Alma Mater Studiorum - Università di Bologna

in cotutela con

UNS - Université de Nice - Sophia Antipolis

dottorato di ricerca in

Meccanica e Scienze Avanzate dell'Ingegneria

ciclo XXVII

Settore Concorsuale di afferenza:
09/A2 - meccanica applicata alle macchine

Settore Scientifico disciplinare:
ing-ind/13 - meccanica applicata alle macchine

Kinematics and statics of cable-driven parallel robots by interval-analysis-based methods

Presentata da:
Alessandro Berti

Coordinatore Dottorato:
Prof. Vincenzo Parenti Castelli

Relatore:
Prof. Marco Carricato

Relatore:
Dr. Jean-Pierre Merlet

---

22 Aprile 2015

Abstract

In the past two decades the work of a growing number of researchers in robotics focused on a particular group of machines, belonging to the family of parallel manipulators: the cable robots. Although these robots share several theoretical elements with the better known parallel robots, they still present completely (or partly) unsolved issues. In particular, the study of their kinematics, already a difficult subject for conventional parallel manipulators, is further complicated by the non-linear nature of cables, which can exert only efforts of pure traction. The work presented in this thesis therefore focuses on the study of the kinematics of these robots and on the development of numerical techniques able to address some of the problems related to it. Most of the work is focused on the development of an interval-analysis based procedure for the solution of the direct geometric problem of a generic cable manipulator. This technique, as well as allowing for a rapid solution of the problem, also guarantees the results obtained against rounding and elimination errors and can take into account any uncertainties in the model of the problem. The developed code has been tested with the help of a small manipulator whose realization is described in this dissertation together with the auxiliary work done during its design and simulation phases.

Sommario

Negli ultimi decenni il lavoro di una parte sempre maggiore di ricercatori che si occupano di robotica si è concentrato su un particolare gruppo di robot appartenenti alla famiglia dei manipolatori paralleli: i robot a cavi. Nonostante i numerosi studi al riguardo, questi robot presentano ancora oggi numerose problematiche del tutto (o in parte) irrisolte. Lo studio della loro cinematica nello specifico, già complesso per i manipolatori paralleli tradizionali, è ulteriormente complicato dalla natura non lineare dei cavi, i quali possono esercitare sforzi di sola trazione. Il lavoro presentato in questa tesi si concentra dunque sullo studio della cinematica dei robot a cavi e sulla messa a punto di tecniche numeriche in grado di affrontare parte delle problematiche ad essa legate. La maggior parte del lavoro è incentrata sullo sviluppo di una procedura per la soluzione del problema geometrico diretto di un generico manipolatore a cavi basata sull'analisi per intervalli. Questa tecnica di analisi numeirica, oltre a consentire una rapida soluzione del problema, permette di garantire i risultati ottenuti in caso di errori di cancellazione e arrotondamento e consente di considerare eventuali incertezze presenti nel modello del problema. Il codice sviluppato è stato testato attraverso un piccolo prototipo di manipolatore a cavi la cui realizzazione, avvenuta durante il percorso di dottrato, è descritta all'interno dell'elaborato unitamente al lavoro collaterale svolto durante la fase di progettazione e simulazione.

Résumé

Pendant les dernières décennies, le travail d'une partie toujours croissante de chercheurs qui s'occupent de robotique s'est focalisé sur un groupe spécifique de robots qui fait partie de la famille des manipulateurs parallèles : les robots à câbles. Malgré les nombreux études que l'on a consacré à ce sujet, ces robots présentent encore aujourd'hui plusieurs problématiques complètement ou partiellement irrésolues. En particulier l'étude de leur cinématique, qui se révèle déjà complexe pour les manipulateurs parallèles traditionnels, est rendu encore plus compliqué par la nature non linéaire des câbles qui peuvent seulement exercer des efforts de traction. Le travail présenté dans ma thèse concentre donc son attention sur l'étude de la cinématique des robots à câbles et sur la mise au point de techniques numériques capables d'aborder une partie des problématiques liées à cela. La plupart du travail se concentre sur l'élaboration d'un algorithme pour la résolution du problème géométrique direct d'un manipulateur à câbles général qui se fonde sur l'analyse par intervalles. Cette technique d'analyse permet non seulement de résoudre rapidement le problème mais également de garantir les résultats obtenus en cas d'erreur de cancellation et d'arrondi et de prendre en considération les incertitudes éventuellement presentes dans le modèle du problème. Le code développé a été testé grâce à un petit prototype de manipulateur à câbles dont la réalisation, qui a eu lieu pendant le parcours de doctorat, est décrite à l'intérieur du devoir en accord avec la phase de conception du projet et de simulation.

Contents

List of Figures

List of Tables

Prologue in English

In the late 80s some researchers began to explore the possibility of replacing parallel manipulators' rigid-body actuators with cables, in order to overcome the main weaknesses of this family of robots, such as the limited workspace, a considerably complex manufacturing process and maintenance, and high costs. These robots, defined *cable-driven parallel robots* (CDPRs), or simply *cable robots*, in addition to the advantages deriving from parallel architecture, thanks to cable flexibility can also have extremely large workspaces, higher load capacity, better kinematic performance and enhanced ease of design and assembly. A more detailed presentation of cable robots, the literature on them and the description of the most famous prototypes is reported in the first part of Chapter 1 of this thesis.

However, the already difficult control and kinematic analysis of parallel robots (that took more than 30 years to be completely solved) is for these manipulators further complicated by the inability of cables to withstand compressive loadings. This implies that certain poses (i.e. positions and orientations) and/or loading conditions cannot be achieved because some cables can suddenly become slack, thus modifying the robot configuration, its static and/or dynamic equilibrium and compromising its control. The work done during the three years of the Ph.D. program primarily dealt with the study of these problems and especially on direct kinematics issues.

Direct or forward kinematics refers to compute the end-effector pose from specified values of the joint variables, in this case cable lengths. If a robot with $n \geq 6$ cables and at least $m = 6$ of them are in tension, the 6 end-effector degrees of freedom can be determined by the equations deriving from the geometric constraints imposed by taut cables. When $m < 6$ these geometric conditions are not sufficient to completely define robot pose, which will be determined by the platform static equilibrium. In order to fulfill equilibrium conditions, the 6 equations of the statics and the m unknown cable tensions have to be added to the m geometric equations. Moreover only solutions having cable tensions greater than 0 have to be taken into account.

Consequently the kinematic analysis of these manipulators kinematics has to be conducted together with their statics and taking into account the nonlinear constraints imposed by cables. Indeed, in order to control a cable robot in a completely safe manner it is indeed necessary to ensure that at all times the manipulator configuration does not change due to a loss of tension in one or more cables. So, all the possible configurations that can be obtained if one or more cables become slack must be taken into account. These amount, for a generic n-cable robot, to

$$N = \sum_{m=1}^{n} \frac{n!}{m!(n-m)!}$$

These are the main reasons that make the already challenging kinematic analysis of parallel robots even more complex for CDPRs. These issues, together with the description of the geometric models of cable robots adopted in this manuscript, are discussed in Chapter 2 along with an overview of the literature devoted to direct kinematics problems of cable robots and related methods currently adopted to deal with it.

Chapter 3 presents the work that led to the development of a problem-solving routine that numerically solves the direct kinematic problem of a generic cable driven parallel robot. This method can completely solve the problem, and its results are guaranteed against numerical errors thanks to *interval analysis* computational techniques. Interval analysis is a branch of numerical analysis that includes a variety of tools that solve numeric problems without the risk of errors due to elimination or rounding effects. While a basic description of interval analysis and the methods used in this thesis is reported in the second half of Chapter 1, Chapter 3 provides an in-depth analysis of the structure of the implemented method, its different versions and proposes possible strategies for further developments and real-time applications.

Chapter 4 describes the results obtained during the test phase of the different versions of the procedure and discusses them. The method developed during this Ph.D. program showed very good performances. It allows for a complete analysis of direct geometrico-static problem for a generic robot with n cables with very low computation times. Thanks to interval analysis, it also directly searches for real solutions with positive cable tensions and its results are guaranteed against numerical errors. This chapter also describes the realization of a small prototype that was built in order to test the implementation of the on-line version of the algorithm within a true control scheme.

Chapter 5 summarizes the main goals achieved during this work, and presents the principal lines of development for future research.

Prologue in Italian

Verso la fine degli anni '80 alcuni ricercatori in ambito robotico cominciarono ad esplorare la possibilità di sostituire i tradizionali attuatori a corpo rigido dei manipolatori paralleli con cavi, in modo tale da superare i principali svantaggi di questa famiglia di robot, come ad esempio il limitato spazio di lavoro, la complessità dei processi di fabbricazione, montaggio e manutenzione e i costi elevati. Questi robot, definiti in inglese *cable-driven parallel robots* (CDPRs), o semplicemente *robot a cavi*, oltre ai vantaggi derivanti dall'architettura parallela, grazie alla flessibilità dei cavi sono in grado di operare in spazi di lavoro molto vasti, posseggono maggiori capacità di carico, migliori prestazioni cinematiche e sono di più facile progettazione e assemblaggio. Una dettagliata presentazione dei robot a cavi, della letteratura a loro dedicata e la descrizione dei prototipi più famosi è riportata nella prima parte del del Capitolo 1 di questa tesi di dottorato.

L'analisi cinematica e il controllo di questi robot, però, oltre alle problematiche già complesse proprie dei manipolatori paralleli (risolte solo dopo più di 30 anni di ricerche), sono rese ancora più difficoltose dalla incapacità dei cavi di sopportare sforzi di compressione. Ciò comporta che alcune pose (ovvero posizioni ed orientamenti) e/o condizioni di carico non possono essere raggiunte perché alcuni cavi diventano improvvisamente laschi, modificando quindi la configurazione del robot, il suo equilibrio statico e/o dinamico e compromettendo seriamente il controllo. Il lavoro svolto durante i tre anno del percorso di dottorato è stato incentrato principalmente sullo studio di queste problematiche e specialmente sugli aspetti della cinematica diretta.

Per cinematica diretta si intende il calcolare la posa del membro terminale del robot per assegnati valori delle variabili di giunto, in questo caso quindi le lunghezze dei cavi. Se si considera un robot con $n \geq 6$ cavi di cui almeno $m = 6$ in tensione, i 6 gradi di libertà dell'end-effector possono essere determinati dalle equazioni derivanti dai vincoli geometrici imposti dai cavi tesi. Quando $m < 6$ invece, queste condizioni geometriche non sono sufficienti in numero a definire completamente la posa del robot, la quale

sarà determinata solamente considerando anche l'equilibrio statico della piattaforma. Per poter rispettare questa condizione, le 6 equazioni della statica nelle m incognite rappresentate dalle tensioni nei cavi devono essere aggiunte alle m equazioni derivanti dai vincoli geometrici. Inoltre solo le soluzioni in cui le tensioni nei cavi sono positive sono da prendere in considerazione. Di conseguenza l'analisi cinematica di questi manipolatori deve essere risolta unitamente alla statica e nel rispetto dei vincoli non lineari dei cavi. Infatti, per poter controllare un robot a cavi in maniera del tutto sicura è necessario garantire che in ogni momento la configurazione del manipolatore non cambi a causa di una perdita di tensione di uno o più cavi. Quindi, tutte le possibili configurazioni che è possibile ottenere se uno o più cavi diventano laschi vanno prese in considerazione il che equivale, per un generico manipolatore a n cavi ad analizzare un numero di configurazioni pari a

$$N = \sum_{m=1}^{n} \frac{n!}{m!(n-m)!}$$

Queste sono le ragioni principali che rendono la già complessa analisi cinematica dei manipolatori paralleli ancora più ardua per i CDPR. Queste problematiche, unite alla descrizione della modellazione geometrica dei robot a cavi adottata in questo lavoro, sono discusse nel Capitolo 2 insieme ad una panoramica della letteratura dedicata ai problemi di cinematica diretta e ai relativi metodi impiegati per la sua risoluzione.

Il Capitolo 3 presenta il lavoro che ha portato allo sviluppo di una routine per trovare per via numerica le soluzioni della cinematica diretto di un generico manipolatore a cavi. Questo metodo permette di risolvere il problema in maniera completa e i suoi risultati sono garantiti contro eventuali errori numerici grazie all'uso della tecnica di analisi numerica chiamata *analisi per intervalli*. L'analisi per intervalli è una branca della analisi numerica che racchiude una varietà di strumenti che consentono di risolvere problemi senza il rischio di errori dovuti a cancellazione o arrotondamento. Mentre una descrizione dei concetti di base dell'analisi per intervalli e i metodi richiamati all'interno del manoscritto è riportata nella seconda metà del Capitolo 1, il Capitolo 3 descrive nel dettaglio la struttura del metodo di calcolo implementato, le sue differenti versioni e propone strategie per eventuali futuri sviluppi e applicazioni in real-time.

Il Capitolo 4 mostra i risultati ottenuti durante la fase di test per le diverse versioni dell'algoritmo e presenta alcune considerazioni al riguardo. Il metodo sviluppato durante il percorso di questo dottorato mostra un ottimo comportamento dal punto di vista prestazionale. Consente di condurre

una analisi completa del problema geometrico-statico diretto per un robot a n cavi in tempi molto ridotti. Grazie all'analisi per intervalli, esso consente anche di ottenere direttamente solo le soluzioni reali del problema aventi tensioni positive nei cavi e i suoi risultati sono sicuri dal punto di vista numerico. In questo capitolo è inoltre descritta la realizzazione di un piccolo prototipo utilizzato a scopi didattici per testare l'implementazione delle versioni on-line dell'algoritmo all'interno di un vero e proprio schema di controllo.

Il Capitolo 5 riassume i principali risultati contenuti in questo lavoro e presenta le linee guida per futuri sviluppi e attività di ricerca.

Prologue in French

Vers la fin des années 80, certains chercheurs dans le domaine de la robotique ont commencé à explorer la possibilité de remplacer les traditionnels actionneurs à corps rigide de manipulateurs parallèles avec des câbles, afin de surmonter les principaux inconvénients de cette famille de robots, comme le limitée espace de travail, la complexité des processus de fabrication, l'assemblage et l'entretien et les coûts élevés. Ces robots, appelés en anglais *cable-driven parallel robots* (CDPRs), ou tout simplement *robot à câbles* en plus des avantages propres de l'architecture parallèle, grâce à la flexibilité des câbles sont capables de fonctionner dans des espaces de travail très large, de posséder une plus grande capacité de charge, meilleures performances cinématiques et sont de plus faciles conception et assemblage. La première section du Chapitre 1 de cette thèse de doctorat présent en détail les robot à câbles, la littérature consacrée à ce sujet et la description des les prototypes les plus célèbres.

Toutefois, l'analyse cinématique et le contrôle de ces robots, en plus des problèmes déjà complexes propres de manipulateurs parallèles (résolu seulement après plus de 30 années de recherche), sont rendues encore plus difficiles par l'incapacité des câbles de résister à des efforts de compression. Cela implique que certaines poses (positions et orientations) et/ou conditions de charge ne peuvent être atteints parce que certains câbles deviennent soudainement lâche, modifiant ainsi la configuration du robot, son équilibre statique ou dynamique et compromettant sérieusement le contrôle. Le travail effectué au cours du programme de doctorat a été axée principalement sur l'étude de ces questions, et notamment sur les aspects de la cinématique directe.

La cinématique directe est le calcul de la pose de l'élément terminal du robot pour valeurs attribuées des variables du joints, dans ce cas, les longueurs des câbles. Si l'on considère un robot avec $n \geq 6$ câbles dont au moins $m = 6$ en tension, les 6 degrés de liberté de l'end-effector peuvent être déterminée par les équations dérivées les contraintes géométriques im-

posées par les câbles tendus. Lorsque $m < 6$, ces conditions géométriques ne sont pas en nombre suffisant pour définir complètement la pose du robot, qui peut être déterminée seulement en considérant l'équilibre statique de la plateforme. Afin de respecter cette condition, les 6 équations de la statique dans les m inconnues représentés par les tensions des câbles doivent être ajoutés aux m équations des contraintes géométriques. En outre, les seules solutions dans lesquelles les tensions dans les câbles sont positives doivent être pris en considération. En conséquence, l'analyse cinématique de ces manipulateurs doit être résolu au même temps que la statique et en considérant les contraintes non linéaires dans le câbles. En fait, afin de contrôler un robot à câbles dans un façon totalement sûre est nécessaire de garantir qu'à tout moment la configuration du manipulateur ne change pas en raison d'une perte de tension d'un ou plusieurs câbles. Donc, toutes les configurations possibles qui peuvent être obtenues si un ou plusieurs câbles deviennent lâches devraient être prises en considération ce qui équivaut, pour un générique manipulateur à n câbles, à l'analyser un nombre de configurations égal à

$$N = \sum_{m=1}^{n} \frac{n!}{m!(n-m)!}$$

Telles sont les principales raisons qui font que l'analyse cinématique déjà complexe de manipulateurs parallèles soit encore plus difficile pour les CDPR. Ces questions, ainsi que la description de la modélisation géométrique des robot à câbles utilisée dans ce travail, sont examinées dans le Chapitre 2 avec une revue de la littérature consacrée aux problèmes de cinématique directe et les méthodes utilisées dans sa résolution .

Le Chapitre 3 présente le travail qui a conduit à l'élaboration d'une routine pour trouver numériquement les solutions de la cinématique direct d'un générique manipulateur à câbles. Cette méthode permet de résoudre le problème complètement et ses résultats sont garantis contre les erreurs numériques grâce à l'utilisation d'une technique d'analyse numérique appelé *analysepar intervalles*. L'analyse par intervalles est une branche de l'analyse numérique qui comprend une variété d'outils pour résoudre problèmes sans le risque d'erreurs en cas d'élimination ou arrondissement. Bien que la description des concepts de base et des méthodes d'analyse par intervalles préconisés dans le manuscrit est affichée dans la seconde moitié du Chapitre 1, le Chapitre 3 décrit en détail la structure de la méthode de calcul mise en œuvre, ses différentes versions et propose des stratégies pour les futurs développements et applications en temps réel.

Le Chapitre 4 montre les résultats obtenus au sein de la phase de test

pour les différentes versions de l'algorithme et présente quelques considérations à ce sujet. La méthode développée au cours de ce doctorat montre un excellent comportement du point de vue des performances. Il permet d'effectuer une analyse complète de le problème geometrico-statique d'un robot ayant n câbles en un temps très court. En plus, l'analyse par intervalles permet d'obtenir directement les seules solutions réelles du problème avec tensions positives dans les câbles et d'avoir résultats garantis du point de vue numérique. Dans ce chapitre, il est également décrit la réalisation d'un petit prototype utilisé à des fins éducatives pour tester la mise en œuvre des versions en ligne de l'algorithme dans un vrai système de contrôle.

Le Chapitre 5 résume les principaux résultats de ce travail et présente les lignes directrices pour les futures activités de développement et de recherche.

Chapter 1

Introduction

Throughout history, mankind has kept developing more and more sophisticated mechanisms and machines for a great variety of uses and tasks, many of them being represented by mechanisms capable of moving an object with respect to a fixed base. The position and the orientation of a rigid body in three dimensional space can be described through a maximum of 6 independent coordinates called *degrees of freedom* (dofs). The term *robot* or *manipulator* includes every mechanical system that can move an object with respect to a fixed base and control 2 or more of its degrees of freedom.

During the second half of the last century, the use of robots in industries grew dramatically together with the interest of the scientific community in their challenging issues. Over the last decades of the 1900s, indeed, a fruitful combination of scientific and technological developments in both theoretical and practical scientific fields allowed robotic devices to greatly expand their application domains, even in quite unexpected sectors.

From a technological point of view, for example, in the last 50 years electric motors have greatly increased their power density to the extent that it is possible to install relatively small and lightweight motors for heavy duties. Moreover, thanks to the discoveries in metallurgical and material engineering, great improvements were made from a mechanical point of view to gearboxes, that are coupled to the motors and have become more and more compact and reliable, giving designers more freedom during the development of new robot architectures and geometries. At the same time, the progress made in electronic components combined with the consecutive advent of computer science, made it possible to build and expand what can be defined as the *intelligence* of robotic systems, which definitely shows the differences with respect to traditional mechanisms and redefines the meaning

of automation.

During the same period, the scientific community started to show a growing interest to the more and more challenging issues emerging from robot development. Thanks to more powerful processors and more efficient programs provided by progress in computer science, researchers ceaselessly kept producing innovative solutions and contributions to the growth of robotics and its applications.

Indeed, the first manipulators replaced humans for most of the easier and repetitive operations in assembly lines, or for all those tasks that must be carried out in potentially harmful or unhealthy environments such as painting, welding, waste management, etc. Initially the most common robot model had a structure very similar to that of a human arm and all the manipulators sharing this architecture are defined *serial robots*.

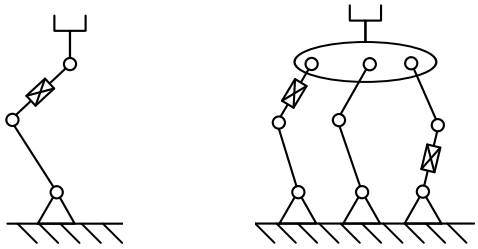
1.1 Serial and parallel manipulators

Nowadays robots are also used in other fields besides the industrial one: there are indeed manipulators for rehabilitation purposes, for entertainment applications, military operations, or domestic duties. Moreover, in the past two decades, a new family of robots consisting of different kinds of autonomous vehicles open up the range of application to an incredible extent. There are actually unmanned vehicles for every environment and condition: flying vehicles, climbing, rolling and walking robots, sailing and underwater autonomous boats. However these robots are not in the scope of this dissertation.

The majority of the robots currently in use have a structure recalling that of a human arm, obtained by consecutively connecting different rigid bodies with joints allowing one degree of freedom between two links. These couplers are generally translational or revolute joints and this sequence of linked bodies is called *open kinematic chain* (Fig. 1.1a). Manipulators having an open kinematic chain architecture are called *serial robots*.

The main advantage of serial robots is a large workspace compared to their fixed base dimension and Fig. 1.2 shows one of the most well known models of these family of manipulators. However, open-chain robots are affected by several weaknesses, accurately described in [Merlet 2006, Angeles 2002], and they can be all attributed to the fact that each member has to support the weight and the inertial forces of the following ones. The most relevant are:

- a small payload to robot weight ratio,



(a) (b)

Figure 1.1 – Open and closed kinematic chains.

- poor positioning accuracy,
- low speed and acceleration.

The drawbacks of the serial structure can be partly solved by connecting the end-effector to the base with multiple open kinematic chains in order to distribute the external load. The resulting structure will be thus formed by several *closed kinematic chains*, like the one depicted in Fig. 1.1b. According to [Merlet 2006]:"a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains is a *parallel manipulator*".

Figure 1.2 – The Kuka® 7-dof robotic arm.

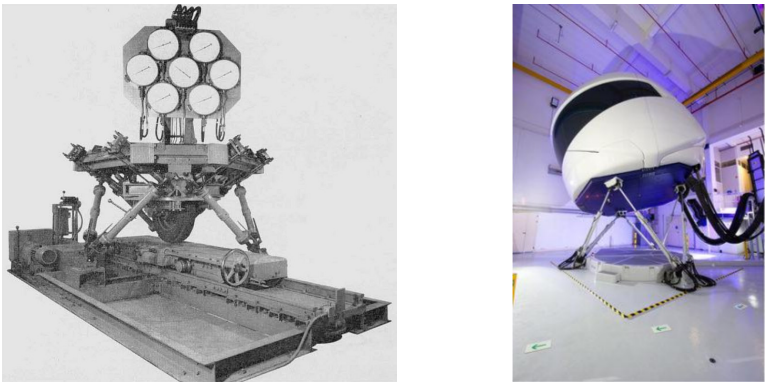
The first advantage of parallel robots that immediately stands out is the redistribution of the load applied to the end-effector on the links forming the closed-loop chains. This implies that bodies and joints constituting the links, also commonly called *legs*, can be lighter provided that the end-effector stiffness remains constant. This fact, together with a lower positioning sensitivity to joint measurement errors due to the fact that actuators are directly

connected to the end-effector, makes parallel manipulators far more accurate than serial robots.

This reduction of weight in robot components has multiple effects on parallel machine performances. First of all, reduced moving mass implies that higher velocities and accelerations are possible. Moreover, the actuator size can be significantly decreased together with, consequently, the machine power consumption.

Another great advantage consists in the modularity of parallel manipulators. Indeed all the legs are generally identical, which means that the components can be manufactured in series. Besides the economic convenience, high modularity positively affects also assembly and maintenance operations.

A fully detailed and comprehensive introduction on parallel robots, on their history and their applications can be found in the first chapter of [Merlet 2006]. In this thesis, only the two most widely used for industrial applications are briefly described: the *Gough* platform and the *Delta* robot.



(a) (b)

Figure 1.3 – The tire testing machine developed by Gough on the left, and a modern flight simulator on the right [Airbus 2010].

Although manipulators with parallel architectures were already known before its appearance, the Gough platform is universally recognized as the first parallel robot with a successful industrial application and it is still one of the most famous. Its name derives from its inventor Vernon Eric Gough, who in 1955 developed a tire testing machine [Gough 1956] (shown in Fig. 1.3a) consisting of a hexagonal platform that moves with respect to a fixed frame by 6 links, each one equipped with a linear actuator and connected to the moving platform with a spherical joint and to the fixed base with a universal

joint. Thanks to this platform, tire position and orientation were precisely controlled like the forces acting upon it. High versatility and stiffness led the Gough platform to be successfully employed in many other domains: flight simulators (like the one depicted in Fig. 1.3b), drilling and milling machines, micro-positioning devices, haptic interfaces, etc.

Another successful model of a parallel manipulator is the Delta robot. It has a limited number of degrees of freedom and it was invented by Clavel in 1988 ([Clavel 1988]). It consists of three links, each one with a parallelogram linkage connected, by means of two revolute joints on its short sides, to the end-effector and to a motorized lever. The Delta is generally equipped with 3 motors and consequently has 3 dofs that sometimes rise to 4 when the end effector is equipped with a rotating head (like the one in Fig. 1.4).

Figure 1.4 – The Delta *IRB 360 FlexPicker* by ABB®.

The Delta became very popular for pick-and-place operations thanks to its high end-effector accelerations (more than 100 m/s^2). Its geometry and architecture allows the Delta to have pure translational motion in the cartesian space, and also the solution of its inverse and forward kinematics is eased by this decoupling between positioning and rotating motion of the moving platform. It is the most common parallel machine for pick-and-place operations.

Parallel robots nowadays have a great variety of architectures and they are successfully employed in several domains: in medicine as surgery devices, as simulator platforms, in manufacturing as milling CNC machines, as mirror positioning systems in telescopes, etc.

As far as the disadvantages of parallel manipulators are concerned, the main one is a reduced workspace compared to their dimensions. The causes are multiple:

- most kinematic chains constituting robot legs have links connected with universal or spherical joints which generally have restricted motion ranges;
- actuators usually have poor strokes and consequently they can provide only limited displacements to the end-effector;
- robots legs can be quite bulky, especially when high power and stiffness are required, leading to a high probability of interference between legs or between legs and platform

An alternative architecture that can overcome this issue has attracted researchers' interest in the last two decades: cable robots. These manipulators can control the end-effector using cables as actuators and the next section describes this family of robots in more detail.

1.2 Cable robots

If rigid body legs are replaced by cables in the kinematic chains connecting the base with the moving platform of a parallel manipulator, then it is denoted as a *cable-driven parallel robot* (CDPR). Thus the end effector pose can be controlled by varying the cable lengths.

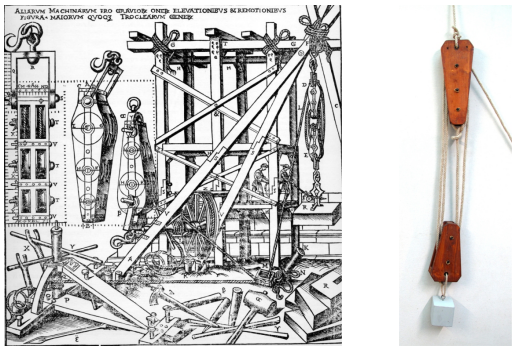
There are several advantages provided by this change in the actuation scheme. First of all, since cables are flexible they can be easily coiled and uncoiled so that actuator strokes in cable robots are way larger than in traditional rigid leg parallel machines. Even though they are flexible, cables can have also a high tensile strength compared to their diameter and their linear density. Thus, adopting cables instead of traditional actuators leads to a significant reduction of moving masses and, as a consequence, to an improvement in terms of kinematic performances. Moreover, they can be more easily connected with the moving platform and thanks to their flexibility many problems caused by interferences or limited joint motion space can be overcome.

Cables also gives CDPRs a lightweight structure which makes them more attractive than traditional parallel robots in different domains and some of them are presented later in this section. Another advantage provided by cables consists in the possibility to easily connect and disconnect them to the end effector, which makes CDPRs highly deployable and reconfigurable. These properties, together with the cable flexibility and reduced mass of the system, make it possible to transport even very large cable-robots with minor effort.

Figure 1.5 – An illustration of a Greek crane during building operations.

1.2.1 History and main applications

Cables started to be used to provide motion and forces to mechanical systems a very long time ago. Traditional cranes and sophisticated block-and-tackle-like arrays of pulleys were already known and extensively employed in construction sites or shipyards (Fig. 1.5 and 1.6) by the Greeks first and the Romans later.



(a) (b)

Figure 1.6 – On the left an image from the "de Architectura" of Marcus Vitruvius Pollio and on the right an ancient block-and-tackle system.

Nowadays, cables and flexible elements in general are currently used in different mechanical domains. Besides the traditional lifting cranes, they are regularly employed also in the automotive industries and for automation purposes (like for example the *bowden cable transmission* shown in Fig. 1.7).

Cable driven chains are employed also in robotics and sometimes the definition of *cable-driven robots* may be misinterpreted. Indeed, cable driving

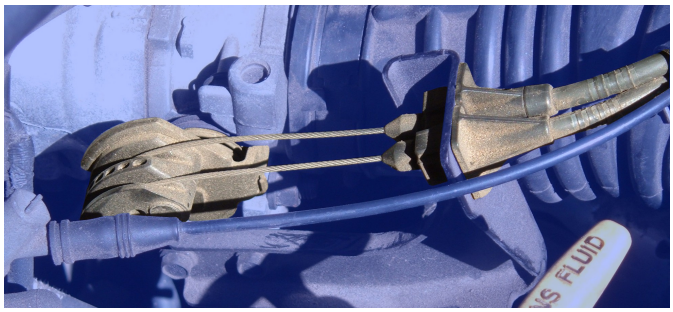
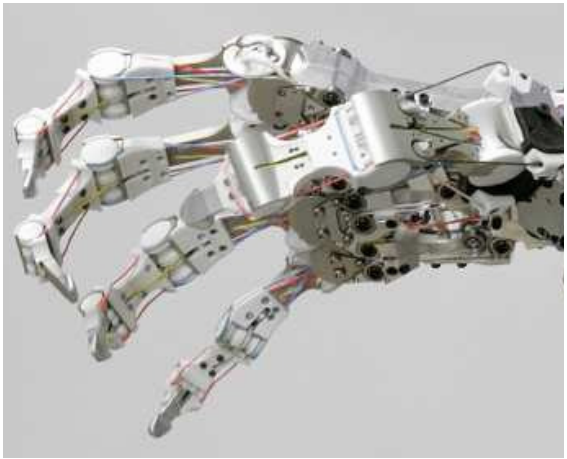


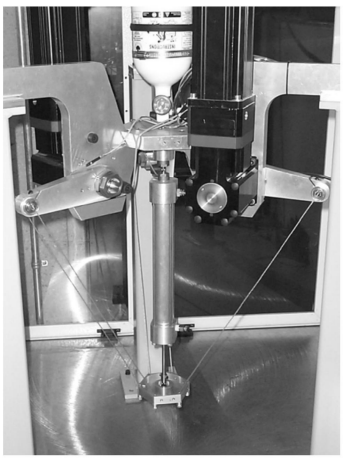
Figure 1.7 – Bowden cables controlling an automobile throttle [Ehardt 2015].

systems are successfully used in prototypes of robotic hands like the one proposed in [Grebenstein et al. 2012] and depicted in Fig. 1.8a, where they act as human tendons to drive finger movements, or in the so-called *tensegrity robots* where cables and rods are interlocked together forming rigid structures. These robots can change their shapes and stiffness or even movements [Paul et al. 2006] by controlling cables lengths.

Another example of an interesting application of cables in robotics is presented in [Dekker et al. 2006, Behzadipour & Khajepour 2006], where cables replace rigid links in a Delta parallel robot (Fig. 1.8b). This choice further reduces the mass of moving elements of the Delta architecture, but necessarily needs an additional extensible passive or active member generally called "spine" between the base and the end-effector that guarantees positive tensions in cables.



(a) The DLR robotic hand presented in [Grebenstein et al. 2012].



(b) The DeltaBot^{TM} prototype [Dekker et al. 2006].

However, even if all these manipulators comprise cables in their structures or actuation schemes they do not belong to the same of family of robots discussed in this thesis. In order to avoid possible misunderstandings, for the remaining part of this manuscript the following definition holds:

Definition. A "cable-driven parallel robot" or simply "cable robot" is a parallel manipulator having the mobile platform connected to the fixed base only by cables and its pose being controlled only by changing the lengths of the cables.

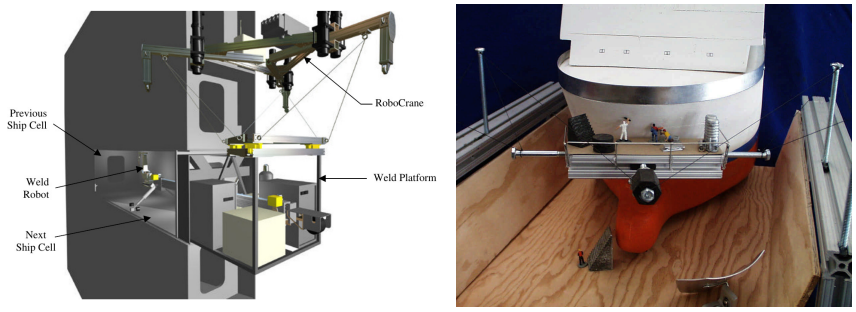
The first manipulator that complies with this definition was developed by the NIST and presented in [Albus et al. 1993]. It is a 6-dof crane (Fig. 1.8) called *RoboCrane* used for lifting and positioning operations or to control position, velocity and force of tools and machinery for cutting or excavating tasks. Among the numerous application proposed for this manipulator in [Bostelman et al. 1994], the one shown in Fig. 1.8b for airplane painting is particularly interesting: the operator works inside a maneuverable platform that uses computer-controlled cables to "float" around the aircraft. This application of the NIST RoboCrane "promises to drastically reduce paint-stripping time per airplane, cut maintenance costs and lessen incidents of operator stress and injury" ([NIST 2006]).

(a) (b)

Figure 1.8 – On the left the first prototype of the NIST RoboCrane and on the right its version for aircraft painting and maintenance ([NIST 2006]).

Another interesting field of application of the NIST RoboCrane is the shipbuilding industry [Bostelman et al. 1999], where it can be used as an appendix of traditional gantry cranes and to provide more precise control of the welding platform (see Fig. 1.9a). Another project, again developed by

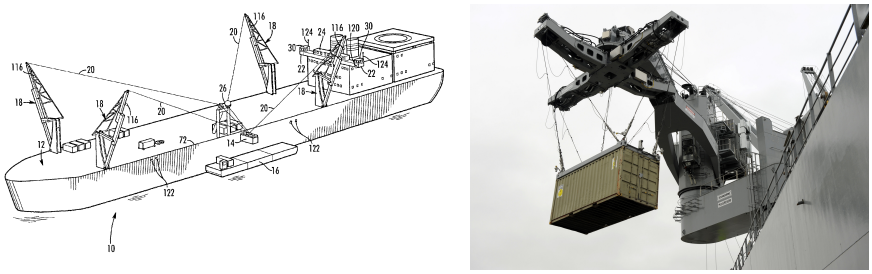
the NIST and still devoted to shipping yards, is the Flying Carpet project [Bostelman et al. 2002], where robot cranes are used to move suspended scaffolding and provide a safer worker access to ships, as shown by the model in Fig. 1.9b.



(a) (b)

Figure 1.9 – On the left the RoboCrane as a controllable welding platform and on the right its version for ship building and maintenance operations [Bostelman et al. 1999].

Figs. 1.10a and 1.10b show how cable robots can be also installed on ships as proposed by the AACTS project (Automated All Weather Cargo Transfer System) [Lee 1992] or by the U.S. Navy [Quadvlieg et al. 2011] to perform ship-to-dock or ship-to-ship operations.

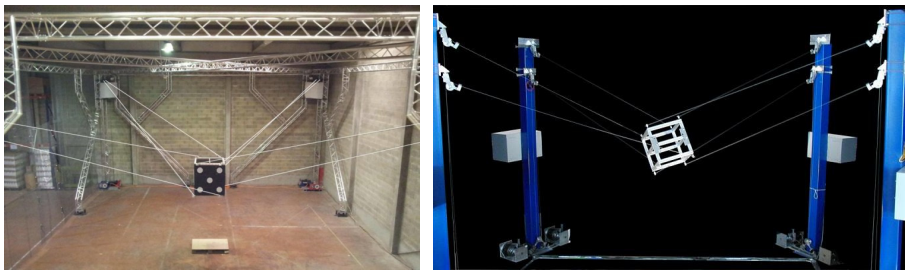


(a) (b)

Figure 1.10 – On the left a drawing representing the functioning of the AACTS ([Lee 1992]) and on the right the Large Vessel Interface Lift On/Lift Off (LVI Lo/Lo) Crane for ship-to-ship container transfer at sea (source: ONR, USA)

Performing freight management and pick and place tasks with cable robots is a potentially winning strategy, especially when operations over

large workspaces have to be performed. Several prototypes are now available and operational that can be applied in these fields and the most important are: the *CoGiRo* and the *ReelAx8* (Fig. 1.11a and 1.11b) both developed by the team Dexter at LIRMM [Dallej et al. 2011, Lamaury & Gouttefarde 2013], the IPAnema family of manipulators from the Fraunhofer Institute of Stuttgart [Pott et al. 2010, 2013] (Fig. 1.12a shows the prototype for solar power-plants installation), the portable and deployable crane realized by Merlet (Fig. 1.12b) and described in [Merlet & Daney 2010] or the cable-robot-based high-bay warehouse from the University of Duisburg-Essen [Bruckmann et al. 2013] (Fig. 1.12c).

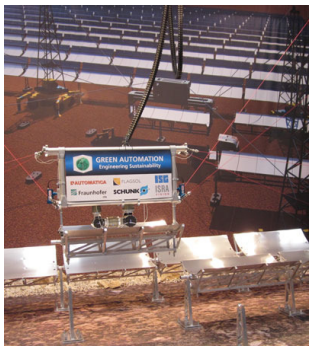


(a) (b)

Figure 1.11 – The CoGiRo robot on the left [Alexandre Dit Sandretto 2013] and the ReelAx8 on the right: the two prototypes available at the LIRMM [Lamaury et al. 2012].

Cable robots provide excellent performances also if they are used as large positioning systems. A huge 6-dof cable crane is currently in service at the NASA to perform simulated low-gravity operations (Fig. 1.13a), and the Chinese government started the construction of a giant telescope (FAST, [Wilcox 2012, Nan 2006]) that can drive the light focus cabin by means of 6 cables (Fig. 1.13b). Cables' intrinsic flexibility and compliance joined with the ease of assembly and reconfiguration make CDPRs particularly well suited also as haptic interfaces [Williams II 1998, Gallina et al. 2001, Murayama et al. 2004] and rehabilitation devices [Merlet 2010, Rosati et al. 2007, Surdilovic et al. 2007] (Fig. 1.14 shows two examples of these robots, one for each application).

Here only the most relevant prototypes and applications for cable robots are reported. Indeed, in the last 10 years CDPRs have become very popular among researchers and the number of teams and publications on this subject keeps growing. But although this technology is very promising for a large set of industrial domains, only a few are already operative and employed



(a)

(b)

(c)

Figure 1.12 – (a) The Fraunhofer Institute prototype for power solar plant construction [Pott et al. 2010], (b) the portable rescue crane Marionet Rescue by Merlet [Merlet & Daney 2010], (c) the cable-driven platform for automated warehouses from the Duisburg University [Bruckmann et al. 2013].

and currently there are no companies selling them except those producing cable-driven film and television cameras like the Skycam (Fig. 1.15) or the Spidercam.

1.2.2 Main components

In order to introduce the main subject of this thesis, a brief description of the structure of a cable robot and of the main elements which make it up it is presented hereafter.

A generic CDPR hardware, schematically represented in Fig. 1.16a, can be divided, like traditional parallel robots, in three main blocks: an end-

(a) (b)

Figure 1.13 – On the left, the Athlete vehicle during a low gravity test [Wilcox 2012] and on the right a rendering of the FAST telescope [Nan 2006].

effector or moving platform, $n > 1$ actuators and a supporting structure or framework.

The end-effector does not usually have a precise shape or geometry, indeed it depends on the particular application. It is important to remember that since cables can be easily attached and detached this is quite a difference with respect to traditional parallel and serial robots. The end-effector is, therefore, generally equipped with eyebolts, rings or has other special features to which cables will be connected (if high precision is required, universal and spherical joints are employed). For every kind of linkage it is possible to find a point, called *anchor point* or *attachment point*, that will define one end of each actuator. These points are part of the robot's geometric model and will be implemented in the control software. Thus it is very important to design and build them so that their position can be easily identified by measurements, and at the same time the reconfigurability and assembly simplicity of these robots are preserved.

Each anchor point is connected to one (or more) cable, whose length can be controlled by a motorized device. The most common cable actuator is obtained by coupling an electric motor to a simple drum that, according to its rotating direction, coils or uncoils the wire. This is probably the easiest actuation scheme, but it also has several drawbacks. First, cable coils are directly in contact and often rolled one above the other, and this fact may cause premature wire failures due to wear. Then, if wires are coiled untidily, their deployment speed may have sudden variations due to overlapping coils and slack cable loops. Moreover, overlapping strongly affects cable measurements made by reading the motor angular positions given by the encoders,

(a) (b)

Figure 1.14 – (a) The haptic interface Spidar G-G [Murayama et al. 2004] and (b) the Marionet Rehab by Merlet [Merlet 2010].

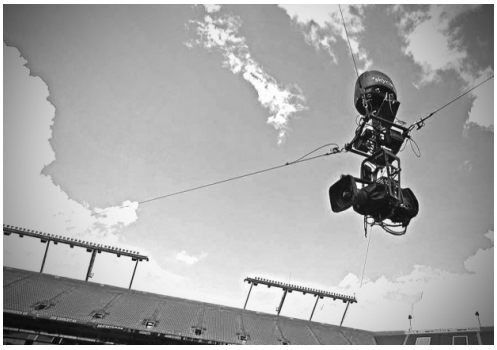
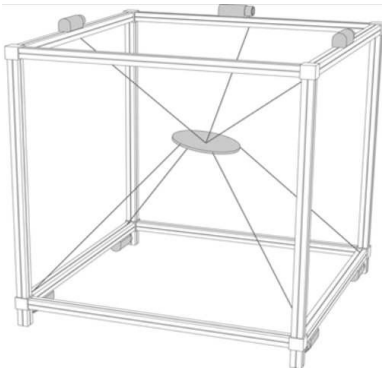


Figure 1.15 – The SkyCam [SkyCam 2015].

an essential element for accurate robot control. More refined actuators can be realized by equipping simple reels with a level-wind wire guide system (1.16b) in order to have more precise and organized cable coiling. Furthermore, the most sophisticated prototypes have added a level-wind wire guide system to specially grooved reel drums (see Fig. 1.16b). Since cable overlapping is mechanically prevented, this architecture is more suitable in all those application where higher precision (and cost) is preferred compared to minor stroke lengths. An alternative to cable winches as actuators for these robots is represented by the solution proposed by Merlet in [Merlet 2008], where cable lengths are controlled by pulley systems mounted on linear motors (Fig. 1.16).

From the actuators to the platform anchor points, cables are generally routed through pulleys, rings or more complex systems in order to define

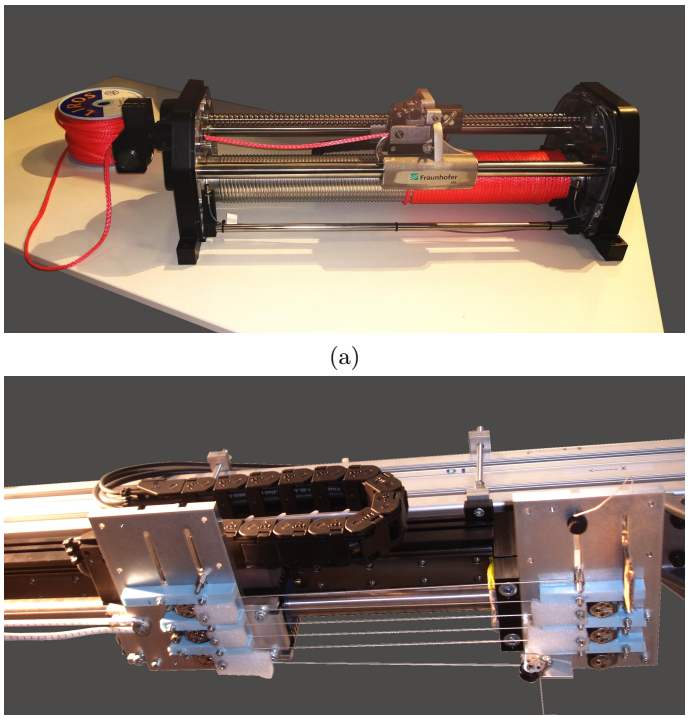


(a) Scheme of a generic cable-driven parallel robot [Bruckmann et al. 2008a].

(b) A SkyCam motorized reel [SkyCam 2015].

possibly unique and fixed points, called *exit points*, that together with the anchor points are fundamental geometric parameters for robot control. The design of these components, indeed, may have a great influence on robot performances, since exit points must allow cables to freely pass through them and at the same time their position must remain possibly the same. They are usually realized with orientable pulleys, like for example the ones shown in Fig. 1.17, or with small ceramic eyes, although this latter solution is practically feasible only in the case of small prototypes because it causes wear and abrasion issues.

Then, a few words must be spent on wires, i.e. the components that physically provide motion to the end-effector. Cable properties and application domains are well known both in mechanics and civil engineering. Cable-robot prototypes can be equipped with any sort of wires, ropes and lines but the most common choices are essentially two: traditional steel-wire ropes or aramid-fiber lines. Steel-wire ropes are generally preferred when high load capacities are required, mainly because of the better knowledge of their physical properties which makes it possible to easily pick the right diameter for given safety coefficients. However, steel cables need maintenance and constant lubrication, winches must be designed in order to be extremely resistant to wear and pitting, coil overlapping is totally undesired because it severely damages cable surfaces. Steel cables also have a high linear density, so that mass effects like sagging must be taken into account when dealing with large workspaces, and they have high linear thermal expansion coefficients which implies that cable lengths vary considerably with temperature. Aramid-fiber lines, like Kevlar or Dyneema ropes and cords,



(a)

(b)

Figure 1.16 – (a) The latest prototype of winches developed by the Fraunhofer Institute IPA [Schmidt et al. 2015] and the cable actuators with linear motors mounted on the Marionet Rehab [Merlet 2010].

have lower linear densities and practically inexistent thermal and humidity effects. They can still provide high tensile strength and stiffness within a lower weight than steel wire ropes and they guarantee lower wear issues. They can also be fabricated so as to have signal cables woven into them (like two of the four cables in the SkyCam system). The main drawback of aramid-fiber lines is the highly non-linear behavior and the lack of experience regarding their mechanical fatigue life. This latter aspect is very important in the cable-robot design process. Indeed, as previously described, for this application it is necessary to coil and uncoil cables on drums and make them pass through pulley systems. This involves a great number of bending cycles on cable fibers and demands on the one hand a careful choice of pulley and drum diameters. On the other hand, since exit points must be realized so as to be possibly modeled with a fixed point, the knowledge of the minimum bending radius allowed for a given cable in relation of the maximum number

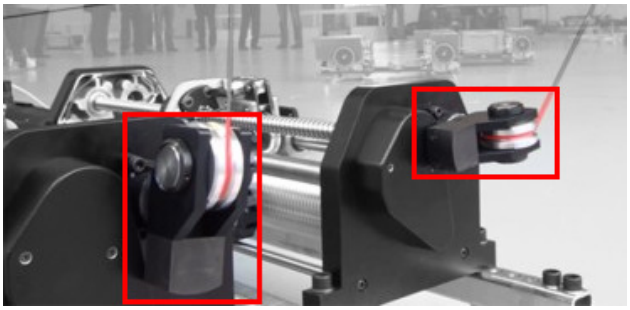


Figure 1.17 – A detail of the winches developed by the Fraunhofer Institute.

of bending cycles is very important.

1.3 Interval analysis

Since methods described in this thesis are based on interval analysis, this section provides some elements of this technique and a general overview of its possible applications in robotics. More details on this technique and its mathematical framework can be found in [Moore 1966, Jaulin 2001, Hansen & Walster 2003].

1.3.1 Basic Notions

The *real interval* $X = [\underline{x}, \overline{x}]$ is defined as the set of real numbers y such that $\underline{x} \leq y \leq \overline{x}$. When this definition is implemented on computers and numbers must be represented generally in double precision format, a real interval $X_r = \left[\underline{x_r}, \overline{x_r}\right]$ is defined as a set containing all the double precision numbers satisfying these two conditions:

- $\underline{x_r}$ is the nearest double precision number such that $\underline{x_r} \leq x$
- $\overline{x_r}$ is the nearest double precision number such that $\overline{x_r} \geq x$.

As an example, the interval representation of π on a personal computer is

$$\Pi = [3.1415926535897931, 3.1415926535897932]] \tag{1.1}$$

The extrema of an interval $\underline{x_r}$ and $\overline{x_r}$ are returned by the functions $\mathrm{Inf}(X)$ and $\mathrm{Sup}(X)$ respectively. The *width* of an interval is the quantity given by

$$\mathrm{w}(X) = \overline{x} - \underline{x}, \tag{1.2}$$

while the mean value of its extrema, i.e.

$$\mathrm{mid}(X) = (\underline{x} + \overline{x})/2 \tag{1.3}$$

is called *mid-point* of an interval.

An *interval vector* $\mathbf{B} = \{X_1, \ldots, X_h\}$, also called a *box*, is a list of intervals. The *mid-point* and the *width* of a box are vectors whose components are respectively the mid-points and the widths of their interval components:

$$\mathrm{w}(\mathbf{B}) = \{\mathrm{w}(X_1), \ldots, \mathrm{w}(X_n)\} \tag{1.4}$$

$$\mathrm{mid}(\mathbf{B}) = \{\mathrm{mid}(X_1), \ldots, \mathrm{mid}(X_n)\} \tag{1.5}$$

Interval arithmetic

The sum among intervals is defined as

$$X + Y = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right] \tag{1.6}$$

while the product between a real number and an interval is

$$aX = [a\underline{x}, a\overline{x}] \tag{1.7}$$

Consequently, the opposite of an interval is

$$-X = [-\overline{x}, -\underline{x}] \tag{1.8}$$

which yields to the definition of the subtraction between intervals

$$X - Y = \left[\underline{x} - \overline{y}, \overline{x} + \underline{y}\right] \tag{1.9}$$

If an interval X does not contain 0, the opposite of an interval can be defined as

$$\frac{1}{X} = [1/\overline{x}, 1/\underline{x}] \tag{1.10}$$

when $\underline{x} > 0$, while the product of two intervals can be computed as follows

$$XY = \left[\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})\right] \tag{1.11}$$

It is worth noting that the interval sum and product operations are *commutative* and *associative* since the following properties hold:

$$0 + X = X + 0 = 0, \quad 1X = X1 = X, \quad 0X = X0 = 0. \tag{1.12}$$

On the contrary, *distributive* property is not generally applicable. For example, let the following expression be considered

$$[1,2]\,(1-1)=0 \tag{1.13}$$

which, expanding it, leads to:

$$[1,2]\ 1-[1,2]\ 1=[-1,1]\neq 0. \tag{1.14}$$

So, the equality $X\,(Y+Z)=XY+XZ$ is not always true. However, the *distributive* property holds in the following cases:

$$a\,(Y+Z)=aY+aZ,\quad \forall a\in\mathbb{R} \tag{1.15}$$

$$X\,(Y+Z)=XY+XZ\quad \text{if } XY>0 \tag{1.16}$$

From this introduction to interval arithmetic it is possible to make some interesting considerations about this numerical technique. First of all, an interval is defined by a couple of real numbers so its representation in a computer system requires twice the memory of a floating point number. Moreover, as is clear from eqs. (1.6), (1.7) and (1.9) to (1.11), even basic arithmetical operations are more expensive from a computational point of view than traditional ones, and the product in particular requires a careful implementation in order to be effective. Also the form of expressions involving intervals has to be chosen carefully: for example, it is important to avoid, if possible, divisions between intervals since they can easily lead to overflow errors.

1.3.2 Evaluation function

Let $f(\mathbf{x})$, with $\mathbf{x}=[x_1,x_2,\ldots,x_h]$, be a function in h unknowns and $\mathbf{B}=[X_1,X_2,\ldots,X_h]$ a box comprising an interval for each unknown. The *interval evaluation* $F(\mathbf{B})$ of f over $\mathbf{B}$ is an interval $\left[\underline{F},\overline{F}\right]$ such that, for any $\mathbf{x}\in\mathbf{B}$, $\underline{F}\leq f(\mathbf{x})\leq\overline{F}$, i.e. $\underline{F}$ and $\overline{F}$ are lower and upper bounds for f when the unknowns lie within $\mathbf{B}$. An evaluation function is defined

- *convergent* if, for any sequence of boxes $\mathbf{B}_k$, if $\lim_{k\to\infty}\mathrm{w}(\mathbf{B}_k)=0$ then $\lim_{k\to\infty}\mathrm{w}(F(\mathbf{B}_k))=0$
- *minimal* if for any $\mathbf{B}$, $F\,(\mathbf{B})$ is the smallest interval containing every $f(\mathbf{x})$, $\forall\mathbf{x}\in\mathbf{B}$.
- *inclusion monotonic* if when $\mathbf{B}_1\in\mathbf{B}_2$ then $F(\mathbf{B}_1)\in F(\mathbf{B}_2)$

- the evaluation of a list of function $\mathbf{F} = [F_1, \ldots, F_p]$ is convergent only if all the F_i are $\forall\, i = 1, \ldots, p$.

Interval analysis provides several tools to implement an interval evaluation of a function [Moore 1979], but the simplest one is the *natural evaluation*, in which basic arithmetic operations and elementary mathematical functions are substituted by interval equivalents. For example, if $f(x) = x^2 - 2x + 1$ and $X = [4, 5]$, the natural evaluation of f over X is:

$$f([4,5]) = [4,5]^2 - 2\,[4,5] + 1 = \\ [16,25] - [8,10] + [1,1] = [6,17] + [1,1] = [7,18] \quad (1.17)$$

A natural evaluation function is:

- inclusion monotonic
- usually not minimal

Indeed, the bounds provided by the natural evaluation of f are generally not exact: the upper (lower) bound may be larger (lower) than the actual maximum (minimum) of the function image, namely $f(\mathbf{B}) = \{f(\mathbf{x}) | x \in \mathbf{B}\} \subseteq F(\mathbf{B})$. For example, the image of the function f in eq. 1.17 over the interval $X = [4, 5]$ is $[9, 16] \in [7, 18]$.

Such an overestimation dramatically affects the performances of interval-analysis methods. This is the reason why methods that provide the tightest interval evaluation of functions are of fundamental importance. As an example, as shown in [Merlet 2007], using the *centered form* and the *mean-value* theorem sharper bounds may be obtained:

$$f(X) \subseteq f(\mathrm{mid}(X)) + f'(X)(\mathrm{mid}(X) - X) = [8.25, 16.25] \quad (1.18)$$

Anyway this improvement came at the price of computing the derivative of the function, thus making the evaluation slightly more demanding from the computational point of view, and this is a general trend for the strategies of interval evaluation of functions. An interesting property is that the bounds of the interval evaluation F are exactly the minimum and the maximum of $f(\mathbf{B})$ when f may be expressed so as to contain a single occurrence of each unknown $x_i (i = 1 \ldots n)$ [Jaulin 2001]. On the base of this property, the expression in eq. 1.17 may be more effectively evaluated as

$$f(X) = X^2 - 2X + 1 = (X - 1)^2 = [9, 16] \quad (1.19)$$

which are the sharpest bounds for f over $X = [4, 5]$.

This property implies that when dealing with interval analysis methods, the choice of the parameterization is extremely important. Indeed, if the problem is modeled by a set of parameters that lead to expressions having a large number of multiple occurrences, the interval evaluation of each function will be heavily overestimated, affecting algorithm performances.

In summary, it is possible to extend every mathematical function to interval analysis but its implementation has to be done carefully. As mentioned before, the distributive property does not generally hold so it is better to simplify expressions manually. Moreover, function interval extensions are more computationally expensive with respect to traditional ones implemented in computer systems since they are built so as to provide the tightest, but at the same time guaranteed, bounds. As a consequence, natural function evaluation is the best choice in order to limit the computational burden, but at the same time functions have to be formulated so as to contain the minimum number of multiple occurrences, which is not always possible. However, when dealing with interval analysis based methods, if multiple parameterizations are available, it can be better to consider simpler expressions, even at a price of a larger number of unknowns, and avoid overestimation on the function evaluation.

1.3.3 Problem-solving algorithm

Interval analysis may be applied for different problems in robotics, and several procedures based on interval analysis were successfully developed in this domain (for a detailed discussion, see [Merlet 2009]). The main part of the work presented in this thesis is aimed at solving systems of nonlinear equations, so that only the algorithm for this particular application is discussed hereafter.

The advantages provided by interval methods are many. One of the most important is the capability to obtain results guaranteed against numerical errors. Indeed, there are libraries that allow basic interval analysis tools to be easily implemented on computer systems. Moreover, interval-analysis algorithms have the advantage of directly searching for real solutions, which are the only ones of practical interest. In addition to that, interval analysis searches for solutions only within a predetermined domain. This property turns out to be extremely useful in robotic applications, and in particular when dealing with kinematics problems. Indeed, for any kind of robot, the solutions of the forward kinematics have to lie within the specific workspace of the robot or, when inverse kinematics is considered, within the ranges of motion of its actuators. As will be explained below, in the particular case of

CDPRs, taking advantage of this peculiar feature of interval-analysis-based algorithms, only those solutions with positive cable tensions can be obtained.

The simplest problem-solving algorithm that can be derived from interval arithmetic relies on a simple branch-and-bound (or branch-and-prune) scheme, but this approach generally leads to unsatisfying performances. Indeed, a great number of tools and techniques must be integrated in order to develop an effective algorithm. This fact represents one of the main drawbacks of this technique, since choosing the right heuristics and the best way to implement them has, on the one hand, a dramatic impact on the performances, but on the other it requires a lot of experience, and the literature devoted to the applications of interval analysis in robotics is limited.

The structure of the algorithm used in the code to solve a system of n equations in n unknowns is as follows. Let $\mathbf{B}_1 = [X_1, X_2, \ldots, X_n]$ be a box and $\mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})] = \mathbf{0}$ a vector equation to be solved within $\mathbf{B}_1$. $\mathcal{L}$ is a list of boxes, initially set as $\mathcal{L} = \{\mathbf{B}_1\}$. An index i, initialized to 1, indicates which box $\mathbf{B}_i$ in $\mathcal{L}$ is currently being processed, while N denotes the number of boxes in $\mathcal{L}$. $\mathcal{S}$ is another list, initially empty, storing the solutions. The interval evaluation of f_j over $\mathbf{B}_i$ is denoted as $F_j(\mathbf{B}_i)$, with $j = 1, \ldots, n$.

A key component of the algorithm is the evaluation operator $\mathcal{E}$, which takes a box $\mathbf{B}_i$ as an input and it returns:

- 1, if for any j, both $\mathrm{w}(F_j(\mathbf{B}_i))$ is smaller than a given threshold ε and $F_j(\mathbf{B}_i)$ includes 0; in this case, $\mathbf{B}_i$ is deemed a solution and it is stored in $\mathcal{S}$;
- -1, if $F_j(\mathbf{B}_i)$ does not include 0 for at least one j;
- 0, otherwise.

A poorly chosen parameterization and a consequent large overestimation of interval functions in the system of equations will lead thus to a large number of bisections in order to assess whether a box may contain a solution or not. On the other hand, using a redundant parameterization to avoid multiple occurrences means a greater number of functions to be evaluated at each step and a larger number of bisections to be made to meet the stopping criterion of the algorithm for a possible solution box. Another key component is the filter operator $\mathcal{F}$, which takes a box as an input and it returns:

- -1, if there is no solution in the input box;
- a box smaller than the input one, if the filter determines that the removed part of the input box cannot contain a solution;

Algorithm 1 Problem-solving algorithm scheme

1: $i = 1$, $N = 1$, $\mathcal{L} = \{\mathbf{B}_1\}$, $\mathcal{S} = \{\}$
2: **if** $i > N$ **then return** $\mathcal{S}$;
3: **end if**
4: **if** $\mathcal{F}(\mathbf{B}_i) = -1$ **then**
5: $i = i + 1$
6: **go to** 2
7: **else**
8: $\mathbf{B}_i = \mathcal{F}(\mathbf{B}_i)$
9: **end if**
10: compute $\mathcal{E}(\mathbf{B}_i)$
11: **if** $\mathcal{E}(\mathbf{B}_i) = -1$ **then**
12: $i = i + 1$
13: **go to** 2
14: **end if**
15: **if** $\mathcal{E}(\mathbf{B}_i) = 1$ **then**
16: add $\mathbf{B}_i$ to $\mathcal{S}$
17: $i = i + 1$
18: **go to** 2
19: **end if**
20: **if** $\mathcal{E}(\mathbf{B}_i) = 0$ **then**
21: pick x_k such that $\mathrm{w}(X_k) = \max(\mathrm{w}(\mathbf{B}_i))$
22: bisect X_k in the middle point
23: create two new boxes $\mathbf{B}_i'$ and $\mathbf{B}_i''$ from $\mathbf{B}_i$
24: replace $\mathbf{B}_i$ with $\{\mathbf{B}_i', \mathbf{B}_i''\}$ in $\mathcal{L}$
25: $N = N + 1$, $i = i + 1$
26: **go to** 2
27: **end if**

- the input box, otherwise.

This is the most common implementation for the evaluation operator. However, a more sophisticated version of this operator is presented in Section 3.3.2. A box which is neither discarded nor deemed to be a solution is bisected. The strategy adopted in the code to select which variable has to be bisected consists in picking the variable contained in the box $\mathbf{B}_i$ having the largest width. The complete scheme of the algorithm is described in Algorithm 1. A strength of an algorithm of this kind is that it always terminates, since the size of a box always decreases after a bisection. Furthermore, provided that the new boxes emerging from a bisection are put at the top of the list, there is usually no problem of memory storage.

1.3.4 Application in robotics

Interval analysis represents a powerful tool for several problems in robotics. One of the main strengths is that it allows for guaranteed computations with computer systems. Indeed, when a problem is solved by a computer algorithm, its solutions are necessarily affected by errors that come from the fact that machines represent numbers only up to a certain precision.

Interval implementation in software libraries can take *round-off* and *elimination* errors into account and thanks to its arithmetic and function extensions it is possible to guarantee results coming from calculations.

The relevance of this feature can seem secondary since rounding or elimination error magnitudes are usually very small quantities. However, the literature devoted to numerical analysis is full of examples of catastrophic calculations due to floating point representation errors (some real cases are available at [Douglas 1998]). Some of these issues can be solved with minor modifications of the code, others require much more expertise, others still are impossible to predict. Interval analysis cannot obviously fix ill-conditioned algorithms but can absolutely be very useful to identify strange behaviors in solving procedures (for example returning large-width intervals as results).

Kinematics and control

Solving robot kinematics problems means dealing with systems formed by both linear and nonlinear equations. In addition, certainty of obtaining safe results, interval analysis offers several peculiar advantages with respect to other techniques when robotic systems are considered. First, interval-analysis-based methods generally do not work with complex numbers, so only real solutions are returned by its algorithms which are the only ones of

practical interest. Moreover, as will be more clearly explained in the next chapters, since interval-analysis-based algorithms require the initial search domain of the solution to be specified, it is possible to take into account geometrical and physical constraint directly in the initialization phase of algorithms. In addition to that such constraints can be implemented into the filter operator $\mathcal{F}$ in order to look directly for feasible solutions.

In control routines where solutions of forward kinematics have to be computed in very short times (a few milliseconds) the most common choice is to rely on Newton-based algorithms that are fast but not guaranteed against numerical errors nor to provide the right solutions. On the contrary, there are interval-based methods that can find, guarantee and assess the uniqueness of forward-kinematics solutions in real-time.

Another great advantage that this technique may have in robotic kinematics problems consists in considering uncertainties affecting some robot parameters directly in the solution algorithms. The pose of a robot end-effector is in practice known up to a certain precision, mainly because of clearances and/or tolerances in the assembly and manufacturing process of mechanical components. These uncertainties can be implemented in an interval analysis algorithm and from its solutions robot precision bounds may be determined.

Workspace analysis and trajectory planning

Calculating workspaces is important both during the design phase and in order to measure robot performances. However, it is generally impossible to obtain their analytical expression and they must be computed through numerical techniques. The majority of numerical procedures computing robot workspaces basically obtain it by checking the feasibility of a finite number of end-effector poses. In order to have a good approximation of the workspace volume this number has to be very large and thus long computation times may be required. Moreover, it is very difficult with this approach to detect whether the workspace contains small unfeasible regions or singularities. Through interval analysis, instead, it is possible to obtain workspace approximation as a list of boxes in which all the end-effector poses contained within are feasible. With the same approach the complete feasibility of a given trajectory can be assessed and it is also possible to include information about physical interferences between the robot structure and the environment or self collisions or uncertainties in robot geometric parameters.

Calibration

Once the trajectory planning is completed and a given task is assigned to the robot, the controller starts transmitting to the motors the appropriate motion law obtained by means of the inverse kinematic model. Thus, the kinematic model has a great influence on robot performances. However, a manipulator is a mechanical system whose behavior can be affected by a large number of phenomena and uncertainties that are extremely hard or even impossible to include in the model. The main origins of these uncertainties are errors in the assembling or manufacturing of robot components, their deformation due to mechanical stress or temperature variations, and joint backlash. The influence of these factors on robot precision and accuracy can be considerably reduced by means of the calibration process. It basically consists in determining a more accurate value of the geometrical parameters of the robot, the input being external measurements of the end-effector pose at various locations. Interval-analysis-based techniques, as shown in [Daney et al. 2004, 2006, Wu & Rao 2007], gave very interesting results on robot calibration routines, and recently some effective algorithms and methods were developed for CDPRs [Alexandre Dit Sandretto 2013].

Robot design

Besides the analysis problems, i.e. determining for a given robot its performances and/or verifying if they fulfill the requirements even in the presence of uncertainties, interval analysis can be helpful during the design stage, also called *synthesis*, of a robot. Design algorithms usually employ optimization procedures to numerically determine a single set of geometric parameters that minimize (or maximize) a user defined *cost function*. However, the choice of the best cost function and its implementation are difficult tasks. Through the interval-analysis based approach it is possible to obtain certified continuous sets of solutions that satisfy all requirements and also allow the management of uncertainties in physical realization. The structure of one very simple algorithm for these purposes proceeds as follows. Let $\mathbf{B}$ be a box of intervals in which geometric parameters can vary. Then $\mathbf{B}$ is processed by considering one system of inequalities representing the first requirement and at the end a set S_1 of feasible geometries is obtained. After this process has been repeated for every requirement, the intersection between all the feasible sets, if it exists, is the set of configurations satisfying design constraints. More information and details about this kind of applications can be found in [Merlet 2009].

1.3.5 Advantages and disadvantages

Interval-analysis strengths and critical issues are summarized here. The first advantage is that interval operations are guaranteed against rounding and elimination errors and so are the procedures based on this technique. Since all the interval-based algorithms rely on branch-and-bound-like techniques and they stop when certain criteria on box widths are attained, they always terminate. Another advantage is the possibility to specify the solution search domain and include constraints within the solving procedure. It can manage problems affected by uncertainties and its algorithms are particularly well suited for parallel computing implementations.

The main drawback of this technique is the lack of knowledge and literature about its methods and its possible applications. Another disadvantage of interval-analysis algorithms is that their effectiveness cannot be a-priori determined. This fact, combined with the aforementioned lack of expertise about this method in certain domains (e.g. robotics), often leads to very poor performances. Moreover the number of parameters, methods, and heuristics that can strongly affect its effectiveness is very high. As a consequence, satisfying implementations of interval-analysis-based methods are often difficult to achieve.

Chapter 2

Geometrico-static model for CDPRs

2.1 General definitions

This chapter presents how a generic cable-driven robot was modeled in order to develop the methods described in this thesis and describes the main assumptions made. To do so, a few definitions about CDPRs are now introduced.

Cable robots can be grouped into two families: *fully-constrained* and *underconstrained*. This distinction depends on how many degrees of freedom of the end-effector can be controlled by robot actuators. The end-effector of a generic robot that operates in a three-dimensional space has, like any other rigid body, 6 degrees of freedom. As a consequence, in order to control all of them at least 6 actuators are needed.

However, if cable flexibility provides on the one hand all the advantages characterizing cable robots, on the other hand it introduces strong nonlinearity into CDPR analysis problems. Indeed, cables can transmit only tensile forces and this implies that 6 cables will not generally be sufficient to completely control the end-effector.

Due to this fact, fully-constrained (or *completely-restrained*) cable robots are equipped with additional actuators in order to prevent cables from becoming slack [Bosscher et al. 2007, Hiller et al. 2005, Kawamura et al. 2000, Kossowski & Notash 2002, Ming & Higuchi 1994, Tadokoro et al. 2002]. Some CDPRs may not need redundant cables to be fully-constrained and this happens when a constant external force with convenient magnitude and direction acts on the moving platform, like an additional wire. This is the

case for example of *cable-suspended parallel robots* (CSPRs), i.e. CDPRs where all forces exerted by cables have no components pointing in the same direction of gravity and acting like multi-degree-of-freedom cranes (e.g. the Robocrane Fig. 1.8b, the CoGiRo Fig. 1.11a, the SkyCam 1.15. etc.). Fig. 2.1 shows, instead, a CDPR prototype with a fully-constrained architecture:

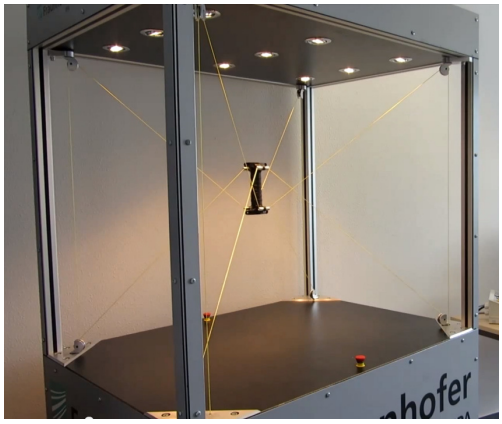


Figure 2.1 – The 8-cable haptic device developed by the Fraunhofer Institute [Yang Ho et al. 2015].

Several studies were made on fully-constrained CDPRs and CSPRs, and most of the developed prototypes have these kinds of architectures [Albus et al. 1993, Behzadipour & Khajepour 2006, Bouchard et al. 2010, Gouttefarde et al. 2011, Lamaury & Gouttefarde 2013, Kawamura et al. 2000, Merlet 2008, 2010, Pott et al. 2010, 2013, Pusey et al. 2004, Bruckmann et al. 2006, 2008a,b].

Otherwise, if the moving platform preserves some freedoms even if all cable actuators are locked, the robot is referred to as *underconstrained*. This happens, for example, when a CDPR has a smaller number of cables than the end-effector dofs. Although these robots are quite overlooked in the literature, they presents some interesting features. First, a limited number of cables implies reduced costs and easier assembly and setup operations. Second, having fewer cables means fewer interferences between cables or collisions between cables and robot framework or platform. Obviously, these advantages come at a price of reduced maneuverability, but underconstrained CDPRs may be successfully employed for all those applications where a limited dexterity is accepted in exchange for a reduced complexity of the robot.

Let, for example, a CDPR with $n = 3$ cables be considered, with the

anchor points on the platform being distinct. Thus, only 3 dofs of the end-effector can be controlled, e.g. the position of a point of the platform, while the remaining 3 (the orientation) can not. It is, however, possible to design such a CDPR and plan its trajectories in order to move the platform along the three directions while orientation deviations remain inside certain safety ranges.

Another important reason that motivates the study of this family of cable-robots is that also fully-constrained CDPRs may act like under-constrained ones if the external wrench applied to the end-effector requires a negative tension in more than $n - 6$ cables, where 6 is the number of robot dofs. Conversely, this fact may be also considered as an advantage and used to extend robot workspaces to all those regions where only a subset of cables are in tension, thus improving CDPRs reconfigurability attitudes.

Although all these reasons should encourage a careful study of under-constrained CDPRs, little attention has been dedicated to them [Collard & Cardou 2013, Fattah & Agrawal 2005, Fink et al. 2011, Gao et al. 2012, Heyden & Woernle 2006, Jiang & Kumar 2010b, Yamamoto et al. 2004, Jiang & Kumar 2010a]. This may be caused by the complexity of their kinematic study, which makes their control very difficult. As previously stated, since only n dofs may be controlled (with $n < 6$), the platform is still movable even if cable lengths are assigned. Thus the actual pose of the end-effector will be ultimately determined by the wrenches acting upon it. As a consequence, the set of equations deriving from the geometric constraints imposed by cable lengths must be simultaneously solved with the relations emerging from static equilibrium. Hereinafter, due to this coupling between kinematic and static equations, the displacement analysis of a generic CDPR will be defined as a *geometrico-static problem*. In particular the problem of finding all the possible equilibrium poses with a given set of lengths is called a direct geometrico-static problem (DGP).

2.2 Working assumptions

This section describes the main assumptions made to develop the model of a generic CDPR and the procedures are then presented in the next chapters. As described in the previous chapter, cable-robots can be built with rather simple mechanical components. However, including a realistic model of eyebolts, pulleys, winch drums and cables in particular makes geometrico-static analyses extremely complex.

CDPRs actuators are usually modeled from exit points to platform an-

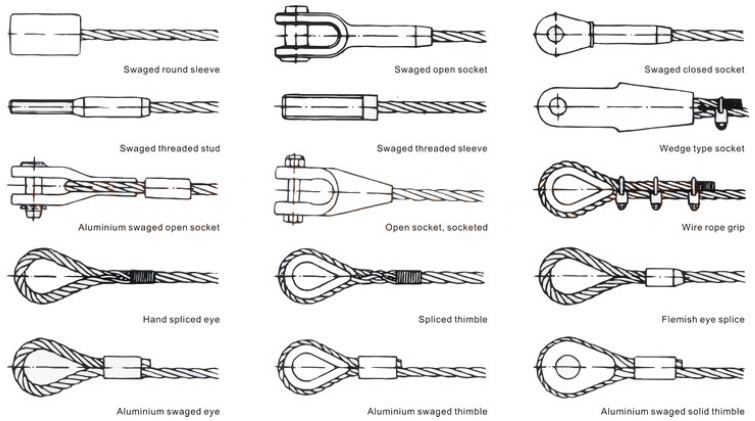


Figure 2.2 – Some wire-end terminals.

chor points like rigid body extensible legs in traditional parallel manipulators, i.e as SPS or SPU kinematic chains, where S denotes a spherical joint, P a prismatic joint and U an universal joint. Thus, anchor points on the platform are usually assumed to behave like spherical or universal joints. Indeed, in some prototypes, they are made with real universal or ball-and-socket joints. However, in order to preserve cable robot reconfigurability and ease of assembly they are more often realized with hooks or carabiners connected to eyebolts or rings, or with specific wire terminal fittings (see Fig. 2.2) or even with simple knots. As usually assumed, hereinafter anchor points are considered to be fixed to the end effector and corresponding, for example, to the center of connection rings or the end of threaded sleeves (Fig. 2.3).

The methods developed and presented in this thesis are based also on the assumption of dealing with massless and perfectly stiff cables. The main reason for this simplification is that the primary purpose of this work is to develop an algorithm that can rapidly solve the full direct-geometrico static problem of a generic CDPR, considering all the possible combinations of slack cables. This is an already complex task and the effects that both sagging and elasticity may have on it deserve more in-depth investigations. Some studies on cable elasticity are available in the literature (see [Pott 2010, Merlet 2008, 2010]) and also on kinematic analysis with real weighting cable models (see [Kozak et al. 2006, Riehl et al. 2009]). However, a general approach for the solution of the full DGP of cable robots has only recently been presented[Merlet & Alexandre-dit Sandretto 2015]. Moreover, if a real cable model is adopted, the part of cable that is comprised between each winch and the corresponding exit point should also be considered.

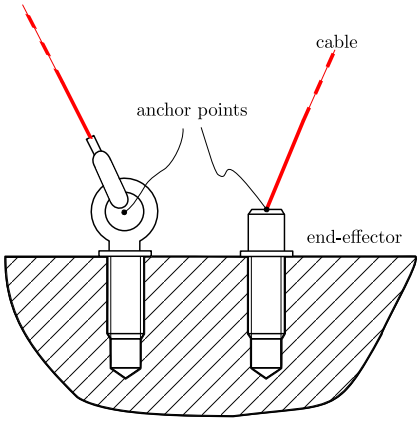


Figure 2.3 – Assumed position of the anchor points on the platform.

As previously mentioned, cable exit points are usually modeled like spherical joints, so they can be assumed as fixed points attached to the robot framework. However, this assumption is stronger than the one made for the anchor points on the moving platform. Indeed, cable exit points must allow cable length changes. Generally they are made by steering roller guidance (like the ones in Fig. 1.17) but this solution conflicts with the assumption of considering them fixed. Indeed, as shown in Fig. 2.4, the actual exit points vary according to end-effector pose [Gouttefarde et al. 2014, Bruckmann et al. 2008a]. A possible solution consists in routing cables through fixed rings with slightly bigger diameters than the cables. However, this solution can be adopted only for small prototypes and very light loads. Otherwise, friction effects severely modify cable behavior, thus making the analysis even more complex with respect to the one with roller guidances.

2.3 Geometrico-static model

In CDPRs, the unilateral nature of the constraints imposed by cables does not allow one to guarantee that in every configuration all cables remain taut. As a consequence, in order to obtain the complete set of possible solutions of the problem, all configurations with one or more slack cables must be considered. Accordingly, for a generic n-n CDPR, a complete solution of its DGP comprises not only the equilibrium configurations with n cables in tension, but also those with $1, 2, \ldots, n-1$ cables being slack. Hence, the

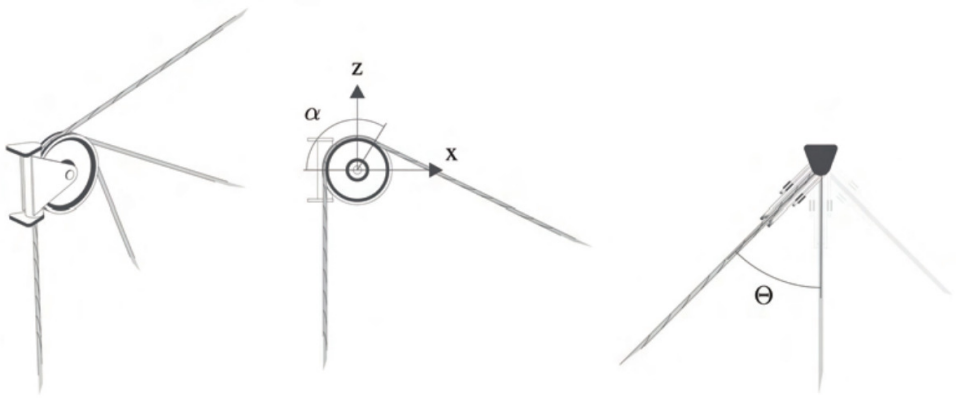


Figure 2.4 – Displacements of exit points when orientable roller guidances are employed [Bruckmann et al. 2008a].

number of DGPs that must be solved is

$$N_{DGP} = \sum_{k=1}^{n} \frac{n!}{k!\,(n-k)!} \tag{2.1}$$

This section describes how the DGP is modeled for robots with m cables in tension, $m \leq n$, and which parameterizations and equation sets are used for each value of m. Hereinafter, an equilibrium configuration is defined *admissible* if all cable tensions are positive or zero, and *feasible* if it is also stable.

2.3.1 Fundamental geometric and static equations

A mobile platform is connected to a fixed base by n cables. The ith cable, $i = 1, \ldots, n$, exits from the base at point A_i, and it is connected to the platform at point B_i (Fig. 2.5). The platform is acted upon by a force of constant magnitude Q applied at point G, e.g. the platform weight acting through its center of mass. This force is described as a 0-pitch wrench $Q\mathcal{L}_e$, where $\mathcal{L}_e$ is the normalized Plücker vector of its line of action. $Oxyz$ is a Cartesian coordinate frame fixed to the base, and $O'x'y'z'$ is a Cartesian frame attached to the end-effector (vector components in $O'x'y'z'$ are denoted with the prime mark). Without loss of generality, the fixed reference frame is chosen in such a way that $O \equiv A_1$, A_2 lies on the xz plane and z is directed as $\mathcal{L}_e$. Moreover $\mathbf{a}_i = A_i - O = [a_{ix}, a_{iy}, a_{iz}]$, $\mathbf{b}_i = B_i - O = [x_i, y_i, z_i]$, $\mathbf{g} = G - O = [g_x, g_y, g_z]$,

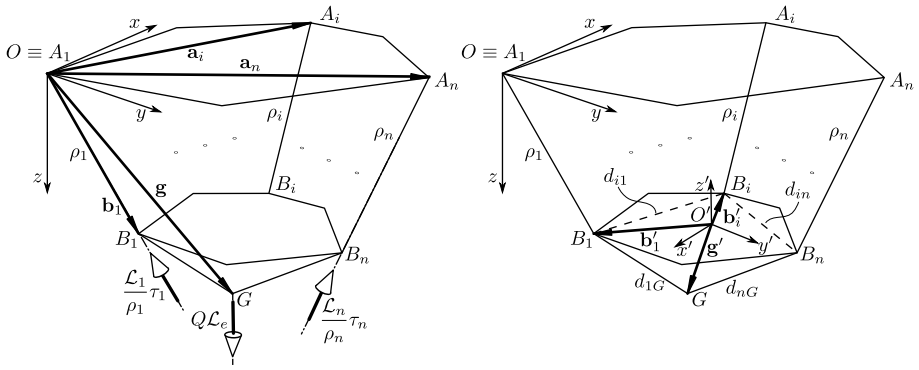


Figure 2.5 – The geometric model of a cable driven parallel robot with n cables.

$\mathbf{b}_i' = B_i - O' = [x_i', y_i', z_i']$, $\mathbf{g}' = G - O' = [g_x', g_y', g_z']$, $d_{iG} = \|\mathbf{b}_i' - \mathbf{g}'\|$, and $d_{ij} = \|\mathbf{b}_i' - \mathbf{b}_j'\|$.

If ρ_i is the assigned length of the *ith* cable (taken as strictly positive), when m cables are *active* (i.e. in tension) the set of geometrical constraints imposed on the platform is formed by the following relations:

$$\|\mathbf{b}_i - \mathbf{a}_i\| = \rho_i, \quad i = 1, \ldots, m \tag{2.2}$$

Since the platform has 6 dofs, if $m < 6$ its pose is ultimately determined by mechanical equilibrium. The normalized Plücker vector of the line associated with the ith cable is $\mathcal{L}_i/\rho_i$, where, in axis coordinates, $\mathcal{L}_i = \big[(\mathbf{a}_i - \mathbf{b}_i)\,; \quad \mathbf{p}_i \times (\mathbf{a}_i - \mathbf{b}_i)\big]$ and $\mathbf{p}_i$ is any vector from an arbitrarily-chosen reference point P (called, for brevity, *moment pole*) to the cable line. Accordingly, the wrench exerted by the ith cable on the platform is $(\tau_i/\rho_i)\mathcal{L}_i$, with τ_i being a positive scalar representing the intensity of the cable tensile force. Static equilibrium may then be expressed as

$$\underbrace{\begin{bmatrix}\mathcal{L}_1 & \ldots & \mathcal{L}_m & \mathcal{L}_e\end{bmatrix}}_{\mathbf{M}(P)} \begin{bmatrix}\tau_1/\rho_1 \\ \vdots \\ \tau_m/\rho_m \\ Q\end{bmatrix} = \mathbf{0}\,, \tag{2.3}$$

with $\tau_i \geq 0$, $i = 1, \ldots, m$.

2.3.2 Parameterization

The choice of the parameters has a great influence on the effectiveness of interval-analysis-based methods. Indeed, if on the one hand it is better to

have a small number of unknowns in order to limit the number of boxes generated by the bisection process that have to be processed, on the other hand a greater number of variables generally means simpler expressions which lead to lower overestimation of the system equations.

Before introducing the adopted parameterization for the geometrico-static model of cable robots, it is interesting to review the parameterizations tested during the development of the algorithms and list their main issues. Although the set of parameters currently adopted was derived from the one presented in [Merlet 2004], the first part of the research was focused on reviewing other parameterizations and measuring their performances when dealing with DGPs of cable robots.

Representation of the robot pose

Let $\mathbf{X}$ be the list of variables representing the end-effector pose. The first parameterization tested consists in 3 parameters that are the components of the position vector $\mathbf{p}$ of a point of the platform with respect to the fixed frame $Oxyz$, and 3 angles $\mathbf{\Theta} = [\phi_x, \phi_y, \phi_z]$ (e.g. the Euler angles for three consecutive rotations about x', y' and z' axis) describing the end-effector orientation. So, by this parameterization $\mathbf{X} = \left[\mathbf{p}^{\mathrm{T}}, \mathbf{\Theta}^{\mathrm{T}}\right]$, equations (2.2) for a generic cable i becomes

$$\|\mathbf{p} + \mathbf{R}(\mathbf{\Theta})\mathbf{b}'_i - \mathbf{a}_i\| = \rho_i \tag{2.4}$$

and, expanded, has the following expression:

$$\begin{aligned}
&\big[p_x + \cos\phi_y \left(\cos\phi_z b'_{ix} - \sin\phi_z b'_{iy}\right) + \sin\phi_y b'_{iz} - a_{ix}\big]^2 \\
&\qquad + \big[p_y + \left(\sin\phi_x \sin\phi_y \cos\phi_z + \cos\phi_x \sin\phi_z\right) b'_{ix} \\
&\quad + \left(-\sin\phi_x \sin\phi_y \sin\phi_z + \cos\phi_x \cos\phi_z\right) b'_{iy} - \sin\phi_x \cos\phi_y b'_{iz} - a_{iy}\big]^2 \\
&\qquad + \big[p_z + \left(-\cos\phi_x \sin\phi_y \cos\phi_z + \sin\phi_x \sin\phi_z\right) b'_{ix} \\
&\quad + \left(\cos\phi_x \sin\phi_y \sin\phi_z + \sin\phi_x \cos\phi_z\right) b'_{iy} + \cos\phi_x \cos\phi_y b'_{iz} - a_{iz}\big]^2 = \rho_i^2
\end{aligned} \tag{2.5}$$

However, this parameterization, which is quite common in robotics, presents two main disadvantages when dealing with interval analysis methods. The first one consists in the great number of multiple variable occurrences in the equation system. Indeed, in (2.5) there are 32 total occurrences of the six $\mathbf{X}$ parameters, i.e.:

- 1 for p_x, 1 for p_y, 1 for p_z,

- 10 for ϕ_x, 9 for ϕ_y, 10 for ϕ_z.

This means, as explained in Sec. 1.3, that interval evaluations of these expressions are affected by large overestimations and, thus, operators $\mathcal{F}$ and $\mathcal{E}$ take more time to assess whether a box can can be a solution or not. Moreover, equations coming from (2.3) are even more involved and have higher numbers of variable occurrences than (2.5).

The second issue is that determining the tightest bounds for the initial search domain of $\boldsymbol{\Theta}$, which have a great influence on algorithm performances, is a challenging task. By this parameterization, the DGP with a CDPR with m cables in tension is formed by a system of $m+6$ relations in the $6+m$ unknowns $\left[\mathbf{X}, \boldsymbol{\tau}^{\mathrm{T}}\right]$.

As a first attempt to deal with these issues, the following substitutions were made

$$\begin{aligned}
\cos(\phi_x) = c_x;\ \sin(\phi_x) = s_x;\\
\cos(\phi_y) = c_y;\ \sin(\phi_y) = s_y;\\
\cos(\phi_z) = c_z;\ \sin(\phi_z) = s_z;
\end{aligned}$$

and consequently the three trigonometric identities were added to the system of equations

$$c_x^2 + s_x^2 = 1; \tag{2.6}$$
$$c_y^2 + s_y^2 = 1; \tag{2.7}$$
$$c_z^2 + s_z^2 = 1; \tag{2.8}$$

The system now has $m+6+3$ equations in the $9+m$ unknowns $\mathbf{Y} = [\mathbf{p}^{\mathrm{T}}, c_x, s_x, c_y, s_y, c_z, s_z, \boldsymbol{tau}^{\mathrm{T}}]$. By this parameterization, (2.4) has a simpler expression

$$\begin{aligned}
&\left[b'_{ix}c_yc_z - b'_{iy}c_ys_z + b'_{iz}s_y - a_{ix} + p_x\right]^2\\
&\quad + \left[p_y + (c_zs_xs_y + c_xs_z)b'_{ix} + (-s_xs_ys_z + c_xc_z)b'_{iy} - s_xc_yb'_{iz} - a_{iy}\right]^2\\
&+ \left[p_z + (-c_xc_zs_y + s_xs_z)b'_{ix} + (c_xs_ys_z + c_zs_x)b'_{iy} + c_xc_yb'_{iz} - a_{iz}\right]^2 - rho_i^2
\end{aligned} \tag{2.9}$$

but there are still 32 occurrences of variables. However, while p_x, p_y and p_z still appear only once, the number of multiple occurrences for orientation parameters drops to

- 5 for c_x, 4 for c_y, 5 for c_z,

- 5 for s_x, 5 for s_y, 5 for s_z.

This not only reduces overestimation in function evaluation, but makes it possible to easily set bounds on the orientation variables. Indeed, each $c_{[x,y,z]}$ and $s_{[x,y,z]}$ can be initialized with the interval $[-1, 1]$. However, also for these it is difficult to find the tightest initial search domain and their width still has a great influence on computation times.

Better results were obtained by describing platform orientation with 4-parameter representations. Quaternions, Euler-Rodrigues, and axis-angle parameterization were tested and they all shown similar performances.

However, none of the listed parameterizations gave satisfying results (the solution of a single DGP with these parameterizations takes more than 25 minutes), even if multiple occurrences for all the described parameterizations may be reduced by rewriting and manipulating the equations.

All these expression, indeed, were not implemented exactly as they are usually written in the literature because this does not assure the equation expressions to have the lowest number of multiple occurrences. Thus, a great deal of expertise and tests are required in order to choose the right set of parameters for an interval-analysis-based method. Moreover, all these parameterizations were all repeatedly tested during the development of the codes because it may happen that improvements on filter and evaluation operators $\mathcal{F}$ and $\mathcal{E}$ make older parameterizations more effective than the current one.

Formulation of the static constraints

The DGP for CDPRs requires simultaneously solving the relations emerging from both the geometrical and the static constraints. The m unknowns of cable tensions (where m is the number of taut cables) may be eliminated from the system of equations by observing that the system in (2.3) is linear in $\boldsymbol{\tau}$. Thus, m linearly independent relationships may be selected with the system and solved for the tensions. The expressions calculated this way may be substituted back into the remaining $6 - m$ equation in (2.3), and, added to the (2.2), form a system in $\mathbf{X}$ only.

An alternative, more elaborate, strategy to eliminate cable tensions, presented in [Carricato 2013], may be designed by observing that (2.3) admits a solution only if

$$\text{rank}[\mathbf{M}(P)] \leq m \tag{2.10}$$

Hence, by setting all $(m + 1) \times (m + 1)$ minors of $\mathbf{M}(P)$ equal to zero and by conveniently changing the moment pole, a set of linearly independent

relations only comprising the platform-pose variables may be derived, i.e.

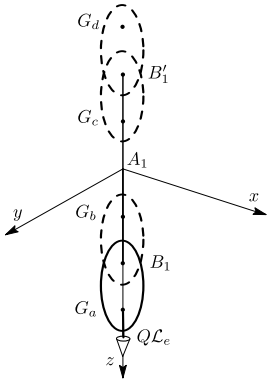
$$p_k(\mathbf{X}) = 0, \quad k = 1, \ldots, N_p \tag{2.11}$$

where N_p is an integer significantly larger than the number N_X of variables contained in $\mathbf{X}$. For the DGP to admit a solution, the above equations must be dependent, though in a non-linear way. When added to the equations deriving from cable length constraints (2.2), (2.11) allows the pose $\mathbf{X}$ to be directly solved.

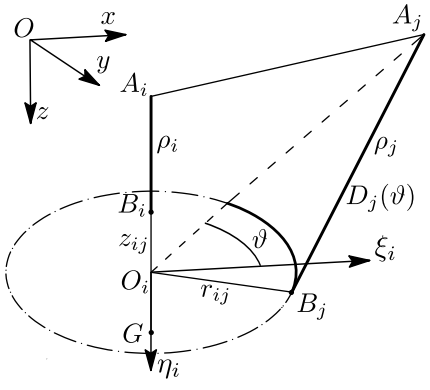
However, eliminating tension with any of the two approaches presented, as described in [Berti et al. 2013], is not convenient. The first reason is that, by doing this, the algorithm will search for all the equilibrium configurations, including the ones having negative cable tensions. It is still possible to filter them by adding specific procedures to the operator $\mathcal{F}$ but this will increase the computational burden. Moreover, the equations deriving from both the approaches are remarkably more complex than the one in (2.3), and in particular they have a higher degree, more terms and a larger number of variable multiple occurrences. Again, this causes large interval evaluations due to overestimation and poor performances of the algorithm.

2.3.3 DGP for robots with one cable in tension

When the platform is suspended by a single cable, under the effect of a force $Q\mathcal{L}_e$, it has 4 equilibrium configurations (Fig. 2.6a), but only in two of them, i.e. G_a and G_b, is cable tension positive. Obviously, the only stable solution is the one having the minimum potential energy, i.e. G_a.



(a) The 4 possible solutions of the DGP with one active cable.



(b) Geometric model of the DGP with one cable in tension.

When only cable i is in tension, at equilibrium the end-effector can still rotate along the z axis. However, for $n \geq 2$, this rotation is bounded by the other cables attached to the platform (Fig. 2.6b). The distance D_j between points B_j and A_j, $j \neq i$, depends on the rotation angle ϑ and has the following expression:

$$D_j(\vartheta) = (a_{ix} + r_{ij}\cos\vartheta - a_{jx})^2 + (a_{iy} + r_{ij}\sin\vartheta - a_{jy})^2 + (a_{iz} + \rho_i + z_{ij} - a_{jz})^2 \tag{2.12}$$

where O_i is the projection of B_j on line B_iG, and quantities

$$\begin{aligned} z_{ij} &= \|B_i - O_i\| = \frac{d_{ij}^2 - d_{jG}^2 + d_{iG}^2}{2d_{iG}} \\ r_{ij} &= \|B_j - O_i\| = \sqrt{d_{ij}^2 - z_{ij}^2} \end{aligned} \tag{2.13}$$

only depend on the platform geometry. The minimum and maximum values of $D_j(\vartheta)$ are obtained by setting $\partial D_j/\partial\vartheta = 0$, which yields

$$\vartheta_{Dj\,min}, \vartheta_{Dj\,max} = \arctan\left(\frac{a_{jy} - a_{iy}}{a_{jx} - a_{ix}}\right) + k\pi, \;\; k \in \{0, 1\}, \tag{2.14}$$

where $D_{j\,min} = D_j(\vartheta_{Dj\,min})$, $D_{j\,max} = D_j(\vartheta_{Dj\,max})$ and $D_{j\,min}$ and $D_{j\,max}$ are distinguished by evaluating the corresponding sign of the second derivative

$$\frac{\partial^2 D_j}{\partial\vartheta^2} = 2r_{ij}\left[(a_{jy} - a_{iy})\sin\vartheta + (a_{jx} - a_{ix})\cos\vartheta\right] \tag{2.15}$$

The routine outlined in Algorithm 2 computes $D_{j\,min}$ and verifies whether $D_{j\,min} < \rho_j$. If the latter condition is satisfied for every $j = 1, \ldots, n$, $j \neq i$, then the equilibrium configuration with the single ith cable in tension is deemed feasible.

2.3.4 DGP for robots with two cables in tension

This problem was analytically investigated by Carricato and Merlet [Carricato & Merlet 2013], and was proven to admit at most 24 real solutions.

If the taut cables are labeled with 1 and 2, then, at equilibrium, A_1, B_1, A_2, B_2 and G must necessarily rest in a plane parallel to the z axis. Two planar operation modes are possible, characterized by opposite orientation of the vector $(\mathbf{b}_2 - \mathbf{g}) \times (\mathbf{b}_1 - \mathbf{g})$ with respect to $\mathbf{k} \times (\mathbf{a}_2 - \mathbf{a}_1)$ (Fig. 2.6). Out-of-the-plane movements, which are very likely to occur in practice, can cause operation-mode changes. The platform pose is described by the 9 components of the position vectors $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{g}$. By conveniently rotating

Algorithm 2 DGP 1-1 for the ith cable in tension

for $(j = 1 \dots n)$ **do**
  if $(j \neq i)$ **then**
    if $(a_{jx} = a_{ix})$ **and** $(a_{jy} = a_{iy})$ **then**
      $D_{j\,min} = r_{ij}^2 + (a_{iz} + \rho_i + z_{ij} - a_{jz})^2$
    else
      if $(a_{jx} = a_{ix})$ **then**
        if $(a_{jy} > a_{iy})$ **then**
          $\vartheta_{min} = \pi/2$
        else
          $\vartheta_{min} = 3\pi/2$
        end if
      else
        if $(a_{jx} > a_{ix})$ **then**
          $\vartheta_{min} = \arctan\left(\frac{a_{jy}-a_{iy}}{a_{jx}-a_{ix}}\right)$
        else
          $\vartheta_{min} = \arctan\left(\frac{a_{jy}-a_{iy}}{a_{jx}-a_{ix}}\right) + \pi$
        end if
      end if
      $D_{j\,min} = D_j(\vartheta_{min})$
    end if
    if $(D_{j\,min} > \rho_j)$ **then**
      The configuration is unfeasible
      Exit
    end if
  end if
end for
The configuration is feasible
Exit

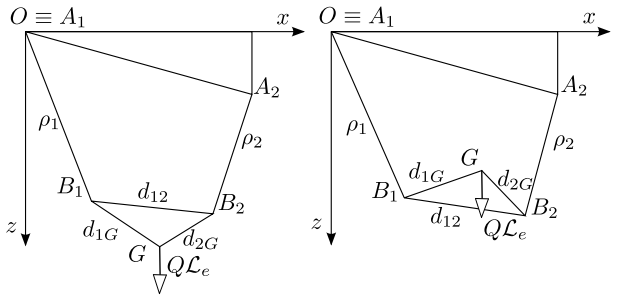


Figure 2.6 – Geometric model of the DGP with two cables in tension.

the fixed reference frame along the z axis in order to have the exit points A_1 and A_2 lying on the xz plane, the number of variables can be reduced to 6, since, at the equilibrium, the y component of $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{g}$ must necessarily vanish. These 6 parameters are not independent, since they have to satisfy the distance relations imposed by the geometry of the platform, namely:

$$\begin{aligned} \|\mathbf{b}_2 - \mathbf{b}_1\| &= d_{12} \\ \|\mathbf{b}_1 - \mathbf{g}\ \| &= d_{1G} \\ \|\mathbf{b}_2 - \mathbf{g}\ \| &= d_{2G} \end{aligned} \tag{2.16}$$

By the aforementioned parameterization, (2.2) assumes the form

$$(x_i - a_{ix})^2 + (z_i - a_{iz})^2 = \rho_i^2, \quad i = 1, 2 \tag{2.17}$$

and, by choosing O as the moment pole, matrix $\mathbf{M}$ in (2.3) may be written as

$$\mathbf{M} = \begin{bmatrix} x_1 & x_2 - a_{2x} & 0 \\ z_1 & z_2 - a_{2z} & -1 \\ 0 & a_{2z}x_2 - a_{2x}z_2 & x \end{bmatrix} \tag{2.18}$$

Substituting (2.18) into (2.3) and considering relations (2.16) and (2.17), a system of 8 equations in the 8 variables comprised in the array $\mathbf{Y} = \left[\mathbf{X}^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}}\right]^{\mathrm{T}}$ is finally obtained, where $\mathbf{X} = [x_1, z_1, x_2, z_2, x_g, z_g]^{\mathrm{T}}$ and $\boldsymbol{\tau} = [\tau_1, \tau_2]^{\mathrm{T}}$. This parameterization allows one to simultaneously search for solutions in both operation modes.

The number of unknowns (and consequently the number of equations) could be reduced by observing that the system in (2.3) admits a solution only if $\mathrm{rank}(\mathbf{M}) \leq 2$. Thus, cable tensions $\boldsymbol{\tau}$ could be eliminated from equilibrium equations by imposing $\det(\mathbf{M}) = 0$. This strategy, applied in [Carricato &

Merlet 2013], is not adopted here, because it does not allow an initial search domain for the cable tensions to be specified. In this case, negative-tension solutions should be filtered by a subsequent procedure that would increase computation time.

2.3.5 DGP for robots with three cables in tension

The geometrico-static analysis of this family of robots was investigated in [Carricato & Merlet 2011, 2013, Abbasnejad & Carricato 2012], where the DGP of a robot suspended by 3 active cables was proven to admit 156 solutions in the complex field. Preliminary results obtained by solving the same problem by interval analysis were presented in [Berti et al. 2013].

If the taut cables are labeled with 1, 2 and 3, the parameters $\mathbf{X}$ adopted to describe the platform pose are the components of the position vectors $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$. They are not independent, since the following relations must hold:

$$\|\mathbf{b}_i - \mathbf{b}_j\| = d_{ij}, \;\; i,j \in [1,2,3], \;\; i \neq j \tag{2.19}$$

The distance equations in (2.2) assume the form

$$(x_i - a_{ix})^2 + (y_i - a_{iy})^2 + (z_i - a_{iz})^2 = \rho_i^2, \;\; i = 1,2,3 \tag{2.20}$$

and the position vector of $(G - B_1)$ in $Oxyz$ may be expressed as

$$\mathbf{g} - \mathbf{b}_1 = \alpha_2 \mathbf{b}_{12} + \alpha_3 \mathbf{b}_{13} + \alpha_4 (\mathbf{b}_{12} \times \mathbf{b}_{13}) \tag{2.21}$$

where $\mathbf{b}_{ij} = \mathbf{b}_j - \mathbf{b}_i$, and α_k, $k = 2, \ldots, 4$, are known constants obtained by solving the system

$$\alpha_2 \mathbf{b}'_{12} + \alpha_3 \mathbf{b}'_{13} + \alpha_4 (\mathbf{b}'_{12} \times \mathbf{b}'_{13}) = \mathbf{g}' - \mathbf{b}'_1 \tag{2.22}$$

From (2.21), one infers

$$\mathbf{g} = \sum_{k=1}^{3} \alpha_k \mathbf{b}_k + \alpha_4 (\mathbf{b}_{12} \times \mathbf{b}_{13}) \tag{2.23}$$

where $\alpha_1 = 1 - \alpha_2 - \alpha_3$. If G lies on the same plane identified by points B_1, B_2 and B_3, α_4 could be set equal to zero and a simpler expression of $\mathbf{g}$ would be achieved.

The relations in (2.19), (2.20) and (2.3) form a square system of 12 equations in 12 unknowns, namely $\mathbf{Y} = \left[\mathbf{X}^{\mathrm{T}}, \; \boldsymbol{\tau}^{\mathrm{T}}\right]^{\mathrm{T}} = \left[\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \mathbf{x}_3^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}}\right]$. All equations are algebraic and of degree at most 2 in the unknowns.

The number of unknowns may be reduced by eliminating cable tensions. Different strategies can be used, and these were all tested and compared in [Berti et al. 2013]. However, the best parameterization proves to be the one described above, including cable tensions. This is mainly due to the reduced overestimation provided by the simpler expression of the equations, to the possibility of excluding negative-tension solutions from the search domain, and to more effective filtering procedures.

2.3.6 DGP for robots with four or five cables in tension

The DGPs for CDPRs with 4 and 5 cables in tension were analytically studied in [Abbasnejad & Carricato 2015], and they were proven to admit 216 and 140 solutions in the complex field, respectively.

Let the taut cables be labeled with $1, \ldots, m$ ($m = 4$ or 5). In the following problem-solving procedure, three different parameterizations are adopted, depending on whether:

1. the platform anchor points B_i, $i = 1, \ldots, m$, and G lie on the same plane;
2. the platform anchor points B_i, $i = 1, \ldots, m$, lie on the same plane, but G does not;
3. The platform anchor points B_i, $i = 1, \ldots, m$, do not lie on the same plane.

The variables $\mathbf{X}$ used to describe the platform pose in case 1 are the position vectors of 3 non-aligned points of the platform, say $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$. These points are called *reference points*. The remaining $m-3$ anchor points and G are denoted as *secondary points*, and they are:

$$\begin{aligned} \mathbf{b}_i &= \sum_{k=1}^{3} \beta_{ik} \mathbf{b}_k \quad i = 4, \ldots, m \\ \mathbf{g} &= \sum_{k=1}^{3} \alpha_k \, \mathbf{b}_k. \end{aligned} \tag{2.24}$$

where $\beta_{i1} = 1 - \beta_{i2} - \beta_{i3}$ and $\alpha_1 = 1 - \alpha_2 - \alpha_3$. The same approach is also adopted in case 2, except for the position vector of G, which is expressed using the relation in (2.23).

The parameters $\mathbf{X}$ in case 3 are the position vectors of 4 points of the platform (not lying on the same plane). Similarly to case 1, the remaining

secondary points can be expressed as:

$$\begin{aligned} \mathbf{b}_5 &= \sum_{k=1}^{4} \beta_{ik} \mathbf{b}_k \\ \mathbf{g} &= \sum_{k=1}^{4} \alpha_k \, \mathbf{b}_k. \end{aligned} \tag{2.25}$$

where $\beta_{i1} = 1 - \beta_{i2} - \beta_{i3} - \beta_{i4}$ and $\alpha_1 = 1 - \alpha_2 - \alpha_3 - \alpha_4$. The latter parameterization is over-redundant, since only three reference points would be sufficient to define the platform pose. However, using only 3 points leads to a more complex expression of the secondary points. The multiple occurrences of variables introduced by the cross product in (2.23) cause an overestimation in the evaluation of the secondary points, and consequently of the geometric-constraint equations. Indeed, the main reason for having three distinct parameterizations, specialized according to the geometry of the moving platform, is to be able to express each geometric constraint so as not to contain multiple occurrences of the unknown variables. Provided that l denotes the number of reference points, with $l \in \{3, 4\}$, the expressions of the distance relations imposed by cable lengths are [Merlet 2004]:

$$(x_i - a_{ix})^2 + (y_i - a_{iy})^2 + (z_i - a_{iz})^2 = \rho_i^2 \quad i = 1, \ldots, l \tag{2.26}$$

$$\left(\sum_{k=1}^{l} \beta_{ik} x_k - a_{ix} \right)^2 + \left(\sum_{k=1}^{l} \beta_{ik} y_k - a_{iy} \right)^2 + \left(\sum_{k=1}^{l} \beta_{ik} z_k - a_{iz} \right)^2 = \rho_i^2 \tag{2.27}$$

with $i = l + 1, \ldots, m$.

Similarly to (2.19), another set of C_2^l relations (namely, 3 for $l = 3$ and 6 for $l = 4$) emerges from the distance constraints between the reference points, i.e.

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = d_{ij}^2 \tag{2.28}$$

for $i, j \in \{1, \ldots, l\}$ and $i \neq j$.

For CDPRs with 4 or 5 active cables, equations (2.26), (2.27) and (2.28) are not sufficient to determine the platform pose, but they must be solved together with relations (2.3) emerging from the mechanical equilibrium, thus adding cable tensions to the variable array, i.e. $\mathbf{Y} = [\mathbf{X}^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}}]^{\mathrm{T}}$, leading to a system of $m + C_2^l + 6$ equations in $3l + m$ unknowns (for $l = 3$ or 4, $C_2^l + 6 = 3l = 9$ or 12).

2.3.7 DGP for robots with six cables in tension

The DGP for this family of robots can be solved by means of the techniques developed for the Gough-Stewart Platform. Indeed, the 6 constraints in (2.2) imposed by the cable lengths in this case are sufficient to completely determine the platform pose. In the literature, there are plenty of studies about the direct kinematics of the Gough-Stewart Platform. In our code, the DGP of a CDPR with 6 active cables is solved by the interval method presented in [Merlet 2004], and solutions with positive cable tensions are obtained by a filtering procedure (see Section 3.1.3).

Chapter 3

DGP solving procedure

The methods presented in this chapter solve the *direct geometrico-static problem* (DGP) of a generic n-n CDPR ($n \leq 6$), namely a robot in which the n exit points of cables on the base and the n cable anchor points on the platform are distinct. The DGP consists in assigning the cable lengths as input variables and in finding *all* possible equilibrium configurations, taking into account the fact that one or more cables may be slack. While the DGP of a CDPR having 6 taut cables can be dealt with using the same tools employed for the Gough-Stewart Platform, the DGP for robots with less than 6 cables in tension, is remarkably more complex. A successful implementation of the methodology proposed by Carricato and Merlet [Carricato & Merlet 2010, 2013], based on exact-arithmetic elimination procedures, allowed the DGP of 2-2, 3-3, 4-4 and 5-5 CDPRs to be solved [Carricato 2013, Carricato & Merlet 2013, 2011, Abbasnejad & Carricato 2015]. In particular, least-degree univariate polynomials in the ideals generated by the equations governing the problems were found. However, the approach used in the aforementioned contributions has the following drawbacks.

- Obtaining univariate polynomials is a challenging and not automatic process. Moreover, they have high orders (12, 156, 216 and 140, for the robots suspended by 2, 3, 4 and 5 taut cables, respectively), so that their reliable solution is very difficult, as the calculation of coefficients is very sensitive to numerical errors.

- It is not possible to incorporate constraints on the unknowns. As a consequence, all roots (both complex and real, regardless of the tension sign and stability) must be calculated and then post-processed in order to discard unfeasible ones.

Effective alternatives are provided by approaches based on floating-point arithmetic, such as homotopy continuation. Continuation methods were shown to be able to robustly provide all solutions for the DGP of a general n-n CDPR [Abbasnejad & Carricato 2012, 2015]. However, this may be quite slow and it is impossible to incorporate constraints on the unknowns.

Interval-analysis-based methods may be very effective in robotic applications [Merlet 2009, 2004] and a specific procedure for the DGP of a 3-3 CDPR was recently developed with promising results [Berti et al. 2013]. Based on these results, the described method improves the algorithm presented in [Berti et al. 2013], thus making it suitable to the DGP of a general CDPR suspended by n cables, with $n = 1 \dots 6$, under the assumption that cables are massless and perfectly stiff. As will be shown in the second text in Chap. 4, the same algorithm can also be used to solve the DGP of robots suspended by more than 6 cables, when cable elasticity is disregarded [Merlet 2012a].

3.1 Code structure

In this section, the most important parts of the interval-analysis code are discussed. The code solves any geometry of cable-robots with $n \leq 6$ cables. In the current version, the geometric parameters are assumed to be known, thus not affected by uncertainties. The ALIAS C++ library [Merlet 2007] is used. ALIAS C++ is large set of interval methods mainly devoted to the solution of robotic and kinematic problems. ALIAS C++ relies on the BIAS/Profil library for basic interval arithmetics operations in double precision.

The main routine of the code reads the geometry data of the n-n CDPR from a user-defined file, thus generating a list containing N_{DGP} sub-problems, each solving the DGP for a subset of m cables in tension, $m \leq n$. Sub-problems are processed one at a time.

3.1.1 Initial search domain

A prerogative of any interval analysis method is that it requires an initial domain where solutions are searched for. For kinematics problems, this feature is an advantage, since it allows the physical constraints of the manipulator to be taken into account, and the code performances to be drastically improved.

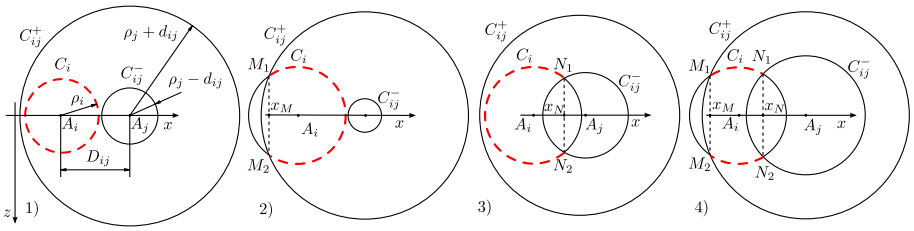


Figure 3.1 – The four different cases for the possible location of B_i according to the location of A_j, the leg length ρ_j and the distance between B_i and B_j. The admissible region zone for B_i is represented by the dashed line.

Initial bounds for the pose variables

A simple method to obtain the initial bounds of the geometrical unknowns is the following. Let d_{ij} be the known distance between two reference points B_i and B_j, $i, j \in \{1, \ldots, l\}$ and let the following quantities be defined:

$$\begin{aligned} X_{ij} &= a_{ix} + \rho_j + d_{ij} \\ Y_{ij} &= a_{iy} + \rho_j + d_{ij} \\ Z_{ij} &= a_{iz} + \rho_j + d_{ij} \end{aligned} \tag{3.1}$$

where $d_{ij} = 0$ when $i = j$. For each j, the coordinates of B_i must be (component-wise) greater than $[-X_{ij}, -Y_{ij}, -Z_{ij}]^{\mathrm{T}}$ and lower than $[X_{ij}, Y_{ij}, Z_{ij}]^{\mathrm{T}}$. Thus, the interval box containing the initial search space for reference point B_i is

$$\mathbf{B}_i = \begin{bmatrix} \left[\max\limits_{j\in\{1,\ldots,l\}} (-X_{ij}) \,,\, \min\limits_{j\in\{1,\ldots,l\}} (X_{ij}) \right] \\ \left[\max\limits_{j\in\{1,\ldots,l\}} (-Y_{ij}) \,,\, \min\limits_{j\in\{1,\ldots,l\}} (Y_{ij}) \right] \\ \left[\max\limits_{j\in\{1,\ldots,l\}} (-Z_{ij}) \,,\, \min\limits_{j\in\{1,\ldots,l\}} (Z_{ij}) \right] \end{bmatrix} \tag{3.2}$$

where $j \in \{1, \ldots, h\}$.

The bounds expressed in (3.2) may be further improved. Consider a pair of cables i, j, and denote the distance between the exit points A_i and A_j with D_{ij}. If the i-th cable is taut, B_i lies on the surface of a sphere S_i centered in A_i and having radius ρ_i. Furthermore, B_i has to lie both inside the sphere S_{ij}^+ centered in A_j with radius $\rho_j + d_{ij}$ and outside the sphere S_{ij}^- centered in A_j but having radius $\rho_j - d_{ij}$. Let, for the sake of simplicity, A_i and A_j

be aligned on the x axis, and let C_i, C_{ij}^+ and C_{ij}^- be the projections of S_i, S_{ij}^+ and S_{ij}^- in plane xz, with M_1, M_2 and N_1, N_2 being the intersection points between C_i, C_{ij}^+ and C_i, C_{ij}^-, respectively. Figure 3.1 represents the following 4 cases:

1. if $\rho_j + d_{ij} > D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \leq D_{ij} - \rho_i$, B_i lies on S_i;
2. if $\rho_j + d_{ij} < D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \leq D_{ij} - \rho_i$, B_i lies on the part of S_i contained within S_{ij}^+;
3. if $\rho_j + d_{ij} > D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \geq D_{ij} - \rho_i$, B_i lies on the part of S_i not contained within S_{ij}^-;
4. if $\rho_j + d_{ij} < D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \geq D_{ij} - \rho_i$, B_i lies on the part of S_i contained within S_{ij}^+ but not contained within S_{ij}^-;

In the first case, no improvement on the bounds of B_i can be obtained with respect to those determined by Eq. (3.2). In the second case, x_M, which is the x coordinate of points M_1 and M_2, is a better lower bound for the first component of $\mathbf{B}_i$ than $-X_{ij}$. In addition to that, if $x_M - a_{ix} \geq 0$, also the third component of $\mathbf{B}_i$ can be updated to the interval $[-z_M, z_M]$. Similarly, in the third case, x_N is a better upper bound for the first component of $\mathbf{B}_i$ than X_{ij}, and if $x_N - a_{ix} \leq 0$ the third component of $\mathbf{B}_i$ can be narrowed to $[-z_N, z_N]$. In the fourth case, both x_M and x_N improve the initial bounds for $\mathbf{B}_i$. Furthermore, thanks to the spherical symmetry of the problem, $[-z_M, z_M]$ and $[-z_N, z_N]$ can be used to improve the bounds for the component y_i of $\mathbf{B}_i$ as well. If A_i and A_j are not aligned with the x axis, a rotation matrix always exists that allows the bounds determined by this procedure to be conveniently rotated, thus improving the actual box $\mathbf{B}_i$. For the DGP of a n-n CDPR with m cables in tension, the above procedure has to be repeated for all pairs of taut cables i, j with $i, j \in \{1, \ldots, m\}$. It is possible to further improve the initial bounds for B_i, $i \in \{1, \ldots, m\}$, by applying the procedure for each slack cable k, with $k \in \{m+1, \ldots, n-m\}$ provided that the radius of S_{ik}^- is set equal to zero. Though extremely simple, this latter refinement may largely speed up the computation, since it sometimes determines an empty initial search space and thus prevents any further calculation.

Initial bounds for cable tensions

The tension variables in $\boldsymbol{\tau}$ require an initial search domain, except when $n = 6$. On the one hand, this can be an advantage since the lower bound

for each element of $\boldsymbol{\tau}$ can be set equal to zero and only solutions having nonnegative cable tensions are consequently looked for. On the other hand, choosing the upper bound of cable tensions a priori is not an easy task. For the particular case of suspended cable robots, an upper bound for tensions can be obtained by observing that each element of the third line of matrix $\mathbf{M}$ in (2.3) is negative, i.e. $M_{3j} < 0$, for $j = 1, \ldots, m$. Thus, by considering the third equation of system (2.3), i.e.

$$Q = -\sum_{j=1}^{m} M_{3j} \frac{\tau_j}{\rho_j} \tag{3.3}$$

the following inequality holds:

$$\tau_j \leq \frac{Q\rho_j}{\underline{-M_{3j}}} \quad j = 1, \ldots, m \tag{3.4}$$

However, for all CDPRs which are not suspended, an upper bound for tensions cannot be determined. Setting a too small value can cause the algorithm to exclude feasible solutions (with high values of cable tensions), that may actually exist within the geometrical workspace. These solutions are the most dangerous from the mechanical point of view and it is important to determine them, since they may possibly occur. Setting a too high value (compared with the intensity Q of the applied load) infers a large interval evaluation of static equations (2.3), and thus a large number of bisections to assess whether mechanical equilibrium is attained. This leads to high computation times.

Section 3.1.3 describes a specific filtering procedure is described, which can drastically reduce the influence of the width of cable tension search domain on the code performances. Thanks to this improvement, an initial search domain for cable tension is no longer required and, when not provided, the algorithm automatically finds all feasible solutions.

Cable numbering

For a generic n-n CDPR, with n l,the l reference points on the platform may be chosen arbitrarily. However, this choice has an influence on both the size of the geometric search space and the value of coefficients α_k and β_{ik} of the secondary points (cf. equations (2.23) through (2.25)), consequently increasing or decreasing their interval evaluation. As shown in [Merlet 2004] for the Gough-Stewart platform, these two factors, have a great influence on the computation speed.

In order to asses the performance of a given combination of reference points, the following index is defined (for each sub-DGP) on an empirical base:

$$\Phi = 5I_\alpha + 3I_{min} + 2I_{Vl} + I_{Vm} \tag{3.5}$$

where I_α, I_{min}, I_{Vl} and I_{Vm} are coefficients, comprised between 0 and 1. I_α is the (normalized) mean value of the absolute value of all coefficients α_k and β_{ik}, I_{min} is the (normalized) diameter of the interval having the minimal width in the geometrical search domain, I_{Vl} and I_{Vm} are, respectively, the (normalized) mean widths of the components forming the initial search domain for the l reference points and for the m active cables. On the basis of (3.5), the algorithm ranks all combinations of reference points and picks the one having the lower Φ. The weights 5, 3, 2 and 1 assigned to, respectively, I_α, I_{min}, I_{Vl} and I_{Vm} were chosen, after several tests, in order to provide the combination of reference points generally leading to the best computation time.

3.1.2 Evaluation operator

The evaluation operator $\mathcal{E}$ is implemented by means of the ALIAS procedure `Solve_General_Gradient_Interval` (`SGGI`). Since the functions involved in Eqs. (2.3,2.16,2.17,2.19,2.20,2.26,2.27,2.28) are at least of class C^2, the Jacobian matrix of the system can be analytically computed. Accordingly, the `SGGI` improves the interval evaluation of the aforementioned functions by conveniently using gradients, and by taking advantage of possible monotonicities [Merlet 2007]. Moreover, `SGGI` exploits derivatives by applying, during the evaluation process, a global filtering method based on the classic interval Newton scheme, as described in [Ratschek & Rokne 1995].

3.1.3 Filter operator

The filter operator $\mathcal{F}$ comprises different procedures. Each of them takes an interval box as an input and tries either to improve its bounds or to eliminate it by using *constraint propagation* techniques [Jaulin 2001]. Most filtering functions included in $\mathcal{F}$ rely on the 2B approach. This method consists in rewriting an equation as the equality of two terms, determining if the interval evaluations of both terms are consistent and, if not, using consistency to improve the width of the interval for one or more unknowns. Let, for instance, the equation $x^2 - 2x + 1 = 0$ be considered. By introducing the new variable $X = x^2$, the original equation may be re-written as $X = 2x - 1$. Now, let $[\underline{u}, \overline{u}]$ be the interval evaluation of $2x - 1$. If $\overline{u} > 0$, then the

inverse function of X indicates that x should lie in $\left[-\sqrt{\overline{u}}, \sqrt{\overline{u}}\right]$ and, from this information, the current interval of x may be updated. If $\underline{u} > 0$, the inverse function of X shows that x should lie outside $\left[-\sqrt{\underline{u}}, \sqrt{\underline{u}}\right]$: if the range of x is included in this interval, then there is no solution of the equation in the current box. The 2B approach can also be applied, with some modifications, to constraints expressed by inequalities.

Filters on geometrical constraints

Thanks to the adopted parameterization, the 2B method applied to the geometrical constraints expressed in (2.26) is very effective. Indeed, the $2B$ method reduces the box $\mathbf{B}_i$ containing the intervals of coordinates x_i, y_i and z_i to the smallest box containing the intersection between $\mathbf{B}_i$ itself and the surface of the sphere centered in A_i and with radius ρ_i. The same scheme is applied for the constraints expressed by the point-to-point distance relations (2.28).

When dealing with equations involving secondary points, such as (2.27), first the interval components X_j, Y_j and Z_j of the j-th secondary point are computed by interval evaluating relations (2.24) and (2.25); then the 2B method is applied to all geometrical relations where secondary points are involved, namely (2.27) and the distance relations between the secondary points and the reference ones. If any improvement is achieved for X_i, Y_i and Z_i, the 2B method is applied back to (2.24) and (2.25).

Since the 2B approach can be applied to inequalities too, the following constraints are also considered and processed:

$$\begin{aligned} |\mathbf{b}_i - \mathbf{a}_j|^2 &\leq (\rho_j + d_{ij})^2 \\ |\mathbf{b}_i - \mathbf{a}_j|^2 &\geq (\rho_j - d_{ij})^2 \end{aligned} \tag{3.6}$$

for any $i \neq j$.

When a sub-DGP is analyzed, further inequalities are added to the filtering procedure, in order to take the constraints imposed by slack cables into account, namely:

$$\begin{aligned} |\mathbf{b}_i - \mathbf{a}_i|^2 &\leq \rho_i^2 \\ |\mathbf{b}_j - \mathbf{a}_i|^2 &\leq (\rho_i + d_{ij})^2 \end{aligned} \tag{3.7}$$

where i denotes one of the $n-m$ cables considered to be slack and j denotes one of the m active cables.

Another procedure that helps to improve the bounds on the pose variables consists in applying the 2B method to all relations like (2.24) and (2.25) that emerge from all possible choices of reference and secondary points (not

necessarily the choice that minimizes function I in (3.5)). Finally, the 2B approach can be applied to the geometric constraint:

$$(\mathbf{b}_j - \mathbf{b}_i) \cdot (\mathbf{b}_k - \mathbf{b}_i) = \varphi_{ijk} \tag{3.8}$$

where $i, j, k \in 1, \ldots, n$, $i \neq j \neq k \neq i$, and $\varphi_{ijk} = \cos(\widehat{B_j B_i B_k})$. The latter is a known constant deriving from the geometry of the moving platform.

Filters on static constraints

For CDPRs having $m \leq 5$ active cables, the 2B method may be applied to the equilibrium equations expressed by (2.3). Moreover, rewriting matrix $\mathbf{M}$ by choosing another exit point on the base as the moment pole, three new equations may be obtained from the last three rows of $\mathbf{M}$. Repeating this process for all base exit points, $3(m-1)$ additional relations can be used in the filter, thus being processed with the 2B approach. However, even if this filtering technique is quite effective, it does not reduce the negative influence that large initial bounds on cable tension may have on the code performances. So, another routine is implemented that almost eliminates this dependence.

The main idea consists in trying to solve the interval linear system obtained by rewriting (2.3) as:

$$\underbrace{\begin{bmatrix}\mathcal{L}_1/\rho_1 & \ldots & \mathcal{L}_m/\rho_m\end{bmatrix}}_{\mathbf{J}} \boldsymbol{\tau} = \underbrace{-Q\mathcal{L}_e}_{\mathbf{Q}} \tag{3.9}$$

where $\mathbf{J}$ is a $6 \times m$ interval matrix and $\mathbf{Q}$ an interval vector. When $m \leq 5$, system (3.9) is not square and admits a solution only when $\text{rank}(\mathbf{J}) \leq m$. An approximate solution of this overdetermined system of linear interval equations may be obtained by left-multiplying both members of (3.9) by an arbitrary $m \times 6$ matrix $\mathbf{K}$ [Rohn 1996], namely

$$\underbrace{\mathbf{ZJ}}_{\mathbf{C}} \boldsymbol{\tau} = \mathbf{ZQ} \tag{3.10}$$

By doing this, a square system is obtained and the interval version of the Gauss elimination scheme provided by ALIAS [Merlet 2007] can be applied. This algorithm, when successful (i.e. when $\mathbf{C}$ contains only regular matrices), returns an interval vector that is the enclosure of the infinite possible solutions of the linear interval system. According to [Rohn 1996], in order to improve the algorithm performances, a good candidate for $\mathbf{K}$ is the

Moore-Penrose pseudoinverse of the mid-point evaluation of $\mathbf{J}$, i.e.

$$\mathbf{K} = (\mathrm{mid}(\mathbf{J})^{\mathrm{T}}\mathrm{mid}(\mathbf{J}))^{-1}\mathrm{mid}(\mathbf{J})^{\mathrm{T}} \tag{3.11}$$

However, this choice is questionable in this case, mainly due to the product $\mathrm{mid}(\mathbf{J})^{\mathrm{T}}\mathrm{mid}(\mathbf{J})$. Indeed, while the first three elements of each column of $\mathbf{J}$ are pure numbers, the remaining three are the result of the cross product between a dimensionless vector and a position vector whose components are lengths. Accordingly, the elements of the matrix $\mathrm{mid}(\mathbf{J})^{\mathrm{T}}\mathrm{mid}(\mathbf{J})$ contains the sum between pure numbers and lengths, which is physically meaningless.

We solve this issue by considering a generalized inverse defined as

$$\mathbf{K} = (\mathbf{W}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{W}^{\mathrm{T}} \tag{3.12}$$

where $\mathbf{W}$ is a $6 \times m$ matrix having the first (last) 3 rows equal to the last (first) 3 rows of $\mathbf{J}$. $\mathbf{K}$ is a true generalized inverse matrix, since it satisfies the following Penrose conditions: $\mathbf{JKJ} = \mathbf{J}$ and $\mathbf{KJK} = \mathbf{K}$ [Penrose 1955]. By adopting $\mathbf{W}$, all operations contained in (3.12) are consistent in terms of units. Moreover, this choice still implies that, as the box size decreases, $\mathbf{Z}$ tends to be the identity matrix, which is the best input for the Gauss elimination scheme. It must be remarked that in order to reduce the overestimation introduced by the matrix product, each element of $\mathbf{Z}$ can be rewritten trying to obtain the lowest number of multiple occurrences of the pose parameters.

A further refinement comes from considering that each element of the interval matrix $\mathbf{Z}$ has the following form

$$Z_{ij} = \sum_{k=1}^{6} K_{ik}J_{kj} \tag{3.13}$$

and that each column of $\mathbf{J}$ is a Plücker vector, so that

$$J_{1j}^2 + J_{2j}^2 + J_{3j}^2 - 1 = 0 \tag{3.14}$$

$$J_{1j}J_{4j} + J_{2j}J_{5j} + J_{3j}J_{6j} = 0 \tag{3.15}$$

for $j = 1, \ldots, 6$. Thus, the bounds for each element of $\mathbf{Z}$ may be improved by searching the minimum and the maximum of the function

$$Z'_{ij}(\mathbf{q}) = \sum_{k=1}^{6} K_{ik}q_k \tag{3.16}$$

where $\mathbf{q}$ must satisfy the following constraints:

$$\begin{aligned} q_1^2 + q_2^2 + q_3^2 - 1 &= 0 \\ q_1 q_4 + q_2 q_5 + q_3 q_6 &= 0 \\ \underline{M_{kj}} \leq q_k \leq \overline{M_{kj}} \quad k &= 1, \ldots, 6 \end{aligned} \tag{3.17}$$

The optimization problem emerging from relations (3.16) and (3.17) can be analytically solved with the generalized Lagrange multiplier method and, thus, implemented without using numeric optimization algorithms.

By virtue of the optimization of $\mathbf{Z}$, improvements on the bounds of cable tension variables can be obtained even if cable tensions are not involved in the optimization formulation. When CDPRs having $m \leq 5$ are considered, this strategy allows the filtering procedure to partially reduce the coupling between geometric and tension variables and, as a consequence, the influence of the width of the tension search domain on computation time. When dealing with the DGP of a CDPR with 6 taut cables, cable tensions are not included in the parameterization, and the above procedure is applied to discard boxes having negative cable tensions. In this case $\mathbf{Z} = \text{mid}(\mathbf{J})^{-1}$, since the linear system (3.9) is square.

3.1.4 Influence on the bisection process

The effectiveness of the filtering procedures on the static constraints, especially the one relying on the Gauss elimination scheme, allows one to slightly modify the bisection process. Indeed, rather than bisecting the largest variable contained in $\mathbf{Y} = \left[\mathbf{X}^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}}\right]^{\mathrm{T}}$, it is more convenient to bisect the largest variable contained in $\mathbf{X}$ until $\max(\mathrm{w}(\mathbf{X})) \geq \varepsilon$ (where ε is the threshold used by operator $\mathcal{E}$ to stop the bisection process, see Sec. 1.3), and let $\mathcal{F}$ filter the variables contained in $\boldsymbol{\tau}$. For boxes having $\max(\mathrm{w}(\mathbf{X})) < \varepsilon$, the bisection process picks the variable having the largest width all over $\mathbf{Y}$.

The advantages coming from this change on the bisection strategy are:

- a lower number of bisections
- a minimal influence of the width of the initial search domain of cable tension variables on code performances
- small differences between the computation times for the DGP of robots with 3, 4 or 5 cables in tension, even though they involve a different number of variables.

3.2 Code improvements

The code proposed in the previous section, thanks to the adopted parameterization and the attempt to find the best combination of reference and secondary points on the platform, can achieve really good performances.

However it is possible to further improve it by adding some features aiming at obtaining better computation times as well as more reliable solutions. Indeed, as it is presented up to now, the code still presents some issues; these are explained hereafter together with the procedures implemented to deal with them.

3.2.1 Uniqueness of the solution

The main issue of the algorithm presented in Sec. 3.1 is its behavior in the case of two solutions that are close but yet distinct. Although it may be considered only a fact of minor importance, when dealing with robot kinematics or, in this case, geometrico-static equations the two (or more) nearly coincident solutions may cause an operation-mode switch due for example to an external disturbance, which is quite likely to occur in CDPRs.

A second issue is that nothing prevents a box containing singular poses from being considered a solution. Indeed, if the interval evaluation of a box $\mathbf{B}^*$, with $\mathrm{w}(\mathbf{B}^*) < \varepsilon$, over the analyzed sub-DGP system of equations $\mathbf{f} = \mathbf{0}$ (any of the ones presented in Sec. 2.3) contains $\mathbf{0}$, then the algorithm will store it in the solution list, without any warning about possible singularities contained within. Two procedures are now presented that allow these issues to be correctly managed.

Kantorovitch operator

The numerical analysis provide a theorem that can guarantee the convergence of the Newton method[Tapia 1971]. Newton-based methods, indeed, allow the solution of a nonlinear system of equations to be found provided that a convenient solution guess is given as an input. Though very fast, these methods are not guaranteed to converge, and if they succeed they may converge not on the solution closest to the guessed one given as input. Through the Kantorovitch theorem both these issues can be addressed.

Let a system of h equations in h unknowns of any DGP presented in Section 2.3 be considered. Let $\mathbf{Y}_0$ be a robot configuration, r_0, s_0, p_0 be positive real constants, $\mathbf{U} = \{\mathbf{Y} : \|\mathbf{Y} - \mathbf{Y}_0\|_\infty \leq 2r_0\}$ a ball centered in $\mathbf{Y}_0$, and $\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}|$ be the infinity row norm of a generic matrix $\mathbf{A}$. If $\mathbf{Y}_0$ is such that

1. the Jacobian of the system of equations has an inverse $\mathbf{\Gamma_{Y0}}$ in $\mathbf{Y}_0$ such that $\|\mathbf{\Gamma_{Y0}}\|_\infty \leq s_0$,
2. $\|\mathbf{\Gamma_{Y0}}\mathbf{f}(\mathbf{Y}_0)\|_\infty \leq r0$,
3. $\sum_{k=1}^{h} |\frac{\partial^2 f_i(\mathbf{Y})}{\partial Y_j \partial Y_k}| \leq p_0$ for $i, j = 1, \ldots, h$ and $\mathbf{Y} \in \mathbf{U}$,
4. the constants s_0, r_0, p_0 satisfy $2hs_0r_0p_0 \leq 1$,

then there is a unique solution $\mathbf{Y}^*$ of the system of equation $\mathbf{f} = \mathbf{0}$ in $\mathbf{U}$ and the Newton method used with $\mathbf{Y}_0$ as the estimate of the solution will converge toward $\mathbf{Y}^*$.

Example 1: Let the following system be considered:

$$\mathbf{f}(\mathbf{x}) = \begin{cases} x^2 + y^2 - 1 = 0 \\ (x-1)^2 + y^2 - 1 = 0 \end{cases} \tag{3.18}$$

be a system of equation, with solutions $\mathbf{x}_0^* = [1/2,\ \sqrt{3}/2]^{\mathrm{T}}$ and $\mathbf{x}_1^* = [1/2,\ -\sqrt{3}/2]^{\mathrm{T}}$, and be

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 2x & 2y \\ 2(x-1) & 2y \end{bmatrix} \tag{3.19}$$

its Jacobian. Let $\mathbf{x}_0 = [2/5, 4/5]^{\mathrm{T}}$ be a point, so matrix $\mathbf{\Gamma_{x0}}$ is

$$\mathbf{\Gamma_{x0}} = (\mathbf{J}(\mathbf{x}_0))^{-1} = \begin{bmatrix} 1/2 & -1/2 \\ 3/8 & 1/4 \end{bmatrix} \tag{3.20}$$

and the constant s_0 may be determined as

$$s_0 = \|\mathbf{\Gamma_{x0}}\|_\infty = 1 \tag{3.21}$$

By multiplying $\mathbf{\Gamma_{x0}}$ by $\mathbf{f}(\mathbf{x}_0)$, the constant r_0 is

$$r_0 = \|\mathbf{\Gamma_{x0}}\mathbf{f}(\mathbf{x}_0)\|_\infty = \frac{1}{10} \tag{3.22}$$

The third constant p_0 needed by the theorem is computed by taking the max between the infinity norms of the Hessian matrices from the two functions in $\mathbf{f}$. Neither of the Hessian matrices depends on $\mathbf{x}$, since functions contained

in $\mathbf{f}$ are of degree 2 in the unknowns, and have infinity norm equal to 2, which yields:

$$p_0 = 2 \tag{3.23}$$

Since the last condition required by the theorem $2hr_0s_0p_0 = 4/5 \leq 1$ is satisfied, the theorem guarantees that in the ball $\mathbf{U} = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_0\|_\infty \leq 1/5\}$ there is a unique solution. Indeed, $\mathbf{x}_0^* \in [1/5, 3/5], [3/5, 1]$.

Then the evaluation operator $\mathcal{E}$ may be upgraded as follows:

- the Kantorovitch theorem is applied to $\text{mid}(\mathbf{B}_i)$.
- if it succeeds
  - through the Newton method, the point solution $\mathbf{B}_i^*$ is computed and then added to the solution list,
  - from now on, all the boxes processed by $\mathcal{F}$ are checked for possible interferences with the box $\mathbf{B}_i^+ = \{\text{mid}(X_1) + [-2r_0, 2r_0], \ldots, \text{mid}(X_h) + [-2r_0, 2r_0]\}$, and eventually filtered;
- if it fails the box is returned to the evaluation operator unchanged.

Implementing this method in the algorithm has three main advantages:

- By this algorithm, the code can guarantee that the solution is unique in the given box and it can calculate it within an arbitrary accuracy and it fails only when, with the current computer precision, the value of at least one equation cannot be determined over a box reduced to a point: in this case, it may be shown that no other algorithm is capable of solving the system;
- in practice, boxes no longer need to be bisected until their width reaches the threshold ε, since Kantorovitch usually assesses the uniqueness of a solution earlier.
- filtering the remaining boxes with $\mathbf{B}_i^+$ may be quite effective.

As explained in [Merlet 2004], if all the equations of the solving system are at most of degree 2 in the unknowns, implementation of the Kantorovitch procedure can be simplified. By the adopted parameterization, the equations involved in all DGPs fulfill this requirement, so that second-order derivatives are constants and p_0 can be precomputed at the beginning of the problem-solving algorithm and thus, the computational burden reduced.

Inflation procedure

The ball containing a single solution determined by Kantorovitch may be widened using the *inflation procedure* described in [Neumaier 2001]. Let $\mathbf{Y}_0^*$ be a solution of the system of equations $\mathbf{f}(\mathbf{Y}) = \mathbf{0}$ and that $\mathbf{Y}_1^*$ is another solution close to $\mathbf{Y}_0^*$, so that $\mathbf{f}(\mathbf{Y}_0^*) = \mathbf{f}(\mathbf{Y}_1^*) = \mathbf{0}$. The mean value theorem yields

$$\mathbf{f}(\mathbf{Y}_1^*) = \mathbf{f}(\mathbf{Y}_0^*) + \mathbf{J}(\mathbf{Y})(\mathbf{Y}_1^* - \mathbf{Y}_0^*) \tag{3.24}$$

where $\mathbf{J}$ is the Jacobian matrix of the system and $\mathbf{Y} \in \{\mathbf{Y}_0^*, \mathbf{Y}_1^*\}$. Since (3.24) is true only when $\mathbf{J}(\mathbf{Y})(\mathbf{Y}_1^* - \mathbf{Y}_0^*) = \mathbf{0}$ which admits a solution only if $\mathbf{J}(\mathbf{Y})$ is singular. Therefore, to be certain that there is no other solution in a box centered in $\mathbf{Y}_0^*$ it must be checked that $\mathbf{J}(\mathbf{Y})$ is regular for any point of the box.

This box can be obtained by means of the H-matrix theory described in [Neumaier 2001, 1990]. Let $\mathbf{B}_0 = \left\{[Y_1^{0*} - \delta, Y_1^{0*} + \delta], \ldots, [Y_h^{0*} - \delta, Y_h^{0*} + \delta]\right\}$ be a box centered in $\mathbf{Y}_0^*$ and δ a small real number, and $\mathbf{S}$ be an interval matrix, obtained as follows

$$\mathbf{S} = \mathbf{J}(\mathbf{Y}_0^*)^{-1}\mathbf{J}(\mathbf{B}_0) \tag{3.25}$$

where $\mathbf{J}(\mathbf{B}_0)$ is the interval evaluation of the Jacobian matrix over the box $\mathbf{B}_0$. Let the following interval operators also be defined

$$\text{high}(X) = \max(|\underline{X}|, |\overline{X}|) \tag{3.26}$$

$$\text{low}(X) = \min(|\underline{X}|, |\overline{X}|) \tag{3.27}$$

and the following quantities be introduced

$$m_S = \min_{i \in \{1,\ldots,h\}} (\text{low}(S_{ii})) \tag{3.28}$$

$$M_S = \max_{i \in \{1,\ldots,h\}} \left(\sum_{j=1}^{h} \text{high}(S_{ij}) \right) \quad \text{with} j \neq i \tag{3.29}$$

If $m_S > M_S$, then $\mathbf{J}$ is regular over $\mathbf{B}_0$ and the whole process is repeated for the new box $\mathbf{B}_1 = \left\{[Y_1^{0*} - 2\delta, Y_1^{0*} + 2\delta], \ldots, [Y_h^{0*} - 2\delta, Y_h^{0*} + 2\delta]\right\}$. When for the box $\mathbf{B}_k = \left\{[Y_1^{0*} - 2^k\delta, Y_1^{0*} + 2^k\delta], \ldots, [Y_h^{0*} - 2^k\delta, Y_h^{0*} + 2^k\delta]\right\}$ this regularity test is not satisfied, the process stops and sets $\mathbf{B}_{k-1}$ as a new single-solution box. This box can be used, again, to filter the other boxes through the $\mathcal{F}$ operator taking advantage of possible interferences.

Also the inflation process can be simplified in the case of system of equations formed by second-degree polynomials. In this case, indeed, each component of the Jacobian matrix is linear in the unknowns. Let Y_i^{0*} be the elements of $\mathbf{Y}_0^*$ and $\mathbf{Y}_\Delta^* = \left\{Y_1^{0*} + \Delta, \ldots, Y_h^{0*} + \Delta\right\}$, where $\Delta = [-\delta, \delta]$. Each element J_{ij} of the Jacobian matrix at $\mathbf{Y}_\Delta^*$ can be rewritten as

$$J_{ij}(\mathbf{Y}_\Delta^*) = \mu_{ij} + \eta_{ij}\Delta \tag{3.30}$$

where μ_{ij} and η_{ij} depends only on $\mathbf{Y}_0^*$. Left-multiplying $\mathbf{J}$ by $\mathbf{J}(\mathbf{Y}_0^*)^{-1}$ yields

$$\underbrace{\mathbf{J}(\mathbf{Y}_0^*)^{-1}\mathbf{J}}_{\mathbf{T}} = \mathbf{I}_h + \mathbf{Z} \tag{3.31}$$

where $\mathbf{I}_h$ is the identity matrix of dimension h and $\mathbf{Z}$ an interval matrix whose elements have the form $Z_{ij} = \xi_{ij}\Delta$ with ξ_{ij} functions of η coefficients and of the elements of $\mathbf{J}(\mathbf{Y}_0^*)^{-1}$. Quantities m_T and M_T are introduced and computed as

$$m_T = \min_{i\in\{1,\ldots,h\}}(\text{low}(T_{ii})) = \min_{i\in\{1,\ldots,h\}}(1 - |\xi_{ii}|\delta) \tag{3.32}$$

$$M_T = \max_{i\in\{1,\ldots,h\}}\left(\sum_{j=1,j\neq i}^{h}\text{high}(T_{ij})\right) = \max_{i\in\{1,\ldots,h\}}\left(\delta\sum_{j=1,j\neq i}^{h}|\xi_{ij}|\right) \tag{3.33}$$

and the following relations must be satisfied for $\mathbf{T}$ to be regular over $\mathbf{Y}_\Delta^*$

$$m_T \geq M_T \tag{3.34}$$

which yields

$$\begin{aligned}
\min_{i\in\{1,\ldots,h\}}(1 - |\xi_{ii}|\delta) &\geq \max_{i\in\{1,\ldots,h\}}\left(\delta\sum_{j=1,j\neq i}^{h}|\xi_{ij}|\right)\\
1 - \max_{i\in\{1,\ldots,h\}}(|\xi_{ii}|\delta) &\geq \max_{i\in\{1,\ldots,h\}}\left(\delta\sum_{j=1,j\neq i}^{h}|\xi_{ij}|\right)\\
1 - \delta\max_{i\in\{1,\ldots,h\}}(|\xi_{ii}|) &\geq \delta\max_{i\in\{1,\ldots,h\}}\left(\sum_{j=1,j\neq i}^{h}|\xi_{ij}|\right)\\
\delta_{min} = \delta &\leq \frac{1}{\max\limits_{i\in\{1,\ldots,h\}}\left(\sum\limits_{j=1,j\neq i}^{h}|\xi_{ij}|\right) + \max\limits_{i\in\{1,\ldots,h\}}(|\xi_{ii}|)}
\end{aligned} \tag{3.35}$$

where $2\delta_{min}$ is the width of the box $\mathbf{B}_{\Delta\,min}$ guaranteed to contain only solution $\mathbf{Y}_0^*$.

Example 2: This example shows the practical application of this latter version of the inflation process, i.e. the one for the system of equations consisting of functions that are at most of degree 2.

Let the same system of equation proposed in Example 3.2.1 be considered. Assume that the solutions of the system are not known: at a certain step of the algorithm, the Kantorovitch-theorem-based procedure finds a unique solution in the box $\mathbf{B}_{\mathbf{x}0} = \{[1/5, 3/5], [3/5, 1]\}$ and then Newton returns the solution $\mathbf{x}_0^* = [1/2, \sqrt{3}/2]^{\mathrm{T}}$. Let $\mathbf{B}_\Delta^* = \{x_0^* + \Delta, y_0^* + \Delta\}$, the Jacobian calculated over $\mathbf{B}_\Delta^*$ is

$$\mathbf{J}(\mathbf{B}_\Delta^*) = \begin{bmatrix} 2x_0^* + 2\Delta & 2y_0^* + 2\Delta \\ 2(x_0^* + \Delta - 1) & 2y_0^* + 2\Delta \end{bmatrix} = \tag{3.36}$$

$$= \mathbf{J}(\mathbf{x}_0^*) + \begin{bmatrix} 2\Delta & 2\Delta \\ 2\Delta & 2\Delta \end{bmatrix} \tag{3.37}$$

Left multiplying both sides of (3.37) by $\mathbf{J}(\mathbf{x}_0^*)^{-1}$ brings

$$\underbrace{\mathbf{J}(\mathbf{x}_0^*)^{-1}\mathbf{J}(\mathbf{B}_\Delta^*)}_{\mathbf{T}} = \mathbf{I}_2 + \underbrace{\mathbf{J}(\mathbf{x}_0^*)^{-1} \begin{bmatrix} 2\Delta & 2\Delta \\ 2\Delta & 2\Delta \end{bmatrix}}_{\mathbf{Z}} \tag{3.38}$$

where, computing the product

$$\mathbf{Z} = \begin{bmatrix} 1/2 & 1/2 \\ \sqrt{3}/6 & \sqrt{3}/6 \end{bmatrix} \begin{bmatrix} 2\Delta & 2\Delta \\ 2\Delta & 2\Delta \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ (2\sqrt{3}/3)\Delta & (2\sqrt{3}/3)\Delta \end{bmatrix} \tag{3.39}$$

Thus, δ_{min} can now be computed applying (3.35), which yields

$$\delta_{min} = \frac{1}{2\sqrt{3}/3 + 2\sqrt{3}/3} = \frac{\sqrt{3}}{4} \simeq 0.4330 \tag{3.40}$$

This result (as shown also in Fig. 3.2) guarantees that in the box $\mathbf{B}_\Delta^* = \left\{ \left[(2-\sqrt{3})/4, (2+\sqrt{3})/4\right], \left[\sqrt{3}/4, 3\sqrt{3}/4\right] \right\}$ there is a unique solution and provides a much wider uniqueness domain. From this example it is possible to show that if the equations in $\mathbf{f} = 0$ are at most of degree 2 in the unknowns, once the inverse of the Jacobian matrix $\mathbf{J}(\mathbf{Y}_0^*)$ is computed it is possible to immediately obtain the largest bound of the box where the solution is guaranteed to be unique, since all the μ and η coefficients can be precomputed. All the described parameterizations in Sec. 2.3 fulfill this requirement so this simplified version of the inflation process can be implemented and the evaluation operator $\mathbf{E}$ further upgraded.

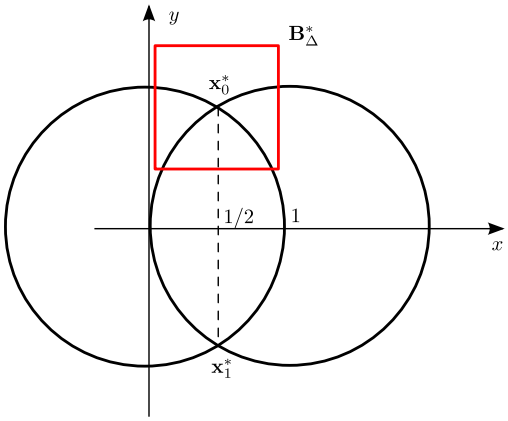


Figure 3.2 – The solution box after the inflation process.

- the Kantorovitch theorem is applied to mid($\mathbf{B}_i$).
- if it succeeds
  - through the Newton method, the point solution $\mathbf{B}^*_i$ is computed and then added to the solution list,
  - the inflation process is applied in $\mathbf{B}^*_i$ and a larger box $\mathbf{B}_{i\Delta}$ is computed where the solution is unique
  - from now on, all the boxes processed by $\mathcal{F}$ are checked for possible interferences with the box $\mathbf{B}_{i\Delta}$ and eventually filtered;
- if it fails the box is returned to the evaluation operator unchanged.

3.2.2 Stability

A third important refinement consists in checking the solution for stability. All the configurations that are deemed solutions by the algorithm are indeed equilibrium equations. However, only stable ones are feasible in practice. Thus each solution identified by $\mathcal{E}$ can be processed by a procedure that determines whether the corresponding equilibrium configuration is stable or not. A sufficient condition for the equilibrium to be stable consists in having the (reduced) Hessian matrix $\mathbf{H}_r$ being positive definite. If $\mathbf{I}_3$ denotes the 3×3 identity matrix, and $\tilde{\mathbf{n}}$ denotes, for a generic vector $\mathbf{n}$, the skew-symmetric matrix associated with the operator $\mathbf{n}\times$, $\mathbf{H}_r$ may be computed as (Carricato & Merlet [2013])

$$\mathbf{H}_r = \mathbf{N}^T\mathbf{H}\mathbf{N}, \tag{3.41}$$

where

$$\mathbf{H} = \sum_{i=1}^{m} \frac{\tau_i}{\rho_i} \begin{bmatrix} \mathbf{I}_3 & -(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}}) \\ (\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}}) & \frac{1}{2}\left[(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}})(\tilde{\mathbf{g}} - \tilde{\mathbf{a}}_i) + (\tilde{\mathbf{g}} - \tilde{\mathbf{a}}_i)(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}})\right] \end{bmatrix}, \tag{3.42}$$

and $\mathbf{N}$ is a $6 \times (6-m)$ matrix whose columns generate the null space of the Jacobian matrix $\mathbf{J}^{\mathrm{T}}$:

$$\mathbf{J}^{\mathrm{T}} = \begin{bmatrix} (\mathbf{b}_1 - \mathbf{a}_1)^{\mathrm{T}}/\rho_1 & (\mathbf{b}_1 - \mathbf{g}) \times (\mathbf{b}_1 - \mathbf{a}_1)^{\mathrm{T}}/\rho_1 \\ \vdots & \vdots \\ (\mathbf{b}_m - \mathbf{a}_m)^{\mathrm{T}}/\rho_m & (\mathbf{b}_m - \mathbf{g}) \times (\mathbf{b}_m - \mathbf{a}_m)^{\mathrm{T}}/\rho_m \end{bmatrix}. \tag{3.43}$$

in this case computed assuming G as the moment pole. Finally, the final upgraded version of the evaluation operator $\mathcal{E}$ is

- the Kantorovitch theorem is applied to mid($\mathbf{B}_i$).
- if it succeeds
  - through the Newton method, the point solution $\mathbf{B}_i^*$ is computed and then added to the solution list,
  - then $\mathbf{B}_i^*$ is checked for stability, and if the test succeeds it is marked as feasible
  - the inflation process is applied in $\mathbf{B}_i^*$ and the a larger box $\mathbf{B}_{i\Delta}$ is computed where the solution is unique
  - from now on, all the boxes processed by $\mathcal{F}$ are checked for possible interferences with the box $\mathbf{B}_{i\Delta}$ and filtered if necessary;
- if it fails the box is returned to the evaluation operator unchanged.

3.2.3 Parallel implementation

Most interval-analysis-based algorithms are appropriate for a distributed implementation, since processing a given box does not generally depend on processing of the other boxes in the list. A master computer manages the list and it sends a box to a slave computer. The slave executes the algorithm, by performing a few bisections. Then, it returns the remaining boxes to the master, and it waits for a new box to process. Such a scheme may be easily implemented in a network of workstations. The decrease in computation time will be, in general, less than proportional to the number of slaves, due to the overhead of data transmission between the master and the slaves.

This approach may also take advantage of modern multi-core CPU architectures. By following this scheme and by using POSIX thread libraries, a distributed implementation of the DGP code was set up on a single workstation with a multi-core CPU. In the first step, an instance of $\mathcal{E}$ generates a few boxes and it stores them in the list $\mathcal{L}$. Then, a number of threads equal to the number of CPUs is created, with each one taking a box from $\mathcal{L}$. A local instance of $\mathcal{E}$ performs an assigned number of bisections and it appends the generated boxes to $\mathcal{L}$. The solutions that are found, if any, are appended to the solution list $\mathcal{S}$.

3.3 Code structure: beta version

The code described in Sec. 3.1, together with the improvements presented in Sec. 3.2, was extensively tested and shown very good performances. However, its implementation can be further enhanced and a first beta version of the current algorithm was developed in order to address the following issues:

- since the filter operator $\mathcal{F}$ of the actual code was developed incrementally, some modules repeat computations made by previous parts of the code, so an in-depth review of the methods implemented within may lead to huge simplifications;
- the actual version of the code processes each single DGP problem separately and the solution of the full DGP is made up of the sum of all sub-DGPs, then this new version presents an approach that considers all DGPs at the same time.

The main idea for this new version of the code consists in trying to consider the DGP of a generic CDPR as a single problem and whose solution can be computed within a unique interval algorithm. During the implementation of this approach, as will be explained below, several improvements to the original code were made.

3.3.1 Initialization

Let a CDPR with n cables be considered. Its platform is parameterized as described in Sec. 2.3 and a set of l reference and $n - l$ secondary points is conveniently chosen.

For each of the N_{DGP} configurations with different sets of taut cables, using the procedure described in Sec. 3.1.1, the code computes the existence bounds for the platform points $\mathbf{b}_i$ and $\mathbf{g}$ and the cable tensions $\boldsymbol{\tau}$ and

generates an initial box $\mathbf{\Omega}_1$ as

$$\mathbf{\Omega}_1 = \bigcup_{i=1}^{N_{DGP}} \left\{\mathbf{B}_1^{\mathrm{T}}, \ldots, \mathbf{B}_n^{\mathrm{T}}, \mathbf{B}_g^{\mathrm{T}}\right\}_i \tag{3.44}$$

where $\mathbf{B}_g$ is the box containing the initial search domain for $\mathbf{g}$.

All the configurations with $m < n$ cables in tension are preprocessed and, for each one, the code computes the set of reference points and stores the values of coefficients α and β for the corresponding geometry of the platform.

Finally, a constant vector $\mathbf{w}$, with N_{DGP} components is generated. Each element of $\mathbf{w}$ identifies a sub-DGP configuration and it is initialized with 1 if the corresponding search domain $\left\{\mathbf{B}_1^{\mathrm{T}}, \ldots, \mathbf{B}_n^{\mathrm{T}}, \mathbf{B}_g^{\mathrm{T}}\right\}_i$, with $i = 1, \ldots, N_{DGP}$, is not empty, and with 0 otherwise.

Provided that $\mathbf{\Psi}_1$ is the box containing the search domain for the n cable tension, at the end of the initialization process the algorithm returns the first input box for the algorithm, i.e.

$$\mathbf{V}_1 = \left\{\mathbf{\Omega}_1, \mathbf{\Psi}_1^{\mathrm{T}}, \mathbf{w}_1\right\} \tag{3.45}$$

During this initialization process, all the constants needed by the Kantorovitch procedure and the inflation process are computed.

3.3.2 Description of the algorithm

The algorithm structure, that was modified with respect to the one of algorithm 1, taking into account the improvements made on the evaluation operator $\mathcal{E}$, proceeds as in algorithm 3. Its main components are described hereafter.

The filter operator $\mathcal{F}_{platform}$

This operator filters the intervals of the $n+1$ platform anchor points and $\mathbf{g}$. It applies the 2B method, explained in Sec. 3.1.3, to all the constraints concerning the moving platform geometry, namely

- the distance relations between platform anchor points $\mathbf{b}_i$, with $i = 1, \ldots, n$ and $\mathbf{g}$

$$\|\mathbf{b}_i - \mathbf{b}_j\| = d_{ij}^2 \tag{3.46}$$

$$\|\mathbf{b}_i - \mathbf{g}\ \| = d_{iG}^2 \tag{3.47}$$

  with $i, j = 1, \ldots, n$, $i \neq j$.

- all the relations (2.21) and (2.25) for any possible combination of reference and secondary points
- all the constraints deriving from (3.8)

If the 2B procedure finds any of these constraint to be inconsistent, $\mathcal{F}_{platform}$ returns -1. Otherwise it returns 1 if $\mathbf{\Omega}_i$ bounds are improved or 0 if the box remains unchanged.

The filter operator $\mathcal{F}_{admissible}$

This filter checks if the actual pose $\mathbf{\Omega}_i$ contained in the analyzed box $\mathbf{V}_i$ is compatible with the following n cable constraints

$$\|\mathbf{b}_i - \mathbf{a}_i\| \leq \rho_i \tag{3.48}$$
$$\|\mathbf{b}_j - \mathbf{a}_i\| \leq \rho_i + d_{ij} \tag{3.49}$$

The filter returns -1 if at least one of the two conditions is not fulfilled, i.e. when for at least for one i or for a couple i, j, $\underline{\|\mathbf{b}_i - \mathbf{a}_i\|} > \rho_i$ or $\underline{\|\mathbf{b}_i - \mathbf{a}_j\|} > \rho_j + d_{ij}$.

Moreover, from (3.48), if $\overline{\|\mathbf{b}_i - \mathbf{a}_i\|} < \rho_i$ then cable i is certainly slack. As a consequence, the filter sets to zero all configurations in $\mathbf{w}_i$ containing cable i, and the corresponding cable tension search domain $\mathbf{\Psi}(i)$. Then the filter tries to improve bounds for $\mathbf{\Omega}_i$ by applying 2B method to inequalities (3.48) and (3.49), and if it succeeds it returns 1. If the box $\mathbf{V}_i$ remains unchanged it returns 0.

The procedure $\mathcal{P}_{solve}$

When there is only one admissible configuration in the $\mathbf{w}_i$ vector contained in the box $\mathbf{V}_i$, this procedure is called by the main algorithm and determines whether $\mathbf{V}_i$ contains solutions for this single sub-DGP and eventually updates the solution list $\mathcal{S}$. As previously explained, during the initialization procedure the code has already stored the parameterizations and equation sets for each sub-DGP, with the corresponding set of reference and secondary points and relative coefficients α and β. The procedure proceeds like Algorithm 1, except for the evaluation operator $\mathcal{E}$, which behaves as follows.

- It computes few iterations of the Newton-Raphson scheme, taking the mid($\{\mathbf{\Omega}_i, \mathbf{\Psi}_i\}$) as guess point.

- If the approximate solution is still in the box, the Kantorovitch-theorem-based procedure is applied in order to check whether there is a single solution of the system in a box centered in the approximate solution with a known width.
  - if it succeeds, then the Newton-Raphson scheme is continued and, thanks to the Kantorovitch theorem, it is now guaranteed to converge toward a unique solution. Once the solution is found and it is stored in $\mathcal{S}$, the inflation procedure is applied in order to obtain a box where the solution is guaranteed to be unique. Finally, this box is stored in the memory and used by the filter operator $\mathcal{F}$ to reduce the size of the adjoining boxes and $\mathcal{E}$ returns 1. The solution box is processed by the stability procedure.
  - if the Kantorovitch test fails, then no guarantees can be provided for the approximate solution so the $\mathcal{E}$ returns 0.
- It returns 0 if after a few iterations the approximate solution is outside the considered box,

The first main difference in this version of operator $\mathcal{E}$ is that the evaluation of the system of equation $\mathbf{f} = 0$ is avoided, because it is already included in $\mathcal{F}$. Second, the system of equations for the statics of each sub-DGP was slightly updated in order to take into account a generic external wrench acting upon the platform, and not only a single force. Third, a module that filters boxes by intersecting them with the enlarged solution boxes coming from the inflation process is added to $\mathcal{F}$.

3.3.3 Remarks

The beta version of the code structure can handle the DGP of CDPR with n cables as a unique problem and gives interesting, but preliminary, results. However, it is in an early development stage, its procedures are not definitive since relevant changes on the algorithm structure were made and other specific and more efficient procedures may be added to it. Moreover, it has not completed the test phase yet and still presents minor bugs.

Algorithm 3 Problem-solving algorithm scheme: beta version

1: $i = 1$, $N = 1$, $\mathcal{L} = \{\mathbf{V}_1\}$, $\mathcal{S} = \{\}$
2: **if** $i > N$ **then return** $\mathcal{S}$;
3: **end if**
4: $\mathbf{V}_{old} = \mathbf{V}_i$
5: **if** $\mathcal{F}_{platform}(\mathbf{V}_i) = -1$ **then**
6: $i = i + 1$
7: **go to** 2
8: **else**
9: $\mathbf{V}_i = \mathcal{F}_{platform}(\mathbf{V}_i)$
10: **end if**
11: **if** $\mathcal{F}_{admissible}(\mathbf{V}_i) = -1$ **then**
12: $i = i + 1$
13: **go to** 2
14: **else**
15: $\mathbf{V}_i = \mathcal{F}_{admissible}(\mathbf{V}_i)$
16: **end if**
17: **if** $\mathbf{w}_i^{\mathrm{T}}\mathbf{w}_i = 1$ **then**
18: $\mathbf{V}_i$ has a single admissible configuration
19: $\mathcal{P}_{solve}(\mathbf{V}_i, \mathcal{S})$
20: **else**
21: **if** $V_i = V_{old}$ **then**
22: pick k such that $\mathbf{w}[k] = 1$
23: create two new boxes $\mathbf{V}_i' = \mathbf{V}_i'' = \mathbf{V}_i$
24: $\mathbf{w}_i' = \{\dots, w_{k-1}, 0, w_{k+1}, \dots\}$
25: $\mathbf{w}_i'' = \mathbf{w}_i - \mathbf{w}_i'$
26: replace $\mathbf{V}_i$ with $\{\mathbf{V}_i', \mathbf{V}_i''\}$ in $\mathcal{L}$
27: $N = N + 1$, $i = i + 1$
28: **go to** 2
29: **else**
30: pick x_k such that $\mathrm{w}(X_k) = \max_{k\in\{1,\dots,3h\}}(\mathrm{w}(\mathbf{\Omega}_i[k]))$
31: bisect X_k in the middle point
32: create two new boxes $\mathbf{V}_i'$ and $\mathbf{V}_i''$ from $\mathbf{V}_i$
33: replace $\mathbf{V}_i$ with $\{\mathbf{V}_i', \mathbf{V}_i''\}$ in $\mathcal{L}$
34: $N = N + 1$, $i = i + 1$
35: **go to** 2
36: **end if**
37: **end if**

Chapter 4

Experiments and Results

In this section the results obtained during the test phase on the developed algorithms are presented. As previously stated, the beta version of the code is currently under development so computation times are not suited to make comparisons between methods.

In order to investigate the feasibility of solutions found by the algorithm and to test the real likelihood of operation mode changes, a small prototype for teaching and experimental purposes was built during the last year of the Ph.D. program. On this prototype, some preliminary tests were done in order to implement a real-time compliant version of the developed DGP solving procedures.

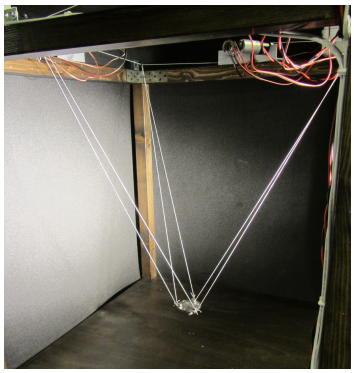
The first part of this chapter describes the teaching prototype and gives an overview of the results obtained by the real-time procedures developed for its control. In the second part some results from tests of the two versions of the DGP solving algorithm are reported and discussed.

4.1 Description of the teaching prototype

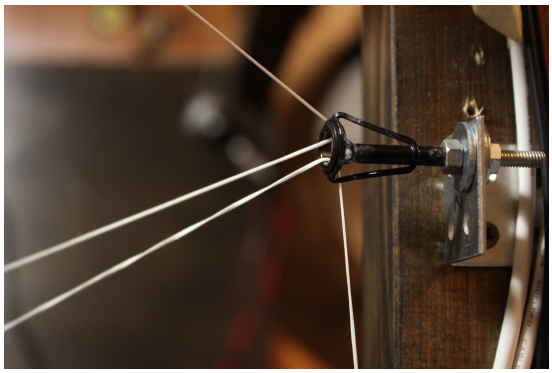
4.1.1 Hardware

The prototype consists of a wooden framework (Fig. 4.1a) built on a square base with sides of 800mm, and height 700mm. The exit points are realized by means of fishing rod top guides (Fig. 4.1b, that have inner rings in ceramic material in order to have a low coefficient of friction.

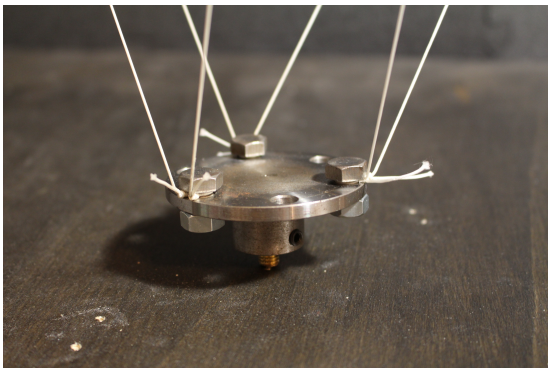
Cables are common single braided Nylon fiber lines and their diameter is 1mm. The end-effector is a 70mm diameter steel flange and cables are connected to it through knots that are held firm by means of nut and bolt (Fig. 4.1a).



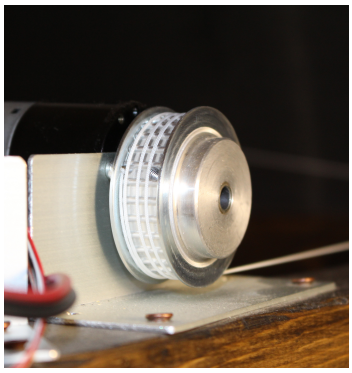
(a) An overview of the teaching prototype.



(b) A detail of the exit points.



(a) The end-effector.



(b) A winch.

The winches are realized by coupling timing belt pulleys (Fig. 4.1b) to the 6 Phidget® DC gear motors whose characteristics are reported in Table 4.1. Each motor is actuated by a Phidget® driver, whose characteristics are listed in Table 4.2. All the drivers are powered by an AC/DC adapter from an old PC and connected through a USB hub to a personal computer that runs the control algorithm.

The winches are located and oriented with respect to the exit points in order to avoid cable overlapping on pulleys and, as shown by experiments, if this positioning is accurate, there are generally no cable coiling issues. However, this badly affects the reconfigurability of cable exit-points.

Motor properties	
Motor type	DC Motor with encoder
Mechanical output power	7W
Maximum speed at rated voltage	73RPM
Rated torque	0.61Nm
Stall torque	7.04Nm
Electrical properties	
Rated voltage	12V DC
Rated current	1.1A
Stall current	5.0A
Gearbox specifications	
Gearbox type	Spur
Gear ratio	50 : 1
Number of gear trains	4
Maximum strength of gears	1.2Nm
Shaft diameter	6mm
Shaft maximum axial load	35N
Shaft maximum radial load	25N
Encoder specifications	
Encoder resolution	360CPR
Connector type	E4P

Table 4.1 – Properties of the Phidgets® 3260E_0 gear motors.

4.1.2 Software

The control routine of the manipulator is managed by a personal computer connected to the motor drivers through a USB cable. A user software interface was developed in order to simplify testing operations and allows several parameters to be controlled also during robot movements. Fig. 4.1 shows a screenshot of the online monitoring module of the graphical user interface. Among the other features, it makes it possible to slightly adjust coefficients of the PID controller and switch between the different routines that compute the forward kinematics.

Controller properties	
Motor type	DC Motor
Velocity resolution	0.39% duty cycle
Acceleration resolution	0.39% duty cycle/s
Acceleration min	24, 5% duty cycle/s
Acceleration max	6250% duty cycle/s
Acceleration time min	32ms
Acceleration time max	8.2s
Encoder interface	
Number of encoder inputs	1
Count rate max	500000 cycles/s
Encoder interface resolution	×1
Update rate	125 samples/s
Time resolution	0.33ms
Electrical properties	
Supply voltage min	9V DC
Supply voltage max	28V DC
Continuous motor current max	5A
Over-current trigger	8A
Current consumption min	20mA
Current consumption max	100mA

Table 4.2 – Properties of the Phidgets® 1065_0 single motor driver.

Inverse kinematics

The control routine has 4 different modules computing the inverse kinematics, one for each underconstrained robot with $m = 3, \ldots, 6$ cables in tension. When the robot has $m < 6$ cables in tension, $6 - m$ degrees of freedom of the platform cannot be controlled.

Let the 6 degrees of freedoms of the moving platform be described by 3 parameters $x_{O'}$, $y_{O'}$ and $z_{O'}$ for the position of the origin O' of the mobile frame in $Oxyz$ and by 3 angles φ_x, φ_y and φ_z, representing the orientation of the platform by three consecutive rotations along the x, y and z fixed axes. When the robot has $m = 3$ cables in tension, the inverse kinematics guarantees the control over the 3 position parameters, when $m = 4$ the user may choose to control also one of the two rotations φ_x or φ_y, and if $m = 5$ both φ_x and φ_y are managed. By means of this parame-

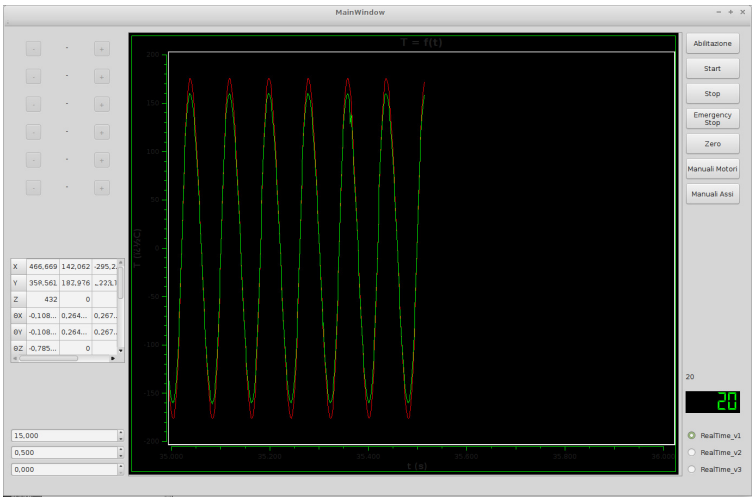


Figure 4.1 – The control user interface of the teaching prototype.

terization, the inverse kinematics is dealt with by solving the m distance relations (2.2) together with the equation from static equilibrium (2.3). This system of equation has $m + 6$ equations in the $m + 6 + m$ variables $[\rho_1, \ldots, \rho_m, x_{O'}, y_{O'}, z_{O'}, \varphi_x, \varphi_y, \varphi_z, \tau_1, \ldots, \tau_m]$. Once m platform pose parameters are assigned, the system becomes square and by solving it, the m cable lengths are obtained.

However, this scheme for the inverse kinematics of underconstrained robot was developed only for testing purposes of the feasibility in real-time of the algorithms for the forward kinematics problems. A new and more refined module for the inverse kinematics is now currently under development, always based on interval analysis.

In the inverse kinematics routine, the working assumptions are the same presented in Sec. 2.2. They are proved to be reasonably correct in most of the robot workspace, considering also the overall (moderate) manufacturing precision of the hardware. However, for heavy payloads and/or in the upper regions of the workspace, where cable tensions are considerably higher, there are relevant deviations on the desired platform pose due to cable elasticity. Moreover, friction between cables and the exit points, during cable coiling, causes the cable segment between the pulley and the exit point to have a higher tension than the segment between the exit point and the corresponding anchor point. Indeed from Fig. 4.2 it is possible to observe that cables have different wrapping angles on the surface of the fishing rod top eyes.

This phenomenon was observed performing a particular lifting operation

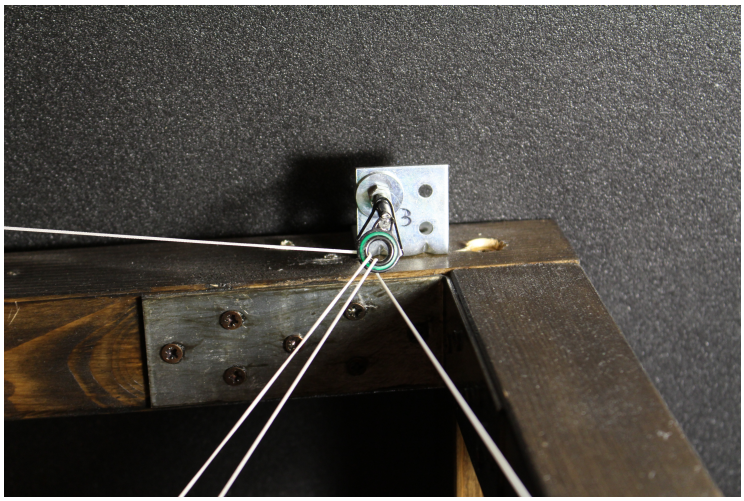
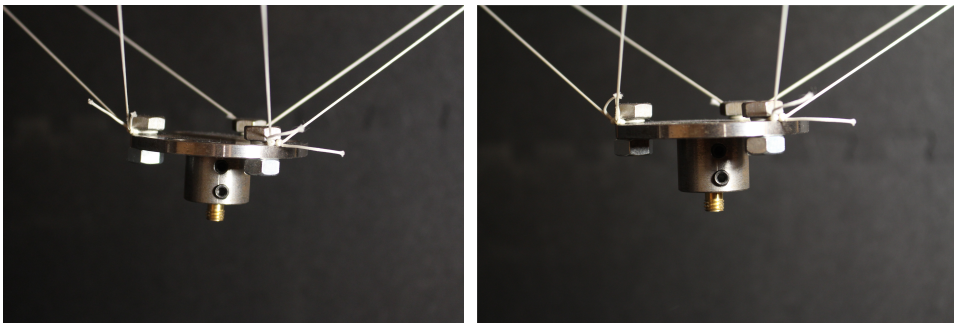


Figure 4.2 – The large wrapping angle of the cable on the right on the fishing rod top eye cause a non uniform tension along the cable.

where both cable tensions and lengths were supposed to be all equal. As the platform height gets closer to one of the exit points, and consequently cable tension become higher, the error on platform orientation increases. As shown in figures 4.3a and 4.3b, this error vanishes if, when the platform is still, the cables are momentarily released (for example by manually tilting the platform).



(a) (b)

Figure 4.3 – The moving platform before (a) and after (b) the manual redistribution of cable tensions

A few attempts were made to model this behavior by introducing the capstan theory, expressed by the Eytelwein formula in (4.1), in the inverse

kinematics. However, the uncertainties on the coefficient of friction ζ between the fiber lines and the rod top guide eyes and on the wrapping angle γ lead to exponential error on the estimate of tension distribution in the two cable sections.

$$\tau_{PA} = \tau_{AB} \mathrm{e}^{\zeta\gamma} \tag{4.1}$$

This problem can be significantly reduced or completely eliminated by employing cables with a higher stiffness and/or by introducing rolling elements on the exit points.

Forward kinematics

The procedure that solves the direct geometrico-static problem is implemented in the control routine as follows. Basically, it relies on the $\mathcal{P}_{solve}$ procedure of the beta version of the code presented in Sec. 3.3.2. Assume that at time t_0 of the control routine the robot has the platform pose $\mathbf{Y}_0$ for a given set of cable lengths $\boldsymbol{\rho}_0$. Then, in order to follow the desired trajectory, the controller computes the inverse kinematics and the motors move to a new set of cable lengths $\boldsymbol{\rho}_1 d$. After a delay of few milliseconds Δ_t, that is the system interrupt time, the controller computes the cable lengths by reading the actuator encoders. Without additional sensors, this estimate of cable lengths is the only measurement available for feedback position control and thus can be considered the real set of cable lengths $\boldsymbol{\rho}_1$ at time $t_1 = t_0 + \Delta_t$. In order to measure the error of the platform pose with respect to the desired trajectory, the actual pose $\mathbf{Y}_1$ satisfying the new cable constraints $\boldsymbol{\rho}_1$ must be computed.

Let a CDPR with 6 cables in tension be considered. According to the parameterization used for the inverse kinematics, assume that the upper bounds on platform velocities $\mathbf{v}_{max}$ and $\boldsymbol{\omega}_{max}$ are known. Thus, it is possible to determine a bounding box for the reference points b_i, with $i = 1, \ldots, h$. Each bounding box is centered in $\mathbf{b}_i^0$, i.e. the coordinates of the ith reference point at $\mathbf{Y}_0$, and its edges are $2\Delta_t(\mathbf{v}_{max} + \boldsymbol{\omega}_{max}\|\mathbf{b}_i'\|)$, with $i = 1, \ldots, h$. The steps of the procedure are described hereafter.

- Kantorovitch is applied to $\mathbf{Y}_0$ to see if there is a box containing the unique solution $\mathbf{Y}_1$.
- If it succeeds, then the Newton-Raphson scheme is guaranteed to converge on the unique solution and then it is computed with an arbitrary precision.
- If it fails, the box $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_h\}$ is processed by filter $\mathcal{F}$.

- If the filter eliminates the box, a solution cannot be found in $\mathbf{B}$. The robot has to be immediately stopped.
- If the filter improves $\mathbf{B}$ bounds, the algorithm is repeated from the beginning but Kantorovitch is applied to the mid($\mathbf{B}$).
- If the filter leaves the box unchanged, then the algorithm creates a list $\mathcal{L}$ and bisects the box. Then the algorithm starts again from the beginning and applies Kantorovitch to the mid-point of each box in the list.

If at the end of the algorithm more than one solution has been found in the box, the robot is stopped.

This approach, from the tests done so far on the teaching prototype, allows the DGP for a CDPR with m cables in tension, with $m = 3, \ldots, 6$, to be solved with performances equivalent to standard Newton methods. Indeed, as explained in Sec. 3.2.1, since the matrix containing the second order derivatives can be precomputed because it is constant, the overhead required by this test is due to the numerical inversion of the system Jacobian matrix. From the tests done on the teaching prototype, Kantorovitch was successful in all tests, in every point of different trajectories.

Then the test were repeated by disabling the Kantorovitch module in the algorithm and with the only action of operator $\mathcal{F}$ and after a few bisections, a solution for the DGP (with an accuracy threshold of $\varepsilon = 1.e - 4$) is found usually within 4 ms, and no more than 7ms. However this is a worst case scenario, since even if Kantorovitch fails at the first step of the algorithm, it usually succeeds after a few bisections and the solutions in the box can still be computed with arbitrary precision.

Remarks on the real-time procedure

Although very effective, this real-time procedure for solving the DGP of cable robots still presents many open issues, described below in order of importance.

The first one is that the above described procedure is not able to deal with the sub-DGPs. Indeed it considers only the case where all the m cables are taut. However, a module was added to check if the robot platform is getting near to a sub-DGP solution. Let a CDPR with m cables in tension be considered and assume that in a certain point of the followed trajectory, the tension of one cable becomes nearly 0. When the tension of a set of k cables, with $k = 1, \ldots, m - 2$ is smaller than a user-defined threshold, e.g. 0.1, the controller starts the real-time solving algorithm for a CDPR

with $m-k$ taut cables on a parallel thread. If, in this parallel instance, Kantorovitch succeeds in finding another unique solution in a bounded box, then the user interface sends out a warning. Moreover, if the main algorithm fails in finding an admissible solution because the k cable tension are 0, the controller automatically switches to the parallel instance of the DGP solver. The preliminary tests are promising, and by exploiting multi-core CPUs there are no measurable differences on algorithm performances. However, this approach still does not offer a comprehensive solution of the DGP problem for cable robots that is real-time compliant.

The second issue concerns the behavior of the real-time module when the Kantorovitch test fails. As described previously, when $m = 6$ and maximum velocities of the platform are available, the solving routine still manages to find a solution, though only up to a certain accuracy. However, giving bounds to the search domain when less than 6 cables are taut is not simple. The first reason is that since not all the platform degrees of freedom are controlled, the end-effector cannot be easily bounded. The second reason is that the cable tension needs a search domain too.

The third challenging aspect related to this algorithm is that its effectiveness is highly dependent on both the choice of the reference points and the problem scale. The first issue can be easily explained by observing for example one equation from (2.27), i.e.

$$\left(\sum_{k=1}^{h}\beta_{ik}x_k - a_{ix}\right)^2 + \left(\sum_{k=1}^{h}\beta_{ik}y_k - a_{iy}\right)^2 + \left(\sum_{k=1}^{h}\beta_{ik}z_k - a_{iz}\right)^2 = \rho_i^2 \quad (4.2)$$

and the first element of the ith row of the Jacobian matrix of the system is

$$J_{i1} = 2\beta_{i1}\left(\sum_{k=1}^{h}\beta_{ik}x_k - a_{ix}\right) \quad (4.3)$$

Thus the first row of the Hessian matrix $\mathbf{H}^i$ of the ith equation will be, if $h = 4$,

$$\mathbf{H}^i_{1,1..12} = \begin{bmatrix} 2\beta_{i1}^2 & 0 & 0 & 2\beta_{i1}\beta_{i2} & 0 & 0 & 2\beta_{i1}\beta_{i3} & 0 & 0 & 2\beta_{i1}\beta_{i4} & 0 & 0 \end{bmatrix} \quad (4.4)$$

The other rows of this hessian matrix contain the same elements but in different positions. Thus the norm of $\mathbf{H}^i$ is given by

$$\|\mathbf{H}^i\|_\infty = 2|\beta_{i1}|\sum_{k=1}^{4}|\beta_{ik}| \quad (4.5)$$

The maximum value of $\|\mathbf{H}^i\|$ among all the equations in the system gives the value of the constant p_0 of the Kantorovitch theorem (see Sec. 3.2.1) and, being considered the fourth condition required by the theorem, if p_0 is small, the theorem has more chances to be successfully applied. With the current parameterization, the lowest value for the constant p_0 is 4, i.e. the norm of the hessian matrices related to point to point distance relations (2.28). Thus, considering also that coefficients β and α appear in the static equations, it is important to choose among the different sets of reference points the combination that gives the lowest coefficients β and α not only to reduce overestimation in secondary points evaluation but also to minimize p_0.

Here, instead, an example is proposed that shows how Kantorovitch success is affected also by the problem scaling. Let a robots with two cables be considered. By the notation presented in Sec. 2.3 the coordinates of its exit points are $a_{1x} = 0$, $a_{1z} = 0$, $a_{2x} = 7$ and $a_{2z} = 1.5$ and platform geometry is defined by $d_{12} = 3$, $d_{1g} = 2$, $d_{2g} = 2$. All the measurements are in meters. For cable lengths $\rho_1 = 3$ and $\rho_2 = 5.5$ the system has a solution $\mathbf{Y}_0$ in

$$\begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1.1197 \\ -2.7832 \end{bmatrix} \begin{bmatrix} x_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1.6640 \\ 0.1679 \end{bmatrix} \begin{bmatrix} x_g \\ z_g \end{bmatrix} = \begin{bmatrix} 0.0909 \\ -1.0681 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 0.9795 \\ 0.3768 \end{bmatrix} \tag{4.6}$$

Suppose that new cable lengths are now given by motors, and they are $\rho_1 = 3.003$ and $\rho_2 = 5.503$. So, the inverse of $\mathbf{J}(\mathbf{Y}_0)$ is computed and its norm is $\|\mathbf{\Gamma}_{\mathbf{Y}0}\|_\infty = 3.5418 = s_0$. Constant r_0 is computed as $\|\mathbf{\Gamma}_{\mathbf{Y}0}\mathbf{f}(\mathbf{Y}_0)\|_\infty = 0.00704$. Since constant $p_0 = 4$ in this case and the dimension of the problem is 8 the following inequality does not hold

$$16 p_0 s_0 r_0 = 1.5957 \not\leq 1 \tag{4.7}$$

However, if the geometric data are all multiplied by a scale factor $s_f = 1/2$, an equivalent problem is obtained. But if all the computations are repeated, since the following constants are now $s_0 = 4.3549$, $r_0 = 0.00211$ and $p_0 = 4$, the inequality in (4.7) is now satisfied

$$16 p_0 s_0 r_0 = 0.5886 \leq 1 \tag{4.8}$$

and the theorem now holds. However, how to determine the best scale factor for a given direct geometrico-static problem is a subject that deserves more investigations, and up to now we have dealt with it only by an empirical approach.

On-line computing of Wrench Feasible Workspace boundaries

In the perspective of developing a user interface which includes several tools for the analysis of the teaching prototype presented previously, a module for the on-line analysis of its constant-orientation workspace was been implemented. However, this is still at an early stage of development and the vast literature devoted to this subject suggests a more in-depth investigation of that topic [Pusey et al. 2004, Gouttefarde et al. 2006, Perreault et al. 2010, Ruiz et al. 2015, Gouttefarde et al. 2011, Bruckmann et al. 2008c].

Let the platform pose be defined by the six parameters $[x_{O'}, y_{O'}, z_{O'}, \varphi_x, \varphi_y, \varphi_z]$. Another real-time module that can be implemented in the control routine for the configuration with $m = 6$ cables in tension consists in a procedure that computes, for a given set of pose parameters $[z_{O'}, \varphi_x, \varphi_y, \varphi_z]$, the boundaries for $x_{O'}$ and $y_{O'}$ of the wrench feasible workspace (WFW). Let us consider the 6 equilibrium equations in (2.3)

$$\underbrace{\begin{bmatrix}\mathcal{L}_1 & \dots & \mathcal{L}_m & \mathcal{L}_e\end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} t_1 \\ \vdots \\ t_6 \\ Q \end{bmatrix} = \mathbf{0}\,, \tag{4.9}$$

where, since $\rho_i > 0$, $t_i = \tau_i/\rho_i \geq 0$, $i = 1, \dots, 6$. Once $[z_{O'}, \varphi_x, \varphi_y, \varphi_z]$ are assigned, the system has 6 equations in the 8 unknowns $[x_{O'}, y_{O'}, t_1, \dots, t_6]$. By observing that, if a given pose has at least one cable tension equal to zero, it is outside the WFW, when only a given unknown $t_i = 0$, the considered pose is lying on the frontier of the WFW.

By recalling that the equations (4.9) are linear in the unknowns $[t_1, \dots, t_6]$ and by setting one unknown $t_i = 0$, it is possible to choose 5 equations among the 6 in (4.9), solve them for the 5 unknowns $[t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_6]$ and then substitute them back to the remaining equation.

This function γ_i will depend on the sole variables $x_{O'}$ and $y_{O'}$ and it represents a part of the WFW boundary. By repeating this approach for t_i, with $i = 1, \dots, 6$, six curves γ_i can be analytically computed and they are all polynomials of the form

$$\gamma_i = c_{i0} + c_{i1}x + c_{i2}y + c_{i3}xy + c_{i4}xy^2 + c_{i5}x^2y + c_{i6}y^3 + c_{i7}x^3 + c_{i8}x^2 + c_{i9}y^2 \tag{4.10}$$

where $i = 1, \dots, 6$ and coefficients c_{ik} contain the geometric data of the robot and the assigned pose parameters. Each curve γ_i (as shown in Fig. 4.4) divides the plane $x_{O'}y_{O'}$ in two regions Ψ_i^- and Ψ_i^+ where t_i is negative

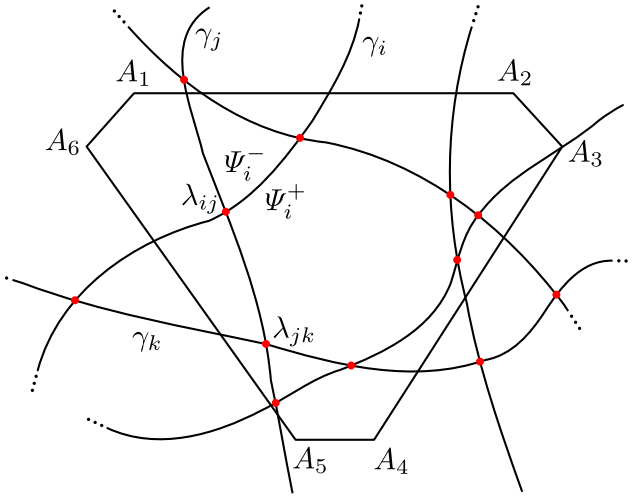


Figure 4.4 – Wrench feasible workspace determination for assigned orientation and $z_{O'}$ of the moving platform.

or positive, respectively. Thus the WFW is given by the intersection of all the Ψ_i regions, with $i = 1, \ldots, 6$.

The first step of the adopted strategy consists in computing all the intersections λ_{ij} between each pair of curves γ_i and γ_j, with $i, j = 1, \ldots, 6,\ j > i$. This can be done by eliminating, through software of symbolic algebra, variable $y_{O'}$ (or $x_{O'}$) from each pair of functions γ_i and γ_j, obtaining $C_2^6 = 15$ univariate polynomials p_{ij} in the only variable $x_{0'}$ (or $y_{O'}$). These polynomials are all of degree 10 and their coefficients depend on c_{ik} and c_{jk}.

Once all the real intersections λ are computed, it is possible to check all the parts of curves comprised between two intersections to determine whether they are part of the WFW boundaries. Let an intersection $\lambda_{ij} = (x_{ij}, y_{ij})$ be considered (see Fig. 4.5) and r_{it} and r_{jt} be the lines tangent to γ_i and γ_j in λ_{ij}. The expression of r_{it} is, for example:

$$\gamma_{ix}(x - x_{ij}) + \gamma_{iy}(y - y_{ij}) = 0 \tag{4.11}$$

where

$$\gamma_{ix} = \left.\frac{\partial \gamma_i}{\partial x}\right|_{x_{ij}, y_{ij}} \tag{4.12}$$

$$\gamma_{iy} = \left.\frac{\partial \gamma_i}{\partial y}\right|_{x_{ij}, y_{ij}} \tag{4.13}$$

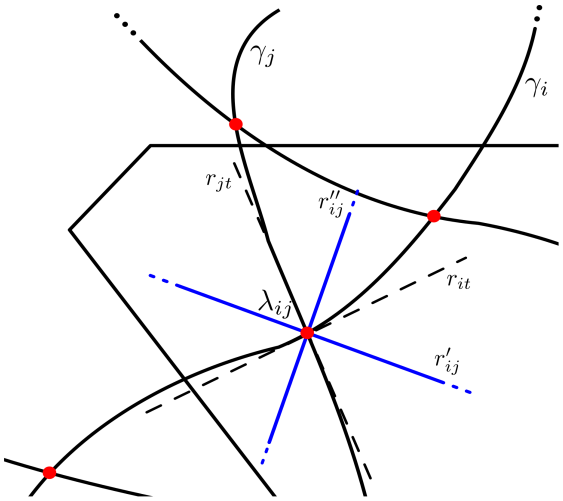


Figure 4.5 – Detail of the two lines bisecting the angles formed by the two tangents of γ_i and γ_j in λ_{ij}.

Then the two lines r'_{ij} and r''_{ij} bisecting the angles formed by the intersection of r_{it} and r_{jt} are derived and their expressions in parametric form are, respectively

$$r'_{ij} : \begin{cases} x = x_{ij} - b'\delta' \\ y = y_{ij} + a'\delta' \end{cases} \tag{4.14}$$

$$r''_{ij} : \begin{cases} x = x_{ij} - b''\delta'' \\ y = y_{ij} + a''\delta'' \end{cases} \tag{4.15}$$

where δ' and δ'' are real parameters and

$$a' = \frac{\gamma_{ix}}{\sqrt{\gamma_{ix}^2 + \gamma_{iy}^2}} + \frac{\gamma_{jx}}{\sqrt{\gamma_{jx}^2 + \gamma_{jy}^2}} \quad b' = \frac{\gamma_{iy}}{\sqrt{\gamma_{ix}^2 + \gamma_{iy}^2}} + \frac{\gamma_{jy}}{\sqrt{\gamma_{jx}^2 + \gamma_{jy}^2}} \tag{4.16}$$

$$a'' = \frac{\gamma_{ix}}{\sqrt{\gamma_{ix}^2 + \gamma_{iy}^2}} - \frac{\gamma_{jx}}{\sqrt{\gamma_{jx}^2 + \gamma_{jy}^2}} \quad b'' = \frac{\gamma_{iy}}{\sqrt{\gamma_{ix}^2 + \gamma_{iy}^2}} - \frac{\gamma_{jy}}{\sqrt{\gamma_{jx}^2 + \gamma_{jy}^2}} \tag{4.17}$$

Then, variables $[t_1, \ldots, t_6]$ are computed in the two pairs of points lying on r'_{ij} and r''_{ij} obtained by substituting $\delta' = \pm\epsilon$ and $\delta'' = \pm\epsilon$ in (4.15), where ϵ is an arbitrary small real number. If for one point, $t_i > 0$, $i = 1, \ldots, 6$, it is possible to consequently determine the corresponding sections of curves γ_i and γ_j constituting part of the WFW boundary. The algorithm proceeds

analyzing all the intersections λ and determines whether all the sections of curves γ deemed part of the WFW boundary form closed loops.

This procedure runs in less than 1.5ms so it is manageable in real-time. It is possible, thus, to assign platform orientation and $z_{O'}$ and computes at each cycle time the boundaries of the WFW of $x_{O'}$ and $y_{O'}$. However, this algorithm is at its preliminary stage of development and its implementation will be further improved in the future. For example, since the expressions of curves γ are known, the area of a section of the WFW determined by following this approach can be computed fairly quickly. It would thus be possible to assess whether the current trajectory of the robot or the external wrenches applied on the moving platform are leading to a dangerous contraction of the WFW or if the current pose of the end-effector is getting near to the WFW boundaries.

4.2 Test of the DGP solving procedure

The results of the tests made with the two versions of the code developed for finding the complete solution of the DGP of cable robots are presented here. These teste are representative of the mean performances of the code. In all examples, lengths are expressed in meters and angles in radians. Each solution reported in the tables at the end of chapter is given as

- a configuration number;
- the number m of taut cables;
- 3 parameters $x_{O'}$, $y_{O'}$ and $z_{O'}$ describing the position of the origin O' of the mobile frame in $Oxyz$;
- 3 angles φ_x, φ_y and φ_z, representing the orientation of the platform by three consecutive rotations along the x, y and z fixed axes;
- n values of cable tensions;
- a symbol that denotes whether the solution is stable ($>$) or not ($<$).

All tests were performed on a PC with an Intel® Xeon W3520 2.67 GHz CPU with 4 cores.

4.2.1 Test on a CDPR with 3 cables in tension

The first test (Table 4.3) is taken from Berti et al. [2013]. The aim is to compare the performances of the current code with respect to the original implementation made at the beginning of the Ph.D. program.

The algorithm presented in [Berti et al. 2013], with the same parameterization described in Section 2.3, finds all equilibrium configurations with 3 cables in tension in 202 seconds, but it does not verify whether solutions with slack cables exist and it does not assess stability. The current version of the code analyzes all possible equilibrium configurations with m cables in tension, $m \leq 3$, in roughly 25 seconds, including stability analysis. With the beta version, computation time is 22 seconds. All computation times refer to single-core executions. The same problem with the same parameterization was solved with HOM4PS 2,0, a software package which implements the polyhedral homotopy continuation method for solving polynomial systems of equations, in 29 seconds [Lee et al. 2008]. It must be noted that the solutions obtained by HOM4PS also include unfeasible and complex ones, and sub-DGPs for CDPRs with less than 3 cables in tension are not considered.

4.2.2 Test on the Cogiro prototype geometry

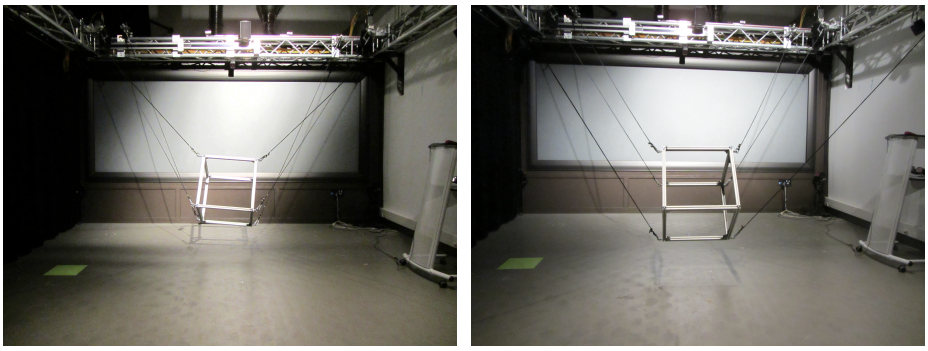
For the second test (Table 4.4), the geometry of the giant cable suspended prototype Cogiro developed by the LIRMM in Montpellier was used. This robot has 8 cables in crane configuration and it belongs to the over constrained family. However, it is possible to show that, under the assumption that cables are perfectly stiff, the maximum number of cables that can be under tension at the same time is 6 [Merlet 2012b]. The code to solve the cable robot DGP was developed on this assumption so it analyzes all the

$$N_{DGP} = \sum_{k=1}^{6} \frac{8!}{k!\,(8-k)!} = 246 \tag{4.18}$$

sub-DGPs. Table 4.4 reports all solutions found by running the code for all the 246 sets of taut cables and the number of solutions retrieved is 46, i.e. 6 with 6 cables in tension, 16 with 5 cables in tension, 15 with 4 cables in tension, 4 with 3 cables in tension and 5 with 2 cables in tension. The computation time is 1343 seconds by a single-core execution, and 348 seconds if the code is distributed over 4 CPUs. The beta version of the code shown better performances: 1219 seconds for single core execution and 314 for the parallel implementation.

4.2.3 Test on the Marionet-VR prototype

The third test (Table 4.5) refers to the large-scale robot MARIONET-VR [Merlet 2010]. This robot uses linear actuators instead of winches to change the cable lengths, and it is employed in a virtual-reality environment as a motion provider and haptic device. The moving platform was designed for demonstration purposes and its anchor points do not belong to the same plane. Table 4.5 reports all solutions found by running the code for all possible sets of taut cables, i.e. for 63 sub-problems. The number of solutions retrieved is 17, i.e. 3 with 6 cables in tension, 1 with 5 cables in tension, 7 with 4 cables in tension, 4 with 3 cables in tension and 2 with 2 cables in tension. The computation time is 702 seconds by a single-core execution, and 178 seconds if the code is distributed over 4 CPUs. Performances of the beta version of the code are practically the same. Feasibility was checked on the prototype shown in Fig. 4.6.



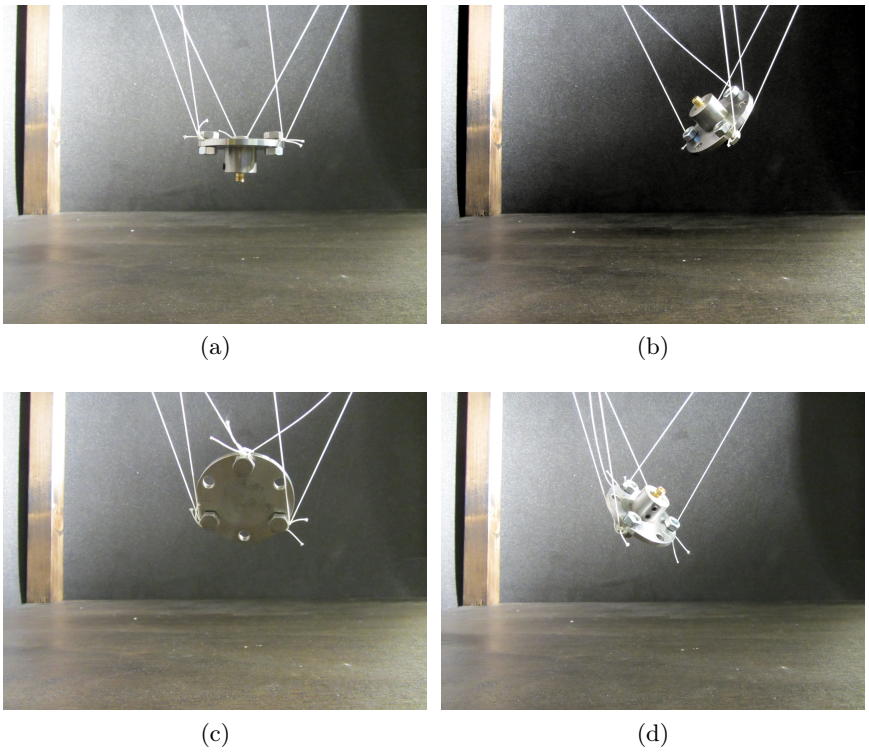
(a) (b)

Figure 4.6 – Validation of Test 3 on the large scale robot MARIONET-VR: (a) configuration No. 1 with 6 cables in tension; (b) configuration No. 3 with 6 cables in tension.

4.2.4 Test on the teaching prototype

This fourth test (Table 4.4) shows the case of a 6-6 CDPR having all cable anchor points on the moving platform lying on the same plane. As shown in Section 2.3, a specific parameterization can be used in this case. The algorithm finds 2 solutions with 6 cables in tension, 11 solutions with 5 taut cables, 12 with 4 and 9 with 3 in 187 seconds by a single-core execution, and in 49 seconds if the code is distributed over 4 CPUs. The beta version of the code finds all the solution in almost the same amount of time, 179 seconds.

Five out of 34 solutions are stable and they are reported in Table 4.4, for the sake of conciseness. Their feasibility was checked on the teaching prototype (Fig. 4.7).



(a) (b)

(c) (d)

Figure 4.7 – From Table 4.6: (a) configuration No. 1, (b) Configuration No. 3, (c) configuration No. 4, (d) configuration No. 5.

4.2.5 Discussion of results

As previously stated, the tests reported above are representative of the average performance of the problem-solving code. The first test shows that the code described in this thesis with the refinements introduced with respect to the first version presented in [Berti et al. 2013] has 4 times lower computation times. Moreover, it computes all the sub-DGPs and assesses stability. The third and the fourth tests present the complete analysis of the DGP of a 6-6 CDPR. For the latter test, thanks to planar disposition of the anchor points on the platform, the code takes advantage of simpler parameterizations for the sub-problems with 4, 5 and 6 cables, as explained

in Section 2.3, with a strong effect on computation times. The second and the third tests analyzes the DGP of robots with spatial disposition of the anchor points on the respective moving platforms. For this kind of geometry, the code needs higher computation times. However, performances are still particularly interesting. For the second example, without considering the 36 subproblems with 1 and 2 active cables, that are processed in less than 2 seconds, the code solves 56 sub-DGPs with 3 cables in tension, 70 with 4 cables in tension, 56 with 5 cables in tension and 28 with 6 cables in tension in 1343 with the standard version of the code (1219 with the beta version), with a mean computation time of nearly 7 (6) seconds for each problem. For the third example, without considering the 21 subproblems with 1 and 2 active cables (they are generally processed in less than 1 second), the code solves 20 sub-DGP with 3 cables in tension, 15 with 4 cables in tension, 6 with 5 cables in tension and 1 with 6 cables in tension in 702 seconds, with a mean computation time of roughly 17 seconds for each problem.

It is worth mentioning that the obtained computation times are not necessarily the lowest ones. It can happen that the procedure that selects the best cable numbering for each DGP fails to find the best combination. Indeed, the ranking criterion expressed in (3.16) is an empirical relation coming from the data gathered during tests and on the results obtained in [Merlet 2004], which deserve further investigation.

In tests 2, 3 and 4, multiple feasible solutions are obtained for a given set of cable lengths. In the third example, the robot (Fig. 4.6a) can easily switch from configuration No. 1 to configuration No. 3 (Fig. 4.6b) (passing through an unstable configuration with 4 taut cables) and vice-versa, while configuration No. 2 cannot be directly reached without detaching cables. Configuration No. 3 with 6 cables in tension is almost coincident with configuration No. 4, with 5 taut cables, as can be noted by the small value of tension in the fourth cable. Accordingly, in this part of the workspace, the robot may easily act as an underconstrained one in an uncontrolled way. In the fourth test, if the platform is in configuration No. 1 (Fig. 4.7a), a potential external disturbance can cause the robot to fall in one of the 3 solutions with 5 cables in tension (Figs. 4.7b, 4.7c and 4.7d show configurations No. 3, 4 and 5), and without additional external sensors it is difficult to determine in which solution branch the robot is operating. Conversely, configuration No. 2, with 6 cables in tension, is not directly achievable with a finite movement of the platform without detaching and reattaching cables.

The current version of the code can determine the stability of equilibrium only after the computation of all statically-admissible solutions. The results obtained during the experimental campaign show that stable solutions are

usually a small subset of statically-admissible poses. Therefore, the implementation of a stability filter during the computation would allow the domain regions where unstable configurations may happen to be discarded, thus allowing the direct computation of stable solutions. A few attempts have been made in order to develop such a filter, but the overestimation introduced by interval-analysis computation severely compromises its effectiveness. On the other hand, the analysis of unstable configurations may be of practical interest. Indeed, an unstable solution may provide a bridge towards stable configurations. The possibility of configuration changes without crossing singularities is indeed specific to CDPRs. In this situation, the set of physical constraints represented by the interference between the platform and taut cables may be reduced, thus disclosing new paths for other operating modes.

DATA

$\mathbf{a}_1 = [0,0,0]$ $\mathbf{a}_2 = [10,0,0]$ $\mathbf{a}_3 = [0,12,0]$
$\mathbf{b}'_1 = [0.817,0,0]$ $\mathbf{b}'_2 = [-0.408,0.707,0]$ $\mathbf{b}'_3 = [-0.408,-0.707,0]$
$\mathbf{g}' = [0,0,-0.577]$ $(\rho_1,\rho_2,\rho_3) = [7.5,10,9.5]$ $Q = 1$

Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	Stability
1	3	2.745	3.979	5.506	3.007	0.340	0.109	0.526	0.511	0.581	$>$
2	3	1.700	3.687	5.809	0.339	−1.036	−2.596	0.676	0.251	0.486	$<$
3	3	3.020	4.757	3.879	−0.038	0.027	0.776	0.590	0.783	0.956	$<$
4	3	1.846	4.074	5.322	2.146	−0.708	2.423	0.684	0.305	0.614	$<$
5	3	2.138	4.287	6.030	−0.482	−0.360	−2.211	0.546	0.325	0.550	$<$
6	3	3.499	5.369	4.709	−2.908	−0.174	−2.659	0.289	0.787	0.912	$<$
–	2	-	-	-	-	-	-	-	-	-	-
–	1	-	-	-	-	-	-	-	-	-	-

Table 4.3 – Solution set for a CDPR suspended by 3 cables.

DATA

$\mathbf{a}_1 = [-7.175, -5.244, 5.462]$ $\mathbf{a}_2 = [-7.316, -5.103, 5.472]$ $\mathbf{a}_3 = [-7.303, 5.236, 5.476]$ $\mathbf{a}_4 = [-7.161, 5.373, 5.485]$
$\mathbf{a}_5 = [7.182, 5.348, 5.488]$ $\mathbf{a}_6 = [7.323, 5.206, 5.499]$ $\mathbf{a}_7 = [7.302, -5.133, 5.489]$ $\mathbf{a}_8 = [7.161, -5.270, 5.497]$
$\mathbf{b}'_1 = [0.503, -0.493, 0.000]$ $\mathbf{b}'_2 = [-0.510, 0.351, 0.998]$ $\mathbf{b}'_3 = [-0.503, -0.270, 0.000]$ $\mathbf{b}'_4 = [0.496, 0.356, 1.000]$
$\mathbf{b}'_5 = [-0.503, 0.493, 0.000]$ $\mathbf{b}'_6 = [0.500, -0.340, 0.999]$ $\mathbf{b}'_7 = [0.502, 0.275, 0.000]$ $\mathbf{b}'_8 = [-0.505, -0.346, 0.998]$
$\mathbf{g}' = [0, 0, 0.5]$ $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \rho_7, \rho_8) = [10.482, 9.839, 10.160, 9.968, 10.310, 9.422, 9.663, 9.656]$ $Q = 1$

Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	Stability
1	6	−0.465	−0.309	1.988	2.177	−0.505	1.499	0	0.655	0	0.680	0.309	0.311	0.632	0.136	>
2	6	−0.182	−0.481	1.631	−0.027	−0.718	−3.121	0	0.874	0.396	0.071	0.445	0.279	0.683	0	>
3	6	0.519	1.329	2.460	1.048	1.339	−0.765	0.454	0.279	0	0.761	0	0.879	0.340	0.305	>
4	6	0.635	0.381	1.697	0.396	0.646	−2.999	0.389	0.269	0.710	0	0	0.818	0.471	0.091	>
5	6	0.921	0.236	1.989	−2.193	0.243	1.421	0.358	0.219	0.617	0.154	0	0.628	0	0.724	>
6	6	0.416	0.564	1.627	2.761	−1.363	−1.210	0.670	0.085	0.027	0.589	0	0.741	0.697	0	>
7	5	0.245	−0.521	1.418	−2.147	1.260	−0.514	0	0	0.570	0.599	0	0.012	0.634	0.543	>
8	5	0.255	−0.039	0.828	0.033	−0.099	0.055	0	0	0.622	0.546	0	0.011	0.626	0.551	>
9	5	−0.836	−0.182	2.462	−1.369	−0.222	−1.386	0	0.236	0	1.459	0	0.147	0.591	0.895	<
10	5	0.339	1.246	2.387	0.688	1.067	−0.883	0	0.205	0	1.422	0	0.381	0.446	0.916	<
11	5	−0.405	−0.664	1.902	1.131	−0.659	2.580	0	1.101	0.328	0	0.728	0.227	0.543	0	<
12	5	0.250	−1.140	2.361	−1.079	−1.201	−0.403	0	0.198	0.607	0.770	0	0.101	0	1.434	<
13	5	−0.111	−0.304	1.747	0.124	−0.892	2.799	0	0.711	0.365	0.295	0.570	0	0.855	0	<
14	5	0.469	1.295	2.572	1.868	1.330	0.053	0.651	0	0	0.773	0	0.826	0.543	0.080	<
15	5	0.577	0.242	1.789	0.462	0.814	−3.137	0.399	0	0.963	0	0	0.585	0.448	0.384	<
16	5	0.588	0.084	1.779	−1.430	0.391	1.450	0.408	0	1.112	0	0	0.436	0.106	0.956	<
17	5	0.712	−0.492	1.990	−1.970	0.959	0.693	0.033	0	1.485	0.079	0	0.034	0	1.579	<
18	5	0.109	0.045	1.346	0.829	−0.476	0.568	0.682	0	0.274	0.306	0	0.660	0.678	0	<
19	5	1.198	0.503	2.442	1.086	0.461	−1.454	0.617	0.720	0	0.308	0	1.370	0	0.352	<

20	5	0.574	1.289	2.541	1.799	1.509	−0.065	0.717	0.121	0	0.604	0	0.993	0.448	0	<
21	5	0.976	0.674	2.135	−1.322	0.548	2.532	0.798	0.233	0.497	0	0	1.176	0.385	0	<
22	5	0.378	−0.289	1.218	−0.981	0.610	0.341	0.435	0.696	0.082	0	1.138	0	0.110	0	<
23	4	0.188	1.161	2.654	1.703	1.021	0.109	0	0	0	1.554	0	0.177	0.705	0.790	<
24	4	−0.831	−0.141	2.560	−1.555	−0.276	−1.367	0	0.111	0	1.558	0	0	0.769	0.820	<
25	4	−0.269	−0.504	2.419	−0.622	−0.445	−1.262	0	0.476	0	1.597	0	0.490	0	1.611	<
26	4	0.219	0.092	2.744	−3.098	−0.006	−1.393	0	0.624	0	0.652	0	0.655	0	0.667	<
27	4	0.748	0.696	2.428	0.628	0.447	−1.260	0	0.438	0	1.601	0	0.549	0	1.582	<
28	4	0.230	0.094	2.685	3.137	−0.016	1.380	0	0.621	0	0.641	0	0.648	0	0.647	<
29	4	−0.163	−0.612	2.814	1.784	0.414	2.682	0	0.460	1.442	0	0	0.258	1.526	0	<
30	4	0.751	0.638	2.900	−1.810	−0.389	2.693	0	0.232	1.556	0	0	0.524	1.485	0	<
31	4	0.275	−0.003	1.747	0.069	0.006	−3.058	0	0.721	0.766	0	0	0.733	0.788	0	<
32	4	−0.256	−0.460	1.875	0.862	−0.728	2.460	0	1.030	0.361	0	0.891	0	0.612	0	<
33	4	0.318	−1.088	2.441	−1.510	−1.140	0.042	0	0.099	0.737	0.720	0	0	0	1.509	<
34	4	0.962	0.456	3.039	1.717	0.820	−0.948	1.540	0	0	0.025	0	1.549	0	0.065	<
35	4	0.509	1.290	2.573	1.965	1.400	0.129	0.718	0	0	0.702	0	0.891	0.541	0	<
36	4	0.795	0.425	2.084	−1.002	0.601	2.378	0.954	0	0.598	0	0	1.096	0.437	0	<
37	4	1.301	0.319	2.574	1.365	0.301	−1.528	0.868	0.686	0	0	0	1.557	0	0.092	<
38	3	−0.414	0.404	2.333	2.198	−0.937	0.499	0	0	0	1.717	0	0	1.519	0.198	<
39	3	0.644	−0.429	1.929	−1.728	0.999	0.879	0	0	1.631	0	0	0.015	0	1.637	<
40	3	0.072	0.031	1.854	0.882	−0.799	1.584	0	0	0.302	1.318	0	0	1.634	0	<
41	3	0.964	0.458	3.036	1.711	0.836	−0.959	1.561	0	0	0	0	1.571	0	0.037	<
42	2	0.123	−0.048	4.169	−3.134	−0.012	−0.027	0	0	0	2.110	0	0	0	2.115	<
43	2	−0.250	0.309	2.232	1.667	−1.008	0.934	0	0	0	1.822	0	0	1.826	0	<
44	2	0.215	−0.098	2.751	−0.007	0.005	2.028	0	0	1.808	0	0	0	1.817	0	<
45	2	0.176	0.171	3.362	0.006	0.002	−1.928	0	4.273	0	0	0	4.277	0	0	<
46	2	0.169	0.170	5.359	−3.139	0.004	−3.128	0	4.273	0	0	0	4.277	0	0	<

Table 4.4 – Complete solution set for the the 8 cable robot Cogiro. Platform anchor points are not coplanar.

DATA													
$\mathbf{a}_1 = [1.886, 0.558, 2.590]$ $\mathbf{a}_2 = [1.921, -2.876, 2.750]$ $\mathbf{a}_3 = [-1.498, 0.809, 2.745]$ $\mathbf{a}_4 = [1.567, 0.895, 2.600]$ $\mathbf{a}_5 = [-1.581, -3.015, 2.740]$ $\mathbf{a}_6 = [-1.692, 0.372, 2.580]$ $\mathbf{b}'_1 = [0.63, 0.59, 0.585]$ $\mathbf{b}'_2 = [0.63, 0.585, 0.015]$ $\mathbf{b}'_3 = [-0.03, 0.05, 0.585]$ $\mathbf{b}'_4 = [0.63, 0.05, 0.585]$ $\mathbf{b}'_5 = [-0.03, 0.585, 0.015]$ $\mathbf{b}'_6 = [-0.03, 0.59, 0.585]$ $\mathbf{g}' = [0.3, 0.3, 0.3]$ $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6) = [2.755, 3.519, 2.849, 2.837, 3.489, 2.609]$ $Q = 1$													
Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	Stability
1	6	−0.270	0.235	0.778	2.554	0.124	0.080	0.398	0.226	0.248	0.078	0.244	0.268	>
2	6	0.253	−0.520	0.338	0.960	−0.105	−3.077	0.262	0.291	0.293	0.278	0.314	0.283	>
3	6	−0.278	−1.470	0.549	−0.670	0.014	−0.043	0.374	0.271	0.156	0.004	0.376	0.220	>
4	5	−0.279	−1.470	0.549	−0.669	0.016	−0.046	0.381	0.267	0.161	0	0.380	0.213	>
5	4	−0.502	−0.452	0.351	0.817	−0.913	−2.873	0	0.300	0	0.492	0.243	0.679	<
6	4	−0.289	−0.592	0.553	0.940	0.069	0.008	0	0.395	0.496	0.464	0.432	0	<
7	4	0.530	−0.505	2.234	−1.986	−0.324	−3.011	1.403	0	0.353	0	1.332	0.134	<
8	4	−0.217	0.025	1.237	−2.420	0.068	0.058	0.545	0.082	0	0	0.177	0.506	<
9	4	0.334	−0.380	2.325	−1.695	0.067	3.134	0.464	1.200	0	0	0.031	1.352	<
10	4	0.151	−0.579	2.411	−2.009	0.333	3.075	0.054	1.337	0	0.395	0	1.409	<
11	4	0.879	−0.455	0.751	0.862	0.757	3.036	0.658	0.229	0.524	0	0.339	0	<
12	3	0.132	−0.597	2.425	−2.065	0.376	3.057	0	1.355	0	0.471	0	1.419	<
13	3	0.386	−0.364	2.294	−1.703	−0.087	−3.059	1.393	0	0	0	1.256	0.480	<
14	3	0.552	−0.525	2.211	−2.131	−0.431	−2.963	1.421	0	0.546	0	1.374	0	<
15	3	0.332	−0.383	2.328	−1.700	0.073	3.132	0.442	1.229	0	0	0	1.373	<
16	2	−0.218	−1.192	1.499	1.220	0.379	1.539	0	1.569	1.612	0	0	0	<
17	2	0.342	−0.676	1.244	1.194	−0.403	−1.458	0	0	0	1.691	1.681	0	<

Table 4.5 – Solution set for the CDPR suspended by 6 cables MARIONET-VR: cable anchor points on the platform are not coplanar.

DATA

$\mathbf{a}_1 = [0.238, 0.063, 0.785]$ $\mathbf{a}_2 = [0.736, 0.551, 0.785]$ $\mathbf{a}_3 = [0.738, 0.551, 0.781]$
$\mathbf{a}_4 = [0.062, 0.729, 0.785]$ $\mathbf{a}_5 = [0.064, 0.729, 0.789]$ $\mathbf{a}_6 = [0.238, 0.062, 0.785]$
$\mathbf{b}'_1 = [0.369, -0.002, 0.025]$ $\mathbf{b}'_2 = [0.369, 0.002, 0.025]$ $\mathbf{b}'_3 = [-0.017, 0.033, 0.025]$
$\mathbf{b}'_4 = [-0.020, 0.031, 0.025]$ $\mathbf{b}'_5 = [-0.020, -0.031, 0.025]$ $\mathbf{b}'_6 = [-0.017, -0.033, 0.025]$
$\mathbf{g}' = [0, 0, 0.019]$ $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6) = [0.799, 0.790, 0.796, 0.787, 0.798, 0.791]$ $Q = 1$

Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	Stability
1	6	0.346	0.453	0.066	−0.003	0.005	−0.518	0.153	0.228	0.156	0.225	0.163	0.219	>
2	6	0.348	0.453	0.088	−0.019	0.004	2.081	0.231	0.165	0.233	0.162	0.239	0.155	>
3	5	0.363	0.454	0.122	−2.420	0.406	−1.341	0.202	0.333	0.105	0.362	0	0.151	>
4	5	0.341	0.437	0.122	2.419	0.345	0.449	0.100	0.362	0	0.159	0.204	0.327	>
5	5	0.335	0.464	0.122	−3.124	−0.823	2.636	0	0.149	0.210	0.326	0.116	0.352	>
6	5	0.366	0.401	0.098	−0.384	0.623	1.265	0.445	0.026	0.382	0.005	0.334	0	<
7	5	0.341	0.539	0.118	−1.401	−0.157	3.104	0.066	0.237	0.180	0.483	0	0.206	<
8	5	0.356	0.436	0.130	3.119	0.448	−0.087	0.023	0.336	0.045	0.373	0	0.392	<
9	5	0.294	0.463	0.099	0.710	−0.028	1.369	0.384	0.002	0.333	0	0.459	0.013	<
10	5	0.424	0.416	0.117	1.019	1.163	−2.600	0.173	0.480	0	0.216	0.072	0.232	<
11	5	0.332	0.456	0.131	2.809	−0.256	2.131	0.022	0.381	0	0.409	0.019	0.338	<
12	5	0.356	0.470	0.131	−2.731	−0.202	−2.250	0	0.388	0.037	0.327	0.058	0.360	<
13	5	0.277	0.404	0.116	1.217	−0.877	2.383	0	0.223	0.067	0.252	0.165	0.469	<
14	4	0.349	0.467	0.117	−1.979	0.416	−1.395	0.331	0.232	0.196	0.387	0	0	<
15	4	0.347	0.463	0.137	−1.899	0.370	1.567	0.384	0.134	0.234	0.429	0	0	<
16	4	0.414	0.396	0.121	0.399	1.451	2.627	0.424	0.461	0	0.142	0.134	0	<
17	4	0.370	0.440	0.138	3.034	0.894	2.441	0.397	0.439	0	0.160	0.173	0	<

18	4	0.361	0.443	0.117	2.019	0.578	0.036	0.201	0.386	0	0	0.341	0.215	<
19	4	0.354	0.449	0.138	1.951	0.612	2.940	0.234	0.426	0	0	0.401	0.120	<
20	4	0.363	0.539	0.122	−1.466	−0.471	2.666	0.127	0	0.433	0.466	0	0.135	<
21	4	0.347	0.480	0.138	−2.262	−0.341	0.231	0.167	0	0.409	0.442	0	0.152	<
22	4	0.324	0.440	0.139	2.339	−0.467	−1.519	0	0.159	0.172	0	0.414	0.426	<
23	4	0.264	0.424	0.122	1.437	−0.567	1.751	0	0.142	0.133	0	0.437	0.448	<
24	4	0.339	0.448	0.137	2.837	−1.247	−0.602	0	0	0.396	0.141	0.228	0.417	<
25	4	0.330	0.445	0.117	2.891	−1.187	2.648	0	0	0.336	0.223	0.210	0.376	<
26	3	0.367	0.476	0.142	−1.991	0.768	2.158	0.364	0.413	0	0.406	0	0	<
27	3	0.349	0.440	0.139	2.006	0.809	2.651	0.376	0.417	0	0	0.387	0	<
28	3	0.359	0.462	0.138	−1.864	0.152	1.207	0.371	0	0.390	0.420	0	0	<
29	3	0.311	0.462	0.096	0.472	−0.036	1.353	0.392	0	0.358	0	0.445	0	<
30	3	0.378	0.429	0.106	0.355	0.320	−2.882	0	0.435	0	0.354	0	0.412	<
31	3	0.295	0.444	0.108	−0.051	−0.653	−2.915	0	0.335	0	0.442	0	0.421	<
32	3	0.355	0.425	0.142	1.877	0.215	−2.738	0	0.404	0	0	0.381	0.398	<
33	3	0.319	0.459	0.142	−2.344	−1.141	−1.026	0	0	0.378	0.412	0	0.394	<
34	3	0.334	0.459	0.139	2.297	−1.121	−0.474	0	0	0.381	0	0.395	0.406	<

Table 4.6 – Solution set for a CDPR suspended by 6 cables: all cable anchor points on the platform are coplanar.

Chapter 5

Conclusions

This thesis presents an approach to deal with displacement analysis problems of cable-driven parallel robots based on interval-analysis methods. In particular, the developed methods aim to solve the direct geometrico-static problem in a comprehensive and guaranteed way.

The first chapter gives an overview on the cable-driven parallel robots state of the art and describes the main advantages offered by that this particular architecture of parallel robots. Indeed, they can provide very high velocities to the end-effector and may have very large workspaces thanks to the possibility of coiling cables on actuated winches. On the other hand, cable flexibility also causes the main issues on their kinematic analysis. In particular, determining the end-effector pose being given the cable lengths is a problem that must be carefully dealt with. Since cables cannot withstand compressive loads, the current robot platform pose may have only a subset m over the n cables that are taut. Moreover, when the number of taut cables $m < 6$, not all the platform degrees of freedom can be restrained and the robot end-effector still preserves some freedoms. This implies that its pose is ultimately determined by static equilibrium. Therefore the direct kinematic problem of cable robots is strictly related to their static analysis and it is called a direct geomtrico-static problem (DGP).

These aspects motivate the development of the codes and procedures presented in this thesis. The main tool used for the implementation of the described algorithm was interval analysis. Interval analysis is a numerical technique that comprehends methods to deal with a large variety of problems and to obtain results guaranteed against elimination and round-off errors. But in the case of cable robot DGPs, interval analysis offers other important advantages. For a better comprehension of the subsequent chapters, the

second part of the first chapter describes some basics elements of this theory, and lists some of its possible applications in robotics.

In the second chapter, the main definitions used through all the manuscript are introduced and the assumptions made in order to model cable-robots and develop the procedures are stated and discussed. Then the geometrico-static model for CDPRs having $m = 1, \ldots, 6$ cables in tension is presented along with the different systems of equations used to solve the related DGPs. The second chapter reports also a brief overview of the different parameterizations used during the development of the code but that turned out to be ineffective.

In the third chapter, the code structure was described in detail together with its main procedures. Then particular attention was devoted to some refinements that are fundamental for the reliability of the code. The last part of the third chapter describes a new version, currently still in a development phase.

The code was tested on real prototypes, and during the Ph.D. program, in order to simplify testing operations, a small teaching prototype was built. Thanks to this prototype it was possible to develop a version of the DGP solving routine that can be used in real-time together with an online algorithm to compute the wrench feasible workspaces boundaries for a given orientation and height of the moving platform. Then, in the second part of the fourth chapter, numerical results of the most relevant tests conducted on the offline algorithm for finding the complete set of DGP solutions were presented and discussed.

5.1 Future perspectives

The first version of the code presented in the third chapter can be considered both a good framework and a starting point for further developments. Its performances are good and it was proven to be reliable, but it can be further improved in several ways.

The first one consists in completing and refining the beta version. Indeed, in order to measure its real effectiveness, the complete revision of all the procedures implemented up to now has to continue in order to avoid all the possible multiple computations of the same functions, code blocks or boxes.

The second family of possible improvements consists in adding further modules to implement more realistic cable models. A first version of such a code was presented in [Merlet & Alexandre-dit Sandretto 2015] that offers several hints for further enhancements of the methods described in this thesis.

Then, more attention will be devoted to the development of real-time procedures. First of all, an interval-based algorithm for the inverse geometrico-static problem is currently under study, combined with a trajectory planning module that could take into account paths with different numbers of taut cables. Second, cable interferences must also be considered [Blanchet & Merlet 2014]. Indeed, cable-cable, cable-platform and cable-environment collision analysis must necessarily be integrated into the trajectory planning and it can also significantly improve the DGP solving procedures. Third, up to now a real-time compliant procedure that computes all the sub-DGPs is not currently available but it remains a long-term objective. In order to achieve that result, the following issues must be addressed first:

- determine the smallest search domain of the platform pose variables after a single cycle-time, including the unknowns on cable tensions when there are less than 6 taut cables;
- investigate if the Kantorovitch theorem and the inflation procedure could be further specialized for cable robot kinematic problems.

Moreover, the study of how uncertainties on robot parameters affects robot control will be dealt with. Finally, as explained at the end of the fourth chapter, implementing the adopted stability check in filter procedures is not manageable. However, especially for real-time control, this is a very important issue so other strategies will be investigated and tested.

Bibliography

G. Abbasnejad & M. Carricato (2012). 'Real solutions of the direct geometrico-static problem of under-constrained cable-driven parallel robots with 3 cables: a numerical investigation'. *Meccanica* **47**(7):1761–1773.

G. Abbasnejad & M. Carricato (2015). 'Direct Geometrico-Static Problem of Underconstrained Cable-Driven Parallel Robots with n Cables'. *IEEE Transactions on Robotics, DOI:10.1109/TRO.2015.2393173* **31**(2).

Airbus (2010). *Simulation | Airbus, a leading aircraft manufacturer*.

J. Albus, et al. (1993). 'The Nist robocrane'. *Journal of Robotic Systems* **10**(5):709–724.

J. Alexandre Dit Sandretto (2013). *Certified calibration of parallel cable-driven robots*. Theses, Université Nice Sophia Antipolis.

J. Angeles (2002). *Fundamentals of robotic mechanical systems*, vol. 2. Springer.

S. Behzadipour & A. Khajepour (2006). 'Stiffness of cable-based parallel manipulators with application to stability analysis'. *Journal of mechanical design* **128**(1):303–310.

A. Berti, et al. (2013). 'Solving the Direct Geometrico-Static Problem of 3-3 Cable-Driven Parallel Robots by Interval Analysis: Preliminary Results'. In T. Bruckmann & A. Pott (eds.), *Cable-Driven Parallel Robots*, vol. 12 of *Mechanisms and Machine Science*, pp. 251–268. Springer Berlin Heidelberg.

L. Blanchet & J.-P. Merlet (2014). 'Interference detection for cable-driven parallel robots (CDPRs)'. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference*, pp. 1413–1418.

P. Bosscher, et al. (2007). 'Cable-suspended robotic contour crafting system'. *Automation in Construction* **17**(1):45–55.

R. Bostelman, et al. (1994). 'Applications of the NIST RoboCrane'. In *Proceedings of the 5th International Symposium on Robotics and Manufacturing*, pp. 14–18.

R. Bostelman, et al. (1999). 'Delivery of an Advanced Double-Hull Ship Welding System using RoboCrane.'. In *IIA/SOCO*. Citeseer.

R. Bostelman, et al. (2002). 'The flying carpet: a tool to improve ship repair efficiency'. In *American Society of Naval Engineers Symposium, Bremerton, WA, Sept*, pp. 10–12.

S. Bouchard, et al. (2010). 'On the ability of a cable-driven robot to generate a prescribed set of wrenches'. *Journal of Mechanisms and Robotics* **2**(1):011010.

T. Bruckmann, et al. (2008a). 'Wire Robots Part I: Kinematics, Analysis & Design'. In J.-H. Ryu (ed.), *Parallel Manipulators, New Developments*. I-Tech Education and Publishing.

T. Bruckmann, et al. (2008b). 'Wire robots part II-dynamics, control & application'. *Parallel Manipulators-New Developments* pp. 133–152.

T. Bruckmann, et al. (2008c). 'Continuous workspace analysis, synthesis and optimization of wire robots'. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 59–66. American Society of Mechanical Engineers.

T. Bruckmann, et al. (2006). 'Calculating force distributions for redundantly actuated tendon-based Stewart platforms'. In J. Lennarčič & B. Roth (eds.), *Advances in Robot Kinematics*, pp. 403–412. Springer Netherlands.

T. Bruckmann, et al. (2013). 'An energy-efficient wire-based storage and retrieval system'. In *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pp. 631–636.

M. Carricato (2013). 'Direct geometrico-static problem of underconstrained cable-driven parallel robots with three cables'. *Journal of Mechanisms and Robotics* **5**(3):031008.

M. Carricato & J.-P. Merlet (2010). 'Geometrico-Static Analysis of Under-Constrained Cable-Driven Parallel Robots'. In J. Lenarčič & M. M.

Stanisic (eds.), *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 309–319. Springer Netherlands.

M. Carricato & J.-P. Merlet (2011). 'Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables'. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pp. 3011–3017.

M. Carricato & J.-P. Merlet (2013). 'Stability Analysis of Underconstrained Cable-Driven Parallel Robots'. *IEEE Transactions on Robotics* **29**(1):288–296.

R. Clavel (1988). 'A fast robot with parallel geometry'. In *Proc. Int. Symposium on Industrial Robots*, pp. 91–100.

J.-F. Collard & P. Cardou (2013). 'Computing the lowest equilibrium pose of a cable-suspended rigid body'. *Optimization and Engineering* **14**(3):457–476.

T. Dallej, et al. (2011). 'Towards vision-based control of cable-driven parallel robots'. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 2855–2860. IEEE.

D. Daney, et al. (2006). 'Interval method for calibration of parallel robots: Vision-based experiments'. *Mechanism and Machine Theory* **41**(8):929–944.

D. Daney, et al. (2004). 'Interval methods for certification of the kinematic calibration of parallel robots'. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2, pp. 1913–1918. IEEE.

R. Dekker, et al. (2006). 'Design and Testing of an Ultra-high-speed Cable Robot'. *Int. J. Robot. Autom.* **21**(1):25–34.

N. Douglas (1998). *Numerical computing disasters.*

S. Ehardt (2015). *Bowden cable.*

A. Fattah & S. K. Agrawal (2005). 'On the design of cable-suspended planar parallel robots'. *Journal of mechanical design* **127**(5):1021–1028.

J. Fink, et al. (2011). 'Planning and control for cooperative manipulation and transportation with aerial robots'. *The International Journal of Robotics Research* **30**(3):324–334.

P. Gallina, et al. (2001). '3-dof wire driven planar haptic interface'. *Journal of Intelligent and Robotic Systems* **32**(1):23–36.

B. Gao, et al. (2012). 'Combined inverse kinematic and static analysis and optimal design of a cable-driven mechanism with a spring spine'. *Advanced Robotics* **26**(8-9):923–946.

V. Gough (1956). 'Contribution to discussion of papers on research in automobile stability, control and tyre performance'. In *Proc. Auto Div. Inst. Mech. Eng*, vol. 171, pp. 392–394.

M. Gouttefarde, et al. (2011). 'Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots'. *IEEE Transactions on Robotics* **27**(1):1–13.

M. Gouttefarde, et al. (2006). 'Determination of the wrench-closure workspace of 6-DOF parallel cable-driven mechanisms'. In J. Lennarčič & B. Roth (eds.), *Advances in Robot Kinematics*, pp. 315–322. Springer Netherlands.

M. Gouttefarde, et al. (2014). 'Kinetostatic Analysis of Cable-Driven Parallel Robots with Consideration of Sagging and Pulleys'. In J. Lenarčič & O. Khatib (eds.), *Advances in Robot Kinematics*, pp. 213–221. Springer International Publishing.

M. Grebenstein, et al. (2012). 'The hand of the DLR Hand Arm System: Designed for interaction'. *The International Journal of Robotics Research* **31**(13):1531–1555.

E. Hansen & G. W. Walster (2003). *Global optimization using interval analysis: revised and expanded*. CRC Press.

T. Heyden & C. Woernle (2006). 'Dynamics and flatness-based control of a kinematically undetermined cable suspension manipulator'. *Multibody System Dynamics* **16**(2):155–177.

M. Hiller, et al. (2005). 'Design, analysis and realization of tendon-based parallel manipulators'. *Mechanism and Machine Theory* **40**(4):429–445.

L. Jaulin (2001). *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer.

Q. Jiang & V. Kumar (2010a). 'The direct kinematics of objects suspended from cables'. In *ASME 2010 International Design Engineering Technical*

Conferences and Computers and Information in Engineering Conference, pp. 193–202. American Society of Mechanical Engineers.

Q. Jiang & V. Kumar (2010b). 'The inverse kinematics of 3-D towing'. In *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 321–328. Springer.

S. Kawamura, et al. (2000). 'High-speed manipulation by using parallel wire-driven robots'. *Robotica* **18**(01):13–21.

C. Kossowski & L. Notash (2002). 'CAT4 (Cable Actuated Truss–4 Degrees of Freedom): A novel 4 DOF cable actuated parallel manipulator'. *Journal of Robotic Systems* **19**(12):605–615.

K. Kozak, et al. (2006). 'Static analysis of cable-driven manipulators with non-negligible cable mass'. *IEEE Transactions on Robotics* **22**(3):425–433.

J. Lamaury & M. Gouttefarde (2013). 'Control of a large redundantly actuated cable-suspended parallel robot'. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4659–4664. IEEE.

J. Lamaury, et al. (2012). 'Design and Control of a Redundant Suspended Cable-Driven Parallel Robots'. In J. Lenarcic & M. Husty (eds.), *Latest Advances in Robot Kinematics*, pp. 237–244. Springer Netherlands.

D. E. Lee (1992). 'Automated all-weather cargo transfer system'. US Patent 5,154,561.

T. Lee, et al. (2008). 'HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method'. *Computing* **83**(2-3):109–133.

J.-P. Merlet (2004). 'Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis'. *The International Journal of robotics research* **23**(3):221–235.

J.-P. Merlet (2006). *Parallel robots*, vol. 128. Springer.

J.-P. Merlet (2007). 'ALIAS-C++'. Available on line at: http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html.

J.-P. Merlet (2008). 'Kinematics of the wire-driven parallel robot MARIONET using linear actuators'. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3857–3862.

J.-P. Merlet (2009). 'Interval analysis for certified numerical solution of problems in robotics'. *International Journal of Applied Mathematics and Computer Science* **19**(3):399–412.

J.-P. Merlet (2010). 'MARIONET, A Family of Modular Wire-Driven Parallel Robots'. In J. Lenarčič & M. M. Stanisic (eds.), *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 53–61. Springer Netherlands.

J.-P. Merlet (2012a). 'The kinematics of the redundant N–1 wire driven parallel robot'. In *(ICRA), 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14-18 May 2012*, pp. 2313–2318.

J.-P. Merlet (2012b). 'The kinematics of the redundant n-1 wire driven parallel robot'. In *IEEE International Conference on Robotics and Automation (ICRA), 2012*, pp. 2313–2318.

J.-P. Merlet & J. Alexandre-dit Sandretto (2015). 'The Forward Kinematics of Cable-Driven Parallel Robots with Sagging Cables'. In *Cable-Driven Parallel Robots*, pp. 3–15. Springer.

J.-P. Merlet & D. Daney (2010). 'A portable, modular parallel wire crane for rescue operations'. In *(ICRA), 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3-8 May 2010.*, pp. 2834–2839.

A. Ming & T. Higuchi (1994). 'Study on multiple degree-of-freedom positioning mechanism using wires. II: Development of a planar completely restrained positioning mechanism'. *International Journal of the Japan Society for Precision Engineering* **28**(3):235–242.

R. E. Moore (1966). *Interval analysis*. Prentice-Hall Englewood Cliffs.

R. E. Moore (1979). *Methods and applications of interval analysis*. SIAM.

J. Murayama, et al. (2004). 'SPIDAR G&G: A two-handed haptic interface for bimanual VR interaction'. In *Proceedings of EuroHaptics*, pp. 138–146. Citeseer.

R. Nan (2006). 'Five hundred meter aperture spherical radio telescope (FAST)'. *Science in China Series G* **49**(2):129–148.

A. Neumaier (1990). *Interval methods for systems of equations*, vol. 37. Cambridge university press.

A. Neumaier (2001). *Introduction to numerical analysis*. Cambridge University Press.

NIST (2006). *RoboCrane for Aircraft Maintenance*.

C. Paul, et al. (2006). 'Design and control of tensegrity robots for locomotion'. *IEEE Transactions on Robotics* **22**(5):944–957.

R. Penrose (1955). 'A generalized inverse for matrices'. *Mathematical Proceedings of the Cambridge Philosophical Society* **51**(3):406–413.

S. Perreault, et al. (2010). 'Geometric determination of the interference-free constant-orientation workspace of parallel cable-driven mechanisms'. *Journal of Mechanisms and Robotics* **2**(3):031016.

A. Pott (2010). 'An Algorithm for Real-Time Forward Kinematics of Cable-Driven Parallel Robots'. In J. Lenarcic & M. M. Stanisic (eds.), *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 529–538. Springer Netherlands.

A. Pott, et al. (2010). 'Large-scale assembly of solar power plants with parallel cable robots'. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–6.

A. Pott, et al. (2013). 'IPAnema: A family of Cable-Driven Parallel Robots for Industrial Applications'. In T. Bruckmann & A. Pott (eds.), *Cable-Driven Parallel Robots*, vol. 12 of *Mechanisms and Machine Science*, pp. 119–134. Springer Berlin Heidelberg.

J. Pusey, et al. (2004). 'Design and workspace analysis of a 6–6 cable-suspended parallel robot'. *Mechanism and machine theory* **39**(7):761–778.

F. Quadvlieg, et al. (2011). 'Using dynamic positioning for side-by-side cargo transfer operations'. In *2nd International conference on ship manoeuvring in shallow and confined water: ship to ship interaction, Trontheim*.

H. Ratschek & J. Rokne (1995). 'Interval methods'. In *Handbook of global optimization*, pp. 751–828. Springer.

N. Riehl, et al. (2009). 'Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms'. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2193–2198.

J. Rohn (1996). 'Enclosing solutions of overdetermined systems of linear interval equations'. *Reliable Computing* **2**(2):167–171.

G. Rosati, et al. (2007). 'Design, implementation and clinical tests of a wire-based robot for neurorehabilitation'. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **15**(4):560–569.

A. Ruiz, et al. (2015). 'ARACHNIS: Analysis of Robots Actuated by Cables with Handy and Neat Interface Software'. In A. Pott & T. Bruckmann (eds.), *Cable-Driven Parallel Robots*, vol. 32 of *Mechanisms and Machine Science*, pp. 293–305. Springer International Publishing.

V. Schmidt, et al. (2015). 'Investigating the Effect of Cable Force on Winch Winding Accuracy for Cable-Driven Parallel Robots'. In P. Flores & F. Viadero (eds.), *New Trends in Mechanism and Machine Science*, vol. 24 of *Mechanisms and Machine Science*, pp. 315–323. Springer International Publishing.

SkyCam (2015). *SkyCam*.

D. Surdilovic, et al. (2007). 'STRING-MAN: Wire-robot technology for safe, flexible and human-friendly gait rehabilitation'. In *ICORR 2007. IEEE 10th International Conference on Rehabilitation Robotics, Noordwijk, Netherlands, 13-15 June 2007*, pp. 446–453.

S. Tadokoro, et al. (2002). 'A motion base with 6-DOF by parallel cable drive architecture'. *IEEE/ASME Transactions on Mechatronics* **7**(2):115–123.

R. A. Tapia (1971). 'The Kantorovich theorem for Newton's method'. *American Mathematical Monthly* pp. 389–392.

B. Wilcox (2012). 'ATHLETE: A limbed vehicle for solar system exploration'. In *Aerospace Conference, 2012 IEEE*, pp. 1–9.

R. L. Williams II (1998). 'Cable-suspended haptic interface'. *International Journal of Virtual Reality* **3**(3):13–21.

W. Wu & S. Rao (2007). 'Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis'. *Reliability Engineering & System Safety* **92**(1):54–64.

M. Yamamoto, et al. (2004). 'Trajectory control of incompletely restrained parallel-wire-suspended mechanism based on inverse dynamics'. *IEEE Transactions on Robotics* **20**(5):840–850.

W. Yang Ho, et al. (2015). 'Haptic Interaction with a Cable-Driven Parallel Robot Using Admittance Control'. In A. Pott & T. Bruckmann (eds.), *Cable-Driven Parallel Robots*, vol. 32 of *Mechanisms and Machine Science*, pp. 201–212. Springer International Publishing.

Acknowledgments

First of all, I would like to thank my supervisors Marco Carricato and Jean-Pierre Merlet, for their support and guidance throughout all these years. Their advice on both research as well as on my career have been priceless.

I would also like to express my deepest gratitude to Tobias Bruckmann, Andreas Pott and Stéphane Caro, who accepted to be the referees and reviewed my manuscript, giving me precious comments and suggestions.

I would also like to thank professor Vincenzo Parenti-Castelli that gave me the pleasure to be president of the jury, and dr. David Daney for being member of the examining committee.

A special thank to my friends and colleagues Claudio, Margherita, Fabrizio, Federica, Irene, Michele, Nicola, Antonio, Daniele, Glauco, Julien, Laurent, Rémy, Karim, Ting.

Finally, words cannot express how grateful I am to my girlfriend Giulia, to my family and to all my friends who always encouraged me and provided unending support during these years.