly, by a suitable homotopy, only the paths originating at the isolated roots of the start system (with N_{sol} such paths) may be tracked; whereas, those corresponding to solutions at infinity may be ignored. In particular, if the N_{sol} isolated roots of a system of equations are known for a generic $\mathbf{P} = \mathbf{P}_0$, the solutions for any other $\mathbf{P} = \mathbf{P}_1$ may be found by tracking the homotopy:

$$\mathbf{F}(\mathbf{Y}, (1-t)\mathbf{P}_0 + t\mathbf{P}_1) = \mathbf{0} \quad (2.24)$$

with t varying from 0 to 1 or, more robustly, along the curve $t = \gamma t' / [1 + (\gamma - 1)t']$, with $t' \in [0, 1]$ and $\gamma \in \mathbb{C}$. For a complete discussion of continuation methods, the reader may refer to Sommese and Wampler [2005].

In this thesis, the software Bertini (Bates et al.) is used as a solver for systems of polynomials by homotopy continuation. The advantages of Bertini over other homotopy-continuation-based packages are its capability to implement user-defined parameter homotopies and its convenient interface with computer algebra systems such as Matlab or Maple.

2.3 Dietmaier’s algorithm

Dietmaier’s algorithm is a continuation procedure for finding a set of robot parameters with the maximum number of real solutions. As with coefficient-parameter homotopy, a start system is defined for a given value of \mathbf{P}_0 and a complete set of solutions is found. Subsequently, an iterative procedure is established to change the system parameters and conveniently vary the solution set. In contrast with parameter homotopy, however, the target parameters \mathbf{P}_1 are unknown *a priori*. The tracked path is adaptively modified in such a way that, at each iteration, the imaginary parts of some complex solutions are decreased and, eventually, as many complex roots as possible are transformed into real

ones.

The start system may be constructed by choosing an arbitrary set of geometric parameters \mathbf{P}_0 and calculating, via general homotopy continuation, the corresponding solution set of $\mathbf{F}(\mathbf{Y}, \mathbf{P}_0)$: $\{\mathbf{Y}_h, h = 1 \dots N_{sol}\}$. Since $\mathbf{F}(\mathbf{Y}, \mathbf{P})$ is algebraic and has real coefficients, the number of real roots is even, whereas complex roots appear in complex conjugate pairs. As shown in Fig. 2.2, \mathbf{P} is iteratively assigned small variations, $\Delta\mathbf{P}$, in such a way that two complex conjugate solutions continually become nearer. This process is achieved by decrementing the absolute value of the imaginary parts until the complex roots are transformed into a double real root and, thereafter, into a pair of distinct real solutions. The procedure is repeated for all pairs of complex conjugate roots. Throughout the process, the algorithm ensures that no two existing real solutions become too close, so as to prevent them transforming back into a double root and, eventually, a complex pair.

2.3.1 Distance functions between solutions

To implement Dietmaier's algorithm, suitable scalar functions must be defined to measure the 'distance' between either a pair of complex conjugate solutions or two real roots.

Accordingly, given a *complex* solution vector comprising n elements, \mathbf{Y}_h , the distance between \mathbf{Y}_h and its complex conjugate, $\bar{\mathbf{Y}}_h$, may be measured as:

$$S_h = \sum_{j=1}^n |Im(Y_{h,j})| \quad (2.25)$$

Indeed, when S_h vanishes, the conjugate solutions coalesce into a double real root.

Correspondingly, for two given *real* solutions, \mathbf{Y}_r and \mathbf{Y}_s ($r \neq s$), the distance between them may be defined by way of a standard 2-norm, namely:

$$D_{rs} = (\mathbf{Y}_r - \mathbf{Y}_s)^T (\mathbf{Y}_r - \mathbf{Y}_s) \quad (2.26)$$

2.3.2 Decreasing the distance between a pair of complex conjugate solutions

Let a complex solution \mathbf{Y}_h be considered. The distance, S_h , between this vector and the corresponding complex conjugate is given by Eq. (2.25). A variation, $\Delta\mathbf{P}$, of the parameter vector must be computed such that S_h decreases.

Differentiating Eq. (2.25) yields:

$$dS_h = \sum_{j=1}^n \text{sgn}[Im(Y_{h,j})] d[Im(Y_{h,j})] = \sum_{j=1}^n \text{sgn}[Im(Y_{h,j})] Im(dY_{h,j}) \quad (2.27)$$

and hence:

$$dS_h = \text{sgn}[Im(\mathbf{Y}_h^T)] Im(d\mathbf{Y}_h) \quad (2.28)$$

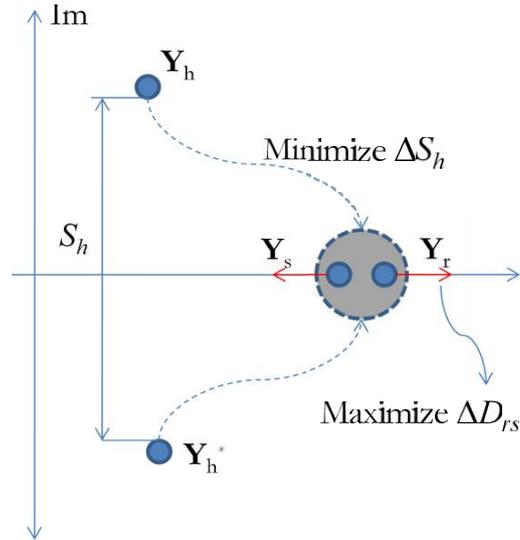


Figure 2.2: Schematic of Dietmier's algorithm

where the operators sgn and Im are applied to all elements of the corresponding vector or matrix.

Indeed, differentiating $\mathbf{F}(\mathbf{Y}, \mathbf{P})$ with respect to \mathbf{P} and \mathbf{Y} yields:

$$\mathbf{A}d\mathbf{P} + \mathbf{B}d\mathbf{Y} = \mathbf{0} \quad (2.29)$$

where $A_{kw} = \partial f_k / \partial P_w$, $B_{kj} = \partial f_k / \partial Y_j$, so that:

$$d\mathbf{Y} = -\mathbf{B}^{-1}\mathbf{A}d\mathbf{P} \quad (2.30)$$

If \mathbf{J} is the matrix obtained from $-\mathbf{B}^{-1}\mathbf{A}$, $d\mathbf{Y}$ immediately emerges from Eq. (2.31) as:

$$d\mathbf{Y} = \mathbf{J}d\mathbf{P} \quad (2.31)$$

and thus:

$$d\mathbf{Y}_h = \mathbf{J}_h d\mathbf{P} \quad (2.32)$$

where \mathbf{J}_h is matrix \mathbf{J} evaluated for $\mathbf{Y} = \mathbf{Y}_h$.

Substituting Eq. (2.32) into Eq. (2.28) finally yields:

$$dS_h = sgn [Im(\mathbf{Y}_h^T)] Im(\mathbf{J}_h d\mathbf{P}) = sgn [Im(\mathbf{Y}_h^T)] Im(\mathbf{J}_h) d\mathbf{P} \quad (2.33)$$

and thus, by replacing differential quantities with small finite variations:

$$\Delta S_h \approx sgn [Im(\mathbf{Y}_h^T)] Im(\mathbf{J}_h) \Delta \mathbf{P} \quad (2.34)$$

By Eq. (2.34), the task of bringing two complex conjugate solutions closer together is reduced to searching for a vector $\Delta\mathbf{P}$ that minimises the objective function, ΔS_h . This is a linear optimisation problem, which may be solved by any algorithm for linear programming, such as the simplex method (Luenberger and Ye [2008]).

Since the formulation given in Eq. (2.34) emerges from the linearisation of a complex nonlinear system of equations, the estimation that it provides is acceptable so long as $\Delta\mathbf{P}$ is sufficiently small; that is:

$$-\Delta\mathbf{P}_{max} \leq \Delta\mathbf{P} \leq \Delta\mathbf{P}_{max} \quad (2.35)$$

where, for the sake of simplicity, all components of $\Delta\mathbf{P}_{max}$ are taken to be equal to a given ΔP_{max} .

Furthermore, when the distance between any two real solutions associated with the current value of \mathbf{P} reaches a certain minimal value, D_{min} , the algorithm must ensure that they do not become any closer so as to prevent them eventually forming a complex conjugate pair. If \mathbf{Y}_r and \mathbf{Y}_s are any two such poses, differentiating Eq. (2.26) and further using Eq. (2.32) yields:

$$dD_{rs} = 2(\mathbf{Y}_r - \mathbf{Y}_s)^T (d\mathbf{Y}_r - d\mathbf{Y}_s) = 2(\mathbf{Y}_r - \mathbf{Y}_s)^T (\mathbf{J}_r - \mathbf{J}_s) d\mathbf{P} \quad (2.36)$$

Accordingly, the condition that \mathbf{Y}_r and \mathbf{Y}_s must not become any closer when assigning the increment $\Delta\mathbf{P}$ may be expressed by the linear constraint:

$$(\mathbf{Y}_r - \mathbf{Y}_s)^T (\mathbf{J}_r - \mathbf{J}_s) \Delta\mathbf{P} \geq 0 \quad (2.37)$$

By considering Eqs. (2.34), (2.35) and (2.37), the linear optimisation problem may be stated as:

$$\begin{aligned} \text{minimise : } & \quad \text{sgn} [Im(\mathbf{Y}_h^T)] Im(\mathbf{J}_h) \Delta\mathbf{P} \\ \text{subject to : } & \quad -\Delta\mathbf{P}_{max} \leq \Delta\mathbf{P} \leq \Delta\mathbf{P}_{max}; \\ & \quad (\mathbf{Y}_r - \mathbf{Y}_s)^T (\mathbf{J}_r - \mathbf{J}_s) \Delta\mathbf{P} \geq 0, \quad \forall (\mathbf{Y}_r, \mathbf{Y}_s) : D_{rs} < D_{min} \end{aligned} \quad (2.38)$$

If the problem in Eq. (2.38) provides a solution, the corresponding increment, $\Delta\mathbf{P}$, is added to \mathbf{P} and the set of geometric parameters is updated. The solutions of the modified $\mathbf{F}(\mathbf{Y}, \mathbf{P})$ are then computed by the help of an iterative Newton-Raphson routine, using $\mathbf{Y}_h + \mathbf{J}_h \Delta\mathbf{P}$, $h = 1 \dots N_{sol}$, as a starting estimate. At this point, the distance-reduction procedure formulated in Eq. (2.38) may be attempted again.

By the iterative application of the aforementioned minimisation procedure, S_h is progressively reduced until the imaginary parts of all elements of \mathbf{Y}_h become smaller than a certain numerical error, ε . In this situation, \mathbf{Y}_h and $\bar{\mathbf{Y}}_h$ may be, from a numerical point of view, either complex conjugate or real. A Newton-Raphson algorithm is then used to search in a sphere of radius ε , centred at $\Re(\mathbf{Y}_h)$, for two distinct real solutions, \mathbf{Y}_u and \mathbf{Y}_v , so that \mathbf{Y}_u and \mathbf{Y}_v may replace \mathbf{Y}_h and $\bar{\mathbf{Y}}_h$. In order to prevent \mathbf{Y}_u and \mathbf{Y}_v changing back to the complex field in the following steps of the algorithm, they are subsequently forced apart in the real domain by a certain distance (see Section 2.3.3). At this point, the algorithm is ready to attempt the transformation of another pair of complex conjugate roots into real ones. The pair

that is chosen is always that which has the shortest distance S_h .

2.3.3 Increasing the distance between a pair of real solutions

Let \mathbf{Y}_u and \mathbf{Y}_v be two real equilibrium poses. A variation, $\Delta\mathbf{P}$, of the parameter vector needs to be computed such that the distance between them, D_{uv} , is increased. If $\Delta\mathbf{P}$ is sufficiently small, the increment of D_{uv} may be expressed as in Eq. (2.36), so that maximising ΔD_{uv} may be reduced to a linear maximisation problem subject to the same constraints (Eqs. (2.35) and (2.37)) discussed in Section 2.3.2; that is,

$$\begin{aligned}
 \text{maximize : } & (\mathbf{Y}_u - \mathbf{Y}_v)^T (\mathbf{J}_u - \mathbf{J}_v) \Delta\mathbf{P} \\
 \text{subject to : } & -\Delta\mathbf{P}_{max} \leq \Delta\mathbf{P} \leq \Delta\mathbf{P}_{max}; \\
 & (\mathbf{Y}_r - \mathbf{Y}_s)^T (\mathbf{J}_r - \mathbf{J}_s) \Delta\mathbf{P} \geq 0, \quad \forall (\mathbf{Y}_r, \mathbf{Y}_s) : D_{rs} < D_{min}
 \end{aligned} \tag{2.39}$$

The procedure in Eq. (2.39) is repeated until either D_{uv} reaches a convenient value, D_r (e.g. $2D_{min}$), or no $\Delta\mathbf{P}$ is found that satisfies the imposed constraints.

Chapter 3

Geometric Static model of under-constrained cable-driven parallel robot

It is well-known in robotics that under conditions of static equilibrium, an external wrench, \mathbf{w} , applied to an end-effector relates to actuation forces, $\boldsymbol{\tau}$, as follows:

$$\mathbf{J}^T \boldsymbol{\tau} + \mathbf{w} = 0 \quad (3.1)$$

where \mathbf{J} is the jacobian matrix that transforms end-effector twists into actuator velocities. As such, for a platform in a given posture with given wrench, \mathbf{w} , the applied force on the cables, $\boldsymbol{\tau}$, may be obtained as the solution of Eq. (3.1). In general, for f -DOF end-effector and n tendons, \mathbf{J} is a matrix of size $f \times n$. Based on the dimensions of the jacobian matrix, \mathbf{J} , a classification of cable-driven parallel robots is defined. This was first proposed by [Ming and Higuchi \[1994\]](#) and discussed in further detail by [Kurtz and Hayward \[1991\]](#); [Roberts et al. \[1998\]](#). When the number of cables is more than the degrees of freedom of the moving platform; that is:

$$n > f \quad (3.2)$$

the forces in the cables are computed as:

$$\boldsymbol{\tau} = -\mathbf{J}^{T+} \mathbf{w} + \lambda \mathbf{h} \quad (3.3)$$

where \mathbf{J}^{T+} is the pseudo-inverse of matrix \mathbf{J}^T , \mathbf{h} is an arbitrary vector in the null space of \mathbf{J}^T and λ is a free parameter. One of the special characteristics of cables that makes the control of the end-effector challenging is that the cable cannot withstand compression. Therefore, in a cable-driven parallel robot with n cables and the moving platform at rest in equilibrium, the following condition must be satisfied:

$$\tau_i \geq 0, i = 1, 2, \dots, n \quad (3.4)$$

where τ_i is the applied force in each cable. Provided that all the components of \mathbf{h} in Eq. (3.6) are positive, λ may be set sufficiently large to ensure that all the applied forces in the cables are positive regardless of \mathbf{w} . Physically, the outcome of this is that the position and orientation of the robot are independent of the imposed load, \mathbf{w} , on the end-effector.

In the case of a robot with the number of cables, n , equal to the number of degrees of freedom, that is:

$$n = f \quad (3.5)$$

the forces in the cables are computed as:

$$\boldsymbol{\tau} = -\mathbf{J}^{-1}\mathbf{w} \quad (3.6)$$

When the robot falls into configurations with positive tension in all cables, the pose of the moving platform is independent of the imposed load and may be controlled by changing the cable lengths. These configurations in which it is impossible for the robot to be moved from its position and orientation without having to change the cable lengths are called fully constrained.

On the other hand, when the number of cables is less than the number of DOFs; that is:

$$n < f \quad (3.7)$$

then, in general, Eq. (3.1) has no solution unless it satisfies the consistency condition. From a mathematical point of view, Eq. (3.1) has solutions only if the external wrench \mathbf{w} falls into the column space of \mathbf{J}^T . Under this condition, from linear algebra, a solution of the forces in the cables, $\boldsymbol{\tau}$, may be computed for an imposed wrench \mathbf{w} . But even if all the forces of the cables be computed as positive, still some freedoms of the platform can not be controlled because of the condition in Eq. (3.7). Physically, this means that the posture of the moving platform depends on the imposed load, in addition to the cable lengths, since when the load changes the pose has to adapt in order to satisfy the consistency condition. In practice, due to the external wrench, the platform posture may change even if the cable lengths are kept constant. Such a system in which the posture of the moving platform depends on applied load is called under-constrained. Due to the increased complexity of displacement analysis for such a robot, where kinematics and statics must be dealt with simultaneously, an appropriate mathematical model for the robot is necessary to solve the problem efficiently. In the following section, modelling of such a robot and derivation of the governing equations are discussed.

3.1 Mathematical geometric static model of CDPRs

As shown in Fig. 3.1, a general under-constrained n - n CDPR comprises a mobile platform connected to a fixed base by $n < 6$ cables. The i th cable ($i = 1, 2, \dots, n$) is attached to the base at point A_i and to the platform at point B_i . It is assumed that the platform is acted upon by a 0-pitch wrench, $\mathbf{w} = Q\mathcal{L}_e$, applied at point G , where Q is the constant magnitude of \mathbf{w} and \mathcal{L}_e is the normalised Plücker vector

of its line of action. Such a wrench may, for example, result from the weight of the platform acting through its centre of mass. To formulate the problem, other relevant definitions are as follows:

- $Oxyz$ is a Cartesian coordinate frame fixed to the base with \mathbf{i} , \mathbf{j} and \mathbf{k} being unit vectors along the coordinate axes.
- $O'x'y'z'$ is a Cartesian coordinate frame fixed to the end-effector with \mathbf{i}' , \mathbf{j}' and \mathbf{k}' being the corresponding unit vectors along the coordinate axes.
- The platform posture is described by $\mathbf{X} = [\mathbf{x}; \Phi]$, where $\mathbf{x} = O' - O$ and Φ is the array grouping the variables that parameterise the platform orientation with respect to $Oxyz$.
- ρ_i is the cable length.
- $\mathbf{a}_i = A_i - O$, $\mathbf{r}_i = B_i - O'$, $\mathbf{s}_i = B_i - A_i$ and $\mathbf{u}_i = (A_i - B_i) / \rho_i = -\mathbf{s}_i / \rho_i$.
- \mathbf{b}_i is the projection of \mathbf{r}_i on $O'x'y'z'$.
- $\mathbf{R}(\Phi)$ is the rotation matrix between the mobile and fixed frame.
- $\mathbf{r}_i = \mathbf{R}(\Phi) \mathbf{b}_i$.
- \mathcal{L}_i / ρ_i is the normalised Plücker vector of the i th cable in axis coordinates.
- \mathbf{p}_i is any vector from an arbitrarily-chosen reference point P (called for brevity moment pole) to the cable line.
- $\mathcal{L}_i = -[\mathbf{s}_i; \mathbf{p}_i \times \mathbf{s}_i]$.
- τ_i is a scalar representing the cable tensile force.
- $(\tau_i / \rho_i) \mathcal{L}_i$ is the wrench exerted by the i th cable on the platform.

For the sake of brevity, the components of \mathbf{x} in $Oxyz$ are denoted as x , y and z . Vector components along the coordinate axes are denoted by righthand subscripts reporting the axes names.

For practical reasons, the following is finally assumed:

1. $\rho_i > 0$ and, thus, $\mathbf{s}_i \neq \mathbf{0}$, $i = 1 \dots n$ (Assumption 1);
2. $0 < \|B_j - B_i\| < \|(A_j - A_i)_{xy}\|$, $i \neq j$ (Assumption 2).

The latter assumption, according to which the segment $B_i B_j$ is strictly smaller than the projection of the segment $A_i A_j$ on the xy -plane, is not conceptually necessary, but it rules out some special configurations, which could be handled with no difficulty, but whose analysis would burden the presentation. In particular, the possibility that any two cables may be simultaneously parallel to \mathbf{k} is discarded.

It will now be shown that, according to the problem under consideration, the governing equations can be simplified by appropriately choosing the reference coordinates and the orientation of the axes.

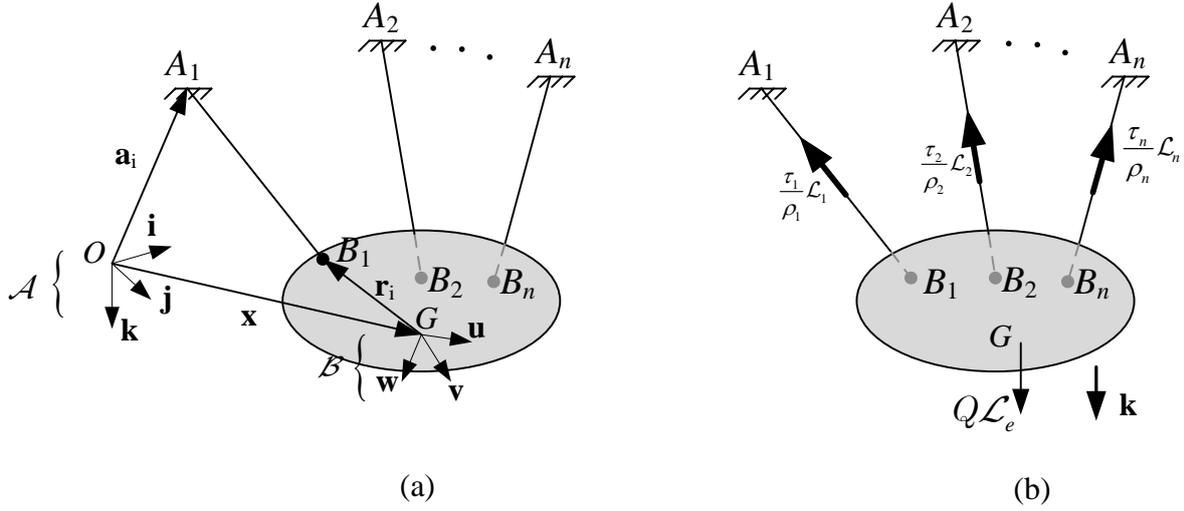


Figure 3.1: A CDP with n cables: (a) geometric model and (b) static model.

If all cables are active, the following n geometric constraints must be satisfied:

$$q_i := \mathbf{s}_i \cdot \mathbf{s}_i - \rho_i^2 = 0, \quad i = 1 \dots n \quad (3.8)$$

Since the platform has 6 DOFs, its posture is ultimately determined by mechanical equilibrium. Static equilibrium may be expressed as:

$$\sum_{i=1}^n \frac{\tau_i}{\rho_i} \mathcal{L}_i + Q \mathcal{L}_e = \underbrace{\begin{bmatrix} \mathcal{L}_1 & \mathcal{L}_2 & \dots & \mathcal{L}_n & \mathcal{L}_e \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} (\tau_1/\rho_1) \\ (\tau_2/\rho_2) \\ \vdots \\ (\tau_n/\rho_n) \\ Q \end{bmatrix} = \mathbf{0} \quad (3.9)$$

Eqs. (3.8) and (3.9) amount to a system of $n + 6$ scalar relations in $2n + 6$ variables; namely, the variables of platform pose, grouped in the array $\mathbf{X} = [\mathbf{x}; \Phi]$, the length of cable $\rho_i, i = 1 \dots n$, and the cable tensions $\tau_i, i = 1 \dots n$. A finite set of system configurations may generally be determined if any n of these variables are known. Depending on the assigned n variables, two types of problem may be considered:

- Inverse Geometric Static Problem (IGP): When n variables concerning the platform posture are assigned.
- Direct Geometric Static Problem (DGP): When n cable lengths are assigned.

To solve the problem, the strategy presented by Carricato and Merlet [2010, 2013] is followed.

Both problems may be simplified by eliminating cable tensions from Eq. (3.9). A convenient elimination strategy emerges by observing that Eq. (3.9) holds only if:

$$\text{rank}(\mathbf{M}) \leq n \tag{3.10}$$

namely if $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n$ and \mathcal{L}_e are linearly dependent. This is a purely geometric condition, since \mathbf{M} is a $6 \times (n + 1)$ matrix depending only on the platform posture. Furthermore, such relations do *not* comprise cable lengths and lead to a partial decoupling of the system equations, with cable lengths appearing only in Eq. (3.8). The IGP takes particular advantage of the aforementioned decoupling, as the platform configuration may be directly computed by relations emerging from Eq. (3.10). By setting all $(n + 1) \times (n + 1)$ minors of \mathbf{M} equal to zero and by conveniently changing the moment pole, a large set of linearly independent relations exclusively comprising the platform pose variables may be derived; that is:

$$p_k(\mathbf{X}) = 0, \quad k = 1 \dots N_p \tag{3.11}$$

where N_p is an integer, usually much greater than the number of variables, N_X , contained in \mathbf{X} . Indeed, for Eq. (3.10) to hold, *all* minors of \mathbf{M} must vanish and this ordinarily results in more equations than the number of variables, N_X . The solution of the IGP problem coincides with the variety, V , of the ideal generated by such equations. When n configuration variables are known (typically, either \mathbf{x} or Φ), any set of n equations from Eq. (3.11) may be chosen and a corresponding (generally zero-dimensional) variety, V , is obtained. Once \mathbf{X} is known, cable lengths may be directly calculated by Eq. (3.8). It is worth emphasising that the IGP only provides *potential* robot configurations, since, when the cable lengths emerging from Eq. (3.8) are fed into the actuators of the actual robot, the end-effector may evolve, *a priori*, in any one of the feasible solutions emerging from the corresponding DGP and nothing guarantees that the desired configuration is actually attained. This will be discussed in more detail later on. The DGP poses remarkably more complex mathematical problems, as in this case, the platform configuration is entirely unknown and it must be determined by simultaneously satisfying both the relations emerging from Eq. (3.11) and those emerging from Eq. (3.8).

The cost of eliminating some of the unknowns is that the relations in Eq. (3.11) are significantly more involved than those in Eq. (3.9). In particular, when they are of polynomial form, as is normally the case, they have a higher degree, more numerous terms and more complicated coefficients. Formulating equilibrium constraints via Eq. (3.11) is particularly favourable if, in order to solve the DGP, variable-elimination strategies are pursued that take advantage of the abundance of linearly independent relations that this approach provides, such as methods based on Groebner bases or Sylvester's dialytic elimination (Carricato [2013a,b]; Carricato and Merlet [2011a,b]). These methods, however, rely on exact-arithmetic calculations and are particularly expensive in terms of computational burden. When working with floating-point computation (such as homotopy continuation or interval analysis), the relations in Eq. (3.11) have two drawbacks. Firstly, they are very sensitive to the accuracy with which the coefficients are calculated, such that solutions are appreciably affected by numerical errors.

Furthermore, in order to compute a finite set of solutions, a subset of $N_X - n$ relations must be selected within Eq. (3.11), so as to form, together with Eq. (3.8), a square system of N_X equations in N_X unknowns. In this way, only a few minors of \mathbf{M} are used and spurious roots are likely to be introduced; that is, solutions for which some of the unused minors do not vanish. Accordingly, all roots must be checked against the original set of equations, thus further increasing the computational burden. Due to these limits, the formulation of static equilibrium via Eq. (3.9) will be preferred for the successful implementation of numerical continuation methods.

Since using different parameterisations to describe the platform configuration causes Eqs. (3.8), (3.9) and (3.11) to yield polynomial relations with varied complexity and degree, alternative parameterisations will now be discussed.

3.2 Rodrigues' parameterisation

According to Rodrigues' parameterisation, the platform posture, \mathbf{X} , may be described by 6 variables; namely, the vector \mathbf{x} identifying the position of a point on the platform (e.g. O') and the array Φ grouping the three Rodrigues parameters defining the platform orientation with respect to the fixed frame. Rodrigues parameters are derived using the theorem of Cayley (Bottema and Roth [1990]). This theorem states that it is possible to represent an orthogonal matrix, \mathbf{R} , using a skew symmetric matrix, \mathbf{S} :

$$\mathbf{R} = (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S}) \quad (3.12)$$

where \mathbf{I} is the 3×3 unit matrix and

$$\mathbf{S} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \quad (3.13)$$

If the eigenvalues of \mathbf{R} do not have values of -1 (meaning that \mathbf{R} does not describe a rotation of angle π), one may compute the matrix \mathbf{S} via $\mathbf{S} = (\mathbf{R} - \mathbf{I})(\mathbf{R} + \mathbf{I})^{-1}$. From the entries e_i of \mathbf{S} , one can directly compute the direction of the rotation axis $\Phi = (e_1, e_2, e_3)$ and the rotation angle $\tan(\frac{\phi}{2}) = \sqrt{e_1^2 + e_2^2 + e_3^2}$. The parameters $e_i, i = 1, \dots, 3$ are called Rodrigues parameters. The case in which \mathbf{R} describes a rotation of angle π must be handled in a different way, as shown by Angeles [2002].

Accordingly, $\mathbf{X}^T = [\mathbf{x}^T; \Phi^T]$. Substitution of these variables into Eqs. (3.8) and (3.9) provides $n + 6$ scalar relations in $2n + 6$ variables; that is, \mathbf{x} , Φ and τ . This parameterisation is especially suitable when the elimination strategy is implemented, as it only introduces the minimum number of variables. Accordingly, the number of elimination steps and the required time are reduced.

It is worth mentioning a drawback of this parameterization when the Dietmaier algorithm, discussed in Section 2.3, is applied. This parameterisation involves variables with mixed units (the components of \mathbf{x} are lengths, whereas those of Φ are dimensionless), making it difficult to conceive a physically-

consistent unit to ‘measure’ the distance between two poses that have been dealt with in the Dietmaier algorithm.

3.3 Study’s parameterisation

Study’s soma coordinates (Bottema and Roth [1990]) provide an 8-parameter homogeneous representation of the pose. Homogenisation of Rodrigues’ parameterisation by performing the substitutions $e_i = \frac{c_i}{c_0}$ for $i = 1, \dots, 3$ and recomputing the rotation matrix \mathbf{R} gives:

$$\mathbf{R} = \frac{1}{\Delta} \begin{bmatrix} c_0^2 + c_1^2 - c_2^2 - c_3^2 & 2(c_1c_2 - c_0c_3) & 2(c_1c_3 + c_0c_2) \\ 2(c_1c_2 + c_0c_3) & c_0^2 - c_1^2 + c_2^2 - c_3^2 & 2(c_2c_3 - c_0c_1) \\ 2(c_1c_3 + c_0c_2) & 2(c_2c_3 + c_0c_1) & c_0^2 - c_1^2 - c_2^2 + c_3^2 \end{bmatrix} \quad (3.14)$$

where $\Delta = c_0^2 + c_1^2 + c_2^2 + c_3^2$. Without loss of generality, one may assume normalisation, meaning that $\Delta = 1$. Considering $g_k, k = 0 \dots 3$, as the components of a quaternion such that:

$$e_0g_0 + e_1g_1 + e_2g_2 + e_3g_3 = 0 \quad (3.15)$$

the platform position may be stated as:

$$\mathbf{x} = \frac{1}{\Delta} \begin{bmatrix} -e_0g_1 + e_1g_0 - e_2g_3 + e_3g_2 \\ -e_0g_2 + e_1g_3 + e_2g_0 - e_3g_1 \\ -e_0g_3 - e_1g_2 + e_2g_1 + e_3g_0 \end{bmatrix} \quad (3.16)$$

Study coordinates add 2 unknowns and 2 equations to Eqs. (3.8) and (3.9), thus yielding a system of $8 + n$ polynomial equations for $8 + 2n$ unknowns; that is, $\mathbf{Y} = [e_0, e_1, e_2, e_3, g_0, g_1, g_2, g_3, \tau_1, \dots, \tau_n, \rho_1, \dots, \rho_n]^T$. As explained in the following sections, in some cases the equations derived by this parameterisation are more stable when homotopy continuation is implemented.

3.4 Dietmaier’s parameterization

Following Dietmaier [1998], the platform posture, \mathbf{X} , may be described by 9 variables all representing vector components with units of length. This is accomplished by describing the platform orientation by way of the unit vectors \mathbf{e}_1 and \mathbf{e}_2 and $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$.

Expressing the components of \mathbf{e}_1 and \mathbf{e}_2 in the fixed frame:

$$\mathbf{e}_1 = [e_{11}, e_{12}, e_{13}]^T, \quad \mathbf{e}_2 = [e_{21}, e_{22}, e_{23}]^T \quad (3.17)$$

the rotation matrix \mathbf{R} can be expressed as $\mathbf{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$. By having $\mathbf{x} = [x, y, z]^T$, the position vectors of points B_1, B_2, B_3 and G in $Oxyz$ may be written in a straightforward manner as functions of the

array:

$$\mathbf{X} = [x, y, z, e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}]^T \quad (3.18)$$

whose components satisfy the conditions:

$$\begin{aligned} e_{11}^2 + e_{12}^2 + e_{13}^2 &= 1, \\ e_{21}^2 + e_{22}^2 + e_{23}^2 &= 1, \\ e_{11}e_{21} + e_{12}e_{22} + e_{13}e_{23} &= 0 \end{aligned} \quad (3.19)$$

When the geometric parameters of a robot are known, Eqs. (3.8), (3.9) and (3.19) form a system of $n + 9$ scalar relations with $2n + 9$ variables, namely $\mathbf{x} = [x, y, z]$, $\rho_i, \tau_i, i = 1 \dots n$, and

$$\mathbf{Y} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}]^T \quad (3.20)$$

Dietmaier's parameterisation introduces a higher number of variables, making it unsuitable for the elimination procedures. On the other hand, when dealing with continuation methods, this type of parameterisation is preferred over the Rodrigues variation due to the fact that it yields simpler lower-order polynomial equations, which prove to be stabler. Furthermore, as explained in Section 2.3, when Dietmaier's algorithm is implemented, parameterisation of the platform pose is homogeneous in terms of units of measurement, which is an advantage for conceiving a physically consistent way of 'measuring' the distance between two poses.

Chapter 4

Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

This chapter presents analysis results for the geometric static problem of under-constrained cable driven parallel robots. By implementing the tools discussed in the previous chapter, inverse and direct geometric static problems for robots with 2, 3 . . . , 5 cables are solved. Within the problem-solving algorithm, the following issues, which are classic challenges in robot analysis, are specifically dealt with:

1. determination of the number of solutions in the (zero-dimensional) algebraic variety defined by the polynomial equations of the problem;
2. reduction of the problem to a single equation with one unknown;
3. numerical computation of the solution set;
4. identification of a specific geometry providing the maximum number of distinct real-valued solutions.

By implementing appropriate elimination strategies for the respective IGPs and DGPs, the first two above mentioned issues have been resolved for robots with $n \leq 5$ cables. The strategies are explained in detail in Section 4.1. It will be shown that when a DGP is dealt with, the elimination strategies yield univariate equations of very high order. Due to the complexity of these equations, numerical procedures are introduced in Section 4.2 to compute the solutions of the DGP efficiently. The last issue mentioned above is especially challenging when dealing with DGPs. Due to the very high order of the univariate polynomials and the complexity of the corresponding equations, it is difficult to identify a robot geometry with the maximum number of real solutions. Accordingly, as discussed in Section 4.3, the Dietmaier algorithm is applied to overcome this problem.

4.1 The elimination approach

Before implementing an elimination procedure, the governing equations are simplified as much as possible by adopting appropriate coordinate systems and suitable parameterisation. With O chosen to coincide with A_1 (i.e. $\mathbf{a}_1 = \mathbf{0}$), \mathbf{k} directed along \mathcal{L}_e , A_2 lying in plane $A_1\mathbf{i}\mathbf{j}$, B_1 lying on axis $O'\mathbf{i}'$ and B_2 lying in plane $O'\mathbf{i}'\mathbf{j}'$, the position vectors of points $A_1, A_2, A_3, \dots, A_n$ in frame $Oxyz$ and of points $B_1, B_2, B_3, \dots, B_n$ in frame $O'x'y'z'$ may be expressed as:

$$\begin{aligned} [A_1]_O &= [0, 0, 0]^T, [A_2]_O = [a_{21}, 0, a_{23}]^T, [A_3]_O = [a_{31}, a_{32}, a_{33}]^T, \dots, [A_n]_O = [a_{n1}, a_{n2}, a_{n3}]^T \\ [B_1]_{O'} &= [b_{11}, 0, 0]^T, [B_2]_{O'} = [b_{21}, b_{22}, 0]^T, \dots, [B_n]_{O'} = [b_{n1}, b_{n2}, b_{n3}]^T \\ [G]_O &= [g_1, g_2, g_3] \end{aligned} \quad (4.1)$$

with a_{ij} , b_{ij} and g_j being known geometric parameters. Using this simplification for each case, the corresponding governing equations are derived. An appropriate elimination strategy is then adopted to eliminate all variables but one from the equations until a polynomial is obtained with a single unknown. If this polynomial has the least possible degree, it provides the exact number of solutions in the complex field. For an elimination procedure to be successful, a formulation containing the least number of unknowns is the most suitable. The formulation in Eqs. 3.11 is thus used, in which cable tension variables are eliminated.

It will be seen in Sections 4.1.1, 4.1.2 and 4.1.3, that the IGP is, in general, less complex compared to the DGP and, by use of relatively simple dialytic elimination strategies, all solutions can be computed.

The DGP will be discussed in detail in Section 4.1.4, since the problem demands more effort. Most of the material regarding cases of 2-2 and 3-3 CDPRs are from Carricato [2013a,b]; Carricato and Merlet [2011a,b, 2013], respectively, and for consistency will be presented here briefly.

4.1.1 Inverse geometric static problem of 2-2 CDPRs

Both direct and inverse geometric static analyses of 2-cable robots are presented in detail by Carricato and Merlet [2013]. The elimination procedure becomes quite simple by remodelling the equations in-plane as shown in Fig. 4.1. Without loss of generality, the coordinate plane xz , parallel to \mathbf{k} , may be allowed to pass through A_1 and A_2 ; whereas $x'z'$ may be chosen so that it contains B_1, B_2 and G . If y and y' point in the same direction, the robot is said to work in operation mode I (Fig. 4.1) and the rotation matrix between $Oxyz$ and $Gx'y'z'$ is:

$$\mathbf{R}_I = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \quad (4.2)$$

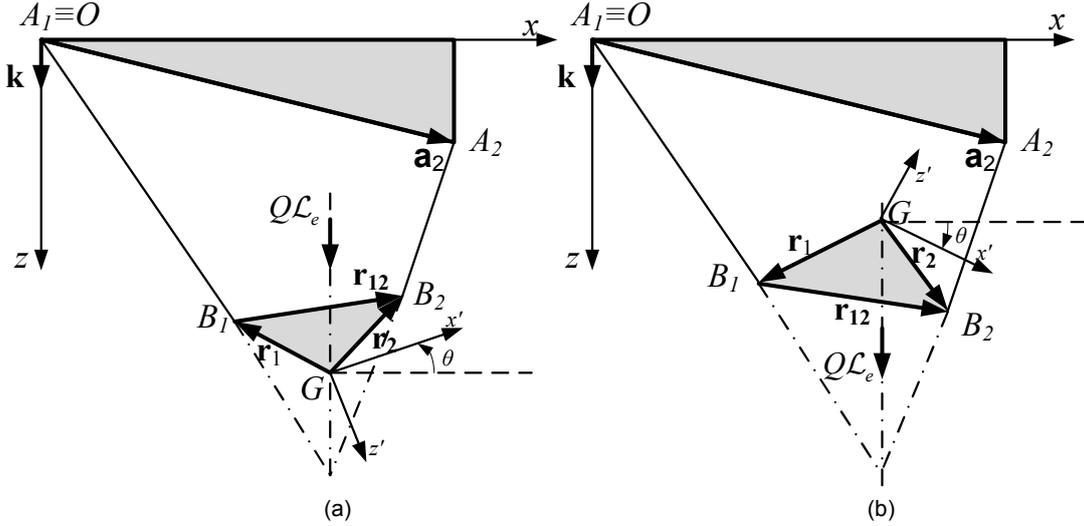


Figure 4.1: Geometric model of a cable-driven parallel robot with 2 cables: (a) operation mode I; (b) operation mode II (Carricato and Merlet [2013])

Whereas if y and y' point in opposite directions, the robot is said to work in operation mode II (Fig. 4.1) and the rotation matrix is:

$$\mathbf{R}_I = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & -1 & 0 \\ s_\theta & 0 & -c_\theta \end{bmatrix} \quad (4.3)$$

where θ is the angle formed by x' and x with a positive rotation about y' and s_θ and c_θ stand for $\sin(\theta)$ and $\cos(\theta)$, respectively. If \mathbf{b}'_i is the coordinate vector of B_i in the $Gx'y'z'$ frame, then $\mathbf{r}_i = \mathbf{R}_I(\theta)\mathbf{b}'_i$ or $\mathbf{r}_i = \mathbf{R}_{II}(\theta)\mathbf{b}'_i$. By noting that in equilibrium conditions the two planes xz and $x'z'$ necessarily coincide and by considering $O = A_1$ as the reduction pole of the moments, the matrix \mathbf{M} in Eq. (3.9) may be reduced to:

$$\mathbf{M}' = \begin{bmatrix} -(\mathbf{x} + \mathbf{r}_1) & \mathbf{a}_2 - (\mathbf{x} + \mathbf{r}_2) & \mathbf{k} \\ 0 & -\mathbf{a}_2 \times (\mathbf{x} + \mathbf{r}_2) & \mathbf{x} \times \mathbf{k} \end{bmatrix} = \begin{bmatrix} x + r_{1x} & r_{21x} - a_{2x} & 0 \\ z + r_{1z} & r_{21z} - a_{2z} & 1 \\ 0 & \mathbf{a}_2 \times (\mathbf{x} + \mathbf{r}_2) \cdot \mathbf{j} & -x \end{bmatrix} \quad (4.4)$$

Accordingly, Eq. (3.10) holds if and only if $\det \mathbf{M}' = 0$; namely:

$$p_1 := (r_{21x}x + a_{2x}r_{1x})z - r_{21z}x^2 + (r_{1z}r_{2x} - r_{1x}r_{2z} + a_{2x}r_{21z} - a_{2z}r_{2x})x + r_{1x}(a_{2x}r_{2z} - a_{2z}r_{2x}) = 0 \quad (4.5)$$

The two constraint equations emerging from Eq. (3.8) may be linearly combined in the form:

$$\begin{aligned} p_2 &:= x^2 + z^2 + 2(r_{1x}x + r_{1z}z) + \|\mathbf{r}_1\|^2 - \rho_1^2 = 0 \\ p_3 &:= 2(a_{2x} - r_{21x})x + 2(a_{2z} - r_{21z})z + \|\mathbf{r}_1\|^2 - \|\mathbf{a}_2 - \mathbf{r}_2\|^2 + \rho_2^2 - \rho_1^2 = 0 \end{aligned} \quad (4.6)$$

comprising the variables x, z, θ, ρ_1 and ρ_2 .

The inverse problem of a 2-2 CDPR may be solved by considering only Eq. (4.5). Following **Caricato and Merlet [2013]**, 4 different cases may be distinguished depending on which pose parameters are assigned:

1. The orientation θ and variable x are assigned:

In this case, Eq. (4.5) provides a single solution in z , except for the case in which $r_{21x}x = -a_{2x}r_{1x}$. If the latter condition occurs, the problem may only be solved if the entire polynomial vanishes, in which case the solution set is one-dimensional and coincides with a line parallel to \mathbf{k} . This occurs if either one of the following conditions holds:

- $r_{ix} = 0$ and $x = a_{ix}$, with $i = 1 \text{ or } 2$;
- $\mathbf{a}_2 \times \mathbf{r}_{21} \cdot \mathbf{j} = 0$, $r_{21x} \neq 0$ and $x = -a_{2x}r_{1x}/r_{21x}$.

In the former case, A_i, B_i and G lie on a line parallel to \mathbf{k} and the robot operates as a 1-DOF crane with the i th cable holding the entire load. In the latter case, the line segments A_1A_2 and B_1B_2 are parallel and the platform may follow a quasi-static linear path parallel to \mathbf{k} , with orientation being constant and the load being sustained by both cables.

2. The orientation θ and variable z are assigned:

In this case, Eq. (4.5) provides, in general, two solutions for x . If all coefficients of p_1 vanish, the solution set is one-dimensional and coincides with a line perpendicular to \mathbf{k} . p_1 is identically nought if $r_{1z} = r_{2z}$, $z + r_{1z} = a_{jz}$ and $r_{ix} = 0$, with $i \neq j$; namely, if points B_1, B_2 and A_j lie on a line perpendicular to \mathbf{k} .

3. The orientation θ and an approximate location of G are assigned:

If the orientation and an approximate desired location, (x_d, z_d) , of G are assigned, x and z must be found such that Eq. (4.5) is satisfied and the error, $\varepsilon = (x - x_d)^2 + (z - z_d)^2$, is minimised. Since both ε and p_1 are continuously differentiable in x and z , the global minimum of ε is a stationary point of the function $L_\varepsilon = (x - x_d)^2 + (z - z_d)^2 + \lambda p_1(x, z)$. Setting the derivatives of L_ε with respect to x and z to zero provides a linear system in x and z , by which both variables may be determined as functions of λ . Upon substituting $x = x(\lambda)$ and $z = z(\lambda)$ into p_1 , a fourth-degree polynomial in λ is obtained. Its real roots are the stationary points of L_ε , among which the global minimum may be determined by direct evaluation of ε . Clearly, this optimal configuration is feasible only if the conditions concerning the sign of cable tensions and stability are satisfied.

4. The position (x, y) is assigned:

Letting $\mathbf{r}_i = \mathbf{R}_I(\theta)\mathbf{b}_i$ or $\mathbf{r}_i = \mathbf{R}_{II}(\theta)\mathbf{b}_i$, with $i = 1$ and 2 , p_1 becomes a quadratic polynomial in s_θ and c_θ . For each operational mode, the resultant of p_1 and the trigonometric identity with respect to c_θ yields a fourth-degree equation in s_θ . For each root in s_θ , Eq. (4.5) provides a single value of c_θ and thus of θ . The problem admits altogether 8 solutions, all of which may be real.

4.1.2 Inverse geometric static problem of 3-3 CDPRs

For the case of robot with 3 cables, Eqs. (3.8) and (3.9) amount to 9 scalar relations in 12 variables. Two cases of the IGP are studied, depending on which set of 3 platform posture variables are assigned:

1. IGP with assigned orientation in which Φ and, correspondingly, all vectors \mathbf{r}_i , $i = 1 \dots 3$, are known.
2. IGP with assigned position in which \mathbf{x} is given and \mathbf{r}_1 , \mathbf{r}_2 and \mathbf{r}_3 are unknown

4.1.2.1 Inverse geometric static problem of 3-3 CDPR with assigned orientation

When Φ is assigned, all vectors \mathbf{r}_i , $i = 1 \dots 3$, are known. If O is chosen as the reduction pole of moments, \mathcal{L}_i and \mathcal{L}_e may be expressed, in axis coordinates, as $-\mathbf{s}_i; \mathbf{a}_i \times \mathbf{s}_i$ and $[\mathbf{k}; \mathbf{x} \times \mathbf{k}]$, respectively. Accordingly, \mathbf{M} in Eq. (3.9) becomes:

$$\mathbf{M}(O) = \begin{bmatrix} -\mathbf{s}_1 & -\mathbf{s}_2 & -\mathbf{s}_3 & \mathbf{k} \\ \mathbf{0} & -\mathbf{a}_2 \times \mathbf{s}_2 & -\mathbf{a}_3 \times \mathbf{s}_3 & \mathbf{x} \times \mathbf{k} \end{bmatrix} \quad (4.7)$$

or, equivalently, by setting $\mathbf{s}_i = \mathbf{x} + \mathbf{r}_i - \mathbf{a}_i$ and by performing elementary column transformations:

$$\mathbf{M}'(O) = \begin{bmatrix} \mathbf{x} + \mathbf{r}_1 & \mathbf{r}_{21} - \mathbf{a}_2 & \mathbf{r}_{31} - \mathbf{a}_3 & \mathbf{k} \\ \mathbf{0} & \mathbf{a}_2 \times (\mathbf{x} + \mathbf{r}_2) & \mathbf{a}_3 \times (\mathbf{x} + \mathbf{r}_3) & \mathbf{x} \times \mathbf{k} \end{bmatrix} \quad (4.8)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $i \neq j$.

It is important to observe that letting $B_i \equiv A_i$, with $i = 1 \dots 3$, causes the i th column of \mathbf{M} to vanish (since $\mathbf{s}_i = \mathbf{a}_i \times \mathbf{s}_i = \mathbf{0}$) and, thus, it causes all 4×4 minors of \mathbf{M} (and of \mathbf{M}') to be zero. It follows that a configuration for which $B_i \equiv A_i$ is a formal solution of the problem, as long as it is compatible with the assigned constraints. We call it a *trivial* solution, and we need to discard it, since ρ is required to be strictly positive:

$$\bar{\mathbf{x}}_i := [\bar{x}_i, \bar{y}_i, \bar{z}_i]^T = \mathbf{a}_i - \mathbf{r}_i, \quad i = 1 \dots 3. \quad (4.9)$$

This observation is particularly important for the IGP with assigned orientation. In this case, in fact, it is always possible to displace the platform (with a given orientation) so as to superimpose B_i onto A_i . Consequently, all trivial solutions appear as solutions of the problem and need to be discarded.

According to Eq. (3.11), the rank of the block matrix $\mathbf{M}'_{123}(O)$ must be equal to either 2 or 3. Discarding the trivial solutions and letting

$$\Delta := \det \mathbf{M}'_{123,234}(O) = (\mathbf{r}_{21} - \mathbf{a}_2)_x (\mathbf{r}_{31} - \mathbf{a}_3)_y - (\mathbf{r}_{21} - \mathbf{a}_2)_y (\mathbf{r}_{31} - \mathbf{a}_3)_x \quad (4.10)$$

two cases may be distinguished depending on Δ being zero or not. It should be noted that Δ depends only on the input parameters when Φ is assigned. For physical interpretation of each case, the reader is referred to Carricato [2013a]; Carricato and Merlet [2011a].

1. $\Delta \neq 0$:

Equations

$$p_1 := \det \mathbf{M}'_{1236}(O) = A_{20}x^2 - (B_{011} + C_{101})xy + A_{02}y^2 + A_{10}x + A_{01}y + A_{00} = 0 \quad (4.11a)$$

$$p_2 := \det \mathbf{M}'_{1235}(O) = B_{110}xy - A_{20}xz + B_{020}y^2 + B_{011}yz + B_{100}x + B_{010}y + B_{001}z + B_{000} = 0 \quad (4.11b)$$

$$p_3 := \det \mathbf{M}'_{1234}(O) = B_{110}x^2 + B_{020}xy - C_{101}xz + A_{02}yz - C_{100}x - C_{010}y - C_{001}z - C_{000} = 0 \quad (4.11c)$$

comprise the lowest-degree polynomials among all minors of $\mathbf{M}'(O)$, with coefficients A_{ij} , B_{ijk} and C_{ijk} being functions of the geometric and orientation parameters only. Since $\text{rank} \mathbf{M}'_{123}(O) = 3$, satisfying Eqs. (4.11a)-(4.11c) assures that $\text{rank} \mathbf{M}'(O) = \text{rank} \mathbf{M}'_{123}(O)$ and thus the variety, V , generated by Eq. (3.11) is equivalent to the variety, V_{123} , generated by the above three equations.

p_1 is quadratic in x and y and does not contain the variable z . p_2 and p_3 are quadratic in x , y and z but do not contain the monomial z^2 . Eliminating z from p_2 and p_3 gives a cubic polynomial, p_{23} , in x and y ; whereas eliminating y from p_1 and p_{23} yields a 4th degree equation in x ; namely:

$$p_{123} := \sum_{h=0}^4 E_h x^h = 0 \quad (4.12)$$

It emerges from Eqs. (4.11a)-(4.11c) that the coefficients of the leading monomials y^2 and y^3 in p_1 and p_{23} are, respectively, A_{02} and $A_{02}B_{020}$ with, in particular, $A_{02} = a_{3x}r_{21x} - a_{2x}r_{31x}$. Since A_{02} appears in both coefficients, it factors all terms of the resultant of p_1 and p_{23} with respect to y . p_{123} is the polynomial obtained by eliminating A_{02} from such a resultant. This simplification is obvious when $A_{02} \neq 0$, but it is still valid when $A_{02} = 0$, with a caveat. In fact, if the resultant of p_1 and p_{23} is calculated after setting $A_{02} = 0$, a 4th-degree polynomial in x is again obtained and it proves to be equal to p_{123} times the constant:

$$\Gamma_0 := B_{020}(C_{001} + A_{01}) - C_{010}B_{011} \quad (4.13)$$

Hence, as long as $\Gamma_0 \neq 0$, p_{123} is still a legitimate elimination ideal for p_1 and p_{23} . The case $\Gamma_0 = 0$ is discussed by Carricato [2013a]. Since three roots of p_{123} necessarily correspond to

trivial solutions, the fourth root, \bar{x}_4 , must be real and may be computed by Vieta's formula:

$$\bar{x}_4 = \frac{E_0}{\bar{x}_1 \bar{x}_2 \bar{x}_3 E_4} = \frac{\hat{E}_0}{\hat{E}_4} \quad (4.14)$$

where the latter expression takes advantage of the fact that it is possible to factorise E_0 and E_4 as $\bar{x}_1 \bar{x}_2 \bar{x}_3 \Delta^2 \hat{E}_0$ and $\Delta^2 \hat{E}_4$, respectively. Once \bar{x}_4 is known, \bar{y}_4 may be calculated, for instance as the greatest common divisor of p_1 and p_{23} , and \bar{z}_4 may be obtained by either Eq. (4.11b) or Eq. (4.11c). As long as $E_4 \neq 0$, the problem thus admits a single real solution: $\bar{\mathbf{x}}_4 = (\bar{x}_4, \bar{y}_4, \bar{z}_4)$. Special combinations of the geometric parameters and input orientations for which Eqs. (4.11a)-(4.11c) admit infinite solutions are presented by Carricato [2013a]. In these cases, G may follow quasi-static paths along assigned curves in space, with the platform orientation remaining constant.

2. $\Delta = 0$:

By expanding the coefficients of p_{123} , it is possible to verify that all of them comprise the factor Δ^2 . Consequently, when $\Delta = 0$, p_{123} degenerates and the procedure described in the previous section is not adequate to solve the problem.

Since it is assumed that $(\mathbf{r}_{21} - \mathbf{a}_2)_{xy}$ and $(\mathbf{r}_{31} - \mathbf{a}_3)_{xy}$ do not vanish, when $\Delta = 0$ these two vectors must be parallel (Eq. (4.10)):

$$(\mathbf{r}_{31} - \mathbf{a}_3)_{xy} = \alpha (\mathbf{r}_{21} - \mathbf{a}_2)_{xy} \quad (4.15)$$

where $\alpha \in \mathbb{R} - \{0\}$.

By enforcing both Eq. (4.15) and $\Delta = 0$, the polynomial relations in Eq. (4.11) may be factored as:

$$p_1 := f_{10}g = 0 \quad (4.16a)$$

$$p_2 := f_{20}g = 0 \quad (4.16b)$$

$$p_3 := f_{30}g = 0 \quad (4.16c)$$

where f_{10} and g are linear polynomials in x and y , and f_{20} and f_{30} are linear polynomials in x , y and z . In particular:

$$g := \det \mathbf{M}'_{123, 124}(O) = (\bar{y}_1 - \bar{y}_2)x - (\bar{x}_1 - \bar{x}_2)y + \bar{x}_1 \bar{y}_2 - \bar{y}_1 \bar{x}_2 \quad (4.17)$$

where the coefficients multiplying x and y cannot vanish simultaneously, since this would infer $(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)_{xy} = (\mathbf{r}_{21} - \mathbf{a}_2)_{xy} = \mathbf{0}$.

Equation (4.16) holds if either $f_{10} = f_{20} = f_{30} = 0$ or $g = 0$. Since the former requirement amounts to a linear non-homogeneous system in x , y and z whose coefficient matrix is singular (and, hence, admits in general no solution), only the latter condition applies. Together with the

requisite $\Delta = 0$, this brings about that $\text{rank} \mathbf{M}'_{123}(O) = 2$, with the first row of $\mathbf{M}'(O)$ linearly dependent on the second and third rows. The problem may thus be solved by considering the system comprising the relation:

$$p_4 := g = 0 \quad (4.18)$$

and any two of the equations:

$$p_5 := \det \mathbf{M}'_{2345}(O) = 0 \quad (4.19a)$$

$$p_6 := \det \mathbf{M}'_{2346}(O) = 0 \quad (4.19b)$$

$$p_7 := \det \mathbf{M}'_{2356}(O) = 0 \quad (4.19c)$$

where p_5 , p_6 and p_7 are cubic polynomials in x , y and z , with p_6 and p_7 being linear in z and p_5 being quadratic in z .

If, for instance, p_6 and p_7 are considered, eliminating z gives a 4th degree polynomial, p_{67} , in x and y . Further eliminating y from p_4 and p_{67} yields a quartic equation in x , $p_{467} = 0$, which admits a single non-trivial solution. This solution formally coincides with the quotient \hat{E}_0/\hat{E}_4 at the right-hand side of Eq. (4.14), once \hat{E}_0 and \hat{E}_4 are expressed as functions of the geometric and orientation parameters. If $\mathbf{M}'_{236}(O)$ has full rank, then $\text{rank} \mathbf{M}'(O) = \text{rank} \mathbf{M}'_{236}(O) = 3$ and $V = V_{467}$. In this case, $\Delta = 0$ and the axes of \mathcal{L}_1 , \mathcal{L}_2 , \mathcal{L}_3 and \mathcal{L}_e form a regulus on a hyperbolic paraboloid.

It may happen that, in special cases, p_{467} degenerates and the procedure described above is not adequate to solve the problem. Typically, this may occur when $\text{rank} \mathbf{M}'_{236} = 2$. A different choice of minors normally allows one to conclude the analysis.

4.1.2.2 Inverse geometric static problem of 3-3 CDPR with assigned position

In the case where \mathbf{x} is assigned and \mathbf{r}_1 , \mathbf{r}_2 and \mathbf{r}_3 are unknown, the relations in Eq. (4.11) assume a particularly favourable structure if the platform orientation is described by means of Rodrigues parameters.

Indeed, by letting $\mathbf{r}_i = \mathbf{R}\mathbf{b}_i$, $i = 1 \dots 3$, and by clearing the nonzero denominator, $1 + e_1^2 + e_2^2 + e_3^2$, p_1 , p_2 and p_3 become quartic polynomials in the Rodrigues parameters:

$$p_h := \sum_{\substack{k=0 \dots 4 \\ m=0 \dots 4-k \\ n=0 \dots 4-k-m}} D_{h,kmn} e_1^k e_2^m e_3^n = 0, \quad h = 1 \dots 3 \quad (4.20)$$

The quartic polynomials in e_1 , e_2 and e_3 that emerge from the minors $\det \mathbf{M}'_{1245}(O)$, $\det \mathbf{M}'_{1246}(O)$ and $\det \mathbf{M}'_{1256}(O)$ depend linearly on p_1 , p_2 and p_3 and may be discarded. A further quartic emerges by setting $\det \mathbf{M}'_{j456}(O) = 0$ for $j = 1 \dots 3$, so that:

$$(\mathbf{x} + \mathbf{r}_1) [\det \mathbf{M}'_{456, 234}(O)] = \mathbf{0} \quad (4.21)$$

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

The variety defined by Eq. (4.21) comprises, other than the trivial solution $\bar{\mathbf{x}}_1 = \mathbf{a}_1 - \mathbf{r}_1 = -\mathbf{r}_1$, the set of configurations for which

$$p_8 := \det \mathbf{M}'_{456, 234}(O) = 0 \quad (4.22)$$

Equation (4.22) is indeed of degree 4 in e_1, e_2 and e_3 (as well as of degree 2 in x, y, z). All minors of $\mathbf{M}'(O)$ not considered so far; namely, $\det \mathbf{M}'_{1345}(O)$, $\det \mathbf{M}'_{1346}(O)$, $\det \mathbf{M}'_{1356}(O)$, $\det \mathbf{M}'_{2345}(O)$, $\det \mathbf{M}'_{2346}(O)$ and $\det \mathbf{M}'_{2356}(O)$, instead yield sextic equations in the Rodrigues parameters. In order to solve such a system of polynomials, Sylvester's dialytic method is implemented as an appropriate elimination strategy.

An effective implementation of the strategy consists of deriving a larger set of linearly independent quartics. Let \mathbf{M} be written by choosing a generic P as the reduction pole of moments; namely:

$$\mathbf{M}(P) = \begin{bmatrix} \cdots & -\mathbf{s}_i & \cdots & \mathbf{k} \\ \cdots & -(B_i - P) \times \mathbf{s}_i & \cdots & (G - P) \times \mathbf{k} \end{bmatrix} \quad (4.23)$$

When $P \equiv B_i$ and $P \equiv A_i$, $i = 1 \dots 3$, the moment vector in the i th column vanishes, so that setting $\det \mathbf{M}_{j456}(B_i) = 0$ and $\det \mathbf{M}_{j456}(A_i) = 0$ for $j = 1 \dots 3$ yields, respectively:

$$\mathbf{s}_i [\det \mathbf{M}_{456, km4}(B_i)] = \mathbf{0} \quad (4.24)$$

and

$$\mathbf{s}_i [\det \mathbf{M}_{456, km4}(A_i)] = \mathbf{0} \quad (4.25)$$

with $k, m \in \{1, 2, 3\} - \{i\}$. In this way, the following equations may be obtained:

$$p_9 := \det \mathbf{M}_{456, 234}(B_1) = 0 \quad (4.26a)$$

$$p_{10} := \det \mathbf{M}_{456, 134}(B_2) = 0 \quad (4.26b)$$

$$p_{11} := \det \mathbf{M}_{456, 134}(A_2) = 0 \quad (4.26c)$$

$$p_{12} := \det \mathbf{M}_{456, 124}(B_3) = 0 \quad (4.26d)$$

$$p_{13} := \det \mathbf{M}_{456, 124}(A_3) = 0 \quad (4.26e)$$

Analogously, by defining a convenient additional point, G_0 , such that $G - G_0 = \mathbf{k}$, and by setting $P \equiv G$ and $P \equiv G_0$, one may also obtain:

$$p_{14} := \det \mathbf{M}_{456, 123}(G) = 0 \quad (4.27a)$$

$$p_{15} := \det \mathbf{M}_{456, 123}(G_0) = 0 \quad (4.27b)$$

All polynomials p_j , $j = 9 \dots 15$, have degree 4 in the Rodrigues parameters (and degree 2 in the components of \mathbf{x}).

No other quartic polynomial that is linearly independent from p_1 , p_2 and p_3 may be obtained from the minors of \mathbf{M} by varying the moment pole. The relations in Eqs. (4.20), (4.22), (4.26) and (4.27) form a system of 11 quartics, comprising 15 monomials in e_1 and e_2 . By multiplying such quartics by e_2 , 11 additional relations may be introduced, which comprise 5 novel monomials in e_1 and e_2 . Among the new equations, 9 may be chosen so as to form, together with the original ones, a linear system of the form:

$$\mathbf{S}(e_3)\mathbf{E}_{23} = \mathbf{0} \quad (4.28)$$

where $\mathbf{S}(e_3)$ is a 20×20 matrix depending only on e_3 , and \mathbf{E}_{23} is a column vector comprising all monomials of e_1 and e_2 with degree ≤ 5 , except e_1^5 . The last three entries of \mathbf{E}_{23} are, in particular, e_1 , e_2 and 1. The determinant of $\mathbf{S}(e_3)$ provides a 24th-degree resultant devoid of spurious roots; namely:

$$\det\mathbf{S}(e_3) = \sum_{h=0}^{24} F_h e_3^h = 0 \quad (4.29)$$

where the coefficients F_h depend only on the position of G and the geometric parameters. For each root of Eq. (4.29), a single value of e_1 and e_2 may be obtained by solving the linear system in Eq. (4.28). As a proof that all solutions of Eq. (4.29) may be *real*, Carricato [2013a] a numerical example with a set of all-real solutions.

Since Rodrigues parameters are unable to describe orientations requiring $e_0 = 0$, this case must be considered separately. For example, by setting $e_0 = 0$ and $e_j = 1$, for some j , the relations in Eqs. (4.20), (4.22), (4.26) and (4.27) become quartics in e_h and e_k , $h \neq j \neq k$. Since these equations contain only 5 monomials in e_k , they are more than sufficient to eliminate e_k and thus obtain a univariate resultant in e_h . If special geometric conditions are satisfied, common solutions may, however, exist.

The equation degrees involved in the solving procedure are rather high and all coefficients are generally nonzero, so that positive-dimensional solution sets are unlikely to be found (except, obviously, when $(x, y) = (\bar{x}_i, \bar{y}_i)$ and $z \neq \bar{z}_i$, $i = 1 \dots 3$, in which case the robot is allowed to operate like a 1-DOF crane, with the i th cable holding the entire load).

4.1.3 Inverse geometric static problem of 4-4 CDPRs

When solving the IGP of an n -dof robot, n platform coordinates need to be assigned. For the scenario involving the IGP of 4-4 CDPRs, two relevant cases are considered, depending on whether: (i) the orientation, Φ , is assigned and G is constrained to lie in a given plane (*IGP with assigned orientation*); or (ii) the position of G is known and a further point, B_5 , on the platform is required to lie in a given plane (*IGP with assigned position*).

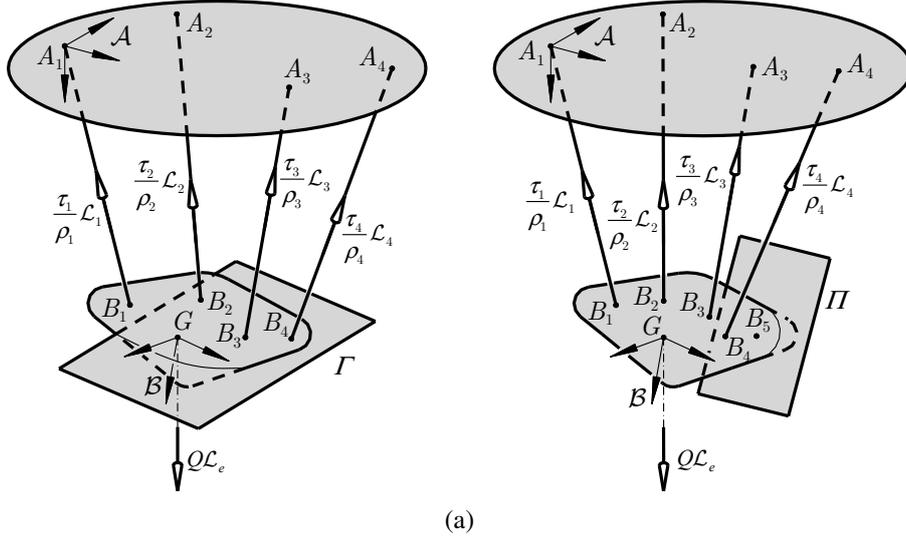


Figure 4.2: A cable-driven parallel robot with 4 cables: (a) model for the IGP with assigned orientation, (b) model for the IGP with assigned position.

4.1.3.1 Inverse geometric static problem of 4-4 CDPRs with assigned orientation

In the case of an IGP with assigned orientation, Φ is known and G is constrained to lie in a plane Γ (Fig. 4.2a). The pose coordinates are subject to 4 linear constraints, $q_i(\mathbf{X}) = 0, i = 1 \dots 4$:

$$e_1 = \bar{e}_1, e_2 = \bar{e}_2, e_3 = \bar{e}_3, \quad \mathbf{g} \cdot \mathbf{n} - d_\Gamma = xn_1 + yn_2 + zn_3 - d_\Gamma = 0 \quad (4.30)$$

where \bar{e}_1, \bar{e}_2 and \bar{e}_3 are known scalars, $\mathbf{n} = [n_1, n_2, n_3]^T$ is a unit vector perpendicular to Γ and $|d_\Gamma|$ is the distance of Γ from A_1 . Two sub-cases may be identified, depending on whether $n_3 \neq 0$ or $n_3 = 0$.

1. $n_3 \neq 0$:

When $n_3 \neq 0$, Γ is a *non-vertical* plane and z may be expressed, from the last relationship in Eq. (4.30), as $z = -(n_1/n_3)x - (n_2/n_3)y + d_\Gamma/n_3$. By taking advantage of this expression and by imposing the first three constraints in Eq. (4.30), the 5 relations $p_j(\mathbf{X}) = 0, j = 1 \dots 5$, emerging from Eq. (3.10) become cubic in x and y , comprising 10 monomials: $[y^3, y^2x, yx^2, x^3, y^2, yx, x^2, y, x, 1]$.

The problem may be efficiently solved by implementing a Sylvester dialytic method; namely, by rewriting the relations $p_j = 0$ as linear equations in all monomials involving the original unknowns except one, which is 'hidden' in the equation coefficients. If these monomials are treated as linear unknowns, a square homogeneous system is obtained and the determinant of the coefficient matrix provides a resultant in the hidden variable. In the case at hand, by hiding y , 4 monomials in x emerge and, thus, *four* relations $p_j = 0, j = 1 \dots 4$ may be used to build up a square Sylvester matrix. The corresponding resultant, however, exhibits a spurious solution. In order to remove the extraneous factor, all *five* relations $p_j = 0, j = 1 \dots 5$, may be linearised in the 5 monomials contained in the array $\kappa_1 = [y^3, x^3, x^2, x, 1]^T$:

$$\mathbf{S}_1(y)\kappa_1 = \mathbf{0} \quad (4.31)$$

Table 4.1: A 4-4 CDPR whose IGP with assigned orientation admits 5 *real* potential solutions ($d_\Gamma = 3/\sqrt{6}$).

$[A_2]_{\mathcal{A}}$	$[A_3]_{\mathcal{A}}$	$[A_4]_{\mathcal{A}}$	$[B_1 - G]_{\mathcal{A}}$	$[B_2 - G]_{\mathcal{A}}$	$[B_3 - G]_{\mathcal{A}}$	$[B_4 - G]_{\mathcal{A}}$	$\sqrt{6}\mathbf{n}$
8	9	1	-3	3	2	3	-2
0	7	8	4	-2	1	1	1
5	6	4	3	1	0	2	1

where $\mathbf{S}_1(y)$ is a 5×5 matrix whose entries are known polynomial functions of y . Letting the determinant of $\mathbf{S}_1(y)$ vanish yields a 5th-degree univariate equation in y . This is obtained in symbolic form and is devoid of spurious roots. For each root, a unique value of x may be obtained by solving the linear system in Eq. (4.31).

Solutions may be either *complex* or *real*, with only the latter being of physical interest. By varying the robot's geometry, the number of real roots may change. Table 4.1 reports an example for which the IGP with assigned orientation admits 5 *real* solutions, though with not all of them necessarily feasible.

2. $n_3 = 0$:

When $n_3 = 0$ and, without loss of generality, $n_2 \neq 0$, Γ is *vertical* and y may be expressed, from the last relation in Eq. (4.30), as $y = -(n_1/n_2)x + (d_\Gamma/n_2)$.

By substituting this expression into the relations $p_j(\mathbf{X}) = 0$, $j = 1 \dots 5$, one obtains 5 cubics in the monomials $[z^2x, zx^2, x^3, z^2, zx, x^2, z, x, 1]$. By a procedure similar to that described for the case $n_3 \neq 0$, a least-degree univariate equation free of extraneous polynomial factors may be obtained by linearising all *five* relations $p_j = 0$, $j = 1 \dots 5$, in the 5 monomials contained in the array $\kappa_2 = [z^2, x^3, x^2, x, 1]^T$ and writing them in the form:

$$\mathbf{S}_2(z)\kappa_2 = \mathbf{0} \tag{4.32}$$

where $\mathbf{S}_2(z)$ is a 5×5 matrix whose entries are known polynomial functions of z . Letting the determinant of $\mathbf{S}_2(z)$ vanish yields a 4th-degree univariate polynomial in z , which is available in symbolic form and devoid of spurious roots. For each root, a unique value of x may be obtained by solving the linear system in Eq. (4.32). Even in this case, it is possible that all roots are *real*.

4.1.3.2 Inverse geometric static problem of 4-4 CDPR with assigned position

In the case of an IGP with assigned position, G is known and a further point on the platform, B_5 , is constrained to lie on an assigned plane, Π (Fig. 4.2a). The constraints $q_i(\mathbf{X}) = 0$, $i = 1 \dots 4$ become:

$$x = \bar{x}, y = \bar{y}, z = \bar{z}, \quad \mathbf{r}_5 \cdot \mathbf{n} - d_\Pi = 0 \tag{4.33}$$

where \bar{x} , \bar{y} and \bar{z} are known scalars, \mathbf{r}_5 is the position vector of B_5 in $Oxyz$, \mathbf{n} is a unit vector perpendicular to Π and $|d_\Pi|$ is the distance of Π from A_1 .

Table 4.2: A 4-4 CDPR whose IGP with assigned position admits 32 *real* potential solutions ($d_{\Pi} = -0.0543588$).

$[A_2]_A$	$[A_3]_A$	$[A_4]_A$	$[B_1]_B$	$[B_2]_B$	$[B_3]_B$	$[B_4]_B$	$[B_5]_B$	\mathbf{n}	$(\bar{x}, \bar{y}, \bar{z})$
0.0096715	0.1602038	0.3227272	0.8585338	0.3187879	0.6598471	0.6273182	0.4579794	0.8894538	0.1339193
0	0.0649423	0.7151215	0	0.4888859	0.6661870	0.9610494	0.5558744	0.4392272	0.2021438
0.5484151	0.6597958	0.5378416	0	0	0.8744523	0.0797477	0.8222827	-0.1262981	0.1180386

If $\mathbf{R}(\Phi)$ is the rotation matrix between $Gx'y'z'$ and $Oxyz$, the position vector of B_i , $i = 1 \dots 5$, in $Oxyz$ may be expressed as $\mathbf{r}_i = [B_i]_{Oxyz} = \mathbf{g} + \mathbf{R}(\Phi) \mathbf{b}_i$, where the position of B_i in $Gx'y'z'$ is known. By substituting these expressions in the 5 relations $p_j(\mathbf{X}) = 0$, $j = 1 \dots 5$, and imposing the first three constraints in Eq. (4.33), one obtains 5 sextic equations in e_1 , e_2 and e_3 . By a similar expansion, the fourth relationship in Eq. (4.33) (i.e. $q_4 = 0$) becomes a quadratic equation in the Rodrigues parameters.

By denoting the ideal generated by the set $J = \{p_1, \dots, p_5, q_4\}$ as $\langle J \rangle$, the solutions of the IGP with assigned position form the variety, V , of $\langle J \rangle$. The high order of the polynomials in J suggests applying elimination procedures based on Groebner bases in order to solve the problem. A Groebner basis, $G[J]$, of $\langle J \rangle$ with respect to a grevlex(e_1, e_2, e_3) may be computed in a very expedited manner; tenths of seconds for the case at hand on a PC with a 2.67GHz Intel Xeon processor and 4GB of RAM. Once $G[J]$ is known, the FGLM algorithm may be called upon to compute a univariate polynomial in $\langle J \rangle$. A more efficient alternative, however, is provided by the Sylvester elimination procedure outlined in Section 2.1.5.3. $G[J]$ comprises 12 polynomials containing 12 monomials in e_1 and e_2 : $\kappa_3 = [e_1 e_2^4, e_2^5, e_1 e_2^3, e_2^4, e_1 e_2^2, e_2^3, e_1^2, e_1 e_2, e_2^2, e_1, e_2, 1]^T$. Accordingly, $G[J]$ may be set up as a square system of homogeneous linear equations of the form:

$$\mathbf{S}_3(e_3) \kappa_3 = \mathbf{0} \tag{4.34}$$

where $\mathbf{S}_3(e_3)$ is a 12×12 matrix polynomial in e_3 . Letting the determinant of $\mathbf{S}_3(e_3)$ vanish yields a spurious-root-free univariate polynomial of degree 32 in e_3 . For each root, unique values of e_1 and e_2 may be obtained by solving the linear system in Eq. (4.34).

As reported in Table 4.2, an example of a set of robot parameters have been found proving that all 32 solutions may be *real*.

4.1.4 Direct geometric static problem of CDPRs

The powerful Groebner-basis elimination strategy will be implemented to perform elimination in the direct geometric static problem due to the complexity of the governing equations. Generally, the method encompasses three steps:

1. a Groebner basis, G , is computed with respect to an efficient monomial order;
2. a subset of the original unknowns is eliminated by computing a Groebner basis, G_l , of a suitable elimination ideal utilising the FGLM algorithm (Faugère et al. [1993]);

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

3. a least-degree univariate polynomial devoid of extraneous factors is computed by applying an elimination strategy.

For a generic CDPR with n cables, $n \leq 6$, it so happens that the number of monomials in the unknown variables, \mathbf{X} , comprised in the ideal of the governing equations, J , usually become very high, in the order of several hundreds. As a consequence, computing a Groebner basis is a nontrivial task. The following strategy is adopted:

- All geometric parameters of the robot are assigned generic rational values so as to ease numerical computation in the computer algebra system, i.e. the GroebnerPackage provided within the software Maple15. Accordingly, $\langle J \rangle \subset \mathbb{Q}[\mathbf{X}]$, where $\mathbb{Q}[\mathbf{X}]$ is the set of all polynomials in \mathbf{X} with coefficients in \mathbb{Q} .
- Groebner bases are computed with respect to grevlex monomial order.
- The *abundance of generators* in J plays a key role, since it greatly speeds up the calculation; a feature already noted by [Dhingra et al. \[2000\]](#)). It is therefore fully exploited here within.

By the above expedients, a Groebner basis, $G_{>grevlex}[J]$, of $\langle J \rangle$ may generally be computed in a fairly expedited way. Once $G[J]$ is known, the normal set, $\mathbf{N}[J]$, of $\langle J \rangle$; that is, the set of all monomials that are *not* multiples of any leading monomial in $G[J]$, may be easily computed. From the properties of Groebner bases ([Corless \[1996\]](#); [Möller \[1998\]](#)), the number of monomials in $\mathbf{N}[J]$ coincides with the number of complex roots in the variety of $\langle J \rangle$ and thus with the number of solutions, N_{sol} , of the problem at hand. This is the order of the least-degree univariate polynomials comprised in $\langle J \rangle$.

4.1.4.1 Direct geometric static problem of 2-2 CDPRs

The DGP is characterised by assigned values of ρ_1 and ρ_2 and an unknown platform pose. Considering x as a linear function of z from the second equation in Eq. (4.6), and substituting into Eq. (4.5) and the first equation in Eq. (4.6), yields two quadratic equations in z ; namely, $p_4 = 0$ and $p_5 = 0$, from which z may be eliminated. By cancelling the nonzero factor, $16(a_{2x} - r_{21x})^4$, a single equation in θ may further be obtained in the form of $p_6 = 0$. Expressing s_θ and c_θ as functions of $t_\theta = \tan(\theta/2)$ and clearing the factor $(1 + t_\theta^2)^6$, a 12th-degree polynomial equation in t_θ is finally obtained:

$$\sum_{k=0}^{12} B_k t_\theta^k = 0 \quad (4.35)$$

As explained by [Carricato and Merlet \[2013\]](#), for each solution of Eq. (4.35), a single value of z is computed by the greatest common divisor of p_4 and p_5 , with a single value of x finally obtained from the second equation in Eq. (4.6). By considering both operational modes, the problems admits up to 24 solutions.

4.1.4.2 Direct geometric static problem of 3-3 CDPRs

In this case too, all cable lengths are assigned and the platform configuration must be determined. In order to resolve the problem, both the equations emerging from the geometric constraints and those inferred from static equilibrium must be solved. Equation (3.10) provides a number of polynomial relations for the platform posture variables. Among them, Eqs. (4.20), (4.22), (4.26) and (4.27) are of degree 4 in the elements of Φ (i.e. the Rodrigues parameters) and degree 2 in the components of \mathbf{x} . Equation (3.8) provides three further relations in \mathbf{X} ; namely, $\|\mathbf{s}_1\|^2 - \rho_1^2 = 0$, $\|\mathbf{s}_2\|^2 - \rho_2^2 = 0$ and $\|\mathbf{s}_3\|^3 - \rho_3^2 = 0$. By subtracting the first relation from both the second and the third one, and by clearing the denominator $1 + e_1^2 + e_2^2 + e_3^2$, one obtains:

$$q_1 := H_{200}x^2 + H_{020}y^2 + H_{002}z^2 + H_{100}x + H_{010}y + H_{001}z + H_{000} = 0 \quad (4.36a)$$

$$q_2 := I_{100}x + I_{010}y + I_{001}z + I_{000} = 0 \quad (4.36b)$$

$$q_3 := K_{100}x + K_{010}y + K_{001}z + K_{000} = 0 \quad (4.36c)$$

where all coefficients H_{kmn} , I_{kmn} and K_{kmn} are quadratic functions in e_1 , e_2 and e_3 . Let $\langle J \rangle$ be the ideal generated by the set of polynomials $J = \{q_1, q_2, q_3, p_1, p_2, p_3, p_8, \dots, p_{15}\}$. q_1 , q_2 and q_3 have, respectively, degrees of 4, 3 and 3 in \mathbf{X} , whereas all other generators in J have degree 6 in the same variables.

In general, 348 monomials in \mathbf{X} are involved and a Groebner basis, $G[J]$, of $\langle J \rangle$ with respect to grevlex(\mathbf{X}) may be computed with variables ordered such that $z > y > x > e_1 > e_2 > e_3$.

Table 4.3: DGP of a 3-3 robot: structure of the Groebner bases of the elimination ideals of $\langle J \rangle$

$G[J_l]$	$\mathbf{X} \setminus \mathbf{X}_l$	N_l	Degrees of the generators	Highest degree in w ,	No. of monomials with variables in
			in $\mathbf{X} \setminus \mathbf{X}_l$	$w \in \mathbf{X} \setminus \mathbf{X}_l$	$\mathbf{X} \setminus \mathbf{X}_l - \{w\}, w \in \mathbf{X} \setminus \mathbf{X}_l$
$G[J]$	$[z, y, x, e_1, e_2, e_3]$	137	3(2), 4(41), 5(94)	4, 4, 4, 4, 5, 5	183, 183, 172, 181, 150, 137
$G[J_1]$	$[y, x, e_1, e_2, e_3]$	126	5(96), 6(30)	5, 5, 5, 5, 6	145, 144, 142, 141, 126
$G[J_2]$	$[x, e_1, e_2, e_3]$	84	6(54), 7(30)	6, 6, 6, 7	98, 98, 94, 84
$G[J_3]$	$[e_1, e_2, e_3]$	45	8(9), 9(36)	8, 8, 9	53, 53, 45
$G[J_4]$	$[e_2, e_3]$	18	17(15), 18(3)	17, 18	19, 18
$G[J_5]$	$[e_3]$	1	156(1)	156	–

A key factor for the efficiency of such a computation is the abundance of generators available in J , which significantly simplifies and speeds up the calculation. By exploiting all 14 generators, the computation of $G[J]$ for the exemplifying 3-3 robot whose dimensions will be reported in Table 4.4 requires roughly 1.3min on a PC with a 2.67GHz Intel Xeon processor and 4GB of RAM. If only 6 generators are used, computation time may be up to 30 times higher and, most significantly, spurious solutions are introduced.

Once $G[J]$ is known, the normal set of $\langle J \rangle$ may easily be computed; namely (in vector format):

$$\mathbf{N}[J] = [1, e_1, e_2, e_3, x, y, z, e_1^2, e_1e_3, \dots, xze_1e_2]^T \quad (4.37)$$

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

Since $\mathbf{N}[J]$ comprises 156 monomials, this is also the number of complex roots in the algebraic variety, V , of $\langle J \rangle$.

In order to actually solve J and thus eliminate the unknowns, Groebner bases are required with respect to the elimination monomial orders (Section 2.1.5). The structure of the Groebner basis, $G[J_l]$, of $\langle J_l \rangle$ with respect to grevlex($\mathbf{X} \setminus \mathbf{X}_l$) as obtained by the FGLM algorithm is illustrated in Table 4.3 for $l = 0 \dots 5$. Column 3 reports the number of generators, N_l , in $G[J_l]$, column 4 reports the degree of such generators in $\mathbf{X} \setminus \mathbf{X}_l$ (the number of generators for each degree is given in brackets), column 5 reports the highest power of w in $G[J_l]$ for each variable $w \in \mathbf{X} \setminus \mathbf{X}_l$ and column 6 reports the number of monomials in $G[J_l]$ having variables in $\mathbf{X} \setminus \mathbf{X}_l - \{w\}$ for each variable. By computing elimination ideals via the FGLM algorithm, a least-degree polynomial in one variable may be obtained. In practical cases, however, the procedure may fail due to an excessively onerous computational burden. As outlined in Section 2.1.5.3, a more effective procedure that can be implemented to obtain the univariate equation is based on the Groebner-Sylvester hybrid approach. For the problem at hand, $G[J]$ comprises 137 generators. By choosing $w \neq e_3$ (e_3 being the ‘smallest’ variable in the monomial ordering chosen to compute $G[J]$), the number of monomials in $\mathbf{X} - \{w\}$ proves to be always greater than 137; however, by choosing $w = e_3$, the number of monomials in $\mathbf{X} - \{e_3\}$ is exactly equal to 137 (see Table 4.3). By this approach, the resultant in w emerges from the expansion of a 137×137 matrix, which is still a very expensive computational task. On the contrary, as Table 4.3 shows, the Groebner basis, $G[J_3]$, of $\langle J \rangle \cap \mathbb{Q}[e_1, e_2, e_3]$ with respect to grevlex(e_1, e_2, e_3) comprises only 45 polynomials (9 of degree 8 in Φ and 36 of degree 9 in Φ), including 45 monomials in e_1 and e_2 (of degree ranging from 0 to 8). It follows that if w is assigned the role of ‘hidden’ variable, the generators of $G[J_l]$ may be written in the form:

$$\mathbf{T}(w) \mathbf{E}_w = \left(\sum_{k=0}^u w^k \mathbf{B}_k \right) \mathbf{E}_w = \mathbf{0} \quad (4.38)$$

where u is the highest power of w in $G[J_l]$, \mathbf{B}_k is a $N_l \times N_l$ numerical matrix, $\mathbf{T}(w)$ is a matrix polynomial of degree u in w and \mathbf{E}_w is a N_l -dimensional vector comprising all monomials in $G[J_l]$ having variables in $\mathbf{X} \setminus \mathbf{X}_l - \{w\}$. Accordingly, the desired resultant, free from extraneous polynomial factors, is:

$$\det \mathbf{T}(w) = \sum_{h=0}^{156} L_h w^h = 0 \quad (4.39)$$

with the coefficients L_h depending only on the input data; namely, the robot geometry and cable lengths.

The advantage gained by applying a dialytic step to a Groebner basis $G[J_l]$ with $l > 0$ is evident in the data presented in Table 4.4. This table reports the CPU time required to compute grevlex bases for the elimination ideals of $\langle J \rangle$ with $l = 0 \dots 5$ on the aforementioned PC. In particular, the third column reports the CPU time, $T_{G[J_l]}$, required to obtain $G[J_l]$ by computing $\langle J \rangle \cap \mathbb{Q}[\mathbf{X} \setminus \mathbf{X}_l]$ or, in brackets, by computing $\langle J_{l-1} \rangle \cap \mathbb{Q}[\mathbf{X} \setminus \mathbf{X}_l]$. The elimination task generally proves to be computationally expensive and time consuming¹. In particular, the ‘deeper’ the elimination process (i.e. the smaller the number of

¹Computation time may significantly increase depending on the complexity of the coefficients of the polynomials in J .

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

Table 4.4: DGP of a 3-3 robot: Computation time to obtain Groebner bases of the elimination ideals of $\langle J \rangle$

Geometric dimensions and load: $\mathbf{a}_2 = [10; 0; 0]$, $\mathbf{a}_3 = [0; 12; 0]$, $\mathbf{b}_1 = [1; 0; 0]$, $\mathbf{b}_2 = [0; 1; 0]$, $\mathbf{b}_3 = [0; 0; 1]$, $(\rho_1, \rho_2, \rho_3) = (7.5, 10, 9.5)$, $Q = 10$.

l	$\langle J_l \rangle$	$T_{G[J_l]}$ [min]	$T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$ [min]
0	$\langle J \rangle$	1.3	1919
1	$\langle J \rangle \cap \mathbb{Q}[y, x, e_1, e_2, e_3]$	19	2159
2	$\langle J \rangle \cap \mathbb{Q}[x, e_1, e_2, e_3]$	42 (27)	579
3	$\langle J \rangle \cap \mathbb{Q}[e_1, e_2, e_3]$	49 (24)	33
4	$\langle J \rangle \cap \mathbb{Q}[e_2, e_3]$	160 (80)	11
5	$\langle J \rangle \cap \mathbb{Q}[e_3]$...	–

variables in $\mathbf{X} \setminus \mathbf{X}_l$, the longer the time taken to perform the computation and, more critically, the larger the amount of memory required. The latter issue is particularly critical; indeed, the final elimination ideal for the example at hand cannot be computed on the given PC due to excessive memory usage¹. The fourth column reports the CPU time, $T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$, required to calculate $\langle J \rangle \cap \mathbb{Q}[e_3]$ by applying Sylvester's dialytic method to $G[J_l]$ for $l = 0 \dots 4$. In this case, computation time depends on the dimension of $\mathbf{T}(w)$ and thus normally decreases with the number of variables in $\mathbf{X} \setminus \mathbf{X}_l$. Memory requirements are modest and the algorithm is normally successful. Deriving $\langle J \rangle \cap \mathbb{Q}[e_3]$ by computing $\langle J_3 \rangle$ and then applying dialytic elimination to $G[J_3]$ requires less than 1/20 of the computation time that would be needed by applying dialytic elimination directly to $G[J]$. It emerges from the above consideration that a hybrid approach, which eliminates a subset of variables by the FGLM algorithm and further applies Sylvester's method on the Groebner basis of the corresponding elimination ideal, provides a profitable strategy to compute a least-degree univariate polynomial in $\langle J \rangle$.

4.1.4.3 Direct geometric static problem of 4-4 CDPR

When all cables of the robot are in tension, Eq.(3.8) yields 4 relations in \mathbf{X} . By subtracting the first relation from the other 3 and clearing the denominator $1 + e_1^2 + e_2^2 + e_3^2$, the following equations are obtained:

$$q_1 := H_{200}x^2 + H_{020}y^2 + H_{002}z^2 + H_{100}x + H_{010}y + H_{001}z + H_{000} = 0 \quad (4.40a)$$

$$q_2 := I_{100}x + I_{010}y + I_{001}z + I_{000} = 0 \quad (4.40b)$$

$$q_3 := J_{100}x + J_{010}y + J_{001}z + J_{000} = 0 \quad (4.40c)$$

$$q_4 := K_{100}x + K_{010}y + K_{001}z + K_{000} = 0 \quad (4.40d)$$

From Eq.(3.11), when $P \equiv A_1$, all equations emerging by setting the 5×5 minors of \mathbf{M} equal to

¹ *Maple* estimated a required memory usage of about 12GB to derive $\langle J_5 \rangle$ from $\langle J_4 \rangle$.

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

zero are linearly independent; namely:

$$p_1 := \det \mathbf{M}_{23456}(A_1) = 0 \quad (4.41a)$$

$$p_2 := \det \mathbf{M}_{13456}(A_1) = 0 \quad (4.41b)$$

$$p_3 := \det \mathbf{M}_{12456}(A_1) = 0 \quad (4.41c)$$

$$p_4 := \det \mathbf{M}_{12356}(A_1) = 0 \quad (4.41d)$$

$$p_5 := \det \mathbf{M}_{12346}(A_1) = 0 \quad (4.41e)$$

$$p_6 := \det \mathbf{M}_{12345}(A_1) = 0 \quad (4.41f)$$

On the other hand, the relations obtained by letting $P \equiv A_i, i = 2 \dots 4$, are linearly dependent on those in Eq.(4.41) and may therefore be discarded. 9 additional linearly independent equations may conversely be obtained by letting $P \equiv B_i, i = 1 \dots 4$; that is:

$$p_7 := \det \mathbf{M}_{23456}(B_1) = 0 \quad (4.42a)$$

$$p_8 := \det \mathbf{M}_{13456}(B_1) = 0 \quad (4.42b)$$

$$p_9 := \det \mathbf{M}_{12456}(B_1) = 0 \quad (4.42c)$$

$$p_{10} := \det \mathbf{M}_{23456}(B_1) = 0 \quad (4.42d)$$

$$p_{11} := \det \mathbf{M}_{13456}(B_1) = 0 \quad (4.42e)$$

$$p_{12} := \det \mathbf{M}_{12456}(B_1) = 0 \quad (4.42f)$$

$$p_{13} := \det \mathbf{M}_{23456}(B_3) = 0 \quad (4.42g)$$

$$p_{14} := \det \mathbf{M}_{13456}(B_4) = 0 \quad (4.42h)$$

$$p_{15} := \det \mathbf{M}_{12456}(B_5) = 0 \quad (4.42i)$$

Two more may be obtained by letting $P \equiv G$; that is:

$$p_{16} := \det \mathbf{M}_{23456}(G) = 0 \quad (4.43a)$$

$$p_{17} := \det \mathbf{M}_{13456}(G) = 0 \quad (4.43b)$$

As outlined in Section 2.1.5, the variety of the ideal $\langle J \rangle$ generated by the polynomial set $J = \{q_1, q_2, q_3, q_4, p_1, \dots, p_{17}\}$ is the solution of the problem. q_1, q_2, q_3 and q_4 have, respectively, degrees of 4, 3, 3 and 3 in \mathbf{X} ; whereas all other generators in $\langle J \rangle$ are of degree 9 in the same variables. In general, 1576 monomials in \mathbf{X} are involved. A Groebner basis, $G[J]$, of $\langle J \rangle$ with respect to grevlex(\mathbf{X}) may be computed with variables ordered such that $z > y > x > e_1 > e_2 > e_3$. As explained previously, a key factor for the efficiency of such a computation is the abundance of generators available in $\langle J \rangle$, which simplifies and speeds up calculation. Numerical tests have shown that the fastest computation is achieved by exploiting the first 19 generators in $\langle J \rangle$: $\{q_1, q_2, q_3, q_4, p_1, \dots, p_{17}\}$. The computation of $G[J]$ for the exemplifying robot whose dimensions are reported in Table 4.6 requires roughly 18min on a PC with a 2.67GHz Intel Xeon processor and 4GB of RAM. If only 6 generators are used, compu-

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

tation time is 4 times higher and, most importantly, spurious solutions are introduced as only two out of six minors of \mathbf{M} are used. Knowing $G[J]$, the normal set of $\langle J \rangle$ may be easily computed; namely (in vector format):

$$\mathbf{N}[J] = [1, e_3, e_2, e_1, x, y, z, e_3^2, e_2e_3, \dots, e_1e_2e_3^2x, e_1e_2e_3^2y]^T \quad (4.44)$$

Since $\mathbf{N}[J]$ comprises 216 polynomials, this is also the number of complex roots (including multiplicity) in the algebraic variety, V , of $\langle J \rangle$ and thus the order of the least-degree univariate polynomials comprised in $\langle J \rangle$.

The structure of $G[J]$ with respect to $\text{grevlex}(\mathbf{X})$, as obtained by the FGLM algorithm, is illustrated in Table 4.5 for $l = 0 \dots 5$. Column 3 reports the number of generators, N_l , in $G[J_l]$, column 4 reports the degree of such generators in $\mathbf{X} \setminus \mathbf{X}_l$ (the number of generators for each degree is given in brackets) and column 5 reports the number of monomials in $G[J_l]$ having variables in $\mathbf{X} \setminus \mathbf{X}_l - \{w\}$ for each variable $w \in \mathbf{X} \setminus \mathbf{X}_l$.

As in the 3-cable case, by computing elimination ideals via the FGLM algorithm, a least-degree polynomial in one variable may be obtained. However, the advantage gained by applying a dialytic step to a Groebner basis, $G[J_l]$, with $l > 0$ is evident in the data presented in Table 4.6. The table reports the CPU time required to compute *grevlex* bases for the elimination ideals of $\langle J \rangle$ with $l = 0 \dots 5$ on the aforementioned PC for an exemplifying 4-4 CDPR. In particular, the third column reports the CPU time, $T_{G[J_l]}$, required to obtain $G[J_l]$ by computing $\langle J_{l-1} \rangle \cap \mathbb{Q}[\mathbf{X} \setminus \mathbf{X}_l]$ via the FGLM algorithm and the fourth column reports the CPU time, $T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$, required to calculate $\langle J \rangle \cap \mathbb{Q}[e_3]$ via application of dialytic elimination to $G[J_l]$ for $l = 0 \dots 4$. As expected, the higher l is (i.e. the more variables are eliminated), the more demanding the FGLM elimination becomes. In particular, the last elimination ideal cannot be computed due to excessive memory usage. Conversely, the computation time of the dialytic step decreases with l , as it depends on the dimensions of $\mathbf{T}(w)$. For the example at hand, $\langle J \rangle \cap \mathbb{Q}[e_3]$ cannot be computed from $G[J_l]$ with $l = 0 \dots 2$ due to excessive computation time. Instead, the univariate polynomial in e_3 may be successfully computed from either $G[J_3]$ or $G[J_4]$. The more efficient computation of $\langle J \rangle \cap \mathbb{Q}[e_3]$ is obtained by eliminating $\{x, y, z\}$ by the FGLM algorithm and $\{e_1, e_2\}$ by the dialytic step.

4.1.4.4 Direct geometric static problem of 5-5 CDPRs

Solutions to the DGP of a robot with 5 cables and the corresponding univariate polynomial are computed by implementing the same elimination strategy. From Eq. (3.8), 5 polynomials may be derived; namely, $\{q_i, i = 1 \dots 5\}$, similar to Eq. (4.40). For the case of 5-5 CDPRs, \mathbf{M} is a 6×6 matrix from which only one equation is obtained by letting its determinant equal zero:

$$p := \det \mathbf{M}(A_1) = 0 \quad (4.45)$$

In contrast to the cases of robots with $n < 5$ cables, all other similar relations obtained by changing the reduction pole of the moment linearly depend on Eq.(4.45) and must therefore be discarded. Thus, in

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

Table 4.5: Structure of the Groebner bases of the elimination ideals of 4-4 CDPRs

$G[J_l]$	$\mathbf{X} \setminus \mathbf{X}_l$	N_l	Degrees of the generators in $\mathbf{X} \setminus \mathbf{X}_l$	No. of monomials with variables in $\mathbf{X} \setminus \mathbf{X}_l - \{w\}, w \in \mathbf{X} \setminus \mathbf{X}_l$
$G[J]$	$[z, y, x, e_1, e_2, e_3]$	195	3(3), 4(5), 5(158), 6(29)	230, 232, 232, 271, 224, 195
$G[J_1]$	$[y, x, e_1, e_2, e_3]$	147	5(36), 6(111)	185, 180, 181, 160, 147
$G[J_2]$	$[x, e_1, e_2, e_3]$	111	5(1), 7(99), 8(11)	127, 127, 117, 111
$G[J_3]$	$[e_1, e_2, e_3]$	61	8(1), 10(60)	66, 61, 61
$G[J_4]$	$[e_2, e_3]$	21	20(15), 21(6)	22, 21
$G[J_5]$	$[e_3]$	1	216(1)	–

Table 4.6: Computation times to obtain Groebner bases of the elimination ideals of 4-4 CDPRs for the robot geometry : $\mathbf{a}_1 = [0; 0; 0]$, $\mathbf{a}_2 = [9; 0; 1]$, $\mathbf{a}_3 = [11; 9; 0]$, $\mathbf{a}_4 = [-2; 8; -1]$, $\mathbf{b}_1 = [-2; -1; -1]$, $\mathbf{b}_2 = [1; -2; 0]$, $\mathbf{b}_3 = [2; 1; -1]$, $\mathbf{b}_4 = [0; 2; -1]$, $(\rho_1, \rho_2, \rho_3, \rho_4) = (6, 7, 8, 9)$, $Q = 10$.

l	$\langle J_l \rangle$	$T_{G[J_l]}$ [min]	$T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$ [min]
0	$\langle J \rangle$	17	...
1	$\langle J \rangle \cap \mathbb{Q}[y, x, e_1, e_2, e_3]$	227	...
2	$\langle J \rangle \cap \mathbb{Q}[x, e_1, e_2, e_3]$	670	...
3	$\langle J \rangle \cap \mathbb{Q}[e_1, e_2, e_3]$	567	340
4	$\langle J \rangle \cap \mathbb{Q}[e_2, e_3]$	1063	67
5	$\langle J \rangle \cap \mathbb{Q}[e_3]$...	–

this case the ideal, $\langle J \rangle$, comprises just 6 polynomials, $\{q_1, q_2, q_3, q_4, q_5, p\}$, in which q_1, q_2, q_3, q_4 and q_5 have, respectively, degrees of 4, 3, 3, 3 and 3 in \mathbf{X} while p , inferred from static equilibrium, has a degree as high as 9 in the same variables. The total number of monomials in \mathbf{X} contained in $\langle J \rangle$ is 1576. Though the computation of $G[J]$ may not take advantage of redundant generators in J , it is relatively fast, mainly due to the fact that a single high-degree polynomial appears in J . For the exemplifying robot reported in subsequent Table 4.8, $G[J]$ may be computed in roughly 3 minutes.

The normal set; namely:

$$\mathbf{N}[J] = [1, e_3, e_2, e_1, x, y, z, e_3^2, e_2e_3, \dots, e_1xy^2, e_1xyz, x^4]^T \quad (4.46)$$

contains 140 monomials such that $N_{sol} = 140$. The structure of $G[J_l]$ with respect to grevlex($\mathbf{X} \setminus \mathbf{X}_l$) is reported in Table 4.7 for $l = 0 \dots 5$. The table is constructed in the same manner as Table 4.5. Notably, in this case, $G[J_l]$ comprises monomials in $\mathbf{X} \setminus \mathbf{X}_l - \{e_3\}$ equal to N_l for only $l = 1, 3, 4$. Hence, Sylvester dialytic elimination may be applied to neither $G[J]$ nor $G[J_2]$. Indeed, these contain more monomials in $\mathbf{X} \setminus \mathbf{X}_l - \{e_3\}$ than available generators.

Table 4.8 reports, for an exemplifying generic robot, the CPU time, $T_{G[J_l]}$, required to compute

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

Table 4.7: Structure of the Groebner bases of the elimination ideals $\langle J_l \rangle$ of $\langle J \rangle$ for a CDPR with 5 cables and a generic geometry

$G[J_l]$	$\mathbf{X} \setminus \mathbf{X}_l$	N_l	Degrees of the generators in $\mathbf{X} \setminus \mathbf{X}_l$	No. of monomials with variables in $\mathbf{X} \setminus \mathbf{X}_l - \{w\}, w \in \mathbf{X} \setminus \mathbf{X}_l$
$G[J]$	$[z, y, x, e_1, e_2, e_3]$	110	3(4), 4(46), 5(60)	159, 156, 147, 168, 141, 118
$G[J_1]$	$[y, x, e_1, e_2, e_3]$	110	4(6), 5(84), 6(20)	130, 128, 139, 124, 110
$G[J_2]$	$[x, e_1, e_2, e_3]$	68	4(1), 5(4), 6(40), 7(23)	87, 96, 83, 69
$G[J_3]$	$[e_1, e_2, e_3]$	31	4(1), 8(11), 9(6), 10(6), 11(6), 12(1)	62, 43, 31
$G[J_4]$	$[e_2, e_3]$	17	16(13), 17(4)	18, 17
$G[J_5]$	$[e_3]$	1	140(1)	–

$G[J_l]$ from $\langle J_{l-1} \rangle \cap \mathbb{Q}[\mathbf{X} \setminus \mathbf{X}_l]$ and the CPU time, $T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$, required to calculate $\langle J \rangle \cap \mathbb{Q}[e_3]$ by applying Sylvester's elimination to $G[J_l]$ for $l = 1, 3, 4$. The most efficient computation is obtained by eliminating $\{x, y, z\}$ via the FGLM algorithm and $\{e_1, e_2\}$ via a dialytic step ($3.3 + 43.0 + 59.5 + 56.7 + 7.5 \approx 170$ min). $G[J_3]$ comprises 11 polynomials (1 of degree 4, 11 of degree 8, 6 of degree 9, 6 of degree 10, 6 of degree 11 and 1 of degree 12 in Φ), including 31 monomials in e_1 and e_2 . Hence, a 140-degree polynomial in e_3 may be computed by expanding a 31×31 matrix.

Table 4.8: Computation times to obtain Groebner bases of the elimination ideals, $\langle J_l \rangle$, for a robot with 5 cables and $\mathbf{a}_1 = [0; 0; 0]$, $\mathbf{a}_2 = [1; 2; -0.75]$, $\mathbf{a}_3 = [3.5; 1; 1]$, $\mathbf{a}_4 = [3.25; -1; 1]$, $\mathbf{a}_5 = [1; -2; -0.5]$, $\mathbf{b}_1 = [-1; 0; -1]$, $\mathbf{b}_2 = [-0.5; 1; -1.25]$, $\mathbf{b}_3 = [0.75; 0.75; -1.25]$, $\mathbf{b}_4 = [0.5; -0.75; -1.25]$, $\mathbf{b}_5 = [-0.25; -0.8; -1.5]$, $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (4.5; 5; 3; 3.75; 4.75)$.

l	$\langle J_l \rangle$	$T_{G[J_l]}$ [min]	$T_{\langle J \rangle \cap \mathbb{Q}[e_3]}$ [min]
0	$\langle J \rangle$	3.3	...
1	$\langle J \rangle \cap \mathbb{Q}[y, x, e_1, e_2, e_3]$	43.0	4042
2	$\langle J \rangle \cap \mathbb{Q}[x, e_1, e_2, e_3]$	59.5	...
3	$\langle J \rangle \cap \mathbb{Q}[e_1, e_2, e_3]$	56.7	7.5
4	$\langle J \rangle \cap \mathbb{Q}[e_2, e_3]$	73.0	10.7
5	$\langle J \rangle \cap \mathbb{Q}[e_3]$...	–

4.2 Numerical computation of the solution set of the DGP

As discussed in Section 4.1.4, for cases of DGP, the elimination strategy yields univariate polynomials of high degree that have no practical use due to the inability to manage such high orders from a numerical point of view. Thus, purely numerical options may be considered to compute the solutions.

4.2.1 Eigenvalue formulation

As discussed in Section 2.1.5.2, once a Groebner basis, $G[J]$, of $\langle J \rangle$ is known, the solutions of the problem may be efficiently computed by constructing a multiplication matrix with the eigenproblem method in Eq. (2.11). Considering $\mathbf{N}[J]$ from Eqs. (4.37), (4.44) and (4.46) for the cases of robots with 3, 4 and 5 cables, one may identify that the first 7 entries of $\mathbf{N}[J]$ are always $1, e_1, e_2, e_3, x, y, z$. For each case, a unique solution in \mathbf{X} emerges from each eigenvector, \mathbf{N}_i , as:

$$e_{1,i} = \frac{t_{i,2}}{t_{i,1}}, e_{2,i} = \frac{t_{i,3}}{t_{i,1}}, e_{3,i} = \frac{t_{i,4}}{t_{i,1}}, x_i = \frac{t_{i,5}}{t_{i,1}} y_i = \frac{t_{i,6}}{t_{i,1}}, z_i = \frac{t_{i,7}}{t_{i,1}}. \quad (4.47)$$

For all cases, computation of $\mathbf{A}[J, e_3]$ and the corresponding eigenvalues may be performed in tenths of a second. Once a robot configuration is found, cable tensions may be obtained via any subset of Eq. 3.9. A drawback of this approach is that it relies on a prior computation of $G[J]$. The efficiency of Groebner basis computation depends heavily on the ‘size’ of the rational coefficients of the polynomials generating the basis. Indeed, for Groebner-basis computation, the coefficients expressing the robot geometric parameters must be assigned rational values, which are normally obtained by converting real values. The higher the number of digits in the original floating-point data, the larger the numerators and denominators of the resulting rationals and the larger the coefficients in the polynomials forming the ideal. As a result, computation becomes slower and more memory-demanding. This is a limitation shared by all computational methods using Groebner bases.

4.2.2 Homotopy continuation

Due to the drawbacks of the eigenvalue formulation discussed in the previous section, homotopy continuation (Sommesse and Wampler [2005]) is chosen to compute the solution set numerically. Continuation has the significant advantage that it requires no prior Groebner-basis computation by a computer algebra system and real-value geometric parameters may be directly handled with floating-point arithmetic. As a consequence, the dependence of computation time on the specific values of the robot parameters is rather modest.

Though formulating equilibrium constraints via Eq. (3.11) is particularly favourable when pursuing elimination strategies, the complexity and high degree of the polynomials emerging from the elimination of cable tensions are a disadvantage for continuation algorithms, as they slow down computation and cause stability problems. For this reason, formulation via Eqs. (3.8) and (3.9) is preferable. For the same reason, pose parameterisation by Study or Dietmaier coordinates (see Sections 3.3 and 3.4) is preferable over the 6-parameter representation described in Section 3.2. Though the resulting system involves more variables and more equations than that in the elimination strategy, it comprises much simpler lower-order polynomials, which are more stable when homotopy continuation is implemented, thus leading to faster computation.

4.2.2.1 General homotopy continuation

As introduced in Section 2.2, polynomial homotopy continuation is a path-tracking technique that transforms a start system of polynomial equations with known solutions to a target system whose solutions must be found (Sommese and Wampler [2005]). The method tracks the evolution of a system such as:

$$\mathbf{H}(\mathbf{Y}, t) = \gamma(1 - t)\mathbf{F}_0(\mathbf{Y}) + t\mathbf{F}_1(\mathbf{Y}) = \mathbf{0} \quad (4.48)$$

where $\mathbf{F}_0(\mathbf{Y})$ and $\mathbf{F}_1(\mathbf{Y})$ are the start and the target system, respectively, γ is a randomly selected complex number and t is a real number called the *continuation parameter*. The concept consists of varying t from 0 to 1 while tracking the solutions of the problem from those of $\mathbf{F}_0(\mathbf{Y}) = \mathbf{0}$, known, to those of $\mathbf{F}_1(\mathbf{Y}) = \mathbf{0}$, unknown.

As previously discussed, the procedure may be classified as *general homotopy continuation* or *parameter homotopy continuation* depending on how the start system is constructed. The former is employed when no information is known *a priori* about the roots of the target system. In this case, a start system yielding the maximum possible number of solutions must be constructed. This number is equal to the minimum multi-homogeneous Bezout number, N_{Bez} , of the target system (Wampler et al. [1990]). Since N_{Bez} coincides with the number of paths to be tracked, it is of paramount importance for computational efficiency.

In the case of a robot with 2 cables, one may take advantage of in-plane formulation by ignoring all variables corresponding to direction \mathbf{j} . Accordingly, the attachment points on the base and moving platform may be expressed as:

$$\begin{aligned} [A_1]_O &= [0, 0, 0]^T, [A_2]_O = [a_{21}, 0, a_{23}]^T, [B_1]_{O'} = [b_{11}, 0, 0]^T, [B_2]_{O'} = [b_{21}, 0, 0]^T, \\ [G]_O &= [x, 0, z]^T \end{aligned} \quad (4.49)$$

Using Dietmaier parameterisation, having $\mathbf{x} = [x, 0, z]^T$ and altering the components of \mathbf{e}_1 and \mathbf{e}_2 to:

$$\mathbf{e}_1 = [e_{11}, 0, e_{13}]^T, \quad \mathbf{e}_2 = [e_{21}, 0, e_{23}]^T \quad (4.50)$$

the position vectors of points B_1 , B_2 and G in O_{xz} may be written as functions of 6 variables:

$$\mathbf{X} = [x, z, e_{11}, e_{13}, e_{21}, e_{23}]^T \quad (4.51)$$

Using this parameterisation, 2 geometric equations from Eq. (3.8) and 3 equilibrium equations from Eq. 3.9 may be derived. By introducing the 3 equations from Eq. (3.19), a system of 8 polynomials in 8 variables (\mathbf{X} , τ_1 and τ_2) is obtained, which governs the DGP of a 2-2 CDPR in the x - z plane. As outlined in Section 2.2.1, when considering general homotopy continuation, a start system must be constructed with the number of solutions equal to the minimum multi-homogeneous Bezout number. For the case at hand, using the multi-homogenisation $\{\tau_1, \tau_2\}, \{e_{11}, e_{13}, e_{21}, e_{23}, x, z\}$ yields a minimum Bezout number of 96.

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

In contrast, in cases of robots with $n > 3$, the smallest N_{Bez} for the governing equations may be obtained using other parameterisations. In Table 4.9, the smallest Bezout number according to different parameterisations are listed. For each case, the partition of variables yielding the smallest Bezout number is as follows:

1. Rodrigues' parameterisation: $\{\tau_1, \dots, \tau_n\}, \{e_1, e_2, e_3\}, \{x, y, z\}$
2. Dietmaier's parameterisation: $\{\tau_1, \dots, \tau_n\}, \{e_{11}, e_{12}, e_{13}\}, \{e_{21}, e_{22}, e_{23}\}, \{x, y, z\}$
3. Study's parameterisation: $\{\tau_1, \dots, \tau_n\}, \{x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3\}$

Table 4.9: Minimum multi-homogeneous Bezout number of equations governing DGP of n - n CDPR according to different parameterisations

n - n CDPR	Rodrigues' parameterisation	Dietmaier's parameterisation	Study's parameterisation
3-3 CDPR	10080	9600	5120
4-4 CDPR	10560	14400	3840
5-5 CDPR	5760	11520	1536
6-6 CDPR	1280	3840	256

Accordingly, since the parameterisation yielding the smallest N_{Bez} for the equations governing the DGP of CDPRs appears to be the one based on Study coordinates, this has been chosen for implementation of the general continuation routine. It is easy to see that N_{Bez} is much larger than the actual number solutions to the problem, N_{sol} , which for the same robots is equal to 156, 216, 140 and 40, respectively. As a consequence, for $t \rightarrow 1$, many paths diverge to infinity and only a limited number converge to finite solutions. Tracking diverging paths causes significant and non-beneficial computational burden. Furthermore, when Study parameterisation is used, the number of converging paths is equal to $2N_{sol}$. This is due to the fact that \mathbf{S} and $-\mathbf{S}$ represent the same platform pose (see Section 3.3). As a result, if \mathbf{S} is a solution to the problem, so is $-\mathbf{S}$. Clearly, repeated solutions must be discarded.

4.2.2.2 Parameter homotopy continuation

Parameter homotopy continuation may be implemented when a start system and a homotopy with the same number of finite solutions can be determined for any value of t . In this case, the burden of tracking diverging paths is ignored from the start. Since the coefficients of the equations governing the DGP are continuous functions of the geometric parameters of the robot, \mathbf{P} , a continuous path through parameter space results in continuous evolution of the coefficients and, generally, continuous solution paths as well. Accordingly, using a suitable homotopy, only N_{sol} paths originating at the isolated roots of the start system must be tracked, while those corresponding to diverging solutions may be ignored. This is the reason why, when implementing parameter continuation, Dietmaier parameterisation is preferred over Study parameterisation. In the latter case, in fact, the number of isolated roots of the equations governing the DGP is $2N_{sol}$ (see Section 4.2.2.1), which is also the number of finite paths that

must be tracked for parameter homotopy to work in a robust manner. This problem does not appear if Dietmaier parameterisation is used, as in this case the number of isolated roots of the modelling equations is exactly N_{sol} .

When N_{sol} isolated roots are known for a generic $\mathbf{P} = \mathbf{P}_0$, the solutions for any other $\mathbf{P} = \mathbf{P}_1$ may be found by tracking the homotopy:

$$\mathbf{F}(\mathbf{Y}, (1-t)\mathbf{P}_0 + t\mathbf{P}_1) = \mathbf{0} \tag{4.52}$$

with t varying from 0 to 1 or, more robustly, along the curve $t = \gamma t' / [1 + (\gamma - 1)t']$, with $t' \in [0, 1]$ and $\gamma \in \mathbb{C}$.

4.3 Number of real-valued solutions of the DGP

As outlined previously, the solution strategies introduced for solving the systems of equations governing the displacement analysis of under-constrained CDPRs provide all solutions to the problems in the complex field. As a matter of fact, the number of solutions, N_{sol} , may be complex or real; however, only the latter are of physical interest. By varying robot parameters, the number of real roots varies. Since there may be roots that always remain complex in the solution set, the maximum number of real solutions in each case may be less than N_{sol} . Determining a tight bound on this number is a challenging task. In the DGP of under-constrained CDPRs, the upper bound on the number of solutions in the complex field was 24, 156, 216 and 140 for robots with 2, 3, 4 and 5 cables, respectively. Except in the case of a 2-2 CDPR, where a robot parameter set was obtained randomly with all 24 solutions being real, there is usually no knowledge of the maximum number of real solutions that may be obtained for a particular parameter set. Due to the very high number of solutions, it is almost impossible to find a robot parameter set with all such solutions being real. It is therefore necessary to use a systematic procedure for finding parameters with the maximum number of real solutions. Accordingly, the Dietmaier procedure introduced in Section 2.3 may be implemented to find sets of CDPR parameters for which the DGP provides the highest number of real configurations. The algorithm begins with a system of equations with known solutions and proceeds by iteratively changing the parameters in such a way that two complex conjugate solutions become continually nearer until they transform into a double real root and, thereafter, a pair of distinct real solutions. This is achieved by decrementing the absolute value of the imaginary parts of the two complex conjugate solutions in question. The procedure is repeated for all pairs of complex conjugate roots until the maximum number of real solutions is obtained. To implement the algorithm successfully, the governing equations of the problem must first be re-derived by choosing an appropriate coordinate system and applying a suitable parameterisation. More explanation on how the equations are derived is provided in the following sections.

4.3.1 Governing equations and parameterisation

It has been shown that, for each solution strategy, the equations governing the DGP may become simpler by adopting a suitable parameterisation and choosing an appropriate coordinate system. For example, Rodrigues' parameterisation with the least number of variables is suitable for elimination strategies, while Study's and Dietmaier parameterisation are the most suitable procedures for numerical methods. Further to parameterisation, the proper definition of geometric parameters has great importance if the algorithm is to work efficiently. Here, the equations governing the DGP of a robot are re-derived by choosing appropriate coordinate axes and reference points. The origin of the Cartesian frame appended to the moving platform, $O'x'y'z'$, is chosen to coincide with B_1 such that the position of the moving platform with respect to the base is described by the vector \mathbf{n} , identifying the position of point B_1 instead of the centre of mass, G . By this choice, the position vectors of points B_1, B_2, \dots, B_n , and G on frame \mathcal{B} may be expressed as:

$$\begin{aligned} [B_1]_{O'} &= [0, 0, 0]^T, [B_2]_{O'} = [b_{21}, 0, 0]^T, [B_3]_{O'} = [b_{31}, b_{32}, 0]^T, \\ &\vdots \\ [B_n]_{O'} &= [b_{n1}, b_{n2}, b_{n3}]^T, \\ [G]_{O'} &= [g_1, g_2, g_3]^T \end{aligned} \tag{4.53}$$

As discussed in Section 2.3, the algorithm is mainly based on the definition of two scalar functions, S_h and D_{rs} , that measure the 'distance' between either a pair of complex conjugate solutions or two real roots. Since the solutions of the DGP, are vectors, any definition of such functions requires appropriate weighting of the vector components. This is difficult to accomplish, however, since the solution vectors are dimensionally non-homogeneous. The non-homogeneity of solutions is due to the fact that they comprise variables corresponding to cable tensions. Parameterisation can also cause non-homogeneity. For example, as discussed in Section 3.2, Rodrigues parameterisation gives solutions of mixed dimensions. Those of $\Phi = (e_1 : e_2 : e_3)$ are dimensionless, while parameters in \mathbf{x} corresponding to the position of the moving platform have dimensions of length. The problem of non-homogeneity may be circumvented by firstly changing the parameterisation. As discussed in Section 3.4, Dietmaier parameterisation provides all posture parameters of the moving platform with the same dimensions of length, which makes it appropriate for defining the scalar functions in the algorithm. Additionally, by observing that a unique solution for the cable tensions, τ , corresponds to any given equilibrium pose, \mathbf{X} , the cable tension variables may be ignored in the aforementioned definition. Accordingly, distance functions may be equivalently defined in the platform configuration space, which is dimensionally homogeneous¹.

By implementing Dietmaier parameterisation and scaling the robot such that $\rho_1 = 1$, the position vector $\mathbf{n} = B_1 - A_1$ has unit magnitude². Eqs. (3.8), (3.9) and (3.19) yield a system of equations,

¹ Indeed, the solutions of the DGP may be computed directly in terms of \mathbf{X} by way of Eqs. (3.8) and (3.11).

² This is somewhat equivalent to establishing a 'characteristic' length for the Euclidean group of rigid-body displacements. The operation is generally physically inconsistent, but it is not so in this case, as ρ_1 is an assigned constant.

$\mathbf{F}(\mathbf{Y}, \mathbf{P}) = 0$, in which \mathbf{Y} and \mathbf{P} are defined as follows:

$$\mathbf{P} = [\rho_2, \rho_3, \dots, \rho_n, a_{21}, a_{23}, a_{31}, a_{32}, a_{33}, \dots, a_{n1}, a_{n2}, a_{n3}, b_{21}, b_{31}, b_{32}, \dots, b_{n1}, b_{n2}, b_{n3}, g_1, g_2, \dots, g_n]^T \quad (4.54)$$

$$\mathbf{Y} = [\mathbf{X}^T, \boldsymbol{\tau}^T]^T = [n_1, n_2, n_3, e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, \tau_1, \tau_2, \dots, \tau_n]^T \quad (4.55)$$

4.3.2 Application of the Dietmaier algorithm

The start system is easily constructed by choosing an arbitrary set, \mathbf{P}_0 , of geometric parameters and computing the corresponding solution set by homotopy continuation. From Eq. (2.29), the increment of a complex solution, \mathbf{Y} , according to a change in \mathbf{P} , may be obtained as:

$$\mathbf{A}d\mathbf{P} + \mathbf{B}d\mathbf{Y} = \mathbf{0} \quad (4.56)$$

The dimensions of matrices \mathbf{A} and \mathbf{B} vary depending on the number of cables in each CDPR case. For example, for a robot with 3 cables, the system of equations $\mathbf{F}(\mathbf{Y}, \mathbf{P})$ comprises 12 equations for 12 unknown variables, \mathbf{Y} , and 13 geometric parameters, \mathbf{P} . Accordingly, the size of matrices \mathbf{A} and \mathbf{B} are 12×13 and 12×12 , respectively. From Eq. 2.31, the increment of \mathbf{Y} may be computed as:

$$d\mathbf{Y} = -\mathbf{B}^{-1}\mathbf{A}d\mathbf{P} \quad (4.57)$$

Since the last n rows of matrix $-\mathbf{B}^{-1}\mathbf{A}$ for CDPRs relate the increment of cable tensions to changes in \mathbf{P} , they may be eliminated from the matrix $-\mathbf{B}^{-1}\mathbf{A}$. Calling this new matrix \mathbf{J} , $d\mathbf{X}$ may be computed as:

$$d\mathbf{X} = \mathbf{J}d\mathbf{P} \quad (4.58)$$

and thus:

$$d\mathbf{X}_h = \mathbf{J}_h d\mathbf{P} \quad (4.59)$$

where \mathbf{J}_h is matrix \mathbf{J} evaluated for $\mathbf{Y} = \mathbf{Y}_h$.

An alternative procedure consists of choosing relations within Eq. (3.11) and complementing them with Eqs. (3.8) and (3.19), thus obtaining an alternative system of 9 equations that depend only on the 9 platform coordinates and the geometric parameters; namely:

$$\mathbf{F}'(\mathbf{X}, \mathbf{P}) = [f'_1(\mathbf{X}, \mathbf{P}), \dots, f'_9(\mathbf{X}, \mathbf{P})]^T = \mathbf{0} \quad (4.60)$$

By differentiating Eq. (4.60), one obtains an alternative formulation, \mathbf{J}'_h , of \mathbf{J}_h . Indeed, by choosing different triplets of equations within Eq. (3.11), one may obtain *several* alternative formulations of \mathbf{J}_h . Switching between \mathbf{J}_h and any other substitute Jacobian, \mathbf{J}'_h , may prove useful when \mathbf{J}_h is ill-conditioned, in order to smooth the numerical computation. By the above observation, the minimisation and maximisation procedure formulated in Eq. (2.38) and Eq. (2.39) may be redefined according to the corresponding pose, \mathbf{X}_h , as:

$$\begin{aligned}
 \text{minimize : } & \quad \text{sgn} [Im(\mathbf{X}_h^T)] Im(\mathbf{J}_h) \Delta \mathbf{P} \\
 \text{subject to : } & \quad -\Delta \mathbf{P}_{max} \leq \Delta \mathbf{P} \leq \Delta \mathbf{P}_{max}; \\
 & \quad (\mathbf{X}_r - \mathbf{X}_s)^T (\mathbf{J}_r - \mathbf{J}_s) \Delta \mathbf{P} \geq 0, \quad \forall (\mathbf{X}_r, \mathbf{X}_s) : D_{rs} < D_{min}
 \end{aligned} \tag{4.61}$$

and:

$$\begin{aligned}
 \text{maximize : } & \quad (\mathbf{X}_u - \mathbf{X}_v)^T (\mathbf{J}_u - \mathbf{J}_v) \Delta \mathbf{P} \\
 \text{subject to : } & \quad -\Delta \mathbf{P}_{max} \leq \Delta \mathbf{P} \leq \Delta \mathbf{P}_{max}; \\
 & \quad (\mathbf{X}_r - \mathbf{X}_s)^T (\mathbf{J}_r - \mathbf{J}_s) \Delta \mathbf{P} \geq 0, \quad \forall (\mathbf{X}_r, \mathbf{X}_s) : D_{rs} < D_{min}
 \end{aligned} \tag{4.62}$$

At this point, the optimisation procedures may be performed efficiently. It may happen that the optimisation procedure fails, returning $\Delta \mathbf{P} = \mathbf{0}$ for $S_h > \varepsilon$. In this case, the algorithm attempts to advance by improving the accuracy with which the solution path is tracked (achieved by decreasing ΔP_{max} or switching between different formulations of the DGP equations) or by relaxing the constraints in Eq. (2.38) (i.e. decreasing D_{min} to allowing real solutions to become closer). By finding an increment, $\Delta \mathbf{P}$, the set of geometric parameters is updated and the solutions of the modified $\mathbf{F}(\mathbf{Y}, \mathbf{P})$ are then computed by help of an iterative Newton-Raphson routine. If the Newton-Raphson routine does not converge or converges twice to the same solution (likely to occur when the paths corresponding to two distinct solutions pass close to each other), a different start guess is attempted (i.e. \mathbf{X}_h) or a different formulation of the DGP equations is utilised. If the Newton-Raphson routine continues to fail, the solution set is updated by a more robust, but slower, solver; namely, a parameter-homotopy-based routine implemented within Bertini [Bates et al.](#). At any iteration step, the algorithm works on the pair of complex conjugate solutions whose mutual distance is the shortest. The general structure of the algorithm is presented in Tab. 4.3.2.

4.3.2.1 Auxiliary parameters of the algorithm

The effectiveness and efficiency of Dietmaier's algorithm depends on the proper adjustment of a number of auxiliary parameters. A brief discussion concerning the most important ones is presented in the following section.

$\Delta \mathbf{P}_{max}$: The smaller $\Delta \mathbf{P}_{max}$, the slower the advancement of the algorithm, but the better the linearisation approximation in procedures (2.38) and (2.39) and, thus, the more accurate the obtained results. A reasonable strategy consists of choosing a not-too-small increment at the beginning of each iteration (e.g. $\Delta P_{max} = 0.01$). Subsequently, if the optimisation step is successful and the imposed constraints are respected after \mathbf{P} and \mathbf{X} are updated, the value of ΔP_{max} is considered acceptable; otherwise, it is

Algorithm 1 Summary of Dietmaier algorithm

Require: A guess for \mathbf{P}_0 and the corresponding solutions computed by homotopy continuation

for All complex solutions **do**

 Choose the complex solution with the shortest mutual distance, \mathbf{X}_i

while Imaginary part of $\mathbf{X}_i \neq 0$ **do**

 Compute $\Delta\mathbf{P}$ according to minimisation 2.38

if minimisation procedure fails **then**

 Try the optimisation procedure by relaxing the constraints, improving the accuracy or switching between different formulations according to Eq. 4.60

if Optimisation continues to fail **then**

 Go to another pair of complex solutions

end if

else

 Update the geometric parameters and compute the new solutions by the Newton-Raphson routine

if Newton-Raphson routine fails **then**

 Use homotopy continuation to compute all solutions

end if

end if

if Two real solutions become too close to each other **then**

 Compute $\Delta\mathbf{P}$ according to the maximisation 2.39 and compute solutions according to the updated \mathbf{P}

end if

end while

end for

decreased. For instance, when the minimisation (2.38) is attempted, ΔP_{max} is decreased when either $\Delta \mathbf{P} = \mathbf{0}$ (i.e. the minimisation algorithm fails) or a nonzero $\Delta \mathbf{P}$ is found but, after \mathbf{P} and \mathbf{X} are updated, D_h is verified to not decrease or the number of real solutions is lower.

D_{min} : The smaller this parameter, the weaker the imposed constraints on the optimisation routines and, thus, the likelier the routines are to be successful. When D_{min} is small, however, real solutions are allowed to become closer together and the risk of two becoming a complex conjugate is higher. Accordingly, at the beginning of each iteration, D_{min} is set equal to a given fraction (e.g. 1/6) of the average distance between all real solutions. Subsequently, if optimisation fails, D_{min} is decreased (i.e. constraints are relaxed); whereas, if a pair of real solutions switches back to the complex domain, D_{min} is increased.

ε : This parameter should be chosen, in theory, small enough to make, from a numerical point of view, both $(\mathbf{X}_h, \bar{\mathbf{X}}_h) \in \mathbb{C}^9 \times \mathbb{C}^9$ and $(\mathbf{X}_u, \mathbf{X}_v) \in \mathbb{R}^9 \times \mathbb{R}^9$ satisfy $\mathbf{F}(\mathbf{Y}, \mathbf{P}) = 0$. Setting ε too small, however, would significantly slow the process. As a rule of thumb, experiments have shown that ε is not required to be smaller than e^{-4} .

4.3.2.2 Case study

The program has been repeatedly executed with different start systems and parameter-tuning configurations for CDPRs with 3, 4 and 5 cables. Accordingly, the maximum number of real solutions obtained for each case is 54, 98 and 74, respectively. An example of the corresponding parameters is as follows:

1. Example of 3-3 CDPR parameters with maximum number of 54 real equilibrium configurations:
 $\mathbf{a}_1 = [0, 0, 0]$, $\mathbf{a}_2 = [0.89744, 0, -0.72651]$, $\mathbf{a}_3 = [0.65671, 0.74636, -0.59091]$, $\mathbf{b}_1 = [1.55665, 0, 0]$, $\mathbf{b}_2 = [0.69695, 0.78429, 0]$, $\mathbf{b}_3 = [-0.46441, -1.38980, -1.19948]$, $(\rho_1, \rho_2, \rho_3) = (1, 1.0805, 2.6025)$ and $Q = 10$.
2. Example of 4-4 CDPR parameters with maximum number of 98 real equilibrium configurations:
 $\mathbf{a}_1 = [5.521454, 7.836054, -1.009788]$, $\mathbf{a}_2 = [-5.366081, 8.252356, 1.959491]$, $\mathbf{a}_3 = [-10.315057, 1.612391, 0.946641]$, $\mathbf{a}_4 = [5.392802, -7.491653, 0.579554]$, $\mathbf{b}_1 = [2.182515, 3.334434, 1.997996]$, $\mathbf{b}_2 = [-2.214432, 3.330068, 1.013029]$, $\mathbf{b}_3 = [-3.659557, 0.265737, 1.111276]$, $\mathbf{b}_4 = [1.404915, -3.195786, 1.219710]$, $(\rho_1, \rho_2, \rho_3, \rho_4) = (14.549082, 14.549329, 15.763856, 10.898894)$ and $Q = 10$.
3. Example of 5-5 CDPR parameters with maximum number of 74 real equilibrium configurations:
 $\mathbf{a}_1 = [8.762837, -1.064001, -0.715711]$, $\mathbf{a}_2 = [6.732934, 8.223691, 0.187221]$, $\mathbf{a}_3 = [-5.094292, 7.798299, 1.258330]$, $\mathbf{a}_4 = [-11.189309, -0.138832, -1.614663]$, $\mathbf{a}_5 = [4.870417, -8.101810, 0.176616]$, $\mathbf{b}_1 = [3.889210, 0.116354, 1.549903]$, $\mathbf{b}_2 = [1.262183, 3.258968, 1.975178]$, $\mathbf{b}_3 = [-1.533217, 2.692867, 1.915029]$, $\mathbf{b}_4 = [-3.991803, 0.020197, 1.186929]$, $\mathbf{b}_5 = [1.257342, -3.138215, 1.992397]$, $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (14.772834, 12.755196, 13.153812, 13.969011, 15.712028)$ and $Q = 10$.

Although the Dietmaier algorithm works according to the new coordinate system presented in Section 4.3.1, all parameters have been provided in the old coordinate system discussed in Section 3.1

4. Problem-solving algorithm for the geometric static problem of under-constrained CDPRs

for consistency throughout the thesis. Tables of real solutions for the DGPs relating to each set of parameters are provided in Appendix-A.

Chapter 5

DGP – Solver

Thus far, the equations governing DGP and IGP of CDPRs have been studied and a number of different solving algorithms have been presented for computation of solutions in the complex field for cases of robots with $n \leq 6$ cables. Using an elimination strategy, a proof of the number of solutions has been provided and the univariate equations corresponding to each robot case have been computed. Subsequently, using numerical algorithms, complete solution sets have been computed efficiently. Obviously, all computed solutions are not necessarily solutions of the physical problem. In fact, most of the solutions are complex and do not have any physical meaning at all. For instance, as discussed in Section 4.3, up to now it has not been possible to find a set of parameters with all real solutions for the DGP, except when considering robots with either 2 or 6 cables. Thus, for a given set of parameters, the complex solutions must be discarded and only the real solutions interpreted as having physical meaning. On the other hand, when cable lengths are assigned as the input, nothing ensures *a priori* that, when the platform reaches an equilibrium pose, all cables are active. Indeed, the final pose may be either a DGP solution for the current $n - n$ CDPR or a valid pose for any $m - m$ CDPR that may be derived from the initial $n - n$ robot with $m < n$, in which $n - m$ cables are slack. Finally, even if all of the aforementioned matters are considered, an investigation into the stability of the computed configuration must take place. By taking advantage of the procedures developed in the present study, a computer program, DGP – Solver (Abbasnejad and Carricato), has been composed that solves the DGP for a generic n - n CDPR, with $n \leq 6$. DGP – Solver receives the robot geometry, cable lengths and external load as inputs, computing all possible equilibrium configurations of the robot that are compatible with the given constraints, under the assumption that cables are inextensible and massless. Two noteworthy features of the software are that the computed configurations include those with slack cables and the program determines whether equilibrium configurations are stable or not.

5.1 Feasible equilibrium configurations

The overall solution set includes all possible solutions to the problem in the complex field. Only the *real* solutions for which all cable tensions are *nonnegative* are, however, of physical interest, since cables exert unilateral constraints. Among these, only those in which static equilibrium is stable are

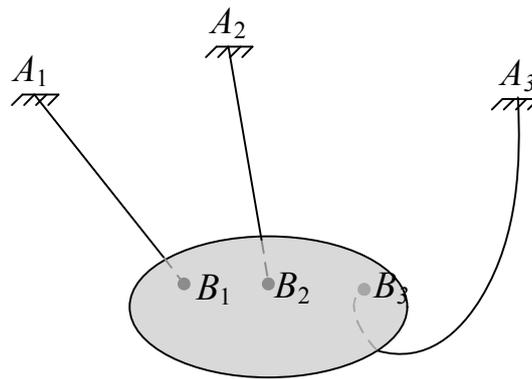


Figure 5.1: Schematic of a robot with one slack cable.

actually *feasible*. The 0-dimensional variety of the ideal generated by Eqs. (3.8) and (3.9) yields all possible solutions of the DGP when n cables are in tension. As previously mentioned, however, the final pose of a robot with n assigned cable lengths may be either a DGP solution for the current $n - n$ CDPR or a valid pose for any $m - m$ CDPR that may be derived from the initial $n - n$ robot, with $m < n$ in which $n - m$ cables are slack. For example, as shown in the schematic in Fig. 5.1, when a robot with 3 cables rests in equilibrium pose, it may happen that a cable becomes slack and the end-effector is supported by just two cables. In the same way, one may suppose that instead of cable 3 being slack, cables 1 or 2, or both, may instead become slack. In general, the number of configurations in which just $m < n$ cables are in tension for a robot with n cables may be calculated as $\binom{n}{m}$ combinations of m out of n .

Accordingly, the overall solution set of a robot with n cables must be obtained by solving the DGP for all possible combinations. The number of combinations for each case is provided in Table 5.1. As an example, for a robot with 5 cables, 31 DGPs must be solved; namely, 1 DGP with 5 active cables, 5 DGPs with 4 active cables, 10 DGPs with 3 active cables, 10 DGPs with 2 active cables and 5 DGPs with 1 active cable.

Table 5.1: Number of different combinations of DGP that must be considered for an n - n CDPR

n - n CDPR	1-1 DGP	2-2 DGP	3-3 DGP	4-4 DGP	5-5 DGP	6-6 DGP
1-1 CDPR	1	-	-	-	-	-
2-2 CDPR	2	1	-	-	-	-
3-3 CDPR	3	3	1	-	-	-
4-4 CDPR	4	6	4	1	-	-
5-5 CDPR	5	10	10	5	1	-
6-6 CDPR	6	15	20	15	6	1

It has been demonstrated that the overall solution set of a robot with n cables is obtained by solving the DGP for all possible combinations. However, all real computed solutions for all possible combi-

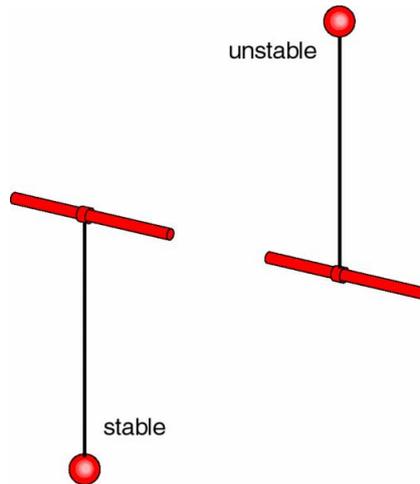


Figure 5.2: Stability of a pendulum

nations may not be physically acceptable. For example, consider the case of the robot with 3 cables shown in Fig. 5.1, where only two cables are active. Resolution of the DGP for a robot with 2 cables, considering cable length constraints and equilibrium, has already been discussed; however, length constraints on the third cable must still be considered since a solution in which $\|A_3 - B_3\| > \rho_3$ is not acceptable. Accordingly, in general, the overall solution set must be obtained by solving the DGP for all possible constraint sets $\{\|A_j - B_j\| = \rho_j, j \in \mathcal{W}\}$, with $\mathcal{W} \subseteq \{1, 2, \dots, n\}$ and $\text{card}(\mathcal{W}) \leq n$, and by retaining, for each solution set, only the solutions for which $\|A_k - B_k\| \leq \rho_k$, with $k \notin \mathcal{W}$. When $m = 6$ and all cables are in tension, the relations in Eq. (3.8) are sufficient to determine the platform pose and the DGP is equivalent to the displacement analysis of the generalised Gough-Stewart manipulator (Merlet [2006]). When a single cable is in tension (i.e. $m = 1$), say the h th one, points A_h , B_h and G must be aligned and $\|A_k - B_k\|$ must be smaller than ρ_k for any $k \neq h$. In this case, the orientation of the platform around the line $A_h B_h$ is undetermined.

After all *admissible* configurations have been found, stability must be assessed. A very well-known example is the two equilibrium positions of a simple pendulum, as shown in Fig. 5.2. As is known, a pendulum can easily deviate from an unstable equilibrium position with a very small perturbation. This is a simple example, which can be extended to the equilibrium position of an under-constrained CDPR. Let an equilibrium configuration, $(\bar{\mathbf{X}}, \bar{\rho}_1 \dots \bar{\rho}_m)$, be considered, with m being the number of cables contributing to support of the platform. By a convenient reordering of indexes, the first m may be assumed to comprise the taut cables, with $m \leq n$. Since the platform preserves $6 - m$ DOFs, it may displace under the effect of a change in the external force acting upon it, while the cable lengths remain unvaried. For the sake of simplicity, it is assumed that the number of cables in tension does not change due to this perturbation, which is reasonable but not necessarily true. Therefore, as noted by Carricato and Merlet [2013], equilibrium solutions in which the pose does not change under the effect of a change in the external force may be accepted as feasible solutions. In particular, while the constraints in Eq. (3.8) hold for $i = 1 \dots m$, G may generally move within a closed region in \mathbb{R}^3 , which in some cases is a surface or curve. If g is the frontier of this region, the equilibrium is stable any

time the potential energy, U , associated with the external wrench, $-Q\mathbf{k} \cdot \mathbf{x}$, is at a local minimum on g . In such a condition, when the platform displaces under the effect of a perturbation, the original configuration is restored if the perturbation ceases. Mathematically, the problem may be defined as a constrained optimisation problem:

$$\begin{aligned} \text{Minimise: } & U = -Q\mathbf{k} \cdot \bar{G}(\mathbf{X}) \\ \text{Subject to:} & \\ q_i := \mathbf{s}_i \cdot \mathbf{s}_i - \rho_i^2 = 0, & \quad i = 1 \dots m \end{aligned} \tag{5.1}$$

Finding the minima of a constrained function is a classic issue in optimisation theory. The problem may be solved by introducing the Lagrange function (Rao [2009]), L , which is defined by Lagrange multipliers λ_j , $j = 1 \dots n$, for each constraint, q_i , as:

$$L = -Q\mathbf{k} \cdot G - \sum_{i=1}^m \lambda_i \rho_i (||\mathbf{s}_i|| - \rho_i) \tag{5.2}$$

The necessary condition for the extremum of L can be obtained by setting its variation equal to zero:

$$\delta L = -Q\mathbf{k} \cdot \delta G - \sum_{i=1}^m \lambda_i \mathbf{s}_i \cdot \delta B_i = 0 \tag{5.3}$$

If $\delta \mathbf{x}$ and $\delta \Phi$ are, respectively, the virtual displacement of G and the virtual rotation of the platform, then:

$$\delta G = \delta \mathbf{x}, \quad \delta B_i = \delta \mathbf{s}_i = \delta \mathbf{x} + \delta \Phi \times \mathbf{r}_i \tag{5.4}$$

and thus:

$$\delta L = -Q\mathbf{k} \cdot \delta \mathbf{x} + \sum_{i=1}^m \lambda_i \mathbf{s}_i \cdot \delta \mathbf{s}_i = \mathbf{f} \cdot \delta \mathbf{x} + \mathbf{m} \cdot \Phi = 0 \tag{5.5}$$

It can be inferred that, by setting \mathbf{f} and \mathbf{m} equal to zero, the same equations in Eq. 3.9 may be obtained from Eq. 5.5. It can further be shown that the physical interpretation of the Lagrange multipliers is the following:

$$\lambda_i = \frac{\tau_i}{\rho_i} \tag{5.6}$$

The sufficient condition for the minimum of L may be assessed by evaluating the definiteness of the reduced Hessian \mathbf{H}_r of L ; that is, the Hessian of L taken with respect to the configuration variables, further restricted to the tangent space of the constraints \mathcal{C} in Eq. 3.8 (Luenberger and Ye [2008]). The

second-order variation of δL is given by:

$$\delta^2 L = -\mathbf{Q}\mathbf{k} \cdot \delta^2 \mathbf{x} + \sum_{i=1}^m \lambda_i \delta \mathbf{s}_i \cdot \delta \mathbf{s}_i + \sum_{i=1}^m \lambda_i \mathbf{s}_i \cdot \delta^2 \mathbf{s}_i \quad (5.7)$$

with:

$$\delta^2 \mathbf{s}_i = \delta^2 \mathbf{x} + \delta^2 \Phi \times \mathbf{r}_i + \delta \Phi \times (\delta \Phi \times \mathbf{r}_i) \quad (5.8)$$

Substituting Eq. 5.8 in Eq. 5.7 and enforcing $\mathbf{f} = \mathbf{m} = 0$ yields:

$$\delta^2 L = \sum_{i=1}^m \lambda_i [\delta \mathbf{s}_i \cdot \delta \mathbf{s}_i + \mathbf{s}_i \cdot \delta \Phi \times (\delta \Phi \times \mathbf{r}_i)] \quad (5.9)$$

and thus:

$$\delta^2 L = \sum_{i=1}^m \lambda_i [\delta \mathbf{x}^T \delta \mathbf{x} - 2\delta \mathbf{x}^T \tilde{\mathbf{r}}_i \delta \Phi + \delta \Phi^T \tilde{\mathbf{r}}_i (\tilde{\mathbf{x}} - \tilde{\mathbf{a}}_i) \delta \Phi] \quad (5.10)$$

where $\tilde{\mathbf{n}}$ denotes, for a generic vector \mathbf{n} the skew-symmetric matrix expressing the operator $\mathbf{n} \times$. $\delta^2 L$ is a bilinear form in the twist space of the platform. If the platform virtual displacement is expressed, in ray coordinates, as $\delta \mathbf{t} = [\delta \mathbf{x}; \delta \Phi]$ and \mathbf{I}_3 denotes the 3×3 identity matrix, the symmetric matrix associated with this form is:

$$\mathbf{H} = \sum_{i=1}^m \lambda_i \begin{bmatrix} \mathbf{I}_3 & -\tilde{\mathbf{r}}_i \\ \tilde{\mathbf{r}}_i & \frac{1}{2} [\tilde{\mathbf{r}}_i (\tilde{\mathbf{x}} - \tilde{\mathbf{a}}_i) + (\tilde{\mathbf{x}} - \tilde{\mathbf{a}}_i) \tilde{\mathbf{r}}_i] \end{bmatrix} \quad (5.11)$$

which represents the pseudo-Hessian of L (\mathbf{H} is not a true and proper Hessian, since $\delta \Phi$ is not generally integrable). The tangent space of \mathcal{C} is obtained by setting the equation

$$\delta (\|\mathbf{s}_i\| - \rho_i) = \delta \|\mathbf{s}_i\| = \frac{\mathbf{s}_i \cdot \delta \mathbf{s}_i}{\rho_i} = \frac{\mathbf{s}_i \cdot \delta \mathbf{x} + \mathbf{r}_i \times \mathbf{s}_i \cdot \delta \Phi}{\rho_i} \quad (5.12)$$

equal to zero for all values of i . In matrix notation, this amounts to:

$$\mathbf{J} \delta \mathbf{t} = \begin{bmatrix} \mathbf{s}_1^T & (\mathbf{r}_1 \times \mathbf{s}_1)^T \\ \vdots & \vdots \\ \mathbf{s}_m^T & (\mathbf{r}_m \times \mathbf{s}_m)^T \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \Phi \end{bmatrix} = \mathbf{0} \quad (5.13)$$

where the i th row of \mathbf{J} coincides with $-\mathcal{L}_i$, expressed in axis coordinates and assuming G is the moment pole. \mathbf{J} is the pseudo-Jacobian of the constraint equations. If \mathbf{N} is any $6 \times (6 - m)$ matrix whose columns generate the null space of \mathbf{J} , the reduced Hessian of \mathcal{C} is the following $(6 - m) \times (6 - m)$ matrix:

$$\mathbf{H}_r = \mathbf{N}^T \mathbf{H} \mathbf{N} \quad (5.14)$$

A sufficient condition for stable equilibrium is \mathbf{H}_r being positive definite.

5.1.1 The software DGP – Solver

DGP – Solver is a software based on the procedures outlined in Sections 4.2.2.1, 4.2.2.2 and 5.1. It solves the DGP for a generic n - n CDPR, with $n \leq 6$ and computes all possible stable configurations that are compatible with the given constraints, including those with slack cables.

5.1.1.1 Input file

The program works according to a text file ‘RobotConfig.txt’, which must be filled in before executing ‘DGP-Solver.exe’. The file is comprised of two parts. The first part specifies the robot geometric parameters and sets a few environment variables that determine the mode in which the program is to run. These parameters consist of the coordinates of the cable exit points on the base, the coordinates of the cable anchor points on the moving platform, the coordinates of the load application point, the magnitude of the external load and the cable lengths. The program interprets the real numbers specified after the string ‘Coordinates Points A:’ as the coordinates of the cable exit points on the base. Coordinates may be specified in an arbitrary fixed Cartesian frame. While the x , y and z coordinates of a single point are separated by commas, distinct points must be separated by a semicolon. The program automatically computes the number of cables in the robot. The real numbers after the string ‘Coordinates Points B:’ define the coordinates of the anchor points on the moving platform in an arbitrarily chosen Cartesian frame on the platform. The syntax is the same as that used for the base exit points. The real numbers after the string ‘Coordinates Point G:’ define the coordinates of the application point of the load (e.g. the centre of mass) on the platform-attached frame. The real numbers after the string ‘Load:’ define the 3 components of the load vector in the fixed Cartesian frame and, finally, cable lengths are specified after the string ‘Cable Lengths:’, separated by commas. After the program is launched, all possible DGPs relating to different combinations of active cables are composed according to the procedure in Section 5.1. Equilibrium configurations may exist in which the platform is supported by m cables, with $m < n$ and $n - m$ cables being slack. By default, the program computes all equilibrium configurations with any possible number, m , of active cables; however, the user may decide to solve the DGP with only a selected number of cables, which is specified after the string ‘Cable Groups:’. In a 5-5 CDPR, for example, entering ‘Cable Groups:’ equal to 2, 3 means the program is supposed to consider the cases in which 2 or 3 cables are active, regardless of the other cases where 1, 4 or 5 cables could be active. After composing the DGP for all the combinations of robot cables, the program begins to solve each case one by one. The numerical procedure, homotopy continuation as discussed in Sections 4.2.2.1 and 4.2.2.2, is implemented to compute all solutions to each problem. In recent years, several software packages have been developed to ease the implementation of homotopy-continuation algorithms. DGP – Solver uses Bertini (Bates et al.) as the computational engine. DGP – Solver uses both parameter and general continuation routines. The former provides the fastest computation, though some (normally complex) solutions may be missed. When this happens, DGP – Solver uses general continuation to correct the outcome. As the numerical computations may be time consuming,

the program provides the possibility of parallel computation if the available hardware has a multicore processor. The integer following the string ‘NCPUs:’ specifies the number of available CPUs. As explained previously, there are two alternatives to performing computation by homotopy continuation: general homotopy continuation (GHC) or coefficient parameter homotopy continuation (PHC). The integer following the string ‘Method Type:’ is a flag giving the user the possibility to choose between these two types of continuation strategies. 0 instructs DGP – Solver to use PHC only. This option provides for the fastest computation, but some solutions (normally complex) may be missed. If the flag is specified as 1, DGP – Solver uses GHC to correct the results obtained by PHC when some solutions are missed. If the flag is specified as 2, DGP – Solver uses GHC in all cases, though computation may be very slow. The integer following the string ‘NSolMiss:’ is effective only when ‘Method-Type’ is set to 1. It may happen that PHC converges to a number of solutions that is smaller than that expected (the number of solutions with m active cables, in the complex field, is known from the previous chapter). If ‘Method-Type’ is set to 1, any time PHC misses a number of solutions equal to ‘NSolMiss’, DGP – Solver repeats computation by using GHC. This gives the user a higher chance of obtaining all solutions to the problem.

The second part of the input file contains optional settings that are passed on to Bertini. These optional settings are those that may be ordinarily specified in a standard Bertini input file, in the configuration part between the strings ‘CONFIG’ and ‘END’. For details on adjusting Bertini settings, the user may refer to the Bertini manual (Bates et al.). It is worth noting that the two parts of *RobotConfig.txt* must be separated by the string ‘CONFIG’, which is necessary even if no Bertini settings are specified. In the latter case, Bertini default values are used.

5.1.1.2 Output files

Given an $n - n$ C DPR, Bertini constructs all $m - m$ C DPRs that may be derived from the initial robot, with $m \leq n$. For any one of these, identified by the cable subset $\mathcal{W} \subseteq \{1, 2, \dots, n\}$, Bertini solves a DGP, thus obtaining a number of solutions in the complex field. At this point, *real* solutions are extracted and, for any $k \notin \mathcal{W}$, the program verifies whether $\|A_k - B_k\| \leq \rho_k$. Real solutions that meet the latter requirement are denoted as *admissible*. Clearly, among these, only those with nonnegative tensions in all active cables have physical interest. DGP – Solver finally verifies whether equilibrium configurations are *stable* or not. Stable, admissible configurations with nonnegative tension in all active cables are called *feasible*.

After computation is completed, the user may access the computation files produced by Bertini in the folder *BertiniOutput* (for more details on these files, the user is referred to Bates et al.). The main results are grouped in a number of text files contained in the folder *Results*. For each group of active cables:

- *RealSols* provides all real solutions;
- *AdmissibleSols* provides all admissible solutions;
- *PositiveTensionSols* provides all admissible solutions with nonnegative tensions in the cables;

- *FeasibleSols* provides all feasible solutions;
- *Summary* provides a prospect with the number of real solutions, admissible solutions, admissible solutions with nonnegative cable tensions and feasible solutions.

Each group of active cables is identified by a sequence of digits, where the first digit is the number, m , of active cables and the other digits are the indexes identifying the cables. For instance, the sequence 3 – 245 identifies a group of 3 cables composed of the 2nd, 4th and 5th cables.

A solution with m active cables and $m > 1$ is given as:

$$S \quad x \quad y \quad z \quad R_{11} \quad R_{12} \quad R_{13} \quad R_{21} \quad R_{22} \quad R_{23} \quad R_{31} \quad R_{32} \quad R_{33} \quad \tau_1 \dots \tau_m \quad (5.15)$$

where $S \in \{-3, -2, -1, 0\}$, x, y, z are the coordinates of the origin of the mobile frame in the fixed frame, R_{hk} is the element in the h th row and k th column of the rotation matrix between the mobile and fixed frames and τ_i is the tensile force in the i th cable, with $i = 1, \dots, m$. S is a flag with the following meaning:

- if $S = -3$: the solution is complex;
- if $S = -2$: the solution is real but not admissible;
- if $S = -1$: the solution is admissible;
- if $S = 0$: the solution is admissible and all cables have nonnegative tensions;
- if $S = 1$: the solution is feasible.

When $m = 1$, the solution is provided as:

$$S \quad x \quad y \quad z \quad (5.16)$$

where S has the same meaning as before, while x, y, z are the coordinates of point G in the fixed frame. The file only lists the equilibrium configuration for which A, B and G are arranged in this order in the direction of the load, as this is the only feasible configuration. The position of the origin of the mobile frame and the rotation matrix are not provided, as in this case, the platform is free to rotate about the line ABG .

The results provided by DGP – Solver for an exemplifying 2-cable CDPR are given in Table 5.2. These results have been extracted from samples that will be discussed in Section 5.2. Due to space limitations, only real solutions with nonnegative tension in all cables are reported. The platform orientation is specified by Rodrigues parameters. As shown in Fig. 5.3, the robot has 6 configurations with all two cables in tension, among which only the first configuration is stable. Due to the simplicity of 2-cable robot configurations, the fact that only the first configuration is stable may be inferred simply from the figures.

In the same manner, Tables 5.3 and 5.4 show the results provided by DGP – Solver for two exemplifying CDPRs with 4 and 5 cables, respectively. The 4-cable robot in Table 5.3 has 2 configurations

Table 5.2: DGP of a CDPR with 2 cables: real solutions with nonnegative tension in all cables

Geometric dimensions and load: $\mathbf{a}_1 = [3.823126, 8.816128, 1.541947]$, $\mathbf{a}_2 = [-6.213115, 7.171288, 0.506044]$, $\mathbf{b}_1 = [1.962229, 3.138447, 1.186822]$, $\mathbf{b}_2 = [-2.104132, 2.813451, 1.176657]$, $(\rho_1, \rho_2) = (10.293260, 11.095270)$, $Q = 10$.

Conf.	(x, y, z)	(e_0, e_1, e_2, e_3)	(τ_1, τ_2)	\mathbf{H}_r
1	-1.6275, 7.9228, 14.4645	1, -1.4622, -0.1857, 0.0264	4.66, 5.77,	>
2	3.3446, 8.7377, 7.9420	1, 0.7157, -0.2150, 0.2468	9.49, 0.63,	<>
3	-5.9784, 7.2098, 7.8948	1, 0.6378, 0.3441, -0.1366	0.28, 9.79,	<>
4	-1.0972, 8.0097, 12.1912	1, -3.2387, 28.1621, -19.2119	6.44, 6.98,	<>
5	-1.0388, 8.0193, 8.0716	1, 0.6787, 0.0506, 0.0646	5.22, 5.21,	<>
6	-1.0159, 8.0231, 5.7940	1, 1.6887, -14.6618, -21.7249	6.50, 6.92,	<>

Table 5.3: DGP of a CDPR with 4 cables: real solutions with nonnegative tension in all cables

Geometric dimensions and load: $\mathbf{a}_1 = [10.830397, -0.462135, -0.418191]$, $\mathbf{a}_2 = [3.074004, 10.213429, 1.739355]$, $\mathbf{a}_3 = [-8.791518, -1.292838, 0.296207]$, $\mathbf{a}_4 = [4.717787, -7.113540, -0.448884]$, $\mathbf{b}_1 = [3.760649, 0.232910, 1.993820]$, $\mathbf{b}_2 = [1.637005, 2.644740, 1.452987]$, $\mathbf{b}_3 = [-3.496261, -0.119685, 1.373177]$, $\mathbf{b}_4 = [1.313785, -3.240876, 1.114063]$, $(\rho_1, \rho_2, \rho_3, \rho_4) = (14.900603, 15.459668, 11.310118, 13.058653)$, $Q = 10$.

Conf.	(x, y, z)	(e_0, e_1, e_2, e_3)	$(\tau_1, \tau_2, \tau_3, \tau_4)$	\mathbf{H}_r
1	-0.1362, -0.9264, 9.7358	1, 0.0190, -0.0905, -0.0005	5.25, 0, 6.11, 0,	<>
2	-1.4177, -4.4700, 8.9256	1, -0.5549, 0.1478, 0.0364	0, 0, 4.87, 5.91,	<>
3	2.1927, -5.0136, 8.9574	1, -0.9246, -0.9035, -0.4648	1.33, 0, 4.65, 7.59,	<>
4	0.0743, -1.6315, 9.7820	1, -0.1070, -0.1094, -0.0116	4.43, 0, 5.65, 1.22,	>
5	2.5358, -6.0922, 8.6514	1, -0.7649, -0.2370, -0.1887	0, 0.10, 2.38, 8.66,	<>
6	-0.9995, 2.5837, 11.2117	1, 1.0384, 0.0012, 0.1941	0.37, 4.80, 4.41, 2.87,	>

with only two cables in tension, both unstable, 3 configurations with 3 cables in tension and only one configuration in which all 4 cables are in tension. As seen in Fig 5.4, it is almost impossible to judge the stability of the 4-cable robot configuration simply by looking at the schematic models. This fact shows the necessity of an efficient mathematic procedure to perform stability analysis. The 5-cable robot in Table 5.4 has 9 unstable configurations with two, three or four cables in tension and 4 stable configurations, of which half have four cables in tension and the other half have five cables in tension. When the DGP admits multiple feasible solutions, the robot may switch (due to inertial forces or external disturbances) across portions of the configuration space characterised by different numbers of taut cables, thus bringing the end-effector onto unpredicted trajectories. Accordingly, the computation of the complete set of equilibrium configurations is essential for robust trajectory planning. This motivates and gives relevance to the algorithms presented in this thesis.

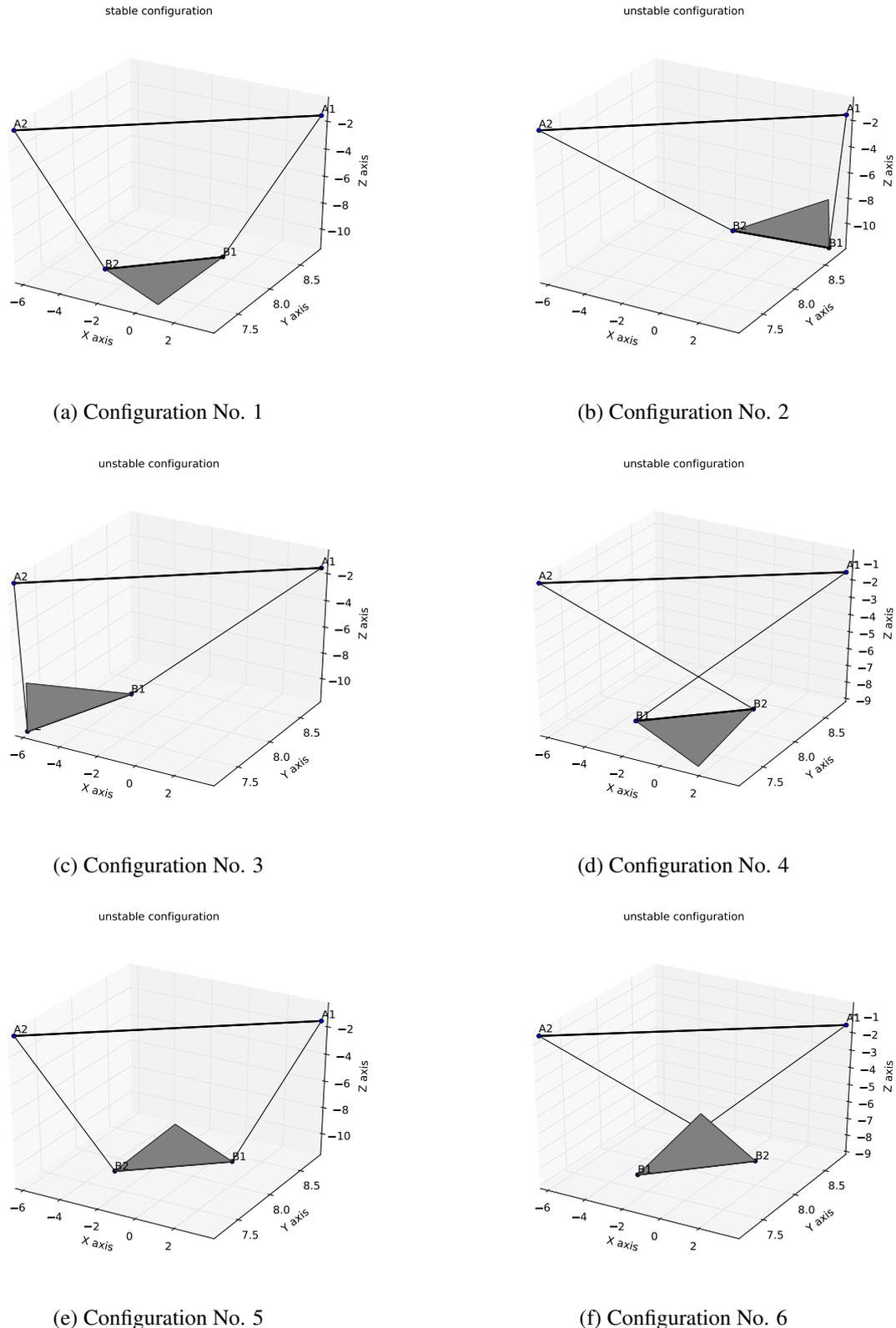


Figure 5.3: Schematic configurations of DGP solutions for 2-cable robot reported in Table 5.2

5.2 Case study

A robot with n cables is designed to control n DOFs of the platform; however, depending on the configuration and the load, only m cables may be active, with $m < n$. Slack cables contribute to neither

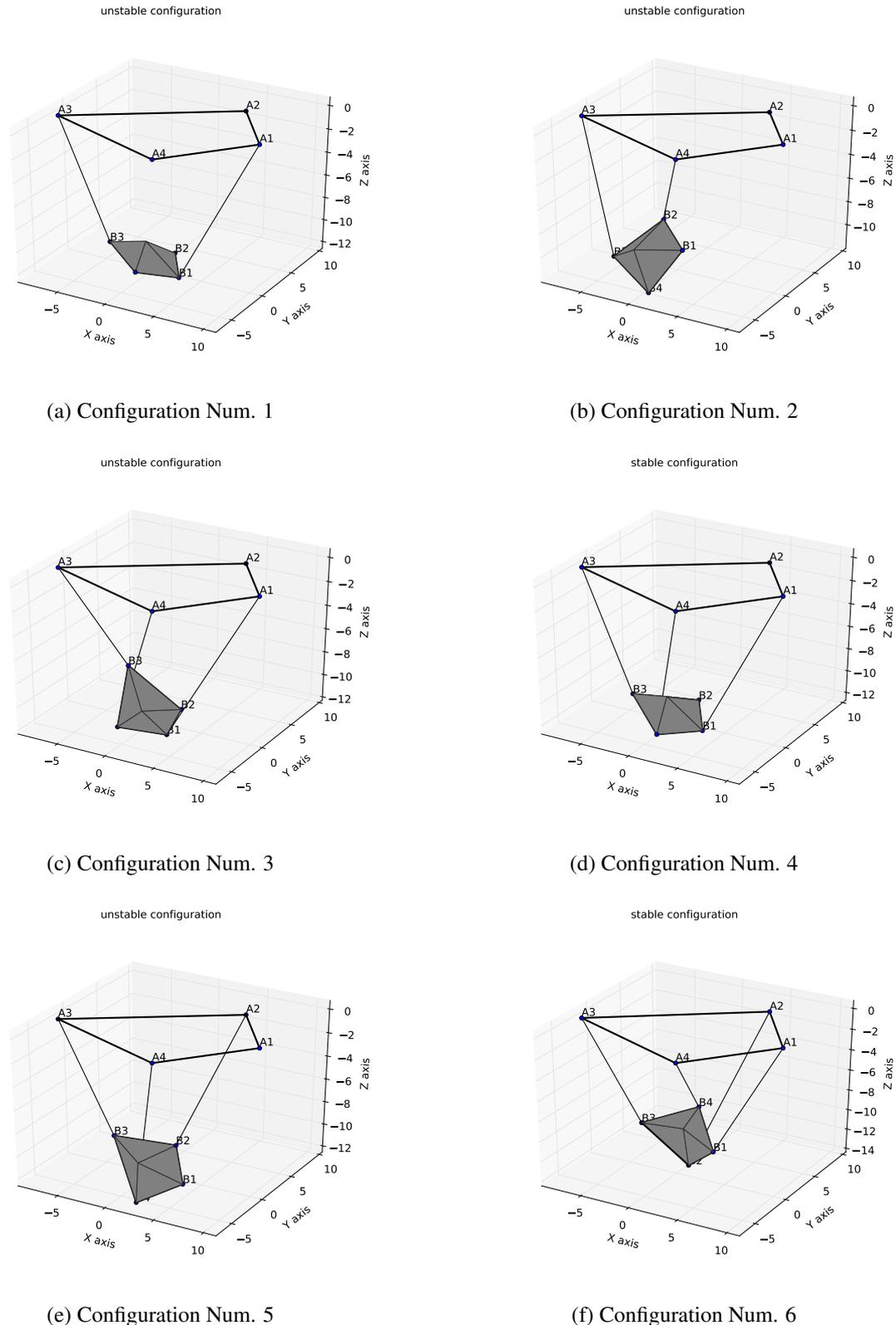


Figure 5.4: Schematic configurations of DGP solutions for robot with 4 cables reported in Table 5.3

controlling the platform pose nor sustaining the load and are thus instantaneously ineffectual. This amounts to a loss of robot capacity. Investigating the most suitable architectures and geometries that guarantee the optimal exploitation of the available actuators is necessary. A stochastic investigation is

Table 5.4: DGP of a CDPR with 5 cables: real solutions with nonnegative tension in all cables

Geometric dimensions and load: $\mathbf{a}_1 = [8.762837, -1.064001, -0.715711]$, $\mathbf{a}_2 = [6.732934, 8.223691, 0.187221]$, $\mathbf{a}_3 = [-5.094292, 7.798299, 1.258330]$, $\mathbf{a}_4 = [-11.189309, -0.138832, -1.614663]$, $\mathbf{a}_5 = [4.870417, -8.101810, 0.176616]$, $\mathbf{b}_1 = [3.889210, 0.116354, 1.549903]$, $\mathbf{b}_2 = [1.262183, 3.258968, 1.975178]$, $\mathbf{b}_3 = [-1.533217, 2.692867, 1.915029]$, $\mathbf{b}_4 = [-3.991803, 0.020197, 1.186929]$, $\mathbf{b}_5 = [1.257342, -3.138215, 1.992397]$, $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5) = (14.772834, 12.755196, 13.153812, 13.969011, 15.712028)$, $Q = 10$.

Conf.	(x, y, z)	(e_0, e_1, e_2, e_3)	$(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$	\mathbf{H}_r
1	(-1.8836, -0.5703, 3.6644)	(1, 1.9122, 103.4291, -2.6272)	(20.26, 0, 0, 20.86, 0)	<>
2	(-0.9683, 4.6303, 8.1748)	(1, 0.4483, 0.1963, 0.0244)	(0, 6.19, 0, 5.44, 0)	<>
3	(-0.5205, 0.5002, 9.6903)	(1, -22.1474, -12.1484, -1.3686)	(0, 0, 10.24, 0, 10.34)	<>
4	(-2.2633, 3.8189, 4.8682)	(1, 14.9241, -32.9020, 16.6404)	(0.47, 21.60, 0, 22.88, 0)	<>
5	(-2.1884, 2.2735, 4.9241)	(1, 4131.8466, -12513.9896, 3408.9760)	(4.72, 13.77, 0, 18.94, 0)	<>
6	(1.5688, -3.0697, 10.5834)	(1, -0.5711, -0.3564, -0.1105)	(1.10, 1.85, 0, 3.66, 5.40)	<>
7	(1.3853, -2.5305, 10.5848)	(1, -0.4701, -0.2959, -0.0865)	(1.22, 2.02, 0, 3.72, 4.87)	>
8	(-1.7403, 0.4235, 10.0329)	(1, 24.6125, 21.5424, 3.9457)	(1.44, 0, 6.64, 2.85, 7.07)	<>
9	(-2.5658, 2.0842, 10.1100)	(1, 2.8702, 2.2467, 0.5195)	(0, 1.62, 5.36, 3.23, 6.15)	<>
10	(1.5754, -2.4698, 10.6232)	(1, -1.6897, -0.3362, -0.1492)	(0, 3.75, 0.50, 3.17, 6.71)	>
11	(1.5476, -3.5523, 10.5849)	(1, -1.0357, -0.4345, -0.1603)	(0, 2.02, 1.14, 2.90, 7.12)	<>
12	(-2.6029, 1.9238, 10.1101)	(1, 3.0172, 2.4254, 0.5732)	(0.02, 1.54, 5.32, 3.34, 6.24)	>
13	(1.5460, -3.4460, 10.6187)	(1, -0.9363, -0.6196, -0.1883)	(0.60, 1.70, 0.77, 3.52, 6.53)	>

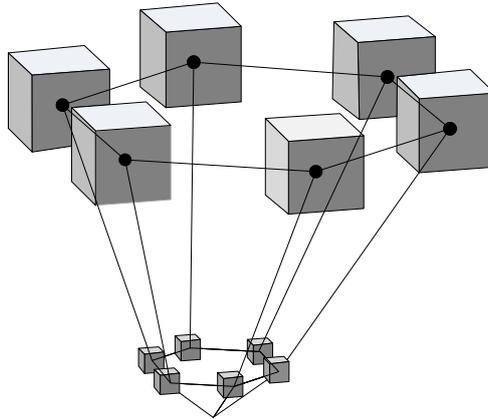


Figure 5.5: Schematic of the robot samples used for the stochastic investigation.

reported hereafter providing preliminary ground in this respect.

Two scenarios are envisaged. In one case, the robot geometry is established and the platform configuration and load orientation are changed. In the other, the load is assigned and the robot geometry and configuration are varied. Since the two scenarios are equivalent, the latter will be described here within. For the sake of simplicity, the anchor points on the base and the platform are chosen inside cubes having centre-points located on the vertices of regular hexagons (Fig. 5.5). The circumradii of the base and platform hexagons are 10 and 3.5, respectively; whereas the side lengths of the base and platform cubes are 4 and 1, respectively. G is located on the line perpendicular to the platform hexagon that passes through its circumcircle, at a distance of 1.5 from the latter. Cable lengths are varied over the interval $[10, 16]$. 500 robot configurations are randomly chosen within the above limits, thus generating

500 samples of 6-cable robots. For each sample, DGP – Solver computes the overall solution set, \mathcal{S} . The latter comprises all solutions of the DGP such that $\{\| \mathbf{s}_i \| \leq \rho_i, i = 1 \dots 6\}$, where the equality holds for taut cables and the inequality for slack cables. By ignoring the constraint $\| \mathbf{s}_i \| \leq \rho_i$ for $6 - k$ cables, $500 \binom{6}{k}$ additional samples of k -cable robots with $k = 2 \dots 5$ may be easily extracted from the DGP – Solver computation¹.

Accordingly, a large sample of robots with m cables, $m < n$, may be derived from the same simulations. For example, if $\mathcal{W} \subseteq \{1, 2, 3, 4, 5\}$, enforcing $\| A_6 - B_6 \| \leq \rho_6$ provides an admissible configuration of the 6-cable robot with only 5 cables in tension; whereas neglecting it provides an admissible configuration of the 5-cable robot with all cables in tension. In particular, among the 500 sample 6-cable CDPR geometries, 7500 robots with 2 cables, 10000 robots with 3 cables, 7500 robots with 4 cables and 3000 robots with 5 cables are derived. The results of the simulation are summarised in Table 5.5. The n th column, with $n = 2 \dots 6$, reports the results obtained for the samples suspended by n cables and, in particular, the number of samples considered (row 1), the number of feasible configurations globally obtained (row 2) and the number of feasible configurations with m cables in tension, $m \leq n$ (row 2 + m).

The data emerging from the table appears to show that, when the number of cables increases, the probability of finding feasible configurations with all cables in tension decreases. For the robots with 2 cables, almost 90% of equilibrium configurations have both cables as active, whereas for the robots with 3 cables, all cables are active in roughly 80% of cases. This trend continues as n increases, until, for the samples with 6 cables, almost no configurations are present with all 6 cables being active. Columns 4 through 6 show that, in most cases, the robots with 4 cables have only 3 taut cables, whereas the robots with 5 and 6 cables have only 3 or 4 cables in tension. It is worth noting that, while equilibrium configurations with slack cables may only occur at the frontier of the geometric workspace² for robots with 2 or 3 cables, this is not the case when $n \geq 4$, where equilibrium configurations with slack cables may occur in the middle of the workspace.

Another interesting issue concerns the probability of a CDPR admitting multiple feasible configurations. When this occurs, the stable equilibrium pose of the platform may change under the influence of external perturbations, which is not favourable in practice. In this context, Table 5.6 investigates the number of multiple solutions encountered when solving the DGP of the samples in Table 5.5. Rows 2, 3 and 4 report the percentages of samples admitting 1, 2 or more feasible solutions, respectively. It is evident that, when the number of cables increases, the probability of having a single solution of the DGP decreases. It is roughly 83% for the 3-cable robots, 53% for the 4-cable samples and a little less than 25% for the 5- and the 6-cable robots.

From the data reported in Tables 5.5 and 5.6, one could argue that, as long as only 3 cables are employed, all of them may reasonably be expected to support the platform with a single stable equilibrium pose being unambiguously determined in most circumstances. When 4, 5 or 6 cables are used,

¹ As explained in Section 5.1, DGP – Solver provides the DGP solutions for all possible constraint sets $\mathcal{C}_{\mathcal{W}} = \{\| \mathbf{s}_j \| = \rho_j, \tau_j > 0, j \in \mathcal{W}\}$, with $\mathcal{W} \subseteq \{1, 2, 3, 4, 5, 6\}$, $m = \text{card}(\mathcal{W}) \leq 6$ and $\{\| \mathbf{s}_k \| \leq \rho_k, \forall k \notin \mathcal{W}\}$. By ignoring the latter requirement, the DGP solutions for $\mathcal{C}_{\mathcal{W}}$ provide the equilibrium configurations of a robot including only m cables.

² When a single cable is taut, the external wrench, $Q\mathcal{L}_e$, must be aligned with it. When $n = 3$ and only two cables are taut, the latter and $Q\mathcal{L}_e$ must be coplanar (Carricato [2013b]; Carricato and Merlet [2013]).

full control of the robot becomes challenging, as it is difficult to take advantage of all available cables. Controlling the platform pose in a deterministic way in this case is therefore not simple.

Table 5.5: Stochastic analysis of random samples of robot geometries with 2, 3, 4, 5 and 6 cables.

n - n CDPRs	2-2 CDPRs	3-3 CDPRs	4-4 CDPRs	5-5 CDPRs	6-6 CDPRs
No. of samples	7500	10000	7500	3000	500
No. of feasible solutions	7500	12048	12006	5912	1114
No. of feasible solutions with 1 active cable	785(10.46%)	70(0.58%)	0	0	0
No. of feasible solutions with 2 active cables	6715(89.53%)	2424(20.11%)	109(0.90%)	3(0.05%)	0
No. of feasible solutions with 3 active cables	–	9554(79.29%)	6987(58.19%)	2362(39.95%)	380(34.11%)
No. of feasible solutions with 4 active cables	–	–	4910(40.89%)	2626(44.41%)	420(37.70%)
No. of feasible solutions with 5 active cables	–	–	–	921(15.57%)	312(28.00%)
No. of feasible solutions with 6 active cables	–	–	–	–	2(0.17%)

Table 5.6: Distribution of multiple feasible configurations as emerging from stochastic analysis.

n - n CDPRs	2-2 CDPRs	3-3 CDPRs	4-4 CDPRs	5-5 CDPRs	6-6 CDPRs
No. of samples	7500	10000	7500	4000	500
No. of cases with 1 feasible solutions	7500(100%)	8320(83.20%)	3941(52.54%)	988(24.70%)	113(22.60%)
No. of cases with 2 feasible solutions	0	1299(12.99%)	2754(36.72%)	1276(31.90%)	213(42.60%)
No. of cases with more than 2 feasible solutions	0	381(3.81%)	805(10.73%)	736(18.40%)	174(34.80%)

Chapter 6

Conclusions

In this dissertation, the geometric static problem of under-constrained cable-driven parallel robots (CD-PRs) supported by n cables, with $n \leq 6$, was studied. The task consisted of finding the overall equilibrium configurations of a robot subject to both geometric and static constraints; namely, with n robot variables and a static load assigned. The kinematics and statics are coupled and must be solved simultaneously, which is a major challenge. When n constraints are imposed on the moving platform, an inverse geometric static problem is concerned, which consists of computing the overall pose of the platform, the cable lengths and the cable tensions. The direct geometric static problem consists of determining the platform posture and cable tensions when the n cable lengths are assigned. In order to propose a sufficient solution strategy, the problem was modelled by a set of algebraic equations in Chapter 3.

In Chapter 4, least-degree univariate polynomials in the ideal corresponding to particular IGPs and DGPs were found for any value of n by using an appropriate elimination procedure, thus setting an exact bound on the number of solutions admitted in the complex field. The IGP was less complex and, by introducing some ingenious strategies, was solved with the attainment of low-order univariate polynomials. The most challenging tasks were associated with the DGP for $n = 3, 4$ and 5 , where the aforementioned polynomials had degrees as high as 156, 216 and 140, respectively. Though these polynomials proved to be too large to be numerically useful, they provided meaningful benchmarks to test the effectiveness of an innovative variable-elimination procedure. It was proven that this procedure may succeed when other methods either fail or are too computationally onerous, thus providing an efficient alternative in calculating a least-degree univariate polynomial in a given ideal; a classic challenge in robot analysis and synthesis. For the numerical computation of the solution set, a numerical procedure based on homotopy continuation was developed. As discussed in Chapter 4, this is a path-tracking technique that transforms a start system of polynomial equations with known solutions to a target system whose solutions must be found. In general, homotopy continuation, with no information about the roots of the target system, sees the construction of a start system yielding the maximum possible number of solutions, equal to the Bezout number. Converting the start system of the equation to the target one and tracking all solution paths, many such paths diverge to infinity, while only a limited number converge to finite solutions. Usually this number of paths is much larger than the finite solutions of the target system of equations in the complex field and, as a result, tracking

diverging paths causes significant and non-beneficial computational burden. It has been demonstrated that the Bezout number and corresponding number of tracked paths may be significantly smaller if the corresponding system of equations are multi-homogenised, yielding faster algorithm execution. On the other hand, when all nonsingular isolated roots of solutions to a general member of a family of equations are known, ‘coefficient-parameter’ homotopy continuation may be implemented to find solutions for any other robots of the same family in a more efficient way. Accordingly, by a suitable homotopy, only the paths originating at the isolated roots of the start system were tracked; whereas those corresponding to solutions going to infinity were ignored.

Another challenge addressed in this thesis was determination of the maximum number of real solutions admitted by the studied systems. The solutions of each problem may be complex or real, but only the latter have physical interest. By varying robot parameters, the number of real roots varies. Due to the very high number of solutions attained during the direct displacement analysis of under-constrained CDPRs, a systematic procedure to find a set of robot parameters with the maximum number of real solutions was necessary. Accordingly, a procedure borrowed from the Dietmaier was introduced in Section 2.3 to find sets of CDPR parameters for which the DGP provided the highest number of real configurations. Dietmaier’s algorithm, like the homotopy method, is a continuation procedure in which a start system is defined for a given set of robot parameters and an iterative procedure is established to change the system parameters and conveniently vary the solution set. In contrast with parameter homotopy, however, the target parameters are unknown *a priori*. The tracked path is adaptively modified in such a way that, at each iteration, the imaginary parts of some complex solutions are decreased and eventually as many complex roots as possible are transformed into real roots. By applying this technique on the DGP of CDPRs with 3, 4 and 5 cables, the maximum number of real solutions obtained for each case was 54, 98 and 74, respectively.

By taking advantage of the homotopy continuation procedure, a program DGP – Solver was composed, which is freely available. The code receives the robot geometry, cable lengths and external load as inputs and computes, under the assumption that cables are inextensible and massless, all possible equilibrium configurations of the robot that are compatible with the given constraints. Distinctive features of the code are that it finds all solutions of the problem, including those with slack cables, and assesses the feasibility of each configuration with an integrated stability analysis.

It was shown that the DGP may admit multiple equilibrium configurations, characterised by different numbers of taut cables. Since slack cables contribute to neither controlling the platform pose nor sustaining the load, they represent a loss of capacity of the robot, which is instantaneously unable to control some of the end-effector degrees of freedom. A preliminary investigation was performed to find the most suitable architectures that guarantee an optimal exploitation of the available actuators. The probability of a CDPR admitting multiple stable equilibrium configurations (which is a critical situation for reliable robot control) was also considered. From the data reported in Chapter 5, one could argue that, as long as only 3 cables are employed, all of them may be reasonably expected to support the platform, with a single stable equilibrium pose being unambiguously determined in most circumstances.

Appendix A

In this section, numerical examples of geometric parameters admitting the maximum number of real solutions for robots with 3, 4 and 5 cables are provided. These examples are the results of the Dietmaier procedure outlined in Section [4.3](#) .

Table 1: All real solutions of DGP of a 3-3 CDPR with maximum number of 54 real solutions

Geometric dimensions and load: $\mathbf{a}_1 = [0, 0, 0]$, $\mathbf{a}_2 = [0.897440, 0, -0.726510]$, $\mathbf{a}_3 = [0.656710, 0.746360, -0.590910]$, $\mathbf{b}_1 = [1.556650, 0, 0]$, $\mathbf{b}_2 = [0.696950, 0.784290, 0]$, $\mathbf{b}_3 = [-0.464410, -1.389800, -1.199480]$, $(\rho_1, \rho_2, \rho_3) = (1, 1.080500, 2.602500)$, $Q = 10$.

Conf.	(x, y, z)	(e_1, e_2, e_3)	(τ_1, τ_2, τ_3)
1	0.0634, 0.1736, -0.5293	-0.5534, -0.9854, -0.8871	10.54, -1.21, -2.76
2	0.9948, -0.0455, -2.1045	1.0390, -0.9800, 1.2955	-9.48, -8.99, 2.33
3	0.5102, -0.1635, -0.5159	-2.1568, -2.2788, -1.9020	4.74, 6.43, 3.13
4	1.4323, 1.0452, -1.6514	2.4261, -2.8663, 3.6663	-50.28, -5.73, 49.47
5	0.2508, -0.0479, -0.6094	-0.2969, -1.0008, -0.0929	6.22, 7.93, 7.49
6	0.6483, 1.1466, -1.8930	-0.8594, -0.3795, -1.1743	-19.92, 14.07, -12.01
7	0.8516, -0.0142, -0.6959	0.1903, -1.3668, -0.5069	0.75, 11.32, 3.08
8	1.1117, -0.1800, -0.4811	6.0423, 13.3840, 4.2951	8.82, 15.20, 19.36
9	0.3744, -0.0308, -0.5111	-1.0816, -1.3568, -1.2971	5.90, 4.25, 0.53
10	0.6098, 0.0192, -1.3549	-0.1640, -0.8795, 0.2065	4.13, -17.24, -16.32
11	0.9362, -0.0509, -1.6747	-0.5647, -3.6481, -0.2073	0.22, -16.05, -13.65
12	0.6129, 0.6724, 0.9651	41.9368, -0.5618, -44.3965	-6.59, 0.34, 7.95
13	0.8858, -0.4903, -0.4509	3.2461, 4.1824, 2.4811	7.60, 12.71, 10.87
14	0.5614, 0.9292, 0.2029	-2.1638, 2.1129, 0.0571	4.71, 6.89, 10.66
15	1.2810, 0.0257, -0.5133	1.1515, 6.1838, 2.4650	5.59, 22.92, 21.84
16	0.8983, -0.0179, -0.6935	0.1435, -1.3928, -0.8091	0.11, 10.46, 0.92
17	0.9443, 0.0077, -2.1090	0.9186, -1.1658, 0.9016	-6.21, -8.26, -1.22
18	0.5369, 0.7693, -1.4696	0.0993, -0.6693, -0.3427	2.19, -5.25, -12.46
19	2.4401, 0.6365, -0.0380	-0.1171, 2.9441, -5.6330	-33.47, 26.48, 8.84
20	0.6796, 0.5035, 0.4579	-1.9056, 7.1397, -7.5311	13.67, -11.31, 6.92
21	0.4697, 0.4552, 0.0534	-0.7309, 2.0689, -0.7835	-3.12, -0.96, 8.20
22	0.5715, 0.7385, 0.1670	-3.1734, 3.2184, -0.9303	1.89, 2.70, 8.87
23	0.8954, 0.1520, 1.1534	0.0756, 0.9074, -1.2974	-1.40, 7.54, 4.01
24	0.8223, 0.6116, -1.6669	-0.1435, -2.9073, -1.4004	1.17, -12.89, -16.32
25	0.6759, 0.4964, 1.0546	-7.6257, -3.8442, 14.9347	-2.93, 6.20, 5.79
26	0.7150, -0.3392, 0.2513	-0.0682, -4.2850, 3.2341	39.94, -33.98, -3.74
27	-0.6475, 0.0685, 0.2574	1.2351, 0.4542, -0.5850	-26.32, -16.34, -1.68
28	0.8120, 0.3450, 1.0225	3.6139, 1.2486, -3.7415	-2.95, 4.97, 4.64
29	0.0000, 0.0003, -0.5566	-2.3817, -0.9993, -2.3831	10.00, -0.01, -0.00
30	0.4932, 0.0208, -0.6134	-0.5848, -1.4390, -0.2065	4.63, 9.31, 10.44
31	1.0589, -0.4306, -0.4556	1.8378, 3.0461, 2.0388	5.60, 14.99, 9.91
32	1.5486, 0.4002, -1.7581	-1.0092, -1.1007, -1.8713	-23.04, 20.76, -4.92
33	1.6482, 1.5973, -0.6567	0.3694, -0.9632, -2.6851	59.39, 2.61, -66.98
34	0.6244, 0.3345, 0.9358	0.4628, 3.4054, -5.7540	-11.53, 15.27, 5.13
35	0.2263, -0.0531, -0.6094	-0.2882, -0.9769, -0.0852	6.43, 7.76, 7.32
36	0.7741, 0.8653, -1.9269	-0.8376, -0.8822, -1.0477	-7.99, 1.73, -9.57

37	0.6231, 0.9516, 0.1995	-1.5319, 1.7594, -0.0227	3.32, 5.05, 11.57
38	0.8369, -0.1204, 0.8292	0.9606, 1.2807, -0.8110	-10.10, 8.23, -4.08
39	1.2552, 0.6976, 0.0004	-6.3727, 9.9899, 5.0805	14.88, 25.43, 9.02
40	2.0431, -0.3251, -1.1174	-0.3020, -4.9995, 3.7384	33.36, -41.90, -3.96
41	0.8970, 0.3504, -1.6633	-0.2785, -3.4817, -1.0557	1.08, -16.97, -17.98
42	2.3001, 0.9783, -0.0311	0.2733, 0.6574, -6.8637	-37.69, 13.64, 27.52
43	2.3215, 0.9513, -0.0311	0.2128, 0.8485, -6.9045	-35.89, 14.32, 25.12
44	0.0102, -0.1188, -0.6065	-0.2191, -0.7297, 0.0307	8.21, 9.26, 8.91
45	0.0089, -0.1397, -0.5596	-4.1698, -1.6030, -3.1349	8.71, 2.54, 2.19
46	-0.4270, -0.6998, 0.0563	-9.6174, -3.5104, 4.3730	-28.59, -4.33, -17.74
47	0.7234, 0.3059, -2.2934	-67.4748, 3.0846, -71.3742	-3.98, -4.75, -3.64
48	0.3095, 0.2323, 0.9427	0.9028, 2.2638, -3.8926	-8.45, 10.71, 4.37
49	-0.5698, 0.4582, -0.0776	-5.6647, 0.6000, 0.2701	29.16, 23.17, 13.96
50	1.1079, 0.4296, 0.9547	0.1429, 8.0783, -15.7334	-7.48, 13.08, 5.55
51	0.7204, 0.7007, 0.1753	0.3915, 4.0488, -2.7628	0.56, -3.00, 10.42
52	0.9818, -0.3338, -1.3954	-0.3452, -1.2753, 0.7052	4.28, -15.57, -8.41
53	0.3393, -0.6917, -0.4700	3.7171, 2.5379, 1.7731	13.38, 10.81, 11.71
54	0.3022, 0.1565, -0.9638	-1.1499, -3.1127, -3.8924	-6.86, 10.82, -1.96

Table 2: All real solutions of DGP of a 4-4 CDPR with maximum number of 98 real solutions

Geometric dimensions and load: $\mathbf{a}_1 = [0, 0, 0]$, $\mathbf{a}_2 = [-0.760549, 0, 0.909316]$, $\mathbf{a}_3 = [-0.716460, 0.680475, 0.079703]$, $\mathbf{a}_4 = [0.025744, 0.704210, 0.873897]$, $\mathbf{b}_1 = [16.548180, 0, 0]$, $\mathbf{b}_2 = [17.163583, 0.777200, 0]$, $\mathbf{b}_3 = [16.222490, 0.751536, 0.768740]$, $\mathbf{b}_4 = [17.386036, 0.095675, 0.817269]$, $(\rho_1, \rho_2, \rho_3, \rho_4) = (1, -1.014940, -1.121710, 1.138240)$, $Q = 10$.

Conf.	(x, y, z)	(e_1, e_2, e_3)	$(\tau_1, \tau_2, \tau_3, \tau_4)$
1	-3.0636, 3.6998, -16.5218	1.2350, -0.9881, 0.8529	28.40, 27.84, 28.89, 28.20,
2	15.5132, 2.5903, -4.9524	1.7397, -8.0431, 7.1683	170.86, -97.16, -112.86, 162.36,
3	13.1838, -8.2574, -3.5469	-0.0401, -1.3812, 2.6625	102.30, -307.78, -337.65, 34.17,
4	11.5670, 6.2772, -9.4221	-3.1177, 1.9410, -8.3627	-741.38, -12.62, 723.44, 128.59,
5	-11.6506, 0.2819, -11.4601	6.8831, -0.2595, 2.9118	-123.21, -115.84, 147.87, -14.07,
6	-10.4613, 11.3382, -8.0156	-4.1399, 1.3489, -1.6696	36.85, -104.40, 186.37, 87.31,
7	10.9196, -11.7787, -1.5815	-30.6642, -70.4630, -12.3521	106.85, -32.59, -268.48, 263.88,
8	-8.6254, -13.1375, -4.6443	3.4338, 1.7411, 1.2351	1101.60, -190.69, -1096.86, 1.21,
9	0.4579, -0.2077, 17.2784	-0.9776, 1.0344, 0.9627	10.16, 0.67, 5.46, -7.69,
10	-7.5948, -12.6090, -6.9715	3.1084, 1.4708, 1.4897	-1025.22, 19.34, 1014.02, 176.33,
11	-9.1237, 0.8618, 14.7504	-0.2707, 0.5708, 0.0957	-167.87, 262.52, 74.43, 93.20,
12	-1.2968, 2.9696, -16.5224	1.2218, -1.0292, 0.9594	23.34, 23.20, 21.14, 21.91,
13	-11.0831, -7.8604, -9.5570	-43.5416, -14.2111, -15.2502	-301.63, 126.92, 287.66, 231.30,
14	-0.7078, 0.3997, 17.3022	-0.8216, 1.0171, 0.7931	3.00, 3.46, -3.42, -2.46,
15	13.3972, -1.1048, -9.7319	0.0425, -2.7658, 0.3565	12.82, 287.02, -114.72, -241.29,
16	-11.6161, 1.7051, -11.4871	19.1728, -0.9528, 8.0644	-88.24, -80.58, 139.66, 20.18,
17	-9.9961, -2.9112, 14.0549	-2.3816, 0.2279, 1.3161	-245.54, 207.84, -70.40, -24.45,
18	-16.9518, 0.3842, 1.2556	2.4621, 0.0476, -0.0431	333.77, -49.79, -115.57, 281.65,
19	-16.9147, 2.8563, 1.8056	1.2138, -0.0294, -0.1204	172.32, -121.63, -107.24, 173.98,
20	10.0205, 13.1293, -5.4691	1.3368, -3.1248, -0.6672	71.81, 203.92, -134.34, -1.42,
21	13.5463, 8.6135, -4.8598	-19.7447, 49.3021, -38.2869	281.05, -39.41, -109.85, 252.90,
22	11.2216, -9.9730, -6.4402	1.2434, 1.4742, 3.4988	-191.86, -102.77, 28.30, 69.28,
23	11.8391, -8.3162, -7.7833	-0.4730, -2.4900, 0.7901	226.98, -1106.38, 41.91, 1115.05,
24	-1.2041, -0.4941, 17.2804	-0.9000, 0.9551, 0.8716	5.26, 12.60, -1.54, 6.27,
25	14.7591, 6.2906, -4.8544	4.2253, -14.6369, 8.9408	261.35, -81.21, -64.15, 272.76,
26	3.4207, -15.5757, -1.7527	3.7214, 4.6109, 1.9663	276.39, 58.21, -434.59, 298.79,
27	-5.4436, 16.2817, -1.3017	-0.0318, -0.0510, -0.7501	189.51, -256.56, -67.01, 163.51,
28	-11.4917, -12.2734, 0.1200	1.0395, 0.4993, 0.4527	263.16, -398.88, -508.04, -63.00,
29	-1.7881, -11.4731, -12.0581	-0.4036, -0.9176, 0.3017	29.77, 103.85, -75.37, -147.94,
30	-1.8657, 11.1139, 13.3556	-0.7322, 1.1125, -0.0938	-361.45, 160.02, -134.48, -209.23,
31	3.2997, 10.6518, 13.3938	14.2253, -9.4774, -13.8594	-575.10, -308.78, 601.30, -1117.08,
32	7.9368, -13.9365, -1.5610	18.0074, 31.2452, 7.0112	136.81, -23.94, -267.28, 233.74,
33	-1.2579, 2.3595, -16.5964	1.1209, -1.0636, 0.9813	16.20, 16.01, 18.16, 16.72,
34	11.6118, -7.0364, -9.2964	-0.3474, -2.3354, 0.5990	-54.28, 1011.71, -204.69, -989.87,

35	-3.6200, -3.1202, -14.9369	0.7934, -0.6334, 0.8057	-24.18, -36.99, -34.93, -20.47,
36	-3.5109, 1.7287, 16.9889	-1.0206, 0.8807, 0.7744	-35.51, 19.94, -39.87, 9.98,
37	9.8118, -11.5275, -3.6059	0.6629, 0.7295, 2.4030	-187.54, -88.75, -41.68, 120.15,
38	1.0196, -0.7691, -16.9923	-0.9582, -1.1682, -1.0171	-13.62, 19.69, -11.30, 6.12,
39	2.3010, -12.2175, 11.1641	0.2460, 1.0457, 0.6962	-595.31, -1236.09, -2306.74, 1318.60,
40	-0.6590, 17.0521, 2.0952	-1.0830, 1.1323, -0.8898	538.10, -342.55, 133.24, 325.24,
41	-8.5796, -5.5863, 14.0797	-1.8980, 0.0910, 1.2490	-191.23, 220.17, -0.54, 44.50,
42	-11.5374, -7.2881, -10.6960	-0.2298, -0.4486, 0.1577	54.06, 43.18, -82.27, -153.07,
43	-16.5082, 1.1372, 5.3901	-0.0148, 0.1572, -0.0467	293.06, -506.20, -315.15, -89.51,
44	14.5033, 6.8223, -4.8527	7.2326, -22.7125, 15.4999	255.41, -67.11, -81.97, 252.18,
45	-0.1189, 0.5187, 17.2900	-1.1053, 0.9706, 1.0528	3.28, 3.39, -2.75, -2.93,
46	-2.3452, 3.4595, -16.5200	1.2721, -1.0089, 0.9245	27.14, 27.38, 25.61, 26.74,
47	5.7669, 0.0877, 16.1842	-2.4653, 1.2553, 3.4840	53.22, 46.46, 126.13, -81.89,
48	-1.6981, -16.4849, 1.2052	-0.8282, -0.7479, 0.9331	285.29, -443.51, -302.77, -109.91,
49	-0.0238, 0.9983, 17.2778	-1.0559, 1.0178, 0.9575	-1.55, -2.90, -6.66, -9.02,
50	-8.4014, 3.5033, 14.7855	-0.3503, 0.6220, 0.0406	-171.12, 209.30, 8.72, 55.17,
51	-0.6585, -16.1920, -3.9849	-1.6116, -1.8141, 0.4874	305.14, -353.11, -26.23, 140.50,
52	-1.1213, -13.8783, -7.8141	2.6998, 1.8657, 2.0903	-366.31, -161.09, 334.68, -16.78,
53	-9.0877, -13.6176, -3.1217	-23.4962, -13.3231, -2.8181	406.46, -270.86, -263.37, -12.96,
54	-6.7220, -7.4720, 14.0653	-1.6268, 0.0259, 1.2758	-167.28, 253.68, 67.75, 84.12,
55	-13.5658, 2.8961, -10.5377	0.2359, -0.3764, -0.0046	177.14, 194.06, 64.10, -188.40,
56	-0.6962, 17.1318, -1.3030	0.0097, -0.0962, -0.9989	118.05, -177.95, -116.98, 166.12,
57	-9.4683, 8.0857, -10.3923	-4.4154, 0.8894, -2.2367	-22.01, -59.33, 133.80, 65.33,
58	12.0232, 8.2022, -7.3781	-2.0126, 1.9488, -5.8751	867.76, -106.67, -871.81, 66.80,
59	-3.0973, 3.9717, -16.5136	1.1367, -1.0085, 0.7741	30.28, 25.70, 32.66, 29.43,
60	-3.5403, 16.7819, -1.3001	-0.0279, -0.0559, -0.8454	159.85, -219.18, -86.78, 157.26,
61	0.9427, 10.9804, -11.9501	-1.6599, 0.3896, -2.1343	-156.04, 26.19, 159.60, 122.53,
62	-16.6690, -0.6791, 4.3865	-17.8995, -0.4171, 2.0004	500.41, -278.68, 90.89, 313.33,
63	0.3626, 15.2344, -8.0626	0.6423, -1.0925, -0.5572	45.36, 222.96, -90.65, -155.07,
64	1.3668, 11.1445, -12.2038	-3.8554, 2.0272, -4.0943	-113.04, -78.00, 160.88, 24.32,
65	2.9583, 2.8202, -15.1643	-0.8981, -1.0091, -1.3430	-26.28, 18.64, -22.35, 36.11,
66	-0.5208, 0.1328, 17.3060	-1.0670, 1.0225, 0.9968	2.11, 3.14, -4.28, -2.67,
67	16.3464, 3.4989, 0.7733	11.3059, -118.4029, -4.6459	211.19, 95.55, -239.05, 316.02,
68	-0.0329, 0.1893, 17.2887	-0.9558, 0.9977, 0.8902	3.68, 2.79, -2.28, -3.59,
69	-17.4454, 1.1040, 0.9311	0.4639, 0.0205, -0.0400	17.37, -358.12, -311.19, 144.65,
70	-0.3857, 0.3993, 17.2735	-0.1550, 1.0215, 0.1251	2.81, 2.81, -3.20, -3.42,
71	-6.8654, 14.5186, -6.7774	0.3779, -0.5253, -0.4865	191.69, 864.91, 16.74, -847.61,
72	-1.5917, -11.5959, -12.0470	-0.5046, -0.9937, 0.2440	0.39, 134.58, -107.10, -162.69,
73	2.0308, -10.5979, 12.9100	0.1284, 1.0251, 0.6358	574.75, 1043.87, 1989.71, -1107.98,
74	4.9629, 1.6947, -15.1507	-0.9950, -1.2040, -1.6246	-39.25, 37.41, -40.34, 55.19,
75	-11.1806, 8.3382, -9.8984	0.0162, -0.3849, -0.2890	-85.71, 227.46, -201.97, -216.17,
76	-4.5377, 5.8776, 15.7660	-7.1976, 2.6089, 4.7423	-136.45, -5.14, -35.08, -127.96,
77	-5.5175, -5.7080, 15.3695	-0.1116, 0.6740, 0.3152	-1.07, 165.58, 136.37, 35.22,
78	11.7485, 12.0228, -1.5498	0.0215, -0.3513, -2.5346	60.83, -71.99, -290.46, 267.97,
79	-5.4578, -13.2439, -7.7592	1.7785, 0.8147, 1.2665	-344.37, 71.25, 334.80, 193.23,

80	-3.7312, 15.5777, -6.8992	0.3902, -0.6362, -0.6047	271.50, 624.55, 112.19, -615.78,
81	3.5662, 2.6486, -15.1679	-0.9527, -1.0377, -1.4724	-30.44, 24.23, -25.60, 41.73,
82	-0.2476, 11.1074, -12.2150	-3.0624, 1.3031, -3.0921	-92.70, -51.69, 144.13, 45.06,
83	-8.4154, 14.8584, -3.7961	0.3745, -0.3752, -0.5197	2.79, -930.91, -192.59, 948.96,
84	-0.4145, 17.2918, -2.5308	0.3939, -0.5353, -0.9287	-179.78, -529.68, -436.36, 708.58,
85	2.9404, 16.8826, -1.2948	0.0736, -0.1894, -1.2342	74.96, -146.05, -162.90, 197.13,
86	-0.3618, 0.4121, 17.3248	-5.0689, 0.9144, 4.9177	2.75, 2.71, -3.67, -3.07,
87	12.6637, 1.7361, -9.6306	77.1195, -31.2701, 238.2116	-314.35, -4.98, 276.00, 116.56,
88	0.1148, 16.5864, -2.7131	-0.5383, 0.3499, -1.1451	374.87, -312.55, -140.79, 30.53,
89	3.3588, 13.5767, 10.4255	3.6731, -2.7332, -3.5815	788.87, 325.61, -800.67, 1419.29,
90	0.8043, -13.7855, -8.1404	1.7219, 1.1359, 1.8345	-224.66, -56.06, 157.58, 71.34,
91	-10.1361, -10.2561, -7.6349	-0.1779, -0.3927, 0.3236	103.97, -16.28, -79.78, -165.79,
92	4.9946, 0.4885, 16.3671	-0.3346, 1.4333, 0.4389	-18.06, -34.52, 74.45, -110.36,
93	-13.0870, -1.2559, -11.1548	-0.0203, -0.3928, 0.0379	56.39, 79.86, -55.87, -144.93,
94	11.7283, 10.0478, -5.7690	0.6982, -2.7080, -1.0165	110.37, 181.38, -74.80, -18.22,
95	-0.0845, -11.3657, -12.0416	-0.2943, -0.9356, 0.4088	59.71, 101.52, -45.58, -146.25,
96	-11.6845, 12.8484, 0.8473	1.0531, -0.4355, -0.4450	347.58, -238.65, 27.03, 448.67,
97	-5.0174, 14.5255, -7.8659	0.5991, -0.7684, -0.4124	-36.81, 309.20, -174.84, -287.03,
98	15.6052, -3.8242, 2.4675	-0.4272, 0.4396, 6.4481	-107.93, -247.56, -371.75, 241.56,

Table 3: All real solutions of DGP of a 5-5 CDPR with maximum number of 74 real solutions

Geometric dimensions and load: $\mathbf{a}_1 = [0, 0, 0]$, $\mathbf{a}_2 = [1.444170, 0, 1.203330]$, $\mathbf{a}_3 = [0.302415, 1.262060, 0.555330]$, $\mathbf{a}_4 = [-0.711127, 0.808726, 0.810451]$, $\mathbf{a}_5 = [0.749568, 0.761578, -0.469085]$, $\mathbf{b}_1 = [2.161690, 0, 0]$, $\mathbf{b}_2 = [-0.125711, 0, 1.326150]$, $\mathbf{b}_3 = [-0.412791, 0.021143, 0.449869]$, $\mathbf{b}_4 = [-0.162650, -0.468249, -0.399945]$, $\mathbf{b}_5 = [1.596530, 1.314460, 0.962240]$, $(\rho_1, \rho_2, \rho_3, \rho_4) = (2.464490, 1.995860, 1.206220, 1.423950, 2.430200)$, $Q = 10$.

Conf.	(x, y, z)	(e_1, e_2, e_3)	$(\tau_1, \tau_2, \tau_3, \tau_4)$
1	-0.2837, -0.1765, 0.2861	-0.9878, -0.6538, -1.0446	11.74, 0.23, -1.94, -0.99, -2.82,
2	0.5312, 0.6906, -0.7303	-0.4333, 0.9140, -0.4121	2.86, 1.88, 17.78, 3.08, -33.92,
3	0.6764, 2.1323, 0.4556	0.1229, 1.4532, -0.3371	-135.72, -60.35, -172.56, 138.78, 189.75,
4	1.2297, 1.1158, 0.6440	-0.6782, 0.4388, -0.5549	27.02, -6.23, 48.62, -46.72, -21.12,
5	0.1548, -0.0888, 0.3115	-1.3644, -1.1945, -1.2275	10.37, 2.35, 4.15, -1.06, 3.29,
6	1.0020, 1.4049, 0.3247	-0.0524, 1.0933, 0.0764	-30.88, -15.22, -18.11, 21.28, 37.16,
7	0.1363, 0.9933, -0.9479	-0.2815, 0.9396, -0.5967	-9.63, -1.45, -25.77, 4.17, 22.46,
8	0.4281, 0.4239, -0.7045	-0.6045, 0.8501, -0.4034	16.06, 3.99, 102.86, -12.02, -118.32,
9	-0.3582, 2.1809, 0.5294	0.3037, 0.2975, -0.1669	-6.97, -16.70, 16.56, 1.98, 7.97,
10	-0.9130, 1.3560, 0.0753	83.0105, -24.9753, 80.8086	2.53, 15.28, -15.22, -6.10, 5.03,
11	-0.4607, 0.0108, 0.6416	0.5670, 4.1847, 1.1578	-82.57, 3.09, 100.70, -89.99, 63.90,
12	-1.3491, 0.9141, 0.1015	-0.9261, 0.5187, -1.8319	18.17, 5.35, 0.08, -16.88, -12.61,
13	0.5107, 0.9502, 1.1408	-0.2853, 0.4373, 0.4091	0.65, 2.09, 10.29, -1.70, -1.52,
14	-0.4694, 0.6436, 1.7056	-9.8403, 3.5712, -1.2696	1.73, 0.47, 2.31, 7.55, 0.48,
15	0.2243, 1.2378, -0.8704	-0.1238, 1.1975, -0.7995	26.07, 6.64, 64.42, -28.30, -77.52,
16	-0.1060, 1.0669, 0.0947	1.7150, -0.3236, 0.7042	7.49, 7.28, -3.92, -5.09, 5.45,
17	-0.1064, 0.1791, 1.0325	0.2008, 53.0584, -8.5358	9.20, -6.32, -1.54, 3.99, -2.06,
18	0.1132, 0.8172, 1.6214	4.1771, -0.3353, -0.6053	5.12, 4.17, 7.05, 5.87, 0.81,
19	0.3336, 0.7842, -0.4449	2.1191, 4.0645, -5.3532	-14.88, 11.74, -9.74, 0.64, -2.86,
20	-0.6706, -0.1342, 0.0699	-0.5845, 1.6081, 0.0576	140.84, -157.06, -178.83, 53.97, 100.65,
21	-0.8354, 1.8229, 1.4399	-0.2594, 2.1498, -0.4546	-4.65, -12.19, 5.94, 4.79, 9.83,
22	0.3662, 0.4199, -0.3881	1.0575, 0.7139, -1.4214	-17.64, 2.08, 13.06, 9.12, -11.98,
23	0.1825, 0.7634, 0.1265	1.1952, -0.0899, 0.0737	0.17, 0.26, -5.00, -6.20, 3.90,
24	-0.3065, 1.4008, 1.3108	0.6942, 2.2769, -4.5700	1.13, -3.68, 10.74, 2.08, 0.46,
25	-1.1314, 1.2503, 0.0753	-3.9748, 1.3324, -4.7641	-8.15, 39.49, -32.44, -9.72, 19.65,
26	-0.4868, 1.9962, 0.5210	0.4409, 0.2077, -0.1097	-15.89, -30.82, 23.22, 14.70, 14.02,
27	-0.0282, 0.5763, -0.1349	12.5513, 10.6915, 15.9334	1.01, 2.85, 14.62, -8.02, 16.18,
28	-0.5173, 1.0320, -0.1333	0.0762, 0.0876, 0.2274	-7.40, -3.38, -2.47, -4.75, 6.80,
29	0.6686, 0.2969, -0.4893	-0.9115, 0.6119, -0.4893	3.24, 5.97, -41.09, -15.81, 43.46,
30	0.1711, 0.9219, 0.0821	2.3583, -0.5316, 1.0148	1.93, 2.36, -6.87, -3.16, 4.28,
31	0.5643, 0.3875, -0.0311	1.1899, 0.3178, -0.8087	11.93, 2.32, -18.47, -23.71, 16.29,
32	-0.0973, 1.6540, 1.4568	-0.9555, 0.5756, -1.7557	6.16, -5.31, 15.58, -6.54, -2.88,

33	0.3553, 0.6625, -0.4498	1.6934, 2.2968, -3.4782	-10.73, 6.76, -3.29, -0.59, -4.13,
34	0.9388, 1.0269, 0.7520	1.9527, 5.9393, 0.7878	0.31, -9.18, -7.10, 7.29, 4.08,
35	0.8958, 1.5893, -0.2146	-0.0721, 1.0494, -0.3239	-32.77, -21.12, 24.89, -51.95, 71.70,
36	-0.0833, -0.1848, -0.3040	-0.9629, 1.0579, -0.2260	-41.55, -8.17, 6.11, 22.41, 12.08,
37	-0.7052, 1.5521, 1.2600	-0.2759, 1.0069, -2.2936	-3.92, -8.55, 12.72, 4.56, 5.97,
38	-1.0635, 1.2974, 0.0753	-6.4385, 2.0797, -7.1895	-1.10, 25.15, -22.27, -7.74, 10.29,
39	0.3027, 0.6760, 0.3765	-0.8220, -2.6709, 2.9390	29.39, -19.91, 27.06, -13.34, -10.95,
40	1.2054, 1.0751, 0.6343	-0.9432, 0.4714, -0.7765	43.29, -11.86, 90.21, -80.86, -40.78,
41	0.4418, 0.1927, 0.6460	-4.6140, -8.4841, -3.7763	6.59, -10.30, -14.74, 7.30, -8.84,
42	0.5834, 0.2172, -0.4841	-0.9146, 0.7383, -0.4486	-11.60, 10.30, -121.68, 4.28, 114.91,
43	-0.2089, 0.0338, 0.7458	2.7490, 10.0872, 3.3235	34.08, -10.69, -35.02, 33.31, -23.92,
44	0.0739, 1.1328, -0.9382	-0.1995, 1.0771, -0.7477	-19.16, -0.48, -54.96, 17.21, 47.24,
45	0.0665, 0.4217, 0.0153	3.5456, 4.9828, 4.6809	-11.19, -4.01, -17.20, -9.17, -9.66,
46	0.6214, 2.1929, 0.3699	0.0734, 1.2226, -0.3357	113.43, 22.03, 164.32, -125.41, -146.08,
47	0.0828, 0.5548, 1.6098	9.3285, -1.3700, -0.9277	12.03, 14.35, 4.69, 8.41, 1.44,
48	0.6561, 1.1819, -0.6655	-0.1956, 1.0172, -0.4598	25.76, 15.92, 21.10, 20.11, -87.87,
49	1.0524, 1.4965, 0.3719	-0.1824, 0.8072, -0.0598	162.26, 30.54, 168.11, -167.41, -172.36,
50	-0.5641, 1.6225, 0.3104	0.5994, 0.0851, 0.0355	19.62, 30.43, -16.93, -30.07, -10.21,
51	1.0278, 1.4705, 0.3274	-0.1251, 0.9397, -0.0072	-98.19, -33.12, -84.40, 89.32, 111.36,
52	0.4918, 2.2898, 0.3568	0.0413, 0.9774, -0.3180	28.91, -5.45, 53.02, -37.75, -32.60,
53	0.4134, 0.1282, -0.5038	-0.8628, 0.8880, -0.3873	-29.84, -7.92, -143.35, 43.19, 128.44,
54	-0.2825, 0.3675, -0.5258	0.4871, 0.5334, -1.1867	-7.76, 0.91, 2.67, -4.82, 0.67,
55	0.3368, 0.1393, 1.2652	37.5553, -5.3037, -5.5255	-20.87, -31.39, 2.14, -5.99, 0.30,
56	-0.1278, 1.1309, -0.9476	-0.1984, 1.1423, -0.8971	-5.36, 1.32, -19.30, 2.49, 10.96,
57	0.9441, 0.6527, 0.2264	-3.3534, 1.3891, -1.8684	4.42, -6.83, -3.60, 3.42, -3.75,
58	-0.0023, 0.0150, 0.6541	-0.1923, -0.1528, -0.2376	10.83, -1.39, 8.46, -11.36, -1.99,
59	-0.8793, -0.0161, 0.1400	-0.4370, 1.4731, 0.2285	-1.67, -25.10, 1.40, -5.47, 19.74,
60	-1.3614, 1.3325, 0.2694	-1.6298, 1.3744, -2.3369	29.45, 40.63, -23.49, -32.05, -17.35,
61	1.0086, 0.4496, -0.0239	-1.3156, 0.7400, -0.7150	103.08, -45.09, 186.07, -103.38, -163.28,
62	0.9379, 1.2051, 0.3295	0.1487, 1.4494, 0.3033	-12.68, -8.10, 0.90, 0.18, 16.09,
63	-0.5244, -0.1694, -0.0583	-0.6830, 1.4965, -0.0419	-213.43, 152.49, 282.18, -67.13, -122.82,
64	-0.7410, 0.5791, -0.1580	-0.1644, 0.4463, 0.4641	-8.02, -4.12, -0.78, -4.74, 5.87,
65	1.1280, 0.9932, 0.4236	-1.5807, 0.5910, -1.2229	-36.84, 14.13, -97.02, 75.50, 49.11,
66	-0.5693, 0.6741, -0.1965	-9.3397, -4.2033, -12.1619	1.05, 1.18, 1.17, -8.09, 3.26,
67	-0.3987, 0.7036, -0.0598	-0.0816, 0.0974, 0.3804	-46.55, 0.93, -33.89, 16.34, 29.91,
68	0.1908, 0.7721, -0.1539	35.9920, 22.1885, 43.1066	0.31, -0.08, 2.01, -5.41, 6.39,
69	-0.1649, 0.5597, 0.1779	-0.1225, -0.0087, 0.3862	66.76, -11.03, 55.50, -40.37, -38.80,
70	-0.8399, 1.3674, 0.0753	12.4335, -3.6207, 11.3678	3.07, 12.97, -13.50, -5.73, 4.12,
71	-1.3369, 1.4526, 0.9523	-1.6895, 1.9967, -1.7759	-36.66, -36.43, 18.11, 29.54, 28.35,
72	-0.0556, 1.0020, 0.1161	1.3905, -0.2024, 0.4061	-4.35, -1.83, -8.71, -5.85, 1.10,
73	-1.2480, 1.1327, 0.0836	-2.0757, 0.7835, -2.8924	50.11, -58.39, 39.99, -2.27, -53.54,
74	-0.3171, 0.0244, 0.7432	1.3636, 6.6077, 1.9086	62.91, -13.07, -69.23, 64.29, -45.60,

References

- ABB. Irb 360 flexpicker. <http://new.abb.com/products/robotics/industrial-robots/irb-360>, 2013. [viii](#), [4](#)
- G. Abbasnejad and M. Carricato. Dgp-solver: Software for the direct geometrico-static analysis of cable-driven parallel robots. <http://grab.diem.unibo.it/DGP-Solver/>. [72](#)
- A. Alikhani, S. Behzadipour, S. A. S. Vanini, and A. Alasty. Workspace analysis of a three DOF cable-driven mechanism. *ASME Journal of Mechanisms and Robotics*, 1(4):041005/1–7, 2009. [viii](#), [8](#), [9](#)
- J. Angeles. *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms, Second Edition*. Springer, 2002. [1](#), [2](#), [38](#)
- Academy Baltic Aviation. Atr 72-500 full flight simulator (ffs). <http://www.balticaa.com/en/facilities-and-services/full-flight-simulators-ffs/atr-72—full-flight-simulator/>, 2013. [viii](#), [3](#)
- Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for numerical algebraic geometry. <http://bertini.nd.edu>. [28](#), [68](#), [77](#), [78](#)
- S. Behzadipour and A. Khajepour. A new cable-based parallel robot with three degrees of freedom. *Multibody System Dynamics*, 13(4):371–383, 2005. [viii](#), [8](#)
- S. Behzadipour, A. Khajepour, R. Dekker, and E. Chan. Deltabot: a new cable-based ultra high speed robot. In *Proc. of the 2003 ASME International Mechanical Engineering Congress*, volume 72, pages 533–537, Washington, DC, United States, 2003. [5](#)
- C. Bonivento, A. Eusebi, C. Melchiorri, and G. Montanari, M. and Vassura. Wireman: A portable wire manipulator for touch-rendering of bas-relief virtual surfaces. In *Proc. of the 1997 International Conference on Advanced Robotics (ICAR97)*, pages 13–18, Monterey, CA, USA, 1997. [10](#)
- P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff. Wrench-feasible workspace generation for cable-driven robots. *IEEE Transactions on Robotics*, 22(5):890–902, 2006. [9](#)
- O. Bottema and B. Roth. *Theoretical kinematics*. Dover Publications, New York, 1990. [38](#), [39](#)
- S. Bouchard, C. Gosselin, and B. Moore. On the ability of a cable-driven robot to generate a prescribed set of wrenches. *ASME Journal of Mechanisms and Robotics*, 2(1):011010/1–10, 2010. [10](#)

- K. Brunthaler. *Synthesis of 4R Linkages Using Kinematic Mapping*. PhD thesis, Institute for Basic Sciences in Engineering, Unit Geometry and CAD, Innsbruck, Austria, 2006. 17
- Cablecam. <http://www.cablecam.com>. 10
- M. Carricato. Inverse geometrico-static problem of under-constrained cable-driven parallel robots with three cables. *ASME Journal of Mechanisms and Robotics*, 5(3):031002/1–11, 2013a. 37, 42, 45, 46, 47, 50
- M. Carricato. Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables. *ASME Journal of Mechanisms and Robotics*, 5(3):031008/1–10, 2013b. 37, 42, 84
- M. Carricato and J.-P. Merlet. Geometrico-static analysis of under-constrained cable-driven parallel robots. In J. Lenarčič and M. M. Stanišič, editors, *Advances in Robot Kinematics: Motion in Man and Machine*, pages 309–319. Springer, Dordrecht, 2010. 36
- M. Carricato and J.-P. Merlet. Inverse geometrico-static problem of under-constrained cable-driven parallel robots with three cables. In *13th World Congress in Mechanism and Machine Science*, pages 1–10, Paper No. A7_283, Guanajuato, Mexico, 2011a. 37, 42, 45
- M. Carricato and J.-P. Merlet. Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation*, pages 3011–3017, Shanghai, China, 2011b. 37, 42
- M. Carricato and J.-P. Merlet. Stability analysis of underconstrained cable-driven parallel robots. *IEEE Transactions on Robotics*, 29(1):288–296, 2013. viii, 36, 42, 43, 44, 54, 74, 84
- R. Clavel. Delta, a fast robot with parallel geometry. *18th International Symposium on Industrial Robots*, pages 91–100, 1988. 4
- J.-F. Collard and P. Cardou. Computing the lowest equilibrium pose of a cable-suspended rigid body. *Optimization and Engineering*, 14(3):457–476, 2013. 15
- R. M. Corless. Gröbner bases and matrix eigenproblems. *ACM SIGSAM Bulletin*, 30(4):26–32, 1996. 54
- D. Cox, J. Little, and D. O’Shea. *Ideals, varieties, and algorithms*. Springer, New York, 2007. 21
- David A. Cox, John Little, and Donal OShea. *Using Algebraic Geometry*. Springer, 2005. 16, 17
- A. K. Dhingra, A. N. Almadi, and D. Kohli. A Gröbner-Sylvester hybrid method for closed-form displacement analysis of mechanisms. *ASME Journal of Mechanical Design*, 122(4):431–438, 2000. 23, 54
- P. Dietmaier. The Stewart-Gough platform of general geometry can have 40 real postures. In J. Lenarčič and M. L. Husty, editors, *Advances in Robot Kinematics: Analysis and Control*, pages 7–16. Kluwer Academic Publishers, Dordrecht, 1998. 16, 39

- J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993. [22](#), [53](#)
- J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *International Journal of Robotics Research*, 30(3):324–334, 2011. [15](#)
- A. Ghasemi, M. Eghtesad, and M. Farid. Workspace analysis for planar and spatial redundant cable robots. *ASME Journal of Mechanisms and Robotics*, 1(4):044502/1–6, 2009. [10](#)
- A. Ghasemi, M. Eghtesad, and M. Mehrdad, Farid. Neural network solution for forward kinematics problem of cable robots. *Journal of Intelligent and Robotic Systems*, 60(2):201–215, 2010. [14](#)
- M. Gobbi, G. Mastinu, and G. Previati. A method for measuring the inertia properties of rigid bodies. *Mechanical Systems and Signal Processing*, 25(1):305–318, 2011. [viii](#), [13](#)
- J. J. Gorman, K. W. Jablokow, and D. J. Cannon. The cable array robot: theory and experiment. *Proceedings of IEEE International Conference on Robotics and Automation*, 3:2804 – 2810, 2001. [10](#)
- C.M. Gosselin and J.-F. Hamel. The agile eye: a high-performance three-degree-of-freedom camera-orienting device. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA*, 45(10):781786, 1994. [4](#)
- V.E. Gough. Contribution to discussion of papers on research in automobile stability, control and tyre performance. *Proceedings of the Automotive Division of the Institution of Mechanical Engineers*, 1956-1957. [3](#)
- M. Gouttefarde and C. M. Gosselin. Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms. *IEEE Transactions on Robotics*, 22(3):434–445, 2006. [10](#)
- 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*, 27(1):1–13, 2011. [10](#)
- Q. Jiang and V. Kumar. The direct kinematics of objects suspended from cables. In *Proc. of the ASME 2010 Int. Design Engineering Technical Conferences*, volume 2, pages 193–202, Paper no. DETC2010–28036, Montreal, Canada, August 15–18 2010. [15](#)
- S. Kawamura, H. Kino, and C. Won. High-speed manipulation by using parallel wire-driven robots. *Robotica*, 18(1):13–21, 2000. [viii](#), [6](#)
- C. Kossowski and L. Notash. CAT4 (Cable Actuated Truss - 4 degrees of freedom): a novel 4 DOF cable actuated parallel manipulator. *Journal of Robotic Systems*, 19(12):605–615, 2002. [viii](#), [6](#), [7](#)

- R. Kurtz and V. Hayward. Dexterity measure for tendon actuated parallel mechanisms. In *Proc. of the Fifth Int. Conf. on Advanced Robotics*, volume 2, pages 1141–1146, 1991. 33
- R. Kurtz and V. Hayward. Dexterity measures with unilateral actuation constraints: the $n + 1$ case. *Advanced Robotics*, 9(5):561–577, 1995. 5
- D. Lau, D. Oetomo, and S.K. Halgamuge. Wrench-closure workspace generation for cable driven parallel manipulators using a hybrid analytical-numerical approach. *ASME Journal of Mechanical Design*, 133(7):071004/1–10, 2011. 10
- Robotics Laboratory Laval University. The agile eye. <http://robot.gmc.ulaval.ca/en/research/theme103.html>, 2013. viii, 3
- D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*. Springer, New York, 2008. 31, 75
- Maplesoft. Maple online help. <http://www.maplesoft.com/support/help/>. 17
- M. Tale Masouleh. *Kinematic analysis of five-DOF (3T2R) parallel mechanisms with identical limb structures*. PhD thesis, Faculté des Sciences et de Génie Université Laval Québec, 2010. 16, 17
- J.-P. Merlet. *Parallel robots*. Springer, Dordrecht, 2006. 2, 74
- J.-P. Merlet and D. Daney. A portable, modular parallel wire crane for rescue operations. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation*, pages 2834–2839, Anchorage, USA, 2010. 5
- N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011. 14
- A. Ming and T. Higuchi. Study on multiple degree-of-freedom positioning mechanism using wires— Part 1: Concept, design and control. *Int. Journal of the Japan Society for Precision Engineering*, 28(2):131–138, 1994. 5, 33
- Electric Europe B.V. Mitsubishi. Melfa roboter rh series. <http://www.mitsubishi-automation.com/>, 2013. viii, 2
- H. M. Möller. Gröbner bases and numerical analysis. In B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*, volume 251 of *London Mathematical Society Lecture Note Series*, pages 159–178. Cambridge University Press, Cambridge, 1998. 54
- So-Ryeok Oh and Sunil K. Agrawal. A reference governor-based controller for a cable robot under input constraints. *IEEE Transaction on Control System Technology*, 13(4):639–645, 2005. 4
- Y. D. Patel and P. M. George. Parallel manipulators applicationsa survey. *Modern Mechanical Engineering*, 2:57–64, 2012. 4

- M. Pfurner. *Analysis of Spatial Serial Manipulators Using Kinematic Mapping*. PhD thesis, Univesrity Innsbruck, Institute for Engineering Mathematics, Geometry and Computer Sciences, Innsbruck, Austria, 2006. 17
- Singiresu S. Rao. *Engineering Optimization Theory and Practice*. JOHN WILEY & SONS, INC., 2009. 75
- R. G. Roberts, T. Graham, and T. Lippitt. On the inverse kinematics, statics, and fault tolerance of cable-suspended robots. *Journal of Robotic Systems*, 15(10):581–597, 1998. 5, 33
- G. Rosati, P. Gallina, and S. Masiero. Design, implementation and clinical tests of a wire-based robot for neurorehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4):560–569, 2007. viii, 11, 12
- S.Kawamura, W.Choe, S.Tanaka, and S.R.Pandian. Development of an ultrahigh speed robot falcon using wire drive system. *IEEE International Conference on Robotics and Automation*, pages 215–220, 1995. 5
- SkyCam. www.august-design.com. 10
- A. J. Sommese and C. W. Wampler. *The numerical solution of systems of polynomials arising in engineering and science*. World Scientific Publishing, Singapore, 2005. 22, 26, 28, 62, 63
- D. Stewart. A platform with 6 degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180:371–386, June 1965. 3
- D. Surdilovic, J. Zhang, and R. Bernhardt. STRING-MAN: wire-robot technology for safe, flexible and human-friendly gait rehabilitation. In *Proc. of the 2007 IEEE Int. Conference on Rehabilitation Robotics*, pages 446–453, Noordwijk, The Netherlands, 2007. viii, 11, 12
- S. Tadokoro, R. Verhoeven, M. Hiller, and T. Takamori. A portable parallel manipulator for search and rescue at large-scale urban earthquakes and an identification algorithm for the installation in unstructured environments. In *Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1222–1227, Kyongju, Korea, 1999. viii, 10, 11
- S. Tadokoro, Y. Murao, M. Hiller, R. Murata, H. Kohkawa, and T. Matsushima. A motion base with 6-DOF by parallel cable drive architecture. *IEEE/ASME Transactions on Mechatronics*, 7(2):115–123, 2002. viii, 6, 7
- C. W. Wampler, A. P. Morgan, and A. J. Sommese. Numerical continuation methods for solving polynomial systems arising in kinematics. *ASME Journal of Mechanical Design*, 112(1):59–68, 1990. 25, 63
- Wikipedia. Wikipedia. <http://en.wikipedia.org/wiki/Manipulator>, May 2013. 1