SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO IN INGEGNERIA ELETTRONICA ED INFORMATICA
XXVIII CICLO

Thesis

Robotic Technologies for Aeronautics Industry

S.S.D. ING-INF/04

Author
Ing. Pasquale Cirillo
Matr. D124/018

Coordinator
Prof. Giovanni Leone

Advisor
Ch.mo Prof. Ing. Ciro Natale

Academic Year 2014-2015

Thesis supervisor:

Prof. Ing. Ciro Natale
Dipartimento di Ingegneria Industriale e dell'Informazione
Seconda Università degli Studi di Napoli

This work has been carried out by Pasquale Cirillo during his PhD. course under the supervision of Prof. Ing. Ciro Natale at the Dipartimento di Ingegneria Industriale dell'Informazione, Seconda Università degli Studi di Napoli, Aversa, Italy. His PhD. has been supported within the European Projects LOCOMACHS and partially supported within the Italian project STEP FAR.

Robotic Technologies for Aeronautics Industry by Pasquale Cirillo

A Valentina.
Alle persone che mi sono accanto
e che mi hanno aiutato a crescere.

A me...

La pura ricerca è quando faccio
quello che non so fare.

Wernher von Braun

Sommario

Oggigiorno, l'industria aeronautica, dato l'aumento dell'uso dei materiali compositi e ibridi nel processo di assemblaggio, sta andando in contro ad un importante aumento delle operazioni a valore non aggiunto nei processi produttivi. Nella progettazione e nella lavorazione, ogni interfaccia tra una parte e un'altra deve essere pensata e realizzata con gaps e steps per far fronte alle variazioni geometriche create durante la fabbricazione delle parti. I vantaggi dell'uso di materiali compositi nella progettazione degli aeroplani sono molteplici: importante risparmio di peso e riduzione del consumo di carburante attraverso l'impiego di materiali più leggeri; riduzione del numero di componenti attraverso la progettazione di parti integrate e più grandi grazie alla tecnologia del composito, nonché riduzione dei costi di ciclo (costi di manutenzione e ispezione) grazie al fatto che i materiali compositi sono più resistenti alla corrosione e offrono una tenuta meccanica maggiore. Questi aspetti hanno giocato un ruolo chiave nella definizione del mercato aeronautico. Infatti, le previsioni sull'andamento del mercato richiederanno ai fornitori tassi di consegna più alti per soddisfare le aspettative degli operatori e dei clienti. Fondamentale il raggiungimento di questo obiettivo, sarà la capacità dei fornitori di sviluppare una linea di produzione più veloce e di fornire tempi di produzione ridotti, riducendo i tempi della fase di assemblaggio, per tutte le parti strutturali dell'aeroplano. Tuttavia, le tecnologie e i processi di assemblaggio odierni non sono abbastanza efficienti per soddisfare le richieste dei clienti della prossima generazione. Pertanto, è essenziale che il processo di produzione e di assemblaggio, che comprendono la progettazione e produzione dei componenti, sia reso più veloce ed economico. Tutto questo si traduce nel realizzare un processo "lean" (snello) grazie alla eleminazione e/o riduzione delle operazioni laboriose di assemblaggio introducendo e sviluppando nuove tecnologie, come quelle robotiche, e integrandole al fine di creare un processo economico e rapido per la lavorazione di strutture in composito, metallo e ibride.

Durante l'assemblaggio, l'industria aeronautica utilizza ancora tool convenzionali progettati per una specifica lavorazione, che offrono poca flessibilità. La foratura è una operazione ripetitiva che richiede grandi quantità di risorse e, spesso, richiede operazioni addizionali di smontaggio e pulizia delle parti assemblate. Inoltre, quando avvengono modifiche al disegno dei componenti, sono richieste modifiche ai tool di assemblaggio che si traducono in perdite importanti di tempo, denaro e ritardi della produzione. Per cui, nasce la necessità di automatizzare i sotto-processi, come la fo-

ratura, e di sviluppare una nuova generazione di tool adattabili per il posizionamento automatico delle parti.

In questo lavoro di tesi, è stato proposto un nuovo sistema di foratura robotizzata basato su robot cooperanti come contributo innovativo nel primo scenario illustrato. L'uso della robotica cooperante permette di monitorare il processo di foratura o di incrementare la rigidità locale nell'intorno dell'area lavorata. In particolare, utilizzando un'analisi termografica, è possibile determinare l'effetto della foratura sulle parti assemblate e di modificare i parametri di processo qualora si verifichino fenomeni di delaminazione e rottura dei materiali. Ancora, la cooperazione è stata sfruttata per evitare la creazione di trucioli tra due o più parti forate applicando forze di clamping esterne sulle parti durante processi di foratura di stack ibridi. L'integrazione di un sensore di forza/coppia, infine, è stato sfruttato per controllare la forza nella direzione di foratura, riducendo al minimo le forze tangenziali evitando il fenomeno di skating. Il problema del riposizionamento delle parti nelle maschere di assembraggio (fixture) è stato affrontato utilizzando un dispositivo di azionamento elettrico flessibile, ossia una piattaforma di Stewart a basso costo progettata per la specifica applicazione. A questo scopo, è stato sviluppato un tool per supportare la progettazione di una piattaforma di Stewart ad-hoc e facilitare la scelta dei componenti, quali, attuatori lineari e plate. All'interno del tool proposto, sono stati utilizzati metodi di ottimizzazione basati sul modello dinamico della piattaforma con lo scopo di massimizzare il payload e migliorare la reiezione delle forze esterne applicate al robot durante le fasi di lavorazione, senza che sia penalizzato il workspace del robot.

Abstract

Today's aeronautics industry has seen an important increase in non-added value operations, with the increased use of composite and hybrid materials in the assembly. In the design and parts manufacturing phase, each interface between the parts must be designed with gaps and steps to cope with the geometrical variations created during the parts manufacture. The benefits of using composites in aircraft design are numerous: important weight saving and fuel reduction through the use of lighter materials; reduction of individual parts through the design of more integrated and larger singular parts provided by composite technology, as well as reduced life cycle costs (such as maintenance and inspection costs) since composite materials are more corrosion and fatigue resistant. These aspects have played a key role in defining the aircraft market. In fact, future aircraft market forecasts will require higher aircraft delivery rates to meet air operators' expectations. One key aspect to achieve this objective will be the supply chain's ability to perform faster production line ramp-ups and provide reduced production lead times, including assembly, for all structural parts of the airframe. Nevertheless, the assembly technologies and processes used today are not efficient enough to fulfil the demands from the customers of the next generation of composite-based aircraft. Therefore, it is essential that the overall assembly production process, including design and part manufacturing, will be rendered more time and cost efficient. That is to say "lean" by the removal of time and labor intensive assembly related operations by developing missing emerging technologies, such as robotic technologies, and integrating them with existing ones to create cost efficient part manufacturing and assembly of composite, metal and hybrid airframe structures. During the assembly of parts, industry still uses conventional tailor-made tooling which allows for little flexibility. Drilling is a repetitive operation that consumes great amounts of resources and, often, requires additional dismantling and cleaning operations on the assembly parts. Moreover, when the design of the assembly components changes, modifications to the assembly tooling is time consuming and expensive and causes major disturbances and delays in production. So, the automation of sub-assembly processes, such as drilling, and the development of a new generation of adaptive and automated tooling for the part positioning are the first step to the next generation of aeronautic manufacturing process.

In this thesis, a new robotized drilling system based on cooperative robots has been proposed as an innovative contribution in the first explained scenario. The use of cooperative robotics solution allows to supervise the drilling operation or to locally increase the stiffness of the assembly components at the drilling point while a robot is performing the drilling operation. In particular, by using a thermographic analysis, it is possible to determine the effect of the drilling on the assembly parts and to change the process parameters to avoid delamination or destruction of the material. Moreover, the cooperation during the drilling has been exploited to avoid the creation of chips between two or more parts by adequately clamping the parts during the drilling of a stack by using a clamping robot. Finally, the integration of a force/torque sensor into the drilling end-effector has allowed to control the force in the drilling direction while minimizing the tangential forces avoiding the skating phenomena. The repositioning of an airframe part in an assembly jig has been addressed by using an electrically driven flexible assembly tooling device, such as an ad-hoc cost-effective Stewart platform. To this aim, a simulation environment to support the design of a Stewart platform-based robot for specific applications and to facilitate the choice of suitable components (e.g., linear actuators, plate sizes) has been developed. Optimization methods based on dynamic models have been adopted to maximize the payload and improve the rejection of external forces exerted on the mobile platform during positioning or manufacturing applications, without penalizing the robot workspace.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Today's aeronautics industry has seen an important increase in non-added value operations, with the introduction and the increased use of composite and hybrid materials in assembly. In the design and parts manufacturing phase, each interface between the parts must be designed with gaps & steps[1] to cope with the geometrical variations created during the parts manufacture. During the assembly of the parts, conventional tailor-made tools are still used, but they often allow for little flexibility. Drilling operations often require additional dismantling and cleaning operations, which can be complicated and time consuming especially for large parts. Inspection operations and shimming operations, especially liquid shimming operations, require often extra-handling tool and long cure times, too, and they represent a bottleneck in the assembly work-flow. Moreover, future aircraft market forecasts will require higher aircraft delivery rates to meet air operators' expectations. The assembly technologies and processes used today are not efficient enough to fulfil the demands from the customers of the next generation of composite based aircraft. So, aeronautics industry should change the actual manufacturing processes, to perform faster production line ramp-ups and to provide reduced production lead times, including assembly, for all structural parts of the airframe. The objective is to achieve a "lean" process, more time and cost efficient, by removing the time and labour intensive assembly operations. This will permit faster production, faster ramp-ups and better reproducibility of operations.

[1]Geometrical deviations in airframe parts which create gaps in the interface between two parts or steps in surface level between two aligning parts.

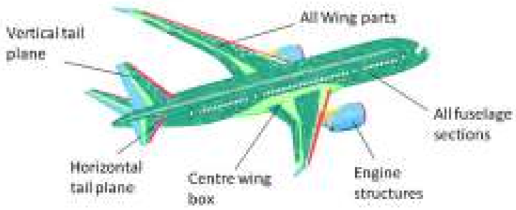


Figure 1.1: Aircraft main parts.

1.1 The Composites in the Aircraft Design

The benefits of using composites in aircraft design are numerous: important weight saving and fuel reduction through the use of lighter materials; reduction of individual parts through the design of more integrated and larger singular parts provided by composite technology, as well as reduced life cycle costs (such as maintenance and inspection costs) since composite materials are more corrosion and fatigue resistant. Currently, the A380 possesses only 25% of composite parts, the Boeing B787 contains 50% and the A350 will contain 52% as larger and more sophisticated assemblies produced using Carbon-Fibre-Reinforced Polymer (CFRP) composite material. Typically, on the A350 and B787, the fuselage and wing structures are primarily made of composite.

All the modern composite-based aircraft are composed of several aircraft structures (wingbox, wings, tail planes, fuselage sections, etc. See Fig 1.1). Each of the main structure is constituted of individual and simpler airframe parts (ribs, spars, stringers, covers). This leads to the existence of hundreds of part interfaces to be managed both in terms of product design as well as design for manufacturing and assembly. Additionally all of the structures and parts require tooling fixtures to ensure their precise positioning within a reference space and jigs to provide the required accuracy and repeatability in the manufacturing of the parts.

In comparison to the relatively easy assembly of machined metal parts, the assembly of composite parts is more challenging. This is due to the anisotropic behaviour of the composite material that causes a shape/geometrical variability of the parts with typically 4% to 7% of thickness dispersion. To compensate for these variations that exist at the joint interfaces between parts, the use of shimming (resin compound or metallic) is integrated into the design of the parts and used during the assembly process. With the increase of the assembly parts in current and future designs, these remaining the major issue in the assembly of composite components lead to important

Figure 1.2: LOCOMACHS logo.

increases in development and manufacturing costs compared to traditional assembly of metal parts.

1.2 The LOCOMACHS and STEPFAR projects

LOCOMACHS (LOw COst Manufacturing and Assembly of Composite and Hybrid Structures) is an European project supported by the European Commission under the 7th Framework Programme. It involves important partners engaged in the European aeronautics industry, i.e. Alenia Aermacchi, Bombardier, Airbus, Saab, Dassault Aviation and research institutes such as MTC, DLR, University of Salerno, Chalmers University, Linköping University, Second University of Naples, etc. The aim of the LOCOMACHS project is to develop missing emerging technologies and integrate them with existing ones to create cost efficient part manufacturing and assembly of composite, metal and hybrid airframe structures focusing on the reduction or the total elimination of the most time consuming and hence expensive non-added value operations (i.e. temporary assembly to check gaps, shimming, dismantling and tool handling). Today, most of the assembly processes (such as shimming, drilling, fastening, etc.) are carried out by skilled operators and, thus, they are especially manual. In order to achieve a lean manufacturing process and ensure the accuracy and the repeatability required in the aeronautics standards, new co-shared manual and automated operations are under investigation. Figure 1.3 shows the comparison in terms of time saving between the manual operations performed today on a typical aircraft structure and a future operations scenario of tomorrow. So, the efforts of the LOCOMACHS project follow the scheme reported in Fig. 1.4 and they are addressed to:

- fully integrate geometrical tolerance and variation management in a representative airframe assembled wingbox structure

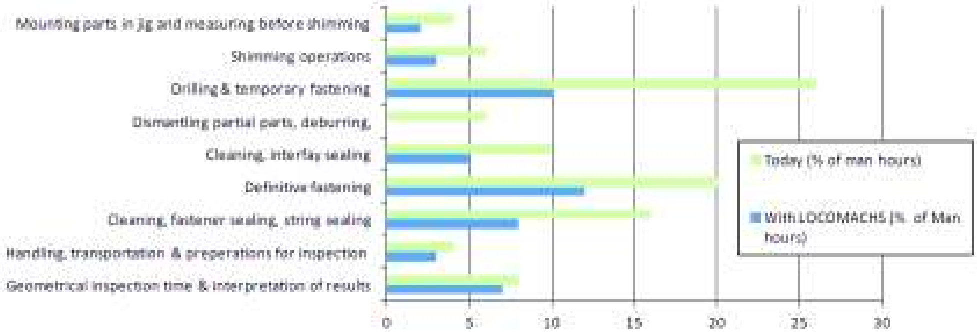


Figure 1.3: Typical time saving through increased use of automation.

- reduce the costs of non-added value shimming operations in structural joints by producing more accurate parts through a better knowledge of the manufacturing process
- reduce the costs of non-added value dismantling operations by developing more cost efficient measurement and verification methodology to avoid temporary assembly operations
- increase the automation of the assembly operations
- reduce the Non Destructive Inspection/Testing (NDI/NDT) times by rendering available novel techniques adapted to the material properties of composite which can be used directly on the production line

The STEP FAR (Sviluppo di materiali e Tecnologie Ecocompatibili, di Processi di Foratura, taglio e di Assemblaggio Robotizzato) project is supported by the DAC (Distretto Tecnologico Aerospaziale della Campania) and it involves as partners Alenia Aermacchi, Atitech, Telespazio, CIRA, CNR, ENEA, University of Salerno, Second University of Naples, etc. The main goal of the project is the study of the issues related to the coupling of hybrid composite-metal structures. In particular, the innovative processes which will be developed in this project are drilling and cutting through a laser source of aluminum alloys, and drilling via machining, using collaborating anthropomorphic robots, off-the-shelf tool end effectors, of hybrid aluminum/-composite stacks and their assembly. Moreover, new monitoring process that uses thermographic data will be investigated to analyze and modify the drilling process in case of delamination and/or deformation of composite parts and to analyze the machining tool wear.

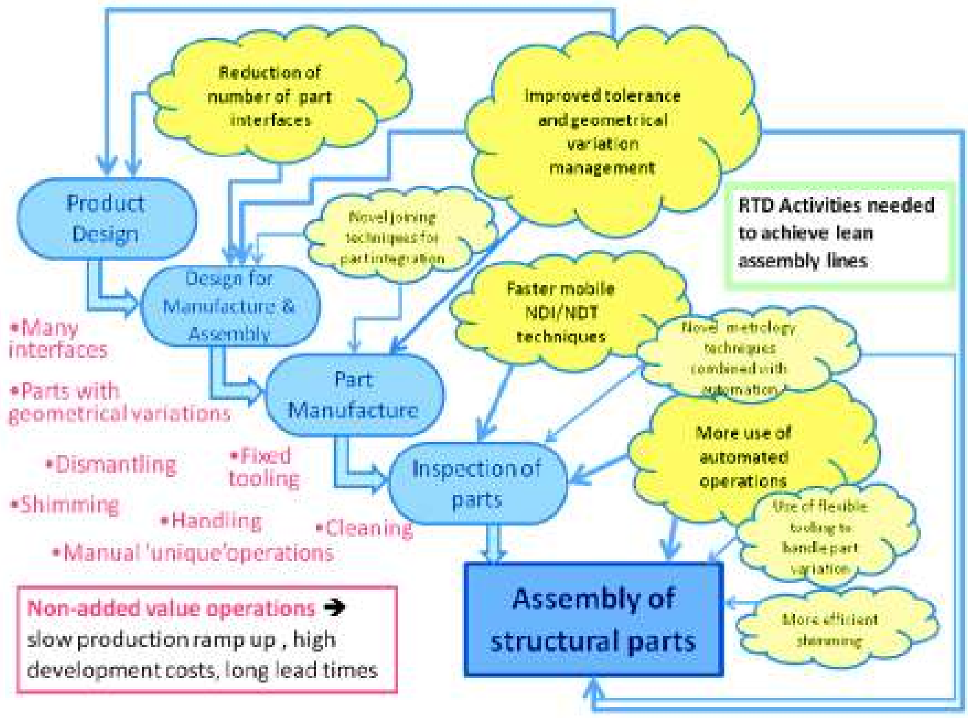


Figure 1.4: Research & Technology Development (RTD) activities to reduce and eliminate non-added value operations.

1.3 Objectives

The objectives of the work of this thesis concerned the development of new methodologies and the integration of existing ones addressed to achieve a "lean" process, more time and cost efficient, for the assembly and sub-assembly operations in the aeronautics industry. The activities have been focused on three main topics: the robotized drilling, the automatic part positioning and the thermographic monitoring of the drilling process of composite and/or metal parts.

Distretto Aerospaziale della Campania
DAC

Figure 1.5: DAC logo.

In the first macro-topic, the efforts regarded the development of a 14 degrees of freedom cooperative dual arm robotic cell. The use of cooperative robotics solution, coupled with the use of a force sensor, is useful for both the analyzed drilling methods: the drilling with jigs and without jigs. In the drilling with jigs the force control allowed the robot to safely enter the end-effector concentric collet in the holes of the jig. The tolerance of such coupling is much less than the positioning accuracy of the industrial robots. Moreover, by imposing the desired moments to zero, the control algorithm will also tolerate errors of alignment between the axis of the concentric collet and the drilling axis allowing the concentric collet to slip inside the holes. Instead, in the drilling without jigs, the force control allowed to reduce, or eliminate, the tangential component of the force to the surface of the panel that arise when the robot is in interaction with the environment in order to avoid the occurrence of the of skating phenomena and control the force along the drilling direction. Moreover, by setting the moments to zero the algorithm ensured that the drilling axis is perpendicular to the panel during the entire drilling operation. Furthermore, the proposed solution allowed to use additional robots to supervise the drilling operation (thermographic monitoring or vision-based positioning), to locally increase the stiffness of the system at the drilling point while a robot is performing the drilling operation allowing to avoid the creation of chips between two or more parts by adequately clamping the parts during the drilling of a stack by using a clamping robot. To this aim, a force/moment control scheme has been designed and implemented. In order to properly design the force controllers, the dynamic model of the robots has been identified.

In the automatic part positioning topic, the activities focused on the development of a new methodology to support the design of parallel robots to be used in the positioning of the aeronautic parts, such as ribs and spars, into a flexible fixture. A simulation environment and an optimization tool to support the design of ad-hoc Stewart platforms have been developed. In particular, a dynamic optimization has been carried out in order to maximize the payload and reduce the actuator forces needed to reject external forces exerted on the mobile platform during positioning or manufacturing applications, e.g., drilling process. Moreover, in order to avoid reduction of the robot workspace, also a kinematic optimality criterion has been considered in the optimization process as well. So, an optimization algorithm has been used to combine the two different optimum objectives by properly defining a cost function to minimize. In order to exploit the anisotropic property of the parallel robot to better optimize the mechanical design given a specific task, the Stewart platform optimization process has been carried out considering both symmetric and unsymmetric geometries. Finally, in order to select the most suitable optimization algorithm for the proposed application, different algorithms have been compared. The performances of the Genetic Algorithm, the Sequential Quadratic Programming algorithm, the Multi-Start algorithm and the GlobalSearch algorithm available in the Matlab Optimization Toolbox, have been analyzed and compared.

The third macro-argument concerned the thermographic monitoring. A multi-task multi-priority control approach can be exploited for the real-time monitoring in the robotized drilling process. The idea is to utilize thermography information provided by a thermal imaging camera to monitor the tool wear or to detect possible damages caused to materials due to the high temperature produced during the machining phase. In this case, the drilling parameters, i.e., speed of rotation of the spindle, feed rate, can be modified in real-time during the drilling process to avoid the delamination and the deformation, especially in the carbon fiber machining. To this aim, an image-based visual servoing algorithm has been developed to keep the features in the FOV of the camera and to keep the optical axis perpendicular to the panel while the drilling robot performs the drilling task. Furthermore, in order to avoid collisions between the monitoring robot and the drilling robot, an obstacle avoidance algorithm has been implemented. Finally, the camera calibration procedure problem of a thermal imaging camera has been addressed.

1.4 Thesis Structure

In this section, the structure of the presented work is illustrated. For each chapter, a brief description of the discussed topics is reported.

- In *Chapter 2* the robotized drilling problem is introduced. In particular, a brief introduction to the basic concept, the today's drilling methodologies used in the aeronautics industry and the proposed robotized system based on a cooperative robotic cell are illustrated. The force control design, simulations and experiments of drilling with jigs and without jigs are presented.

- In *Chapter 3* the robotized part positioning problem of the assembly parts in adaptive and flexible jigs is discussed. After an introduction and a complete description on the methodology on the base of the parallel robots, the proposed simulation environment developed to support the design of ad-hoc cost-effective Stewart platforms for specific applications are described. To demonstrate and to validate the effectiveness of the proposed solution, two specific case studies are reported. Finally, the use of the Stewart platforms has been exploited for the cooperative part positioning, too.

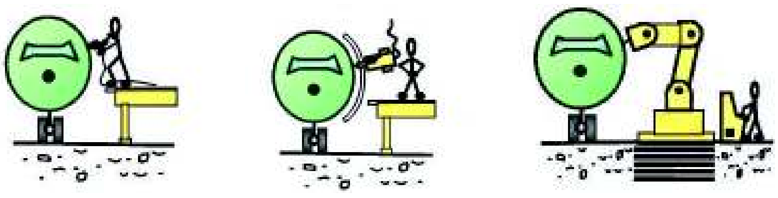
- In *Chapter 4* the image-base visual servoing algorithm developed for the thermographic monitoring of the drilling process is presented. The multi-task multi-priority control approach is introduced and the image processing and camera calibration problems are discussed. Finally, the simulations and the experiments carried out on the Yaskawa SIA5F under ROS and OpenCV environments are presented.

- In *Chapter 5* conclusion and future works are reported.

CHAPTER 2

ROBOTIZED DRILLING

In airframe assemblies, drilling is a repetitive manual operation. Traditionally, the joining of aircraft elements was performed by drilling thousands of holes in each element. In the last years, the research has primarily focused on the design of semi-automatic drilling machines. These machines are secured to heavy and bulky drilling templates, hence relieving the worker from the strenuous tasks of holding in position and pushing the tool. As next step, fully automatic drilling machines, capable of positioning the tool and conducting the whole drilling operation, have been developed but they require often a significantly change of the drilling station, such as fixtures, and the design of ad-hoc tools which result in a very expensive process. Moreover, these robots have to be structurally very rigid in order to achieve sufficient accuracy, which significantly increase the size and weight of a drilling unit. Figure 2.1 shows the three drilling process technologies illustrated above. Two emerging drilling technologies are the “orbital drilling” and the “one shot drilling”. The orbital technology



(a) Manual drilling. (b) Semiautomatic drilling. (c) Drilling by an industrial robot.

Figure 2.1: Drilling process story.

is used for drilling in composite and hybrid stacks. The technology uses a cutter with two rotation axes, one through the centre of the cutter and one slightly offset from the tool centre. This gives a drilling path which can be compared to milling. The technology has great benefit in hole quality but it can not be used in low access areas due to the size of the machine. The one shot drilling, instead, refers to the drilling process of hybrid stacks, where a stack consists of more drilling parts of different materials (composite and metal) and different thickness. In recent years, developments have been achieved on the selection of appropriate tool geometry and process parameters to reduce manufacturing time and to eliminate the difficulties that exist in meeting the tolerances of drilled holes.

2.1 Introduction

The current drilling process is a manual method that makes use of fixtures[2] and jigs[3]. The automation of drilling operations will reduce time and cost of the overall aircraft assembly process, and will increase the competitiveness of the European aerospace industry. Several solutions have been proposed in the literature (see Section 2.2), but all have been focused on the design of a all-in-one drilling end effector. In the LOCOMACHS and STEP FAR projects, instead, the focus has moved to the implementation of a lean system, in which more generic drilling tool, off-the-shelf tool, and low-cost, small-size, and low-accuracy industrial robots are considered.

The proposed solution makes use of two cooperative robot mounted on a sliding track and equipped with two ATI Gamma SI-65-5 force sensors, so, the drilling operation can be performed on a 14 degrees of freedom (DOFs) dual arm robotic cell. The use of cooperative robotics solutions allows to use additional robots to supervise the drilling operation, to locally increase the stiffness of the system at the drilling point while a robot is performing the drilling operation. Moreover, the cooperation during the drilling allows to avoid the creation of chips between two or more parts by adequately clamping the parts during the drilling of a stack by using a clamping robot. The quality of the drilling process is expected to be ensured not only by the use of multiple robots but also by the integration of two different technologies: external

[2]A fixture is a work-holding or support device used in the manufacturing industry. Fixtures are used to securely locate (position in a specific location or orientation) and support the work, ensuring that all parts produced using the fixture will maintain conformity and interchangeability.

[3]A jig is a type of custom-made tool used to control the location and/or motion of another tool. A jig's primary purpose is to provide repeatability, accuracy, and interchangeability in the manufacturing of products.

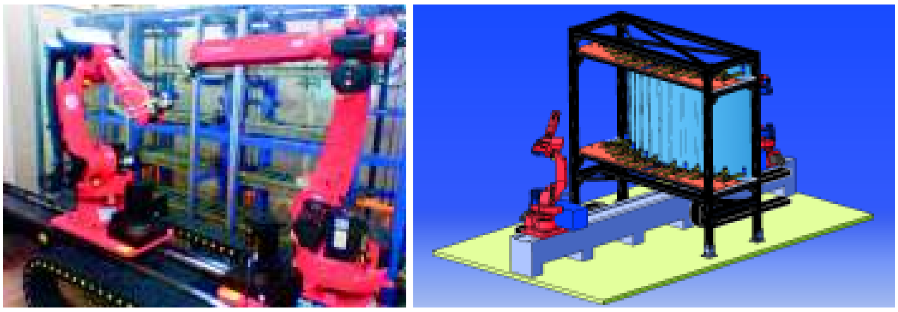
metrology system and force control. For example, by using a thermographic analysis, it is possible to determine the effect of the drilling on the assembly parts and to change the process parameters to avoid delamination or destruction of the material. Proper control strategies based on an external metrology systems could allow the automation of the drilling process by increasing the positioning accuracy of the robot, especially in the drilling process that makes no use of jigs for the end-effector positioning. The integration of a force/torque sensor into the drilling end effector allows to control the force in the drilling direction while minimizing the tangential forces. In particular, the force control capabilities can be exploited in the drilling process by making use of jigs or not. In the drilling with jigs the force control allows the robot to safely enter the end-effector concentric collet[4] in the holes of the jig. The tolerance of such coupling is much less than the accuracy of positioning of the industrial robot. Moreover, by imposing the desired moments to zero, the control algorithm will also tolerates errors of alignment between the axis of the concentric collet and the drilling axis allowing the concentric collet to slip inside the holes. Instead, in the drilling without jigs, the force control allows to reduce, or eliminate, the tangential component of the force to the surface of the panel that arise when the robot is in interaction with the environment in order to avoid the occurrence of the of skating phenomena and control the force along the drilling direction (normal to the panel). Moreover, by setting the moments to zero the algorithm ensures that the drilling axis is perpendicular to the panel during the entire drilling operation.

The considered robotic cell is installed at the robotic laboratory of University of Salerno (UNISA) as shown in Fig. 2.2 (a). In particular, the robots are two COMAU SmartSix (6 dofs anthropomorphic robots). In order to hold the part to drill, a reconfigurable fixture[5] designed by Alenia Aermacchi has been installed at the UNISA robotic laboratory. Figure 2.2 (b) shows the complete cell designed in CATIA environment (CAD provided by Alenia Aermacchi). In Fig. 2.3 the details of the fixture are shown; in the proposed configuration, the fixture allows to keep panel of curvature radius equal to 1650 mm, dimensions 2200 mm $\times$1600 mm, resulting in a working area of 2180 mm $\times$1200 mm.

To support the simulation carried out in Matlab/Simulink environment, the robotic

[4]The concentric collet is a tool installed on a drilling end effector. It offers an ergonomic and quick way of clamping the drill into the jig holes during the drilling process.

[5]The design of reconfigurable tooling is an issue of LOCOMACHS Tg34-5 target. The innovation lies in the ability to change the configuration of an airframe assembly tool in order to assembly different products within a product family. Reconfigurable tooling should reduce the number of tools on the workshop floor and thereby save floor space, it should simplify the build-up and change of assembly tools and it should drastically reduce lead time in tooling design and build-up.



(a) Current setting. (b) Alenia fixture design.

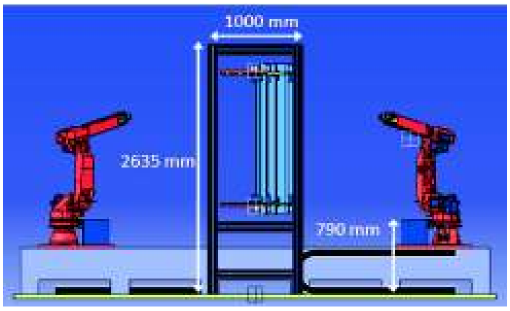
Figure 2.2: UNISA robotic cell.

cell has been modeled and simulated in V-REP[6]. A V-REP view that shows the complete robotic setup is reported in Fig. 2.4.

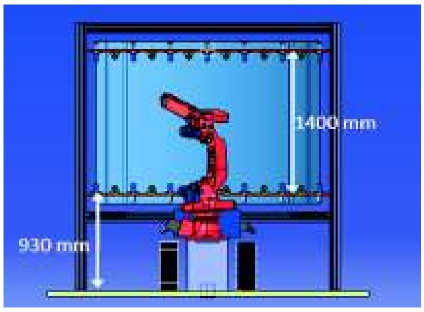
The proposed drilling concept consists of three phases: approaching phase, optimization phase and drilling phase. In the approaching phase, defined the drilling point in the Cartesian space, the "drilling" robot reaches the point of the panel to drill and the other robot reaches the same point from the opposite part of the panel in order to locally increase the stiffness. In case of one shot drilling process, and, so, of drilling of stacks, the "clamping" robot exerts a proper force on the output side of the tool to avoid the creation of chips between the different parts. Once the robot approached the panel, in order to maximize the robot drilling capabilities along the normal direction to the panel an optimization phase is carried out before the drilling process starts and the force manipulability ellipsoid is analyzed and maximized along the drilling direction (see Section 2.6 and Section 2.8.5). Once reconfigured the robots, the drilling phase starts and, in order to minimize the tangential forces with respect to the panel and in order to properly control the force along the drilling direction, a force feedback control is used. Moreover, using a moment control algorithm the drilling tool is kept perpendicular to the panel during the drilling operation.

A brief description of the state of the art of the robotized drilling systems and an introduction on the mostly used robot drilling configurations are reported below.

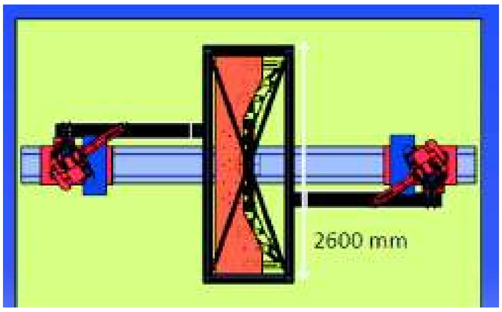
[6]V-REP is an open source robotic simulator developed by Coppelia Robotics. V-REP, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. This makes V-REP very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab, Octave or Urbi.



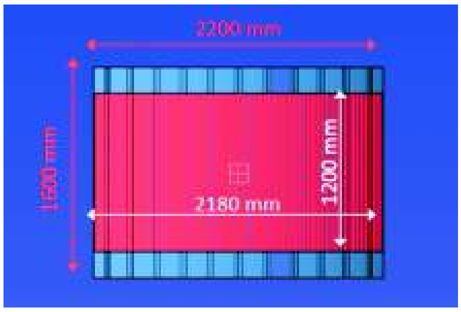
(a) Front view.



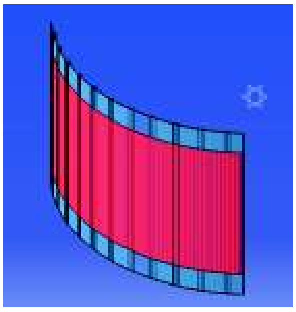
(b) Right view.



(c) Top view.

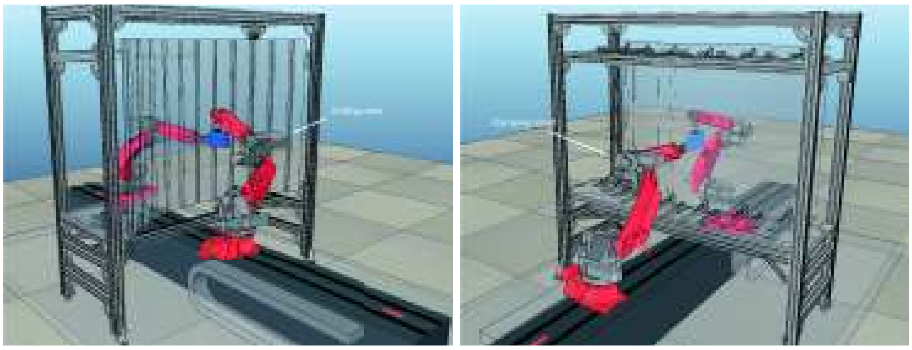


(d) Panel dimensions.



(e) Panel - perspective view.

Figure 2.3: Reconfigurable fixture designed by Alenia Aermacchi.



(a) (b)

Figure 2.4: UNISA robotic cell: V-REP implementation.

2.2 State of the Art

Drilling is an important process of assembling aeroplane components, whose efficiency and quality have an impact on aeroplane assembly cycle and quality. In recent years, industrial robots are increasingly used in the aeroplane assembly process but only as supporting/mobile platforms for coarse positioning of complex drilling systems. In fact, the research activities and the industrial efforts were focused on the development of all-in-one robotic drilling systems. For example, the Electroimpact developed a robotized drilling end effector for Airbus UK Ltd. for use on a Kuka KR350. The end effector is referred to as the DDEE (Drill and Drive End Effector), and incorporates four main functions: push-up of components, drilling with panel detection, hole inspection, bolt insertion [1]. Furthermore, the Electroimpact proposed ONCE (ONe-sided Cell End effector, Fig. 2.5(a)), a more complete system to drill, countersink, and measure fastener holes in the wing trailing edge flaps on the Boeing F/A-18E/F Super Hornet [2]. Hawker de Havilland et al. used the 737 aileron robot cell to drill and countersink as well as trim the trailing edge and tooling lugs from the carbon-fiber-reinforced plastic skins [3]. The second generation of Electroimpact ONCE robotic drilling system [4] (Fig. 2.5(b)), successfully deployed in production, strove toward "one-up" assembly, whereby the product was assembled one time-drilled, countersink, inspected, and ultimately fastened-without removal of components for deburring, cleaning, sealing, etc. [5]. A robotic drilling system, which used orbital hole-drilling technology, was developed by Novator AB in collaboration with Boeing, to overcome the obstacles of drilling holes in a combination of both hard metals and composites [6]. Another robotic drilling system for titanium

(a) First generation. (b) Second generation.

Figure 2.5: Electroimpact ONCE tool.

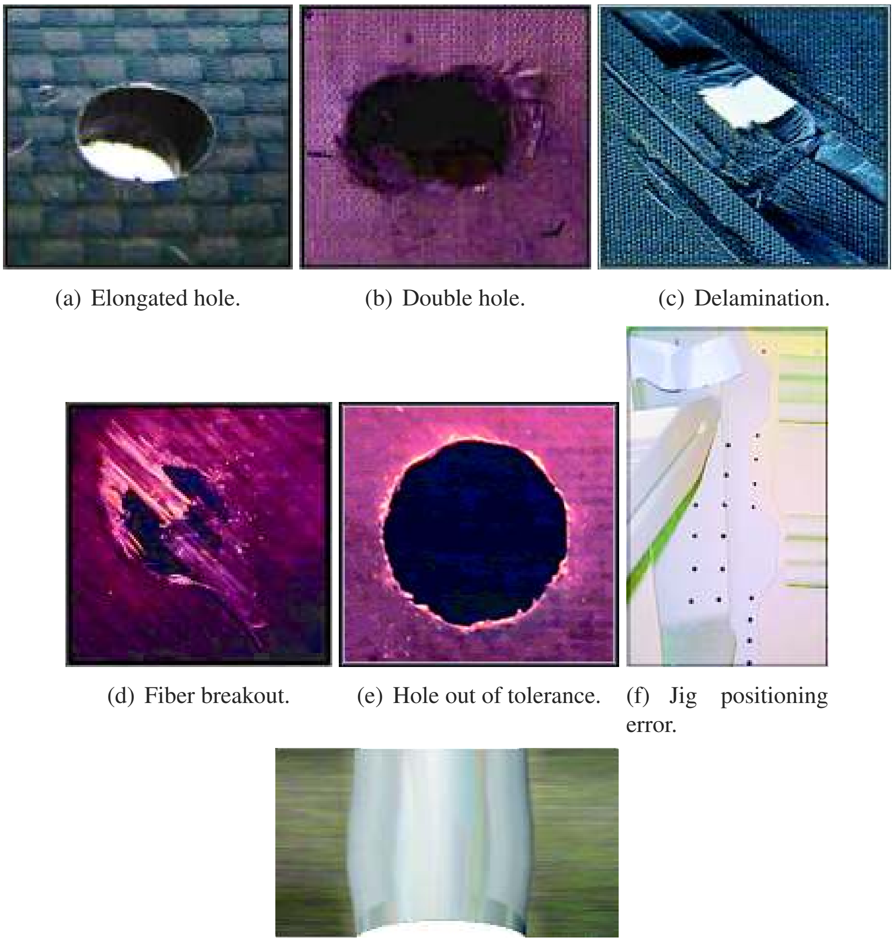
structures was presented by Whinnem et al. [7]. The system functions include locating workpiece with a calibration stick or the vision system, weld mark inspection, one-sided clamping, drilling and reaming hole in material stack combinations of titanium and aluminum, and real-time thrust force feedback. T. Olsson et al. from the Lund University proposed a method for high-precision drilling using an industrial robot with high-bandwidth force feedback, which is used to eliminate the sliding movement (skating) of the end effector during the clamp-up of the end-effector to the work-piece surface [8][9].

Although all of the aforementioned systems are valid solutions, it is very necessary for the aviation industry of different countries to develop their own robotic drilling systems because of intellectual property and various products to assembly.

2.3 Drilling Specification and Problems in Aeronautics

The realization of a hole is one of the most delicate process in the production of an aircraft. If the characteristics of the holes do not fit in the specifications of acceptability, and the part can not be recovered, it can be discarded, resulting in an increase of the costs in the production phase. The characteristics of an acceptable hole are: hole diameter, drilling angle or hole angularity, surface finishing, edge finishing and depth of the flushness. Considering the CFRP parts, the today's requirements dictate that there are no visible scratches or defects around the hole, that there is no tearing of the fiber in entrance or in exit of the hole and that:

- the hole diameter has a tolerance of about ± 0.001 inch on a diameter of 0.5 inch,
- the hole angularity has a tolerance of 2° from the normal to the panel,



(a) Elongated hole. (b) Double hole. (c) Delamination.

(d) Fiber breakout. (e) Hole out of tolerance. (f) Jig positioning error.

(g) Barrel effect.

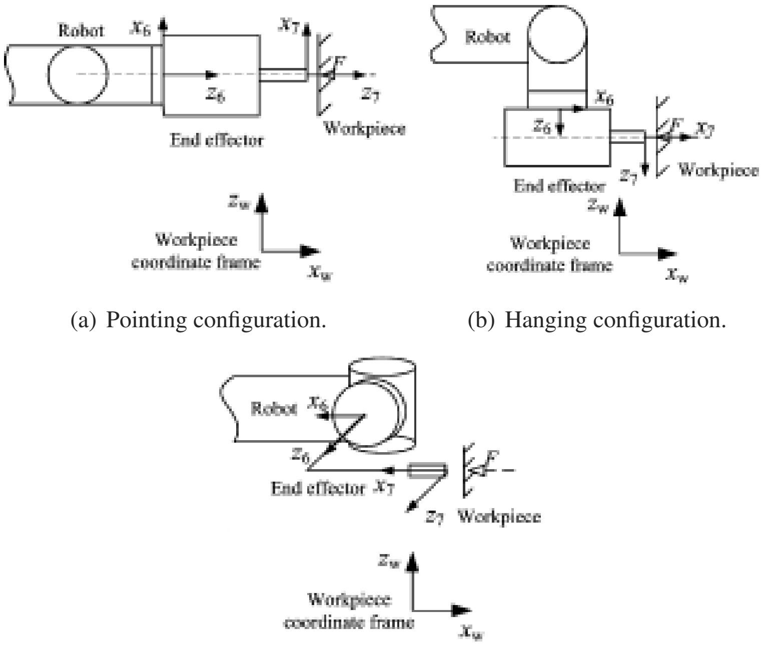
Figure 2.6: Hole defects.

- the center of the hole is from the edge of the panel at least three times the diameter of the hole.

Moreover, during the drilling process, some hole defects related to tool wear, incorrect use of the tool, incorrect process parameters (e.g., great force applied during the drilling) could occur. The most relevant of these defects are reported below and illustrated in Fig. 2.6:

- elongated hole due to the not use of the drill stop or play of the bit in the chuck of the drill when the drill is removed from the hole

- double hole caused by the incorrect positioning of the tool that flows on the



(a) Pointing configuration.
(b) Hanging configuration.
(c) Side configuration.

Figure 2.7: Mounting configuration of the end effector.

part surface

- delamination due to the tool usury or excessive force applied on the exit when no drill stop is used
- fiber breakout due to excessive pressure/thrust during the drilling or tool usury
- hole out of tolerance due to the tool usury
- jig positioning error cause a shift of the holes on the panel which may not be coupled to the other parts
- “barrel” effect caused by the excessive use of lubricant, that retains chips attached to the tip

2.4 Drilling Configurations

There are three different ways to attach the end effector to the robot, which are pointing configuration, hanging configuration, and side configuration [7]. The hanging configuration and the side configuration both improve the manipulability and the motion dexterity of a robot, but with the two configurations, joint 5 of the robot suffers from a high torque, which can cause the joint to deflect and loose the position accuracy and the perpendicular direction to the panel surface. In the pointing configuration, the clamp force goes straight into the robot without creating the unwanted torque. Figure 2.7 shows the three mentioned configurations.

In the proposed simulations, both pointing and hanging configurations have been analyzed.

2.5 CLIK Algorithm

To introduce the Inverse Differential Kinematics [10] for the redundant manipulator, the differential kinematics equation has to be considered. The differential kinematics equation can be formally written as in 2.1 where $\boldsymbol{v}_e$ is meant to be the $(r \times 1)$ vector of end-effector velocity of concern for the specific task and J is the corresponding $(r \times n)$ Jacobian matrix that can be extracted from the geometric Jacobian[7]; $\dot{\boldsymbol{q}}$ is the $(n \times 1)$ vector of joint velocities. If $r < n$, the manipulator is kinematically redundant and there exist $(n - r)$ redundant DOFs.

$$\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.1}$$

The existence of a subspace $N(\boldsymbol{J}) \neq \emptyset$ for a redundant manipulator allows determination of systematic techniques for exploiting the redundant DOFs. In fact, if $\dot{\boldsymbol{q}}^*$ denotes a solution to eq. (2.1) and $\boldsymbol{P}$ is an $(n \times n)$ matrix so that $R(\boldsymbol{P}) \equiv N(\boldsymbol{J})$, the joint velocity vector in (2.2) is a solution to (2.1), with arbitrary $\dot{\boldsymbol{q}}_0$.

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^* + \boldsymbol{P}\dot{\boldsymbol{q}}_0 \tag{2.2}$$

To demonstrate the previous sentence, pre-multiplying both sides of (2.2) by $\boldsymbol{J}$ yields eq. (2.3), since $\boldsymbol{J}\boldsymbol{P}\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$ for any $\dot{\boldsymbol{q}}_0$.

$$\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{J}\dot{\boldsymbol{q}}^* + \boldsymbol{J}\boldsymbol{P}\dot{\boldsymbol{q}}_0 = \boldsymbol{J}\dot{\boldsymbol{q}}^* = \boldsymbol{v}_e \tag{2.3}$$

Note that the effect of $\dot{\boldsymbol{q}}_0$ is to generate internal motions of the structure that do not

[7] The Jacobian describes the linear mapping from the joint velocity space to the end-effector velocity space. In general, it is a function of the configuration. In the context of differential kinematics, however, the Jacobian has to be regarded as a constant matrix, since the instantaneous velocity mapping is of interest for a given posture.

change the end-effector position and orientation.

So, when the manipulator is redundant ($r < n$), the Jacobian matrix has more columns than rows and infinite solutions exist to eq. (2.1). A solution method is to formulate the problem as a constrained linear optimization problem. Let consider a cost functional as in (2.4). This choice provides to minimize the norm of the vector $\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0$, therefore, the objective specified through $\dot{\boldsymbol{q}}_0$ becomes a secondary objective to satisfy with respect to the primary objective specified by the constraint (2.1).

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) \tag{2.4}$$

The problem can be solved with the method of Lagrange multipliers. Consider the modified cost functional in (2.5), where $\boldsymbol{\lambda}$ is an $(r \times 1)$ vector of unknown multipliers that allows the incorporation of the constraint (2.1) in the functional to minimize.

$$g(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) + \boldsymbol{\lambda}^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}) \tag{2.5}$$

The solution has to satisfy the necessary conditions in (2.6).

$$\left(\frac{\partial g}{\partial \dot{\boldsymbol{q}}}\right)^T = \mathbf{0}, \quad \left(\frac{\partial g}{\partial \boldsymbol{\lambda}}\right)^T = \mathbf{0} \tag{2.6}$$

From the first necessary condition it is as in (2.7); from the second one the constraint (2.1) is recovered.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^T\boldsymbol{\lambda} + \dot{\boldsymbol{q}}_0 \tag{2.7}$$

Substituting the (2.7) into (2.1), yields eq. (2.8). Finally, substituting $\boldsymbol{\lambda}$ back in(2.7) gives the solution $\dot{\boldsymbol{q}}$ reported in eq. (2.9), where $\boldsymbol{J}^\dagger = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$ is the right pseudo-inverse of $\boldsymbol{J}$.

$$\boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{J}^T)^{-1}(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}_0) \tag{2.8}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger\boldsymbol{v}_e + (\boldsymbol{I}_n - \boldsymbol{J}^\dagger\boldsymbol{J})\dot{\boldsymbol{q}}_0 \tag{2.9}$$

Note that the obtained solution is composed of two terms: the first is relative to minimum norm joint velocities; the second one, termed homogeneous solution, attempts to satisfy the additional constraint to specify via $\dot{\boldsymbol{q}}_0$. A possible choice of the matrix $\boldsymbol{P}$ introduced in (2.2) is $(\boldsymbol{I} - \boldsymbol{J}^\dagger\boldsymbol{J})$, which allows the projection of the vector $\dot{\boldsymbol{q}}_0$ in the null space of $\boldsymbol{J}$, so as not to violate the constraint (2.1). A direct consequence is that, in the case $\boldsymbol{v}_e = \mathbf{0}$, is possible to generate internal motions described by $(\boldsymbol{I} - \boldsymbol{J}^\dagger\boldsymbol{J})\dot{\boldsymbol{q}}_0$ that reconfigure the manipulator structure without changing the end-effector position and orientation.

2.5.1 Inverse Kinematics Algorithms

In the numerical implementation of the illustrated algorithm, computation of joint velocities is obtained by using the inverse of the Jacobian evaluated with the joint variables at the previous instant of time. It follows that the computed joint velocities $\dot{\boldsymbol{q}}$ do not coincide with those satisfying the same relation in the continuous time. Therefore, reconstruction of joint variables $\boldsymbol{q}$ is entrusted to a numerical integration which involves drift phenomena of the solution; as a consequence, the end-effector pose corresponding to the computed joint variables differs from the desired one.

To overcame the problem, the operational space error defined in (2.10), where $\boldsymbol{x}_d$ and $\boldsymbol{x}_e$ are the desired and effective end-effector position and orientation, can be considered.

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}_e \tag{2.10}$$

Considering the time derivative of (2.10) as in (2.11) and accordingly to the (2.12), yields the expression reported in (2.13).

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_e \tag{2.11}$$

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_\phi(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.12}$$

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.13}$$

In (2.13) with $\boldsymbol{J}_A$ is indicated the analytical Jacobian defined in (2.14) that is different from the geometrical Jacobian. In particular, the relation between the analytical and geometrical Jacobians are reported in (2.15), where $\boldsymbol{T}_A(\boldsymbol{\phi})$ is defined as in (3.9) and $\boldsymbol{T}(\boldsymbol{\phi}_e)$, that is the transformation $\boldsymbol{T}$ relating the angular velocity $\boldsymbol{\omega}_e$ to the time derivative of the ZYZ Euler angles $\dot{\boldsymbol{\phi}}_e$, is defined as in (2.17).

$$\boldsymbol{J}_A(\boldsymbol{q}) = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{2.14}$$

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.15}$$

$$\boldsymbol{J} = \boldsymbol{T}_A(\boldsymbol{\phi})\boldsymbol{J}_A = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}(\boldsymbol{\phi}_e) \end{bmatrix} \boldsymbol{J}_A \tag{2.16}$$

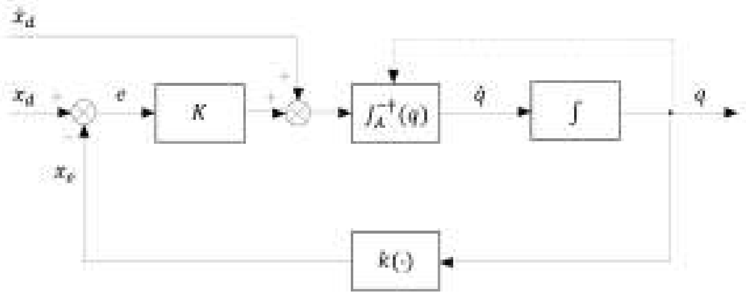


Figure 2.8: Block scheme of CLIK algorithm using the pseudo-inverse of the Jacobian.

$$T(\boldsymbol{\phi}_e) = \begin{bmatrix} 0 & -sin(\varphi) & cos(\varphi)sin(\theta) \\ 0 & cos(\varphi) & sin(\varphi)sin(\theta) \\ 1 & 0 & cos(\theta) \end{bmatrix} \tag{2.17}$$

So, considering the generic solution (2.18) for a redundant manipulator, yields to the equivalent linear system in (2.19).

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e}) \tag{2.18}$$

$$\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = \boldsymbol{0} \tag{2.19}$$

If $\boldsymbol{K}$ is a positive definite (usually diagonal) matrix, the system (2.19) is asymptotically stable. The error tends to zero along the trajectory with a convergence rate that depends on the eigenvalues of matrix $\boldsymbol{K}$. The block scheme corresponding to the inverse kinematics algorithm in (2.18) is shown in Fig. 2.8.

2.6 Manipulability Ellipsoids

In order to define indices for the evaluation of manipulator performances, the manipulability ellipsoids can be used. Such indices, can be helpful both for mechanical manipulator design and for determining suitable manipulator postures to execute a given task in a specific configuration. So, through the velocity manipulability ellipsoid, the attitude of a manipulator to change end-effector position and orientation can be evaluated; through the force manipulability ellipsoid the end-effector forces that can be generated with a given set of joint torques, when the manipulator is in a given posture, can be characterized.

For the definition of such ellipsoids expressions, let consider the set of joint velocities (2.20) of constant unit norm. The equation describes the points on the surface of a sphere in the joint velocity space.

$$\dot{\boldsymbol{q}}^T \dot{\boldsymbol{q}} = 1 \tag{2.20}$$

In the general case of a redundant manipulator ($r < n$) at a nonsingular configuration, the minimum-norm solution (2.21) can be considered which, substituted in (2.20), yields (2.22).

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{v}_e \tag{2.21}$$

$$\boldsymbol{v}_e^T(\boldsymbol{J}^{\dagger^T}(\boldsymbol{q})\boldsymbol{J}^{\dagger}(\boldsymbol{q}))\boldsymbol{v}_e = 1 \tag{2.22}$$

Accounting for the expression of the pseudo-inverse of the Jacobian $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$, gives eq. (2.23).

$$\boldsymbol{v}_e^T(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))^{-1}\boldsymbol{v}_e = 1 \tag{2.23}$$

Meanly, along the direction of the major axis of the ellipsoid, the end effector can move at large velocity, while along the direction of the minor axis small end-effector velocities are obtained. Further, closer the ellipsoid is to a sphere, better the end effector can move isotropically along all directions of the operational space.

The shape and orientation of the ellipsoid are determined by the core $\boldsymbol{J}\boldsymbol{J}^T$, which is in general a function of the manipulator configuration. In particular, the directions of the principal axes of the ellipsoid are determined by the eigenvectors $\boldsymbol{u}_i$, for $i = 1, ..., r$, of the matrix $\boldsymbol{J}\boldsymbol{J}^T$, while the dimensions of the axes are given by the singular values of $\boldsymbol{J}$, $\sigma_i = \sqrt{\lambda_i(\boldsymbol{J}\boldsymbol{J}^T)}$, for $i = 1, ..., r$, where $\lambda_i(\boldsymbol{J}\boldsymbol{J}^T)$ denotes the generic eigenvalue of $\boldsymbol{J}\boldsymbol{J}^T$.

In order to describe the manipulability of the robot with reference to forces, the duality between differential kinematics (2.1) and statics reported in eq. (2.24), where $\boldsymbol{\tau}$ denote the $(n \times 1)$ vector of joint torques and $\boldsymbol{\gamma}_e$ the $(r \times 1)$ vector of end-effector forces, can be considered.

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma}_e \tag{2.24}$$

Thus, considering the sphere in the space of joint torques as in (2.25), accounting for (2.24), the force manipulability ellipsoid can be obtained as in (2.26).

$$\boldsymbol{\tau}^T \boldsymbol{\tau} = 1 \tag{2.25}$$

$$\boldsymbol{\gamma}_e^T(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))\boldsymbol{\gamma}_e = 1 \tag{2.26}$$

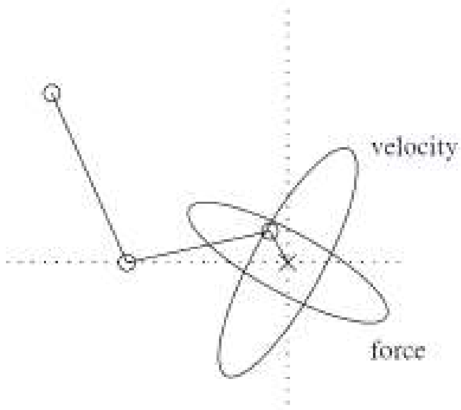


Figure 2.9: Velocity and force manipulability ellipses for a 3-link planar arm.

This ellipsoid characterizes the end-effector forces that can be generated with the given set of joint torques, with the manipulator in a given posture. Note that the core of the quadratic form in (2.26) is constituted by the inverse of the matrix core of the velocity ellipsoid in (2.23). This means that the principal axes of the force manipulability ellipsoid coincide with the principal axes of the velocity manipulability ellipsoid, while the dimensions of the respective axes are in inverse proportion. Then, according to the concept of force/velocity duality, a direction along which good velocity manipulability is obtained is a direction along which poor force manipulability is obtained, and vice versa (see Fig. 2.9).

2.7 Force Control Algorithm

A drilling task requires to control accurately the contact forces generated by the manipulator when it interacts with the environment. Reducing the tangential forces with respect to the drilling object surface allows to avoid the skating problem and, thus, it allows to increase the hole quality. On the other hand, controlling accurately the normal force with respect to the object surface allows to manage the progress of the drilling tool during the drilling operation and to impose the necessary force, that depends on the object material, to drill the considered object. Therefore, in order to directly specify the desired interaction forces, the development of a direct force control system is necessary. A force control scheme is presented below, which is based on the use of an inverse dynamics position control. The effectiveness of a such con-

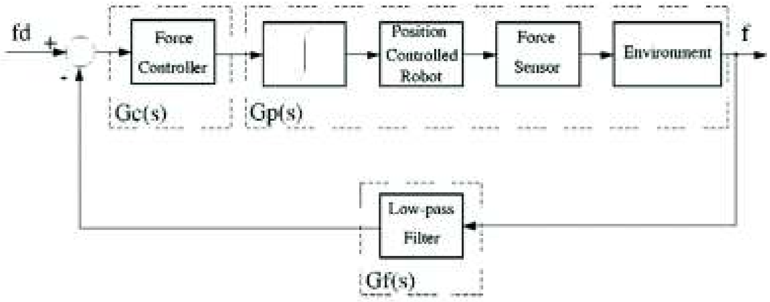


Figure 2.10: Force control reference scheme.

trol scheme depends on the particular interaction cases and, in general, on the contact geometry: a force control strategy is meaningful only for those directions of the operational space along which interaction forces between manipulator and environment may arise. Moreover, the elastic model (2.27) is assumed for the environment, where $\boldsymbol{K}$ is the stiffness matrix of the environment and $\boldsymbol{x}_r$ is the position of the frame attached to the environment. Refer to [11] for a more general explanation of the proposed force control.

$$\boldsymbol{f}_e = \boldsymbol{K}(\boldsymbol{x}_e - \boldsymbol{x}_r) \tag{2.27}$$

2.7.1 Force Control with Inner Position Loop

The approach proposed in [12] has been adopted to design robust force controllers for the considered industrial robots by taking into account the time delay introduced by the communication interface. The cited work proposes a simple and efficient tool for automatic synthesis of a force controller implemented, according to the inner/outer loop technique, on an industrial robot. On the basis of the specifications and by taking into account the system constraints, the first step of the procedure is to design the structure of the compensator acting on the force error; the design is carried out by adopting classical automatic control tools such as stability margins and Nichols charts. The successive step is to compute the parameters of the compensator which guarantees that the specifications are satisfied. One of the key features of the design technique is the handling of the time delay due to the digital implementation of the controller, which is an input deadtime. The designed controller guarantees high gain and phase margins for the closed-loop system, which imply robustness of the control system. The reference control scheme is reported in Fig. 2.10.

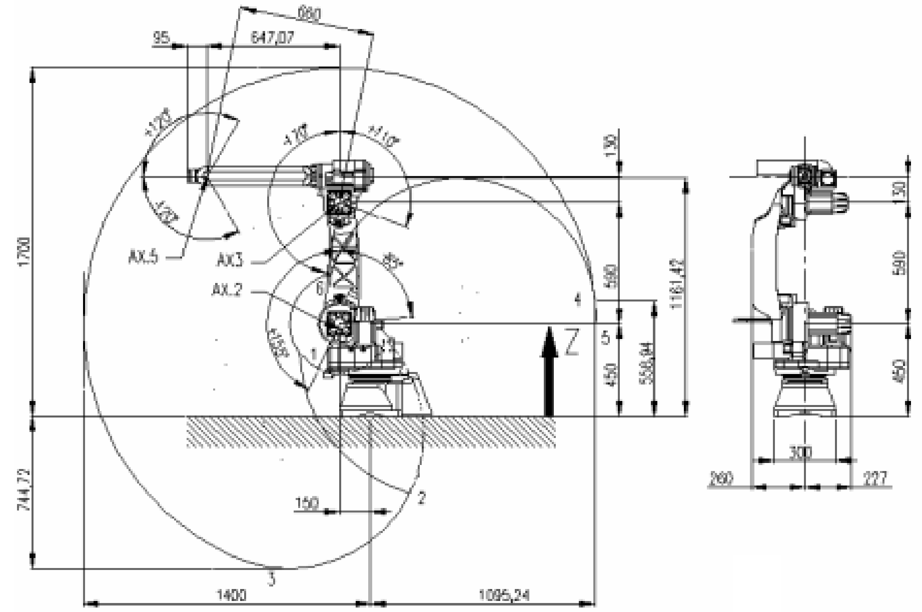


Figure 2.11: Comau SmartSix mechanical scheme.

2.8 Robot Modeling, Control Design and Results

In order to analyze and estimate the robot behavior and performances during the drilling process, force control simulations have been carried out in Matlab/Simulink environment. In the following sections, the force control implementation, the control design and the drilling configuration optimization process are illustrated. Finally, the results of the proposed experiments have been discussed.

2.8.1 Comau SmartSix D-H Table

To develop a positioning algorithm and to compute the direct and inverse kinematics of the 7 dofs robot (1 dof of sliding track plus 6 dofs of the manipulator), a kinematic model of the Comau SmartSix in term of Denavit-Hartenberg (D-H) table, is required. With reference to Fig. 2.11, the D-H parameters are reported in Tab. 2.1. Given the D-H convention reported in Fig. 2.12, in which joint 1 represents a prismatic joint (sliding track), the robot joint values in calibration configuration are:
$q_2^{D-H} = 0$
$q_3^{D-H} = -\pi/2$
$q_4^{D-H} = -0$

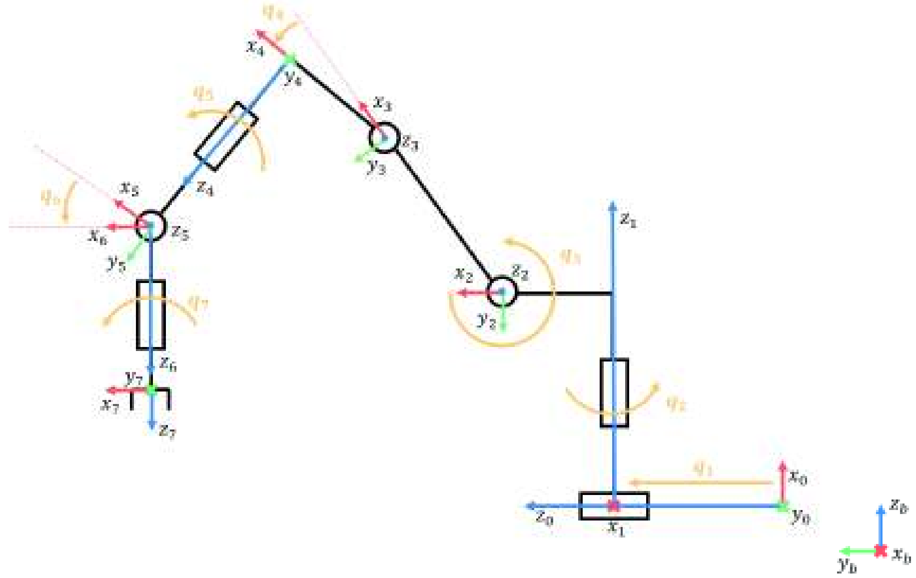


Figure 2.12: Comau SmartSix DH convention scheme.

$q_5^{D-H} = 0$
$q_6^{D-H} = \pi/2$
$q_7^{D-H} = \pi$
While in the considered D-H convention the joint frame are all right-handed, in the Comau convention the frames 2 and 5 are right-handed, but the frames 1, 3, 4 and 6 are left-handed. So the robot joint values in calibration configuration, considering the Comau convention are:
$q_2^{Comau} = 0$
$q_3^{Comau} = 0$

Joint	α_i	$\mathbf{a}_i$ [mm]	$\mathbf{d}_i$ [mm]	$\boldsymbol{\theta}_i$
1	$\pi/2$	0	q_1	$\pi/2$
2	$-\pi/2$	150	450	q_2
3	0	590	0	q_3
4	$-\pi/2$	130	0	q_4
5	$\pi/2$	0	647.07	q_5
6	$-\pi/2$	0	0	q_6
7	0	0	95	q_7

Table 2.1: Comau SmartSix D-H table.

$q_4^{Comau} = -\pi/2$
$q_5^{Comau} = 0$
$q_6^{Comau} = \pi/2$
$q_7^{Comau} = 0$
Therefore, the relations between the Comau convention and the D-H convention are:
$q_2^{D-H} = -q_2^{Comau}$
$q_3^{D-H} = q_3^{Comau} - \pi/2$
$q_4^{D-H} = -q_4^{Comau} - \pi/2$
$q_5^{D-H} = -q_5^{Comau}$
$q_6^{D-H} = q_6^{Comau}$
$q_7^{D-H} = -q_7^{Comau} + \pi$
Similarly, for the joint velocities and accelerations result:
$\dot{q}_2^{D-H} = -\dot{q}_2^{Comau}$
$\dot{q}_3^{D-H} = \dot{q}_3^{Comau}$
$\dot{q}_4^{D-H} = -\dot{q}_4^{Comau}$
$\dot{q}_5^{D-H} = -\dot{q}_5^{Comau}$
$\dot{q}_6^{D-H} = \dot{q}_6^{Comau}$
$\dot{q}_7^{D-H} = -\dot{q}_7^{Comau}$

$\ddot{q}_2^{D-H} = -\ddot{q}_2^{Comau}$
$\ddot{q}_3^{D-H} = \ddot{q}_3^{Comau}$
$\ddot{q}_4^{D-H} = -\ddot{q}_4^{Comau}$
$\ddot{q}_5^{D-H} = -\ddot{q}_5^{Comau}$
$\ddot{q}_6^{D-H} = \ddot{q}_6^{Comau}$
$\ddot{q}_7^{D-H} = -\ddot{q}_7^{Comau}$

In order to be consistent with the real robotic cell and the robotic scene modeled in V-REP simulator, a rotation matrix $\boldsymbol{R}_0^b$ can be considered as below:

$$\boldsymbol{R}_0^b = \boldsymbol{R}_x(-\pi/2)\boldsymbol{R}_z(-\pi/2)$$

2.8.2 COMAU SmartSix modeling

In order to perform force control simulation and ensure an efficient control design, the COMAU SmartSix robot has been identified using Matlab System Identification Toolbox. So, a mathematical model of the robot joints has been obtained from the real dynamic systems utilizing measured time-domain input-output data. In partic-

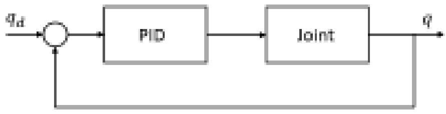
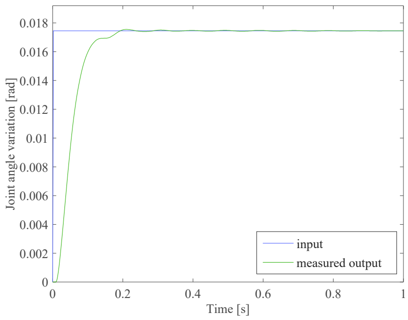


Figure 2.13: Joint Identification Scheme.

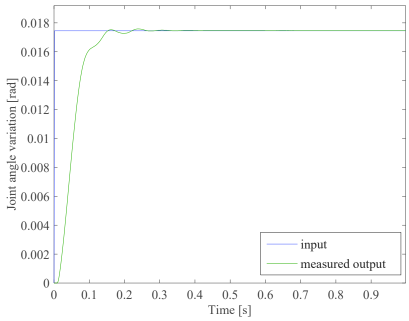
ular, each dynamic joint system has been identified including the low level position controller as shown in Fig. 2.13: the input is the desired joint position $\boldsymbol{q}_d$ and the output is the real joint position $\boldsymbol{q}$; all the data has been acquired with a sample time of 0.002 s. Figure 2.14 shows the input-output data used for the identification process. Figure 2.15 shows the step response of identified systems compared to the measured data. The derived models include sufficient details about the dynamics of the system that should be considered for ensuring an efficient design of the force controllers. Note that the identified models have an input time delay of 0.008 s. The input time delay models the delays due to the communication interface. In fact, the manipulator is connected to a standard desktop PC via ethernet UDP for ad-hoc advanced control algorithms and the considered delay allows to take into account possible delays introduced in the forward chain by the finite computational time of the control computer that communicates with the robot control system.

2.8.3 Force Control Implementation

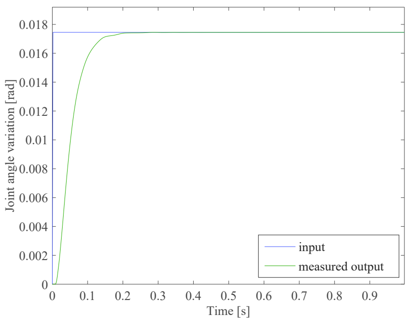
In order to evaluate the performances of the robot during a drilling task, a force/moment control scheme has been implemented as explained in Section 2.7. In particular, the use of a direct force control algorithm is required to minimize the tangential forces with respect to the panel and to control the force along the drilling direction; alike, a moment control algorithm is necessary to keep the drill axis perpendicular to the panel during the drilling operation. In Fig. 2.16 the implemented force/moment control scheme is reported and in Fig. 2.17 the detailed implemented moment control scheme is shown.



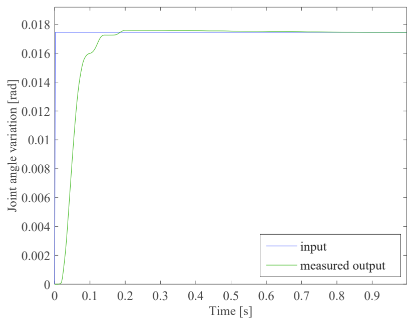
(a) Joint 1.



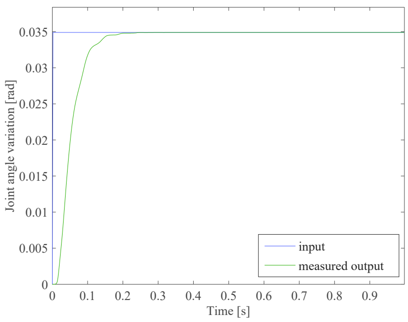
(b) Joint 2.



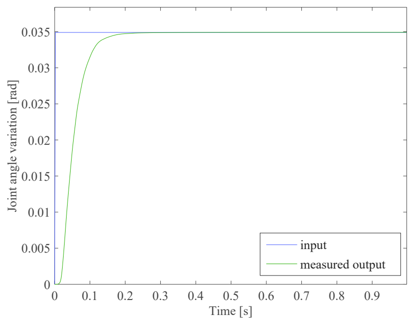
(c) Joint 3.



(d) Joint 4.

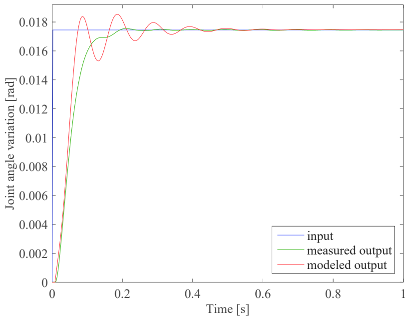


(e) Joint 5.

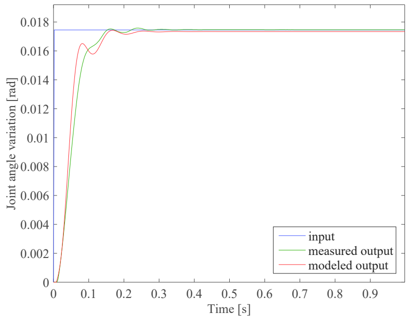


(f) Joint 6.

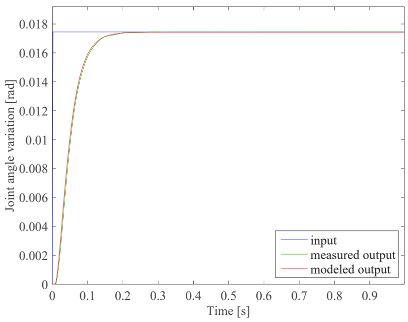
Figure 2.14: Identification Data.



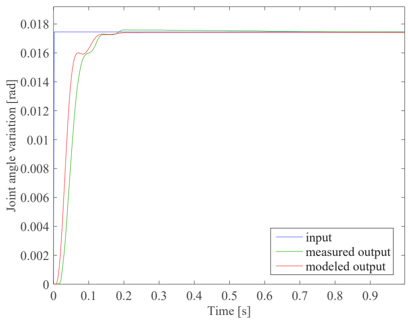
(a) Joint 1 Model: Step Response.



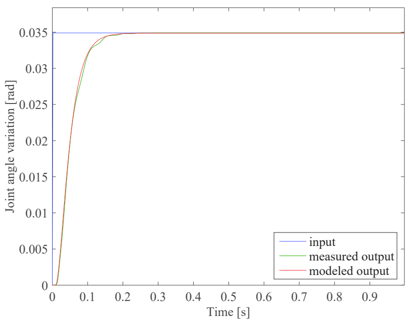
(b) Joint 2 Model: Step Response.



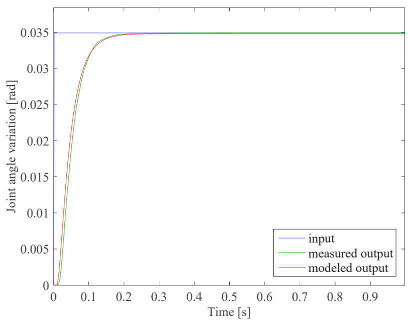
(c) Joint 3 Model: Step Response.



(d) Joint 4 Model: Step Response.



(e) Joint 5 Model: Step Response.



(f) Joint 6 Model: Step Response.

Figure 2.15: SmartSix Joint Model Identification.

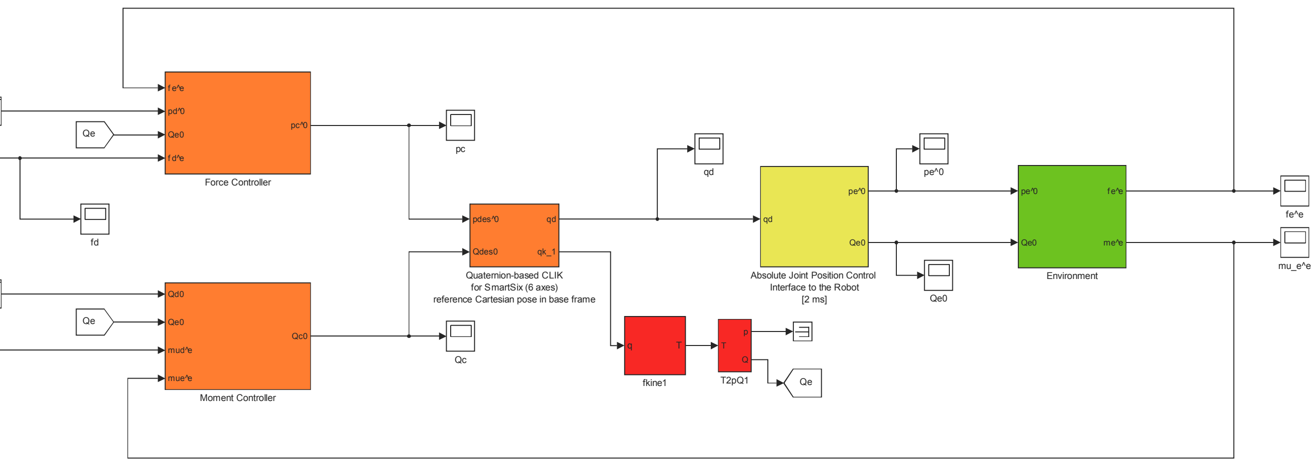


Figure 2.16: Implemented force control scheme.

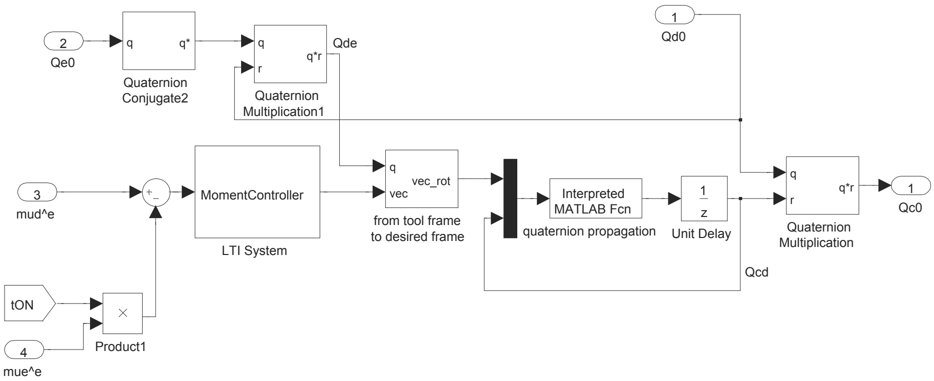


Figure 2.17: Moment control implementation.

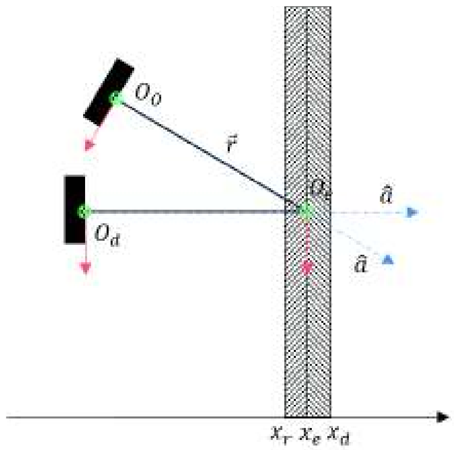


Figure 2.18: Arm in contact with an elastically compliant plane.

Given the desired force vector $\boldsymbol{f}_d^e$ and the desired torque vector $\boldsymbol{\mu}_d^e$, the error vectors can be calculated as $\boldsymbol{f}_d^e - \boldsymbol{f}_e^e$ and $\boldsymbol{\mu}_d^e - \boldsymbol{\mu}_e^e$, for the forces and moments respectively, where $\boldsymbol{f}_e^e$ and $\boldsymbol{\mu}_e^e$ are the forces and torques expressed in end-effector frame and generated on the interaction of the robot with the environment. Given the desired robot pose in Cartesian space expressed as $\boldsymbol{p}_d$ and $\boldsymbol{Q}_d$, the inverse kinematics algorithm can be used to obtain the desired joint configuration vector $\boldsymbol{q}_d$. The vector $\boldsymbol{q}_d$ is the input signals to the robot system obtained identifying the robot joints as in Section 2.8.2. From the effective joint configuration $\boldsymbol{q}$, the effective robot pose ($\boldsymbol{p}_e$ and $\boldsymbol{Q}_e$) can be obtained through the robot direct kinematics algorithm and the forces $\boldsymbol{f}_e^e$ arising on the interaction can be computed considering the elastic environment model in (2.27). Finally, given the end-effector orientation expressed in Euler angles $\boldsymbol{\phi}_e$ and the forces $\boldsymbol{f}_e^e$, referring to Fig. 2.18, the torques arising when the robot interacts with the environment can be computed as in (2.28), where $\boldsymbol{R}_e$ is the rotation matrix that expresses the end-effector frame orientation, $\boldsymbol{R}_r$ represents the orientation of the rest frame (frame attached to the environment), $\boldsymbol{f}_e^r$ is the force vector expressed in rest frame, and $\boldsymbol{r}$ is the vector representing the vector approach of the end-effector frame expressed in the end-effector frame.

$$\boldsymbol{\mu}_e^e = r \times \boldsymbol{R}_e^T \boldsymbol{R}_r \boldsymbol{f}_e^r \tag{2.28}$$

The proposed control scheme allows to keep the drilling axis perpendicular to the drilling object surface imposing the desired torque input $\boldsymbol{\mu}_d^e = \boldsymbol{0}$.

2.8.4 Compensator Design

The compensator design has been performed using Matlab Control System Toolbox[8]. Five compensators have been designed considering the five SISO (Single Input Single Output) systems obtained from the control input signals $\Delta\boldsymbol{x} = [\Delta\boldsymbol{p}, \Delta\boldsymbol{\phi}]$ and the outputs $\boldsymbol{h}_e = [\boldsymbol{f}_e, \boldsymbol{\mu}_e]$. Such systems have been identified using the Matlab Identification Toolbox. Note that no compensator is needed for the sixth identified model because given a variation of the input signal, no variation is obtained on the output. So, given the i-th identified model from the input $\Delta\boldsymbol{x}(i)$ to the output $\Delta\boldsymbol{h}_e(i)$, the compensator C_i has been designed considering as main design specification a settling time less then 0.8 s about. Such compensator results in a lead compensator that ensures a gain margin of 12 dB and a phase margin of 50 degrees at least. Moreover, the astatism with respect to a step input signal has been assured inserting an integral action in the compensator. The transfer functions of the designed compensators are reported below. Figure 2.19 reports the Nichols diagrams of the five closed loop systems.

$$C_1(s) = \frac{75(s^2 + 84s + 4900)}{s(s + 150)^2}$$

$$C_2(s) = \frac{92.932(s + 8)(s^2 + 24s + 3600)}{s(s + 16)(s + 80)(s + 200)}$$

$$C_3(s) = \frac{95.625(s^2 + 36.6s + 3721)}{s(s + 150)^2}$$

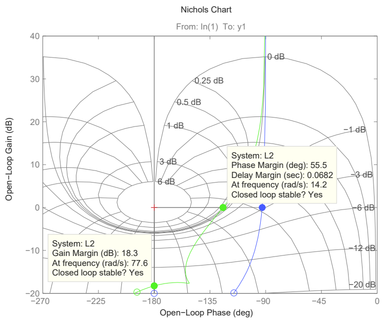
$$C_4(s) = \frac{282.49(s^2 + 30.16s + 3364)}{(s + 150)^2}$$

$$C_5(s) = \frac{250(s^2 + 36s + 3600)}{(s + 150)^2}$$

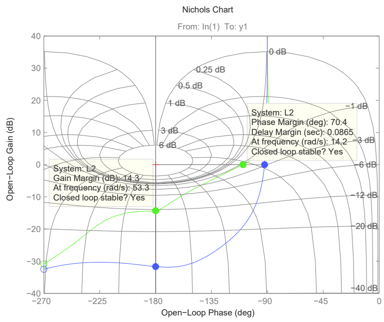
2.8.5 Drilling Configuration Optimization

In order to maximize the drilling capabilities of the robot, an accurate analysis of the force manipulability ellipsoid has been performed. The aim is to reconfigure the robot before the drilling phase. So, given the desired point on the panel to drill expressed in Cartesian space with respect to the base frame of the robot, using the

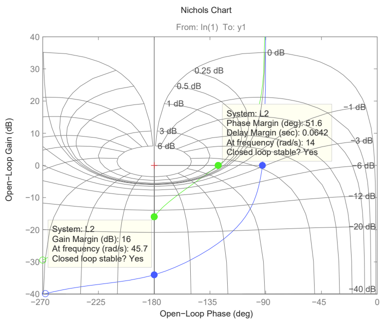
[8]Matlab Control System Toolbox contains two Root Locus design GUI, sisotool and rltool, two interactive design tools for the analysis and design of the single-input single-output (SISO) linear time-invariant (LTI) control systems.



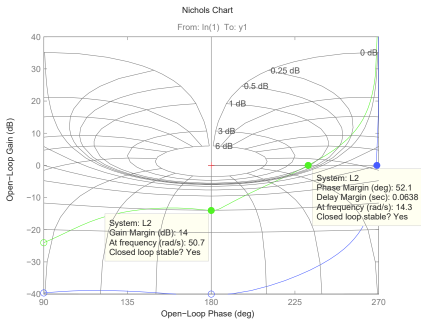
(a) System 1.



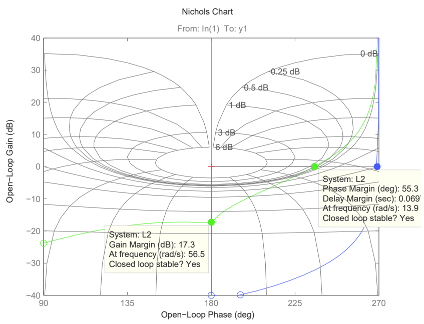
(b) System 2.



(c) System 3.



(d) System 4.



(e) System 5.

Figure 2.19: Designed Compensators: Nichols Charts.

inverse kinematics implemented as explained in Section 2.5 and reported in discrete time in eq. (2.29) (with $\boldsymbol{e}(t_k) = \boldsymbol{x}_d(t_k) - \boldsymbol{x}_e(t_k)$), the redundant dof (sliding track) is exploited to maximize the force manipulability ellipsoid along the drilling direction.

$$\boldsymbol{q}(t_k) = \boldsymbol{q}(t_{k-1})k_1\boldsymbol{J}_A^{\dagger}(\boldsymbol{q}_k)\boldsymbol{e}(t_k) + (\boldsymbol{I} - \boldsymbol{J}_A^{\dagger}(\boldsymbol{q}_k)\boldsymbol{J}_A(\boldsymbol{q}_k))\dot{\boldsymbol{q}}_0 \tag{2.29}$$

The homogeneous term $\dot{\boldsymbol{q}}_0$ can be chosen in order to satisfy the desired constraints specified through the definition of the cost functional W as in (2.30). The cost functional W can be selected as a linear combination of one or more terms that can take into account, e.g., a measure of the robot manipulability, a measure of the distance of the joints from the mechanical joint limits, a measure of the distance of the robot from an obstacle, etc.

$$\dot{\boldsymbol{q}}_0 = k_2\frac{\partial W(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{2.30}$$

In the proposed solution, the cost function has been chosen as a linear combination of three terms as reported in eq. (2.31). The first term allows to align the main axis of the force manipulability ellipsoid to the approach unit vector of the tool; the second one allows to maximize the dimension of the main axis of the force manipulability ellipsoid. In detail, the first objective corresponds to aligning the first eigenvector $\boldsymbol{u}$ of the matrix $(\boldsymbol{J}\boldsymbol{J}^T)$, that is the core of the force manipulability ellipsoid, to the unit vector approach $\boldsymbol{a}$ (or, according to the concept of force/velocity duality, to align the third eigenvector of the matrix $(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$, that is the core of the velocity manipulability ellipsoid, to the unit vector approach $\boldsymbol{a}$). Similarly, the second objective corresponds to maximizing the singular value σ of $\boldsymbol{J}$ or eigenvalue of the matrix $(\boldsymbol{J}\boldsymbol{J}^T)$ corresponding to the main axis of the force manipulability ellipsoid (or, according to the concept of force/velocity duality, minimize the singular value of $\boldsymbol{J}^{-1}$ or eigenvalue of the matrix $(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$ corresponding to the least axis of the velocity manipulability ellipsoid). Finally, denoted with q_{i_M} and q_{i_m} the maximum and the minimum joint limit and $\bar{q}_i$ the middle value of the joint range, minimizing the third term of (2.31) the robot joint variables can be kept as close as possible to the center of their ranges.

$$W = -k_3W_1 + k_4W_2 + k_5W_3 \tag{2.31}$$

$$W_1 = \left\|acos(\boldsymbol{u}^T\boldsymbol{a})\right\|^2$$

$$W_2 = \sigma^2$$

$$W_3 = -\frac{1}{2n}\sum_{i=1}^{n}\left(\frac{q_i - \bar{q}_i}{q_{i_M} - q_{i_m}}\right)^2$$

In order to avoid possible local minimum during the optimization process, the CLIK algorithm can be evaluated considering m pseudo-random initial configurations. The optimal solution $\boldsymbol{q}_{opt}$ is the solution that best optimizes the considered cost function W. Moreover, to assure that the optimization process is correctly performed, the algorithm has to be stopped only if the internal motions $\left\|\dot{\boldsymbol{q}}_0\right\| < \epsilon$, where ϵ is an arbitrary small scalar.

2.8.5.1 Simulations

In the proposed analysis, a vertical panel of $1000 \times 1000 \times 10$ mm has been positioned in the scene. The robot task is to drill a point on such panel preserving the desired forces and the drilling direction aligned to the normal vector of the panel without the use of jigs. So, given the desired pose of the tool on the point to drill, an optimization phase has been performed before starting the drilling process. The panel has been positioned with the center of gravity in $\boldsymbol{p}_{panel_{cog}}^{pointing} = [0, 1601.0, 1169.9]$ mm and $\boldsymbol{p}_{panel_{cog}}^{hanging} = [0, 1201.0, 1169.9]$ mm with respect to the robot base frame and the point to drill is $\boldsymbol{p}_d^{pointing} = [400.0, 1600.0, 1100.0]$ mm and $\boldsymbol{p}_d^{hanging} = [400.0, 1200.0, 1100.0]$ mm expressed in robot base frame for pointing configuration and hanging configuration, respectively. Moreover, the drilling tool has been modeled as a cylinder of length 350 mm. Both the pointing and hanging configurations are considered in the optimization process. For completeness, the transformation matrices $\boldsymbol{T}_{TCP}^{e}$ for the drilling configurations are reported below.

$$\boldsymbol{T}_{TCP_{Pointing}}^{e} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{p}_{TCP_{Pointing}}^{e} \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 350 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{TCP_{Hanging}}^{e} = \begin{bmatrix} \boldsymbol{R}_y(\pi/2) & \boldsymbol{p}_{TCP_{Hanging}}^{e} \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 325 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 25 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The results of the optimization process are shown in Fig. 2.20: referring to the $\boldsymbol{q}_{opt}$, on top the CLIK errors and in the middle the estimated configurations are shown. The optimization has been performed setting the number of CLIK iterations to 100, the number of initial configurations to 10 and the tolerance parameter $\epsilon = 10^{-3}$. Moreover, the Euler angle about the approach axis has been left free because it does not affect the success of the task and, thus, the CLIK error has been computed considering only the three positional coordinates and the first two orientational coordinates in the Cartesian space. The force manipulability ellipsoids corresponding to the obtained optimal configurations are reported in Fig. 2.21 (the figure shows a Matlab Robotics Toolbox view).

2.8.6 Force Control Simulation

The presented control algorithm and the designed compensators have been analyzed and validated by simulations in Matlab/Simulink environment. The reported simulations have been carried out by considering a one hole drilling task without the use of a jig, and supposing that there is not an error in the robot positioning. The system response has been evaluated by changing the desired drilling forces $\boldsymbol{f}_d^e$ and the initial orientation of the tool frame with respect to the environment (rest frame) (see Fig. 2.22), maintaining constants the desired moments $\boldsymbol{\mu}_e^e$. From Fig. 2.23 it can be seen that the moments $\boldsymbol{\mu}_e^e$ generated when the robot interacts with the environment increase by increasing the misalignment of the tool with respect to the panel and by increasing the module of the force $f_{d_z}^e$. However, with a maximum misalignment of $\theta = 15\,^\circ$ and a maximum force $f_{d_z}^e = 200\,\mathrm{N}$, the moments are less than 5 Nm. Furthermore, by increasing the force $f_{d_z}^e$, the tangential forces $f_{e_x}^e$ and $f_{e_y}^e$ increase, although int the presented results, they are less than 12 N and, then, reach the desired values in about 0.5 s. A similar trend can be noted for the normal component $f_{e_z}^e$ of the force, which tends to overshoot when the robot collides with the panel (between 3 s and 4 s) and then converges to the desired value in about 0.5 s. In Figure 2.24 the trend of the angle between the drilling axis and the normal to the surface of the panel for the cases under consideration is shown. In particular, it can be noted as the angle θ starts from the value that represents the initial misalignment, and vanishes after a brief transient (approximately 0.5 s).

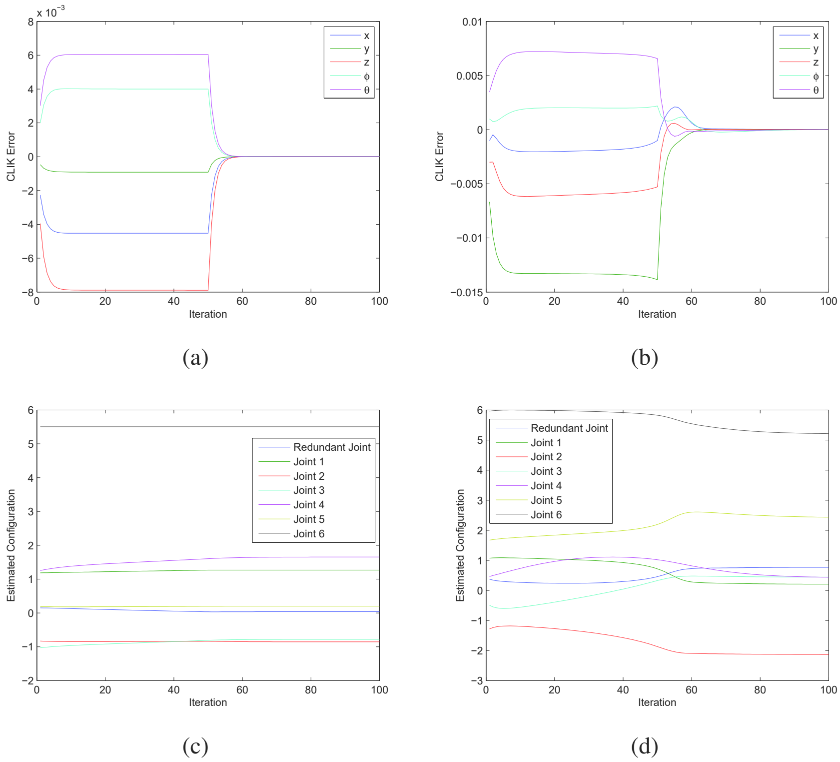
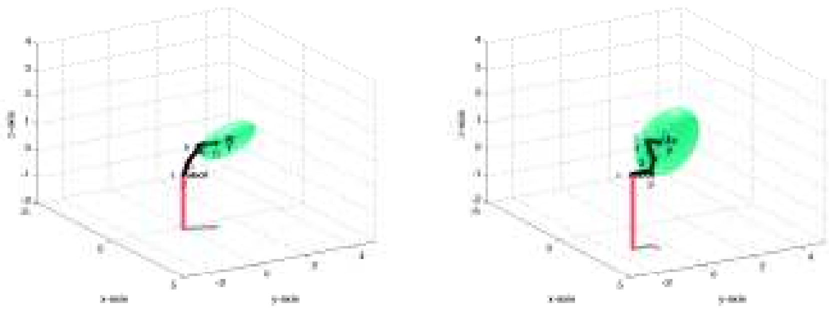


Figure 2.20: Optimization Process Results.



(a) Pointing Configuration. (b) Hanging Configuration.

Figure 2.21: Optimization Process Results: Force Manipulability Ellipsoids.

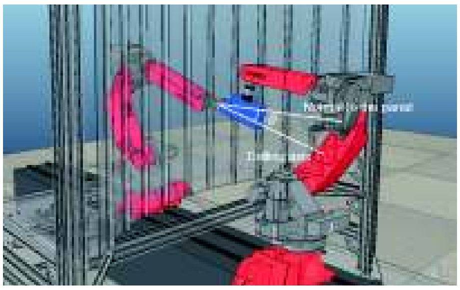
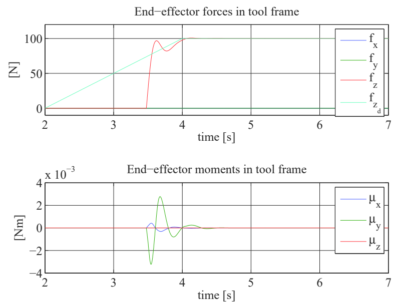


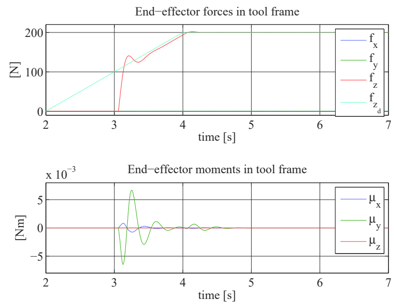
Figure 2.22: Initial misalignment between the drilling axis and the normal to the panel.

2.8.7 Experiments

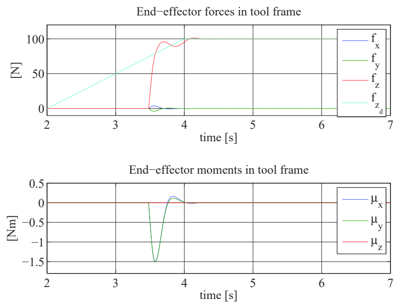
In this section, the results of the experiments carried out at the robotic laboratory of UNISA are shown. In particular, the task consists to drill a fuselage panel in aluminium material with the use of a jig, pre-clamped on the considered panel. The experiments have been carried out by using a pneumatic drilling end effector provided by Alenia Aermacchi, mounted on the Comau SmartSix robot by considering the hanging configuration. The drilling end effector is a Lübbering L.ADU pneumatic, properly adapted to be mounted on the robot, equipped with a concentric collet to assure the perfect clamping of the tool into the jig hole during the drilling process. Such a system, further, allows to balance the forces and the vibrations generated during the drilling on the jig without affecting the robot. Between the end effector and the robot, a 6 dofs force sensor (ATI Gamma SI130-10) has been installed. The mentioned equipment and the experimental setup are reported in Fig. 2.25. The control



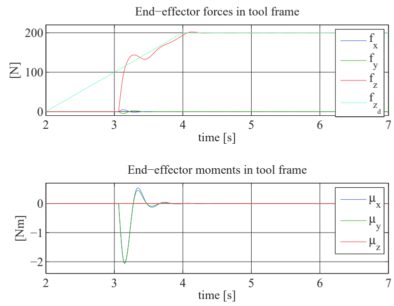
(a) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 0\,\mathrm{deg}$.



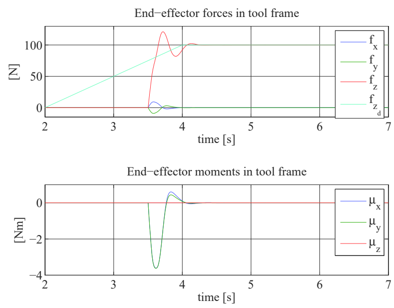
(b) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 0\,\mathrm{deg}$.



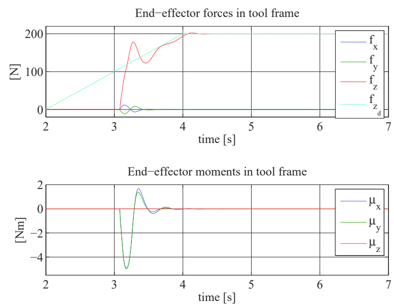
(c) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 6\,\mathrm{deg}$.



(d) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 6\,\mathrm{deg}$.

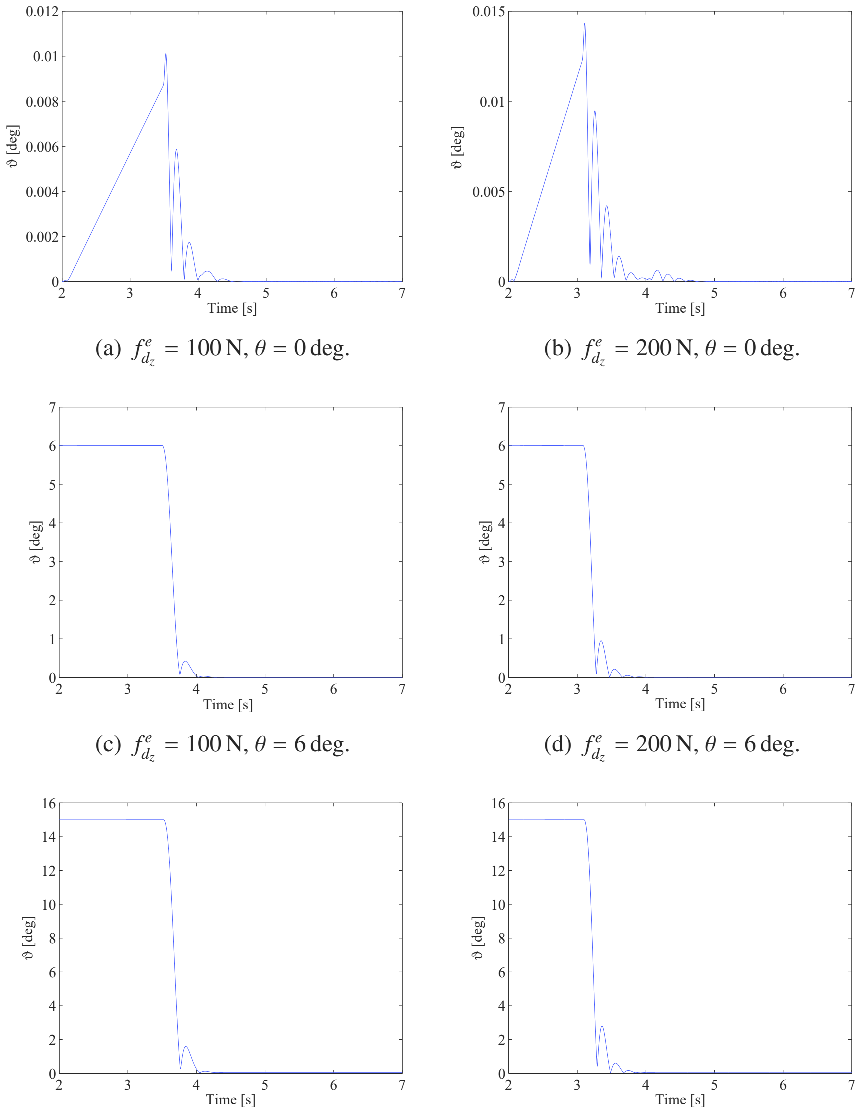


(e) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 15\,\mathrm{deg}$.



(f) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 15\,\mathrm{deg}$.

Figure 2.23: Force/moment control simulation results.



(a) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 0\,\mathrm{deg}$.

(b) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 0\,\mathrm{deg}$.

(c) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 6\,\mathrm{deg}$.

(d) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 6\,\mathrm{deg}$.

(e) $f^e_{d_z} = 100\,\mathrm{N}$, $\theta = 15\,\mathrm{deg}$.

(f) $f^e_{d_z} = 200\,\mathrm{N}$, $\theta = 15\,\mathrm{deg}$.

Figure 2.24: Misalignment between the drilling axis and the normal to the panel.

(a) UNISA laboratory.

(b) ATI Gamma sensor.

(c) Panel with drilling jig.

(d) Assembly fixture.

(e) Pneumatic drilling end effector.

(f) Work-cell top view.

Figure 2.25: Experimental setup at UNISA.

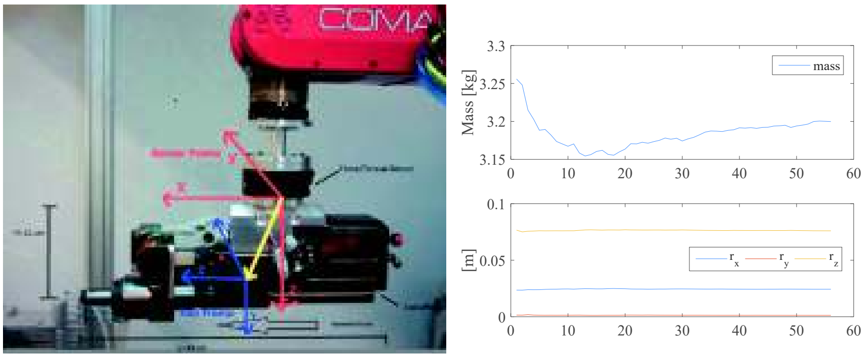
algorithm has been implemented in C++ on a hard real-time Linux operating system by using the RealTime Application Interface (RTAI patch) and the OROCOS (Open RObot Control Software) toolchain. The communication between the control computer and the robot controllers happens via UDP datagrams. The time delay introduced by the communication interface (due to package lost, for example) has been modeled introducing an input time delay of 0.008 s into the dynamic models of the manipulator as explained in Section 2.8.2.

2.8.7.1 End-effector Identification

The designed force control algorithm requires the compensation of the drilling end-effector dynamics in the force sensor measurements in order to correctly estimate the contact forces. The tool dynamics has been considered and the Coriolis and inertia terms have been neglected due to the low operative velocities and accelerations during the drilling process. So, in order to estimate the center of gravity point of the end effector, and its mass, an identification procedure has been carried out. In particular, the robot has been brought in 57 configurations, and the forces and the torques have been measured in static conditions by using the force sensor. Given the robot configurations and the sensor measurements, a least-square problem has been solved. Let consider the vector $\boldsymbol{b}$ containing the sensor measurements, the gravity vector $\boldsymbol{g^b}$ expressed in base frame and the skew matrix $\boldsymbol{S}(\boldsymbol{g}^{sensor})$ where $\boldsymbol{g}^{sensor}$ is computed as in (2.32) and $\boldsymbol{R}^{b}_{sensor}$ is the rotation matrix representing the orientation of the sensor frame with respect to the base frame.

$$\boldsymbol{g}^{sensor} = \boldsymbol{R}^{b^T}_{sensor}\boldsymbol{g}^{b}, \;\; \boldsymbol{g}^{b} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{S}(\boldsymbol{g}^{sensor}) = \begin{bmatrix} 0 & -g_z^{sensor} & g_y^{sensor} \\ g_z^{sensor} & 0 & -g_x^{sensor} \\ -g_y^{sensor} & g_x^{sensor} & 0 \end{bmatrix} \tag{2.33}$$



(a) UNISA laboratory.

(b) ATI Gamma sensor.

Figure 2.26: Pneumatic end-effector identification.

The mass and the COG point of the drilling tool with respect to the sensor frame can be computed as:

$$\begin{bmatrix} m \\ r_x m \\ r_y m \\ r_z m \end{bmatrix} = \boldsymbol{A}^{\dagger}\boldsymbol{b}, \tag{2.34}$$

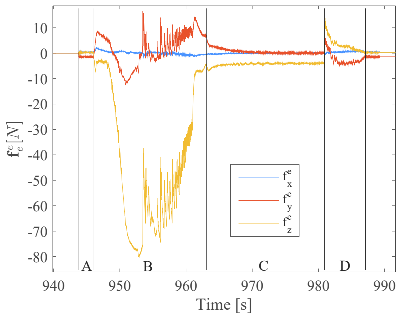
where the regressor $\boldsymbol{A}$ is computed as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{g}^{sensor} & \boldsymbol{0}_3 \\ \boldsymbol{0}^T & \boldsymbol{S}^T(\boldsymbol{g}^{sensor}) \end{bmatrix}. \tag{2.35}$$

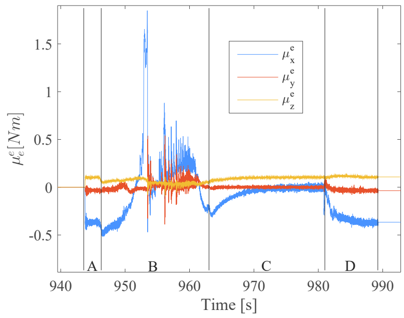
Figure 2.26 shows the considered frames on the left and the estimated parameters with respect to the number of configurations considered in the identification process on the right.

2.8.7.2 Results

The drilling task consists of four operational phases: the approach phase, the insertion phase, the drilling phase and the extraction phase. The approach phase allows to position the robot end effector in front of the hole of the jig where the drilling operation will be performed and it is carried out in position control mode. The rough poses can be obtained from a previous teaching or from the CAD of the panel. The successive three phases allows the robot to insert the concentric collet into the jig, to perform the drilling operation and to extract the concentric collet from the jig, respectively, and they are performed in force control mode. Figure 2.27 shows the forces and the



(a) Forces.



(b) Torques.

Figure 2.27: Sensor measurements during a drilling process.

(a) Drilling process.

(b) Insertion details.

Figure 2.28: Pneumatic end-effector identification.

torques measured during a single cycle of drilling. The reported results show that the forces and torques are higher in the insertion and extraction phases, and decrease in the drilling phase. Figure 2.28 shows the robot during the drilling process on the left and a detailed picture depicting the insertion of the concentric collet into the jig on the right.

CHAPTER 3

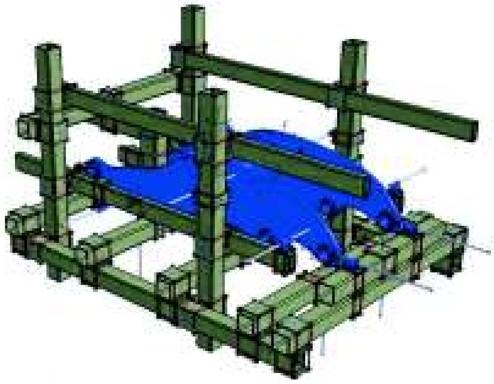
COOPERATIVE ROBOTS FOR PART POSITIONING

Tasks which are difficult to execute with a single manipulator, become feasible when two or more manipulators work in a cooperative way. It is the case of heavy and large work-objects, assembly of multiple parts, and handling of large, articulated or flexible objects. Most of the research works assume that the manipulators tightly grasp a rigid object, but in the last decades the efforts have been focused on the control of cooperative flexible manipulators and on the manipulation of flexible objects [13].

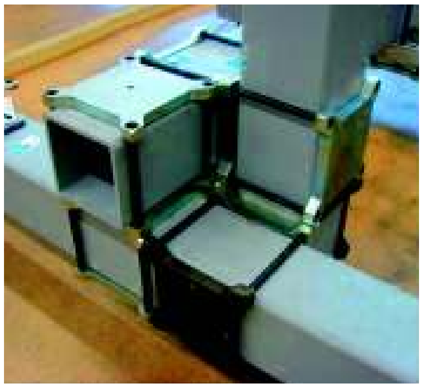
The cooperative manipulation, coupled with the use of force/moment control, can be exploited in the robotized positioning of large and/or flexible assembly parts.

3.1 Robotized Positioning of Assembly Parts

The most common solution today for assembly tooling in the aerospace industry is the use of dedicated tooling consisting of welded structures (fixture), designed and manufactured for a dedicated component and process. When the product design changes, modifications to the assembly tooling are needed and this requires time and it is very expensive, but, especially, it can cause disturbances and delays in the production process. In recent years, with the increase of the demand in the aeronautics industry, more innovative tooling solutions to improve process flexibility, re-configurability and to reduce lead times have been proposed. The most common systems are based on modular aluminium sections, which allow to have enough flexibility to be used for different products. The last generation of reconfigurable tooling systems is known as adaptive tooling: they allow full dimensional flexibility whilst delivering improved



(a) Example.



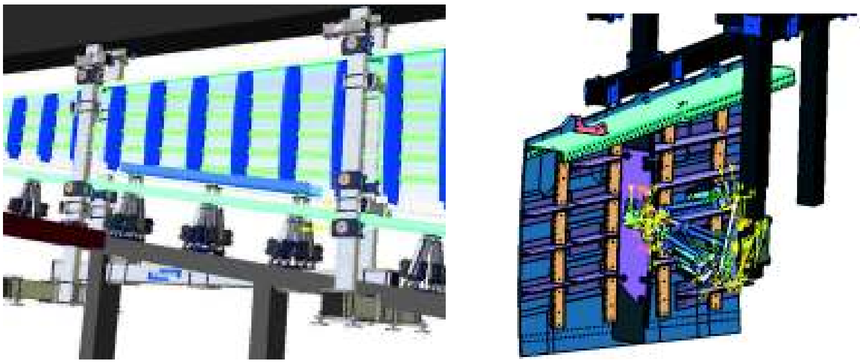
(b) Gough design

Figure 3.1: Box Joint.

rigidity and accuracy. An example of adaptive tooling is proposed by Prodtex (ex DELFOi): the BoxJoint system [14] (Fig. 3.1) provides datum and clamping points that can be located with an infinite resolution on the structure. The same structure can be automated with the use of six degrees of freedom parallel robots (i.e. Stewart platform-based mechanisms) providing the advantage that they are programmable, and therefore, can be rapidly reconfigured to take different products. Some of these solutions [15] are currently being tested for use in production by both the aerospace and automotive sectors, although, they are still one of the hot topics of the aeronautics research.

In this work, the focus is on the development of methodologies and tools to support the design of ad-hoc Stewart platforms to be used in the robotized positioning of the assembly parts. In particular, tailored Stewart platforms can be designed and used to automatize the reconfigurable tooling by taking into account the payload of the parts and/or external forces applied to the parts (i.e., forces applied during a manufacturing process, e.g., drilling).

Two of the main sub-assembly operations of a wing-box, i.e., rib positioning and spar positioning, could benefit of the robotized positioning. Given the high number of components to assembly which constitute a wing-box of an aircraft, and given the limited available space on the fixture due to high number of tools used to manufacture these parts while they are assembled on the fixture (fastening tools, drilling tools, inspection tools, etc.) tailored machines are required to fulfill the aeronautics requirements in terms of accuracy and space. Furthermore, the most of the Stewart platforms available today on the market and designed for pick and place applications,



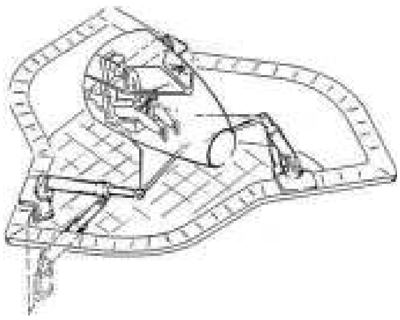
(a) Spar positioning. (b) Rib positioning.

Figure 3.2: Part positioning tasks in the LOCOMACHS project.

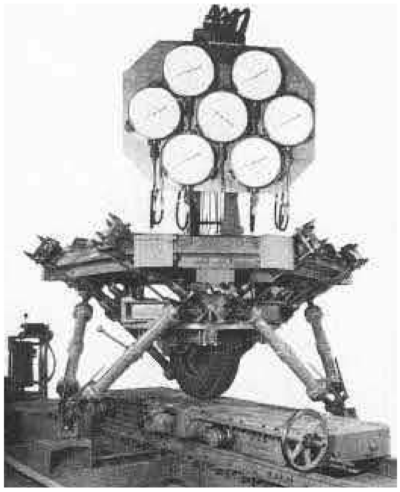
do not meet the sought criteria due to the high cost, the limited linear reach and/or the low payload . Example of existing platforms on the market are the Alio S 6-D Stepper Hybrid and the PI M-840 HexaLight, very accurate but too expensive (60.000 £) and with low payload (20 and 30 kg respectively); the Fanuc F-200iB which reflects the requirements of accuracy and reach but it is not suitable for the part positioning applications due to its high cost (32.000 £); the MOOG HX-P300 with good accuracy but with limited reach (±60 mm); the Sysmetrie Sirus and Notus provide high repeatability but poor leg and joint rigidity, so they are ideal for laboratory research and motion simulator applications. So, an ad-hoc design of parallel robots for these specific applications can significantly reduce the cost of the assembly process[9] over the adoption of existing solutions.

The effort of this work, then, has been to develop a tool to support the design of ad-hoc cost effective Stewart platform for specific applications. The proposed solution has been exploited in the LOCOMACHS project to design and develop two ad-hoc Stewart platforms for the positioning of the ribs and of the lower spar in the wing box assembled in the physical demonstrator. Given the complexity of the fixture and the high number of tools used inside such demonstrator, some mechanical constraints, i.e, plate dimensions, motor strokes and actuator dimensions, have been taken in account during the optimization process. Moreover, the pre-computed trajectory has been considered for each specific application. Figure 3.2 shows a detailed CAD model of the demonstrator of the LOCOMACHS project. In particular, in (a) the spar positioning application is illustrated and in (b) the rib positioning application

[9]The cost of a single platform is estimated to be about 17000 € including the Beckhoff control system. Data provided by Prodtex and MTC.



(a) Stewart design.



(b) Gough design.

Figure 3.3: Stewart and Gough original designs.

is shown (CAD provided by the LOCOMACHS consortium).

3.2 Optimal Design of Parallel Robots for Robotized Positioning

In recent years, great interest has been devoted to parallel manipulators based on the Stewart platform, also named hexapod. The Stewart platform was invented as a flight simulator by Stewart in 1965 [16]. This platform contained three parallel linear actuators. Gough had previously suggested a tire test machine similar to Stewart's model. In the test machine, six actuators were used as a mechanism driven in parallel. Gough was the first person who developed and utilized this type of parallel structure. Therefore, Stewart platform is sometimes named as Stewart-Gough platform in the literature. Stewart's and Gough's original designs [17] are shown in Figure 3.3.

In comparison with a serial manipulator, the Stewart parallel manipulator, capable of providing six degrees of freedom (DOF) movement, come up with some advantages [18]:

- High strength and stiffness-to-weight ratios can be achieved since the links do not carry moment loads but act only in tension and compression
- Positioning of the end effector is performed by actuators acting in parallel,

resulting in a total force and moment capability greater than each individual servomechanism

- Moving only the end effector in space rather than massive servomechanisms results in economy of power, excellent dynamic performance, and low manipulator inertia

- High accuracy and precision is possible since actuator errors are not magnified by lengthy linkages

The Stewart platform is suitable for a wide range of applications such as flight simulation [19], spaceship aligning, radar and satellite antenna orientation [20, 21], rehabilitation applications [22], robots [23], parallel machine tools [24, 25]. Unfortunately, there are factors limiting the application of parallel mechanisms. First, the limited workspace that reduces the number of tasks the robot can execute and the singularity configurations existing inside the workspace in which the manipulator gains one or more degrees of freedom and therefore loses its stiffness. The closed-loop nature of parallel mechanisms generates complex singularities inside the workspace, which makes the workspace analysis and the trajectory planning of parallel mechanisms a very difficult problem. Moreover, although the versatility of the hexapod has been recognized, its acceptance by industry as production equipment has not yet occurred. Some obstacles to this include the high cost and unproven performance in a production environment for a specific task. Hence, development of efficient tools that allow to maximize the robot workspace, to reduce the singularities inside the workspace, to optimize the design of the parallel platform reducing the hexapod costs, e.g. by choosing a suitable set of linear actuators, and in general to achieve good performances becomes a very important issue.

3.2.1 State of the Art

During the past decade, the structural design and optimization of Stewart platform have been carried out by many researchers. Given the number of performance parameters to consider (i.e., workspace volume, manipulability, dexterity, singularity, accuracy, actuators interference, actuation forces) it is still difficult to find an optimal general design for a 6-DOF parallel manipulator. Design optimization and dimensional synthesis of the parallel mechanism has been presented as a multi-criteria constrained problem in [26, 27, 28, 29, 30, 31, 32]. In the optimization process kinematics parameters are usually considered, i.e., workspace [32, 33, 34, 35, 36, 37, 38, 39, 40, 41], stiffness [42, 43, 44, 45], dexterity [32, 46, 47], singularity [21, 48, 36, 37], maximum end-effector velocity [49], manipulability [50]. Many investigations have been done on the dynamic optimal design of parallel robots by analyzing criteria such as balance [23, 51, 52, 53, 54] and torque index [23, 55], but there is few literature that deals with the issues of the anisotropic property [56, 49] of parallel robots or that considers criteria such as acceleration, velocity, gravity and external force components in the considered cost function. Those issues are well explained in [55]. Briefly, in the existing literature, the isotropy property is usually pursued in the dimensional synthesis of the parallel robot. But it is known that most of the parallel robots with symmetrical structures have not isotropic performance in the whole workspace since they have not the same capability in all directions [54, 57]. Also, the performance requirement of the parallel robot is usually not uniform in all directions within the entire desired workspace in practical applications. Thus, the anisotropic property should be considered in the dimensional synthesis of the parallel robot with the aim to obtain a more suitable optimal design. Furthermore, the objective function of the dynamic optimal design of the parallel robot is usually based on the generalized inertia matrix, which describes the mapping between the joint forces/torques and the accelerations. The velocity, gravity and the external force components are not considered in the above objective functions. On the other hand, the velocity components should be taken into account when the parallel robot is used for high speed operation, the gravity component should be considered when the parallel robot is applied to heavy load situation, the external force component should be considered when the parallel robot is used for machining operations.

In the present work, a simulation environment to support the design of a low cost Stewart platform-based mechanism for manufacturing applications is presented. In order to maximize the payload and improve the rejection of external forces exerted on the mobile platform during positioning or manufacturing applications, a dynamic optimization has been carried out. Moreover, in order to avoid reduction of the robot workspace, also a kinematic optimality criterion has been considered in the optimization process as well. To this aim, the design has been optimized by determining the leg attachment points on both top and base plates. In particular, the leg attachment points on either mobile (or top) and fixed (or base) plates have been optimized satisfying the mechanical constraints introduced in the design (such as leg attachment point geometry, distances between the legs/actuators, minimum and maximum top and base plate dimensions, minimum and maximum leg strokes). An optimization algorithm has been used to combine two or more different optimum objectives by properly defining a cost function to minimize. In order to minimize the maximum leg force value and to equally distribute among the legs the forces exerted by the linear actuators during a positioning and/or machining task, the maximum root-mean-square (RMS) value of the forces has been selected as optimum objective. A second optimum objective has been taken into account to maximize (or do not penalize) the robot workspace volume. In order to select the most suitable optimization algorithm for the proposed application, different algorithms have been compared. The performances of the Genetic Algorithm (GA) [58, 59], the Sequential Quadratic Programming (SQP) algorithm [60, 61, 62, 63], the MultiStart algorithm and the GlobalSearch algorithm [64, 65, 66] available in the Matlab Optimization Toolbox, have been analyzed and compared. The comparison shows that the GA provides better results than the other ones in terms of minimum found cost function value and number of cost function evaluation. The SQP algorithm and GlobalSearch algorithm are not suitable for the presented application because they stick in local minima. Differently, the MultiStart algorithm could be a valid alternative to the GA algorithm. Finally, in order to exploit the anisotropic property of the parallel robot to better optimize the mechanical design given a specific task, the Stewart platform optimization process has been carried out considering both symmetric and unsymmetric geometries.

3.2.2 Basic modeling of Stewart platforms

The Stewart platform is a closed kinematic chain manipulator comprising six linear actuators, each connected by a universal joint to the manipulator base and by a spherical joint to the top platform. This arrangement of actuators allows the platform to be placed in any position and orientation within a certain volume of space. Let denote with $l_i, i = 1, ..., 6$, the six actuated prismatic leg lengths, with $\boldsymbol{a}_i = (a_{x_i}, a_{y_i}, a_{z_i})^T$ and $\tilde{\boldsymbol{b}}_i = (b_{x_i}, b_{y_i}, b_{z_i})^T$ the position vectors of the center of the universal (A_i) and spherical (B_i) passive joints given in base and mobile platform reference frames, respectively (see Fig. 3.4).

Given that the pose of the platform can be defined by a position vector $\boldsymbol{p} = (x_p, y_p, z_p)^T$ and either a rotation matrix $\boldsymbol{R}$ or a set of three XYZ Euler angles (φ_p, θ_p, ψ_p), with respect to the base frame, where the rotation matrix $\boldsymbol{R}$ is defined as in eq. (3.1),

$$\boldsymbol{R} = (\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}) = \begin{pmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{pmatrix} \tag{3.1}$$

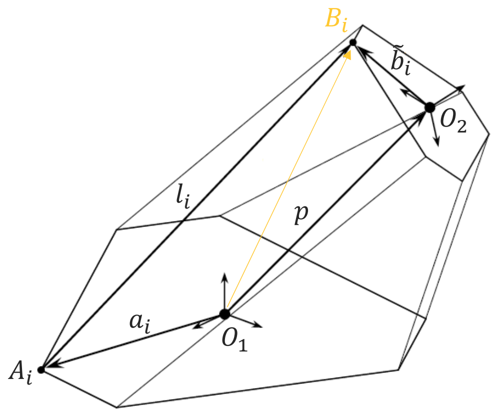


Figure 3.4: Stewart platform manipulator: general scheme.

the platform attachments $\tilde{\boldsymbol{b}}_i$ can be written in the base frame as

$$\boldsymbol{b}_i = \boldsymbol{p} + \boldsymbol{R}\tilde{\boldsymbol{b}}_i, i = 1, ..., 6. \tag{3.2}$$

The following subsections illustrate the inverse kinematics (IK) problem, the Jacobian computation, the statics problem and the inverse dynamics problem of a general Stewart platform.

3.2.2.1 Inverse Kinematics

The inverse kinematics problem of a Stewart platform [67] consists in computing the leg lengths l_i given the position $\boldsymbol{p}$ and the orientation (φ_p, θ_p, ψ_p) of the mobile platform. So, the IK can be computed as described in eq. (3.3), where $\boldsymbol{b}_i$ is calculated as in (3.2).

$$l_i = \left\| \boldsymbol{b}_i - \boldsymbol{a}_i \right\| \tag{3.3}$$

3.2.2.2 Statics

Let $\boldsymbol{x} = [x_p, y_p, z_p, \varphi_p, \theta_p, \psi_p]^T$ be the vector with the six Cartesian coordinates of the mobile platform and $\boldsymbol{q} = [l_1, l_2, l_3, l_4, l_5, l_6]^T$ be the vector with the six leg lengths. If $\boldsymbol{x}(t)$ represents the pose of the mobile frame with respect to the base frame at any time t, the leg lengths can be computed as:

$$l_i = \left\| \begin{matrix} x_p - a_{x_i} + b_{z_i} s_{\theta_p} + b_{x_i} c_{\psi_p} c_{\theta_p} - b_{y_i} c_{\theta_p} s_{\psi_p} \\ y_p - a_{y_i} + b_{x_i}(c_{\varphi_p} s_{\psi_p} + c_{\psi_p} s_{\varphi_p} s_{\theta_p}) + b_{y_i}(c_{\varphi_p} c_{\psi_p} - s_{\varphi_p} s_{\psi_p} s_{\theta_p}) - b_{z_i} c_{\theta_p} s_{\varphi_p} \\ z_p - a_{z_i} + b_{x_i}(s_{\varphi_p} s_{\psi_p} - c_{\varphi_p} c_{\psi_p} s_{\theta_p}) + b_{y_i}(c_{\psi_p} s_{\varphi_p} + c_{\varphi_p} s_{\psi_p} s_{\theta_p}) + b_{z_i} c_{\varphi_p} s_{\theta_p} \end{matrix} \right\| \quad (3.4)$$

where the symbols s_α and c_α denote $\sin(\alpha(t))$ and $\cos(\alpha(t))$, respectively. Note that in the (3.4) the time dependence is omitted.
The analytical Jacobian of the generic Stewart platform [25] is given by

$$\boldsymbol{J}_A(\boldsymbol{x}) = \frac{\partial \boldsymbol{q}(\boldsymbol{x})}{\partial \boldsymbol{x}} \quad (3.5)$$

From the inverse kinematics equation, it is possible to compute the inverse differential kinematics mapping between the vector of the generalized velocities $\dot{\boldsymbol{x}} = [\dot{x}_p, \dot{y}_p, \dot{z}_p, \dot{\varphi}_p, \dot{\theta}_p, \dot{\psi}_p]^T$ and the vector of leg velocities $\dot{\boldsymbol{q}} = [\dot{l}_1, \dot{l}_2, \dot{l}_3, \dot{l}_4, \dot{l}_5, \dot{l}_6]^T$ as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A(\boldsymbol{x})\dot{\boldsymbol{x}}. \quad (3.6)$$

The matrix $\boldsymbol{J}_A$ is the analytical Jacobian to be distinguished from the geometric Jacobian $\boldsymbol{J}$ relating the joint velocity vector to the end-effector velocity $\boldsymbol{v} = [\dot{\boldsymbol{p}}_e, \boldsymbol{\omega}_e]^T$, being $\dot{\boldsymbol{p}}_e$ the linear velocity and $\boldsymbol{\omega}_e$ the angular velocity. The inverse differential kinematics in terms of the geometric Jacobian is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{x})\boldsymbol{v}. \quad (3.7)$$

By comparing (3.6) with (3.7), the relationship between the two Jacobians becomes

$$\boldsymbol{J}(\boldsymbol{x}) = \boldsymbol{J}_A(\boldsymbol{x})\boldsymbol{T}_A^{-1}(\boldsymbol{x}), \quad (3.8)$$

where $\boldsymbol{T}_A(\boldsymbol{x})$, having chosen the XYZ Euler angles, is

$$\boldsymbol{T}_A(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}(\boldsymbol{x}) \end{bmatrix}, \ \boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} 1 & 0 & s_\theta \\ 0 & c_\theta & -s_\varphi c_\theta \\ 0 & s_\theta & c_\varphi c_\theta \end{bmatrix}. \quad (3.9)$$

By virtue of the duality established by the principle of virtual works, the inverse statics mapping between the vector of leg forces $\boldsymbol{\tau}$ and the vector of generalized forces $\boldsymbol{h}_e$ on the mobile base is

$$\boldsymbol{\tau} = \boldsymbol{J}^{-T}(\boldsymbol{x})\boldsymbol{h}_e = \boldsymbol{J}_A^{-T}(\boldsymbol{x})\boldsymbol{T}_A^T(\boldsymbol{x})\boldsymbol{h}_e. \tag{3.10}$$

3.2.2.3 Inverse Dynamics

The dynamic analysis of a Stewart platform is more difficult in comparison with the serial manipulator because of the existence of several kinematic chains all connected by the moving platform. Several methods were used in the literature to describe the problem and obtain the dynamic model of the manipulator, but there is still no consensus on which formulation is the best to describe the dynamics of the manipulator. A Lagrange formulation was presented in [68] to provide an analytical and orderly model, while a closed form solution for the inverse and direct dynamics models based on the Newton-Euler formulation was presented in [69]. In the proposed work, the inverse dynamics (ID) of the Stewart platform has been computed using a physics-based approach by modeling the parallel mechanism in the Matlab/Simulink environment using the SimMechanics Toolbox. This solution has been selected to obtain an easy-to-use software tool also owing to the availability of an optimization toolbox in the same software environment.

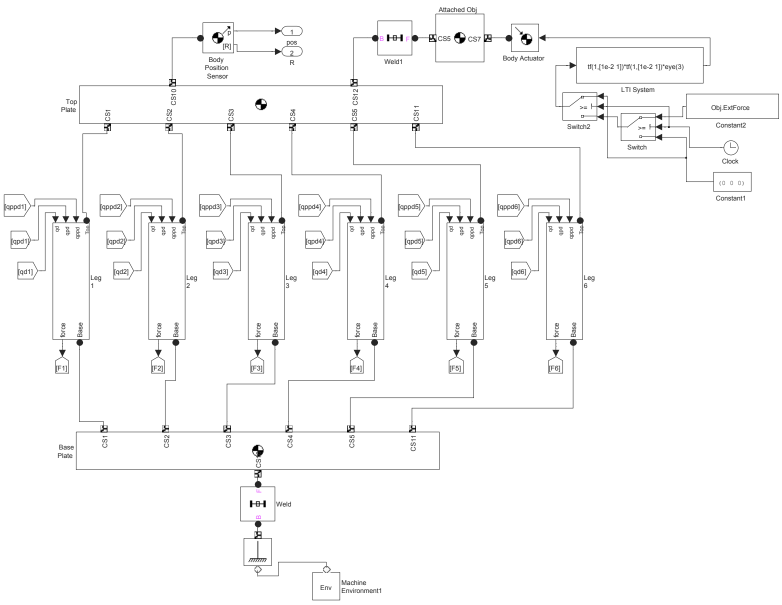
3.2.3 Dynamic simulator for ID computation

The proposed simulator provides a simulation environment to support the design of a Stewart platform-based mechanism for specific applications. A dynamic optimization has been carried out to minimize a cost function that will be defined in Section 3.2.4.1. In particular, the simulator has been developed in Matlab/Simulink environment using the SimMechanics Toolbox and the Global Optimization Toolbox. Given an initial configuration of the Stewart platform in terms of leg attachment points on base and top plates and given two sets of bounds properly defined on the base of the mechanical constraints, a GA is used to solve a non-linear constrained problem. The bounds are defined for each leg attachment point and delimit the area in which the points can be moved during the optimization process. They should be defined to avoid collisions between the legs, actuators and fixtures, to satisfy specified maximum dimensions of the plates or other mechanical constraints.

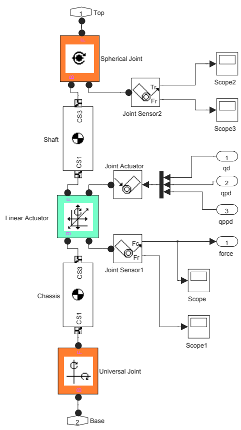
3.2.3.1 Construction of the SimMechanics model

The dynamic model of the parallel mechanism is computed as illustrated below. The base and top plates and all components of the legs have been implemented as rigid bodies with a proper mass and inertia tensor. A leg consists of a linear actuator attached to the base and top plates using a universal joint and a spherical joint, respectively. In detail, an actuator has been implemented in SimMechanics using three elements: a chassis and a shaft, that are rigid bodies, and a prismatic joint that gives a single dof and that connects the chassis to the shaft. Moreover, all the objects have been considered to be of aluminium material, with a mass density of $2700\,\mathrm{kg/m^3}$; the inertia tensors of both top and base plates have been computed considering the cuboids that includes their shapes; the inertia tensors of the chassis and the shaft of the actuators have been computed considering a cylinder properly dimensioned. Alternatively, more accurate inertia tensors and mass values can be estimated using a CAD tool such as Catia.

The ID of the hexapod has been computed by setting the Analysis Mode parameter of the Simulink block scheme as Forward Dynamics [70, 71] rather than as Inverse Dynamics as one would expect. In general, the SimMechanics Inverse Dynamics mode allows to find all the forces on a closed kinematics chain or an open kinematics chain given a model that completely specifies the system's motions. The motivation of such choice is twofold. As well explained on the Mathworks documentation, the ID of a closed kinematics chain can be efficiently computed by setting the SimMechanics Analysis Mode parameter to Kinematics but, in this case, the external forces/torques applied to the body of the hexapod or the work-object can not be specified. Moreover, the motion of all the active and passive joints in terms of position, velocity, and acceleration should be completely specified in advance, and this requires the resolution of the kinematic constrains. Alternatively, the ID can be computed by setting the Analysis Mode parameter to Forward Dynamics at the expense of some performance degradation. In fact, in this case, the motion of the passive joints is computed by the solver by solving the Newton's laws starting from the topology of the bodies connection, the dofs and constraints among dofs, the external forces/-torques applied to the bodies, the mass properties (masses and inertia tensors) of the bodies, the initial condition of all the active joints. Figure 3.5 reports the Simulink block diagram of the Stewart platform. The block scheme on the bottom illustrates the detailed implementation of a leg.

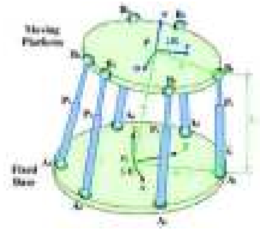


(a) Stewart platform scheme.

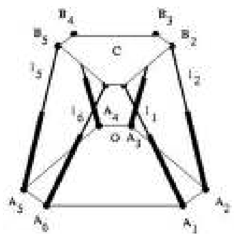


(b) Leg detail.

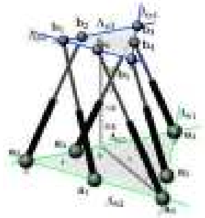
Figure 3.5: Hexapod - SimMechanics block diagram. In cyan active joint (assigned motion); in orange passive joints (computed motion).



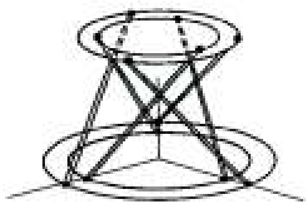
(a) Generic Stewart-Gough platform.



(b) Symmetric Stewart-Gough platform.



(c) Griffis-Duffy platform.



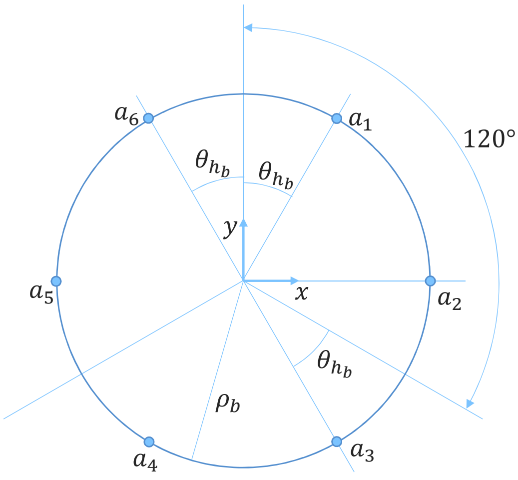
(d) MSP.

Figure 3.6: Hexapod Geometries.

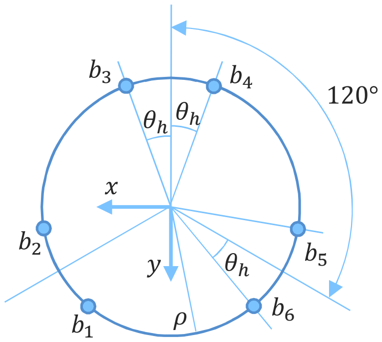
3.2.3.2 Parameterizations of the hexapod geometry

In order to find the best hexapod geometry that should provide the maximum workspace volume and the minimum RMS value of the leg forces exerted by the actuators along a desired trajectory, the optimization process has been carried out by considering symmetric and unsymmetric hexapod geometries. The choice of a geometry affects the time required for the optimization process: the use of a large number of variables allows to optimize the platform minimizing the number of constrains but it increases the computational time. Stewart-Gough geometry, Griffis-Duffy geometry [72], MSP [26] geometry and the more general one-axis geometry [23] have been investigated (see Fig. 3.6). Note that a drawback of the original Stewart platform design is that, due to interference constraints between the legs, the orientations of the legs cannot deviate far from the z axis of the manipulator. Since the static force applied by each leg to the moving platform must act along the axis of the leg, the force capacity in the z direction is considerably higher than in the x-y plane, and the torque capacity about the z axis is limited. Such issue is partially solved in the Griffis-Duffy geometry and in the MSP geometry.

The proposed geometries are briefly recalled in the following.



(a) Base Plate



(b) Top Plate

Figure 3.7: Stewart-Gough Platform 4 Variables Geometry.

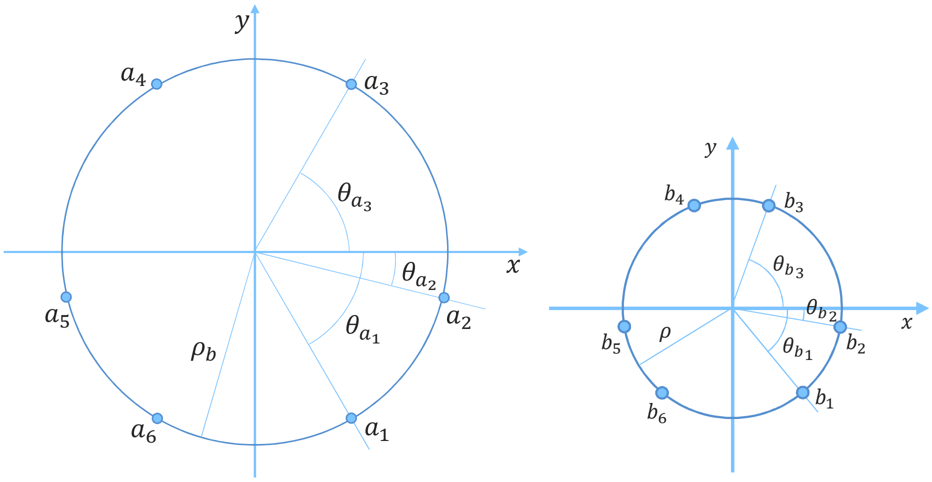
3.2.3.2.1 Stewart-Gough Geometry (4 Variables) – Symmetric geometry (SG4)

The geometry is defined by using two variables for the base plate and two variables for the top plate (see Fig. 3.7). In particular, ρ_b and ρ are the circle radius of the base and top plates, respectively, and θ_{h_b} and θ_h are the half angle between two pairs of joints on the base and on the top plates, respectively. So, the vector of the unknown variables can be written as $\boldsymbol{x} = [\theta_{h_b}\ \theta_h\ \rho_b\ \rho]^T$. The leg attachment point positions $\boldsymbol{a}_i$ and $\tilde{\boldsymbol{b}}_i$, with respect to base frame and mobile frame, can be easily computed in the Cartesian space as a function of the geometry parameters defined above as reported below. Note that only x and y coordinates are considered since the z coordinate is fixed by the mechanical design.

$$
\begin{aligned}
a_{x_1} &= \rho_b \cos\left(\frac{\pi}{2} - \theta_{h_b}\right),\ a_{y_1} = \rho_b \sin\left(\frac{\pi}{2} - \theta_{h_b}\right) \\
a_{x_2} &= \rho_b \cos\left(-\frac{\pi}{6} + \theta_{h_b}\right),\ a_{y_2} = \rho_b \sin\left(-\frac{\pi}{2} + \theta_{h_b}\right) \\
a_{x_3} &= \rho_b \cos\left(-\frac{\pi}{6} - \theta_{h_b}\right),\ a_{y_3} = \rho_b \sin\left(-\frac{\pi}{6} - \theta_{h_b}\right) \\
a_{x_4} &= \rho_b \cos\left(-\frac{5}{6}\pi + \theta_{h_b}\right),\ a_{y_4} = \rho_b \sin\left(-\frac{5}{6}\pi + \theta_{h_b}\right) \\
a_{x_5} &= \rho_b \cos\left(-\frac{5}{6}\pi - \theta_{h_b}\right),\ a_{y_5} = \rho_b \sin\left(-\frac{5}{6}\pi - \theta_{h_b}\right) \\
a_{x_6} &= \rho_b \cos\left(\frac{\pi}{2} + \theta_{h_b}\right),\ a_{y_6} = \rho_b \sin\left(\frac{\pi}{2} + \theta_{h_b}\right) \\
b_{x_1} &= \rho \cos\left(\frac{\pi}{2} - \theta_h\right),\ b_{y_1} = \rho \sin\left(\frac{\pi}{2}\theta_h\right) \\
b_{x_2} &= \rho \cos\left(-\frac{\pi}{6} + \theta_h\right),\ b_{y_2} = \rho \sin\left(-\frac{\pi}{2} + \theta_h\right) \\
b_{x_3} &= \rho \cos\left(-\frac{\pi}{6} - \theta_h\right),\ b_{y_3} = \rho \sin\left(-\frac{\pi}{6} - \theta_h\right) \\
b_{x_4} &= \rho \cos\left(-\frac{5}{6}\pi + \theta_h\right),\ b_{y_4} = \rho \sin\left(-\frac{5}{6}\pi + \theta_h\right) \\
b_{x_5} &= \rho \cos\left(-\frac{5}{6}\pi - \theta_h\right),\ b_{y_5} = \rho \sin\left(-\frac{5}{6}\pi - \theta_h\right) \\
b_{x_6} &= \rho \cos\left(\frac{\pi}{2} + \theta_h\right),\ b_{y_6} = \rho \sin\left(\frac{\pi}{2} + \theta_h\right)
\end{aligned}
\tag{3.11}
$$

3.2.3.2.2 Stewart-Gough Geometry (8 Variables) – Unsymmetric geometry (SG8)

The previous geometry can be modified as in Fig. 3.8 by relaxing some constrains and by defining each leg attachment point position individually. Four variables are used to define the leg attachment points on the base plate and other four variables are used to define the leg attachment points on the top plate. In particular, ρ_b and ρ are the circle radius of the base and top plates, respectively; θ_{a_i} and θ_{b_i} with $i = 1,\ 2,\ 3$ are the angles that define the three joint positions in the half-plane of the positive x axis on the base plate and on the top plate, respectively. The joint positions of the three legs in the left half-plane are calculated by symmetry with respect to the y axis. The vector containing the unknown variables is $\boldsymbol{x} = [\theta_{a_1}\ \theta_{a_2}\ \theta_{a_3}\ \theta_{b_1}\ \theta_{b_2}\ \theta_{b_3}\ \rho_b\ \rho]^T$. The leg attachment point positions, with respect to the base plate frame and top plate frame,



(a) Base Plate (b) Top Plate

Figure 3.8: Stewart-Gough Platform 8 Variables (Unsymmetric) Geometry.

can be computed in the Cartesian space as a function of the parameters just defined.

$$
\begin{aligned}
a_{x_1} &= \rho_b \cos(\theta_{a_1}),\ a_{y_1} = \rho_b \sin(\theta_{a_1}) \\
a_{x_2} &= \rho_b \cos(\theta_{a_2}),\ a_{y_2} = \rho_b \sin(\theta_{a_2}) \\
a_{x_3} &= \rho_b \cos(\theta_{a_3}),\ a_{y_3} = \rho_b \sin(\theta_{a_3}) \\
a_{x_4} &= \rho_b \cos(\pi - \theta_{a_3}),\ a_{y_4} = \rho_b \sin(\pi - \theta_{a_3}) \\
a_{x_5} &= \rho_b \cos(\pi - \theta_{a_2}),\ a_{y_5} = \rho_b \sin(\pi - \theta_{a_2}) \\
a_{x_6} &= \rho_b \cos(\pi - \theta_{a_1}),\ a_{y_6} = \rho_b \sin(\pi - \theta_{a_1}) \\
b_{x_1} &= \rho \cos(\theta_{b_1}),\ b_{y_1} = \rho \sin(\theta_{b_1}) \\
b_{x_2} &= \rho \cos(\theta_{b_2}),\ b_{y_2} = \rho \sin(\theta_{b_2}) \\
b_{x_3} &= \rho \cos(\theta_{b_3}),\ b_{y_3} = \rho \sin(\theta_{b_3}) \\
b_{x_4} &= \rho \cos(\pi - \theta_{b_3}),\ b_{y_4} = \rho \sin(\pi - \theta_{b_3}) \\
b_{x_5} &= \rho \cos(\pi - \theta_{b_2}),\ b_{y_5} = \rho \sin(\pi - \theta_{b_2}) \\
b_{x_6} &= \rho \cos(\pi - \theta_{b_1}),\ b_{y_6} = \rho \sin(\pi - \theta_{b_1})
\end{aligned}
\tag{3.12}
$$

3.2.3.2.3 Generic One-Axis Geometry (12 variables) – Unsymmetric geometry (OA12) The leg attachment points are defined in the Cartesian space (see Fig. 3.9). The geometry parameters are the x and y coordinates a_{x_i}, a_{y_i}, b_{x_i}, b_{x_i} with $i = 1, 2, 3$ of the leg positions in the half-plane of the positive x axis on the base and on the

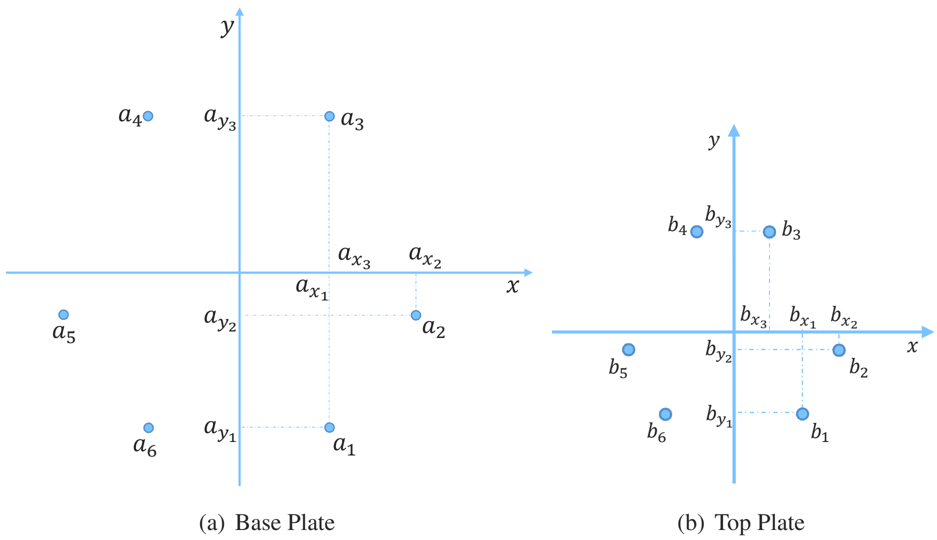
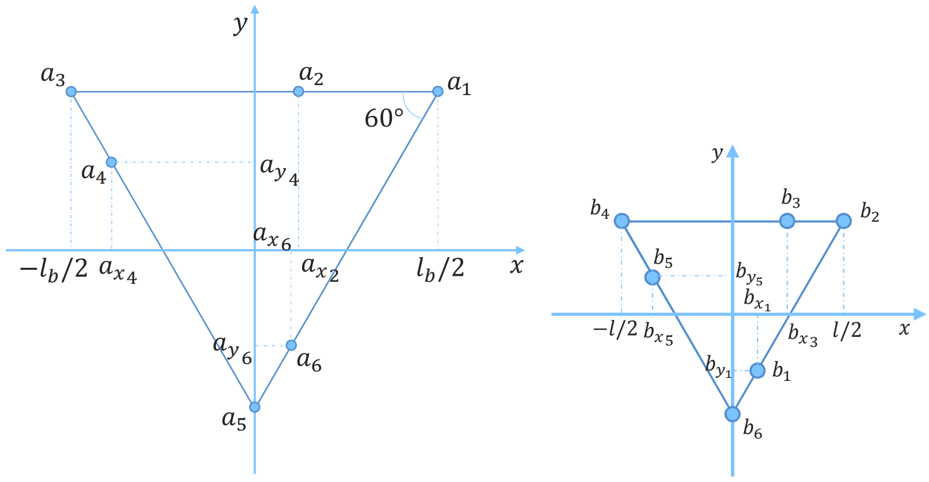


Figure 3.9: Generic One Axis Symmetry (Unsymmetric) Geometry.

top plate with respect to base frame and mobile frame, respectively (12 variables). The joint positions of the three legs in the left half-plane are calculated by symmetry with respect to the y axis. The unknown variable vector can be chosen as $\boldsymbol{x} = [a_{x_1}\ a_{y_1}\ a_{x_2}\ a_{y_2}\ a_{x_3}\ a_{y_3}\ b_{x_1}\ b_{y_1}\ b_{x_2}\ b_{y_2}\ b_{x_3}\ b_{y_3}]^T$.

3.2.3.2.4 Griffis-Duffy Geometry (2+6 variables) – Unsymmetric geometry (GD2+6)

The leg attachment points are defined in the Cartesian space and they are constrained on a triangular shape of known side length (see Fig. 3.10). The variables represent the coordinates a_{x_2}, a_{x_4}, a_{x_6}, b_{x_1}, b_{x_3}, b_{x_5} of the leg attachment points with respect to base plate frame and top plate frame and the sides l_b and l of the base plate and top plate, respectively. The optimal design, in this case, requires a double optimization process because the bounds of the position coordinate a_{x_2}, a_{x_4}, a_{x_6}, b_{x_1}, b_{x_3}, b_{x_5} variables depend on the computation of the side l_b and l variables. The vector of all the

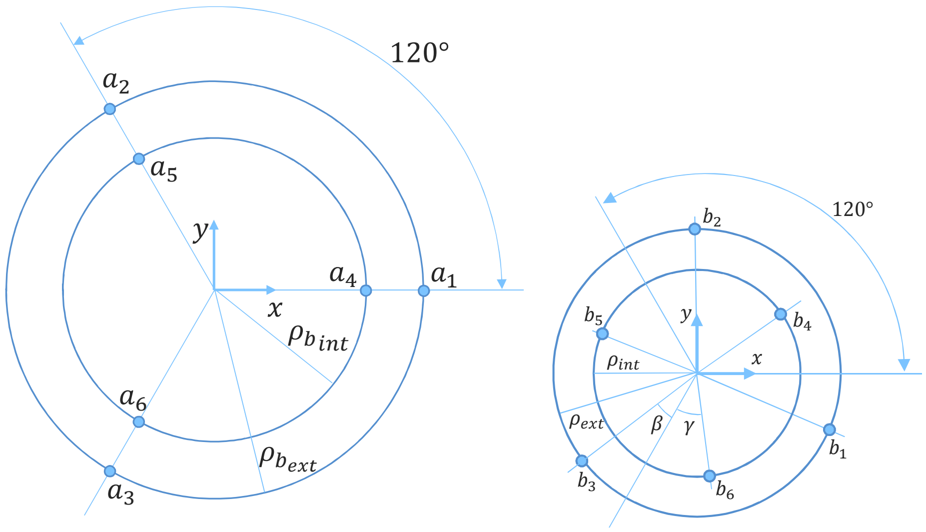


(a) Base Plate (b) Top Plate

Figure 3.10: Griffis-Duffy (Unsymmetric) Geometry.

unknown variables can be written as $\boldsymbol{x} = [l_b\ l\ a_{x_2}\ a_{x_4}\ a_{x_6}\ b_{x_1}\ b_{x_3}\ b_{x_5}]^T$.

$$
\begin{aligned}
a_{x_1} &= \frac{l_b}{2},\ a_{y_1} = \frac{l_b}{4}\sqrt{3} \\
a_{x_2} &= a_{x_1},\ a_{y_2} = \frac{l_b}{4}\sqrt{3} \\
a_{x_3} &= -\frac{l_b}{2},\ a_{y_3} = \frac{l_b}{4}\sqrt{3} \\
a_{x_4} &= a_{x_4},\ a_{y_4} = -\sqrt{3}\left(\frac{l_b}{4} + a_{x_4}\right) \\
a_{x_5} &= 0,\ a_{y_5} = -\frac{l_b}{4}\sqrt{3} \\
a_{x_6} &= a_{x_6},\ a_{y_6} = \sqrt{3}\left(a_{x_6} - \frac{l_b}{4}\right) \\
b_{x_1} &= b_{x_1},\ b_{y_1} = \sqrt{3}\left(b_{x_1} - \frac{l}{4}\right) \\
b_{x_2} &= \frac{l}{2},\ b_{y_2} = \frac{l}{4}\sqrt{3} \\
b_{x_3} &= b_{x_3},\ b_{y_3} = \frac{l}{4}\sqrt{3} \\
b_{x_4} &= -\frac{l}{2},\ b_{y_4} = \frac{l}{4}\sqrt{3} \\
b_{x_5} &= b_{x_5},\ b_{y_5} = -\sqrt{3}\left(\frac{l}{4} + b_{x_5}\right) \\
b_{x_6} &= 0,\ b_{y_6} = -\frac{l}{4}\sqrt{3}
\end{aligned}
\tag{3.13}
$$



(a) Base Plate (b) Top Plate

Figure 3.11: MSP (Unsymmetric) Geometry.

3.2.3.2.5 MSP (6 variables) – Unsymmetric geometry (MSP6) The leg attachment points are placed on two circles at both base plate and top plate: three of the six legs are positioned on an inner circle both at the base and mobile platforms, and the other three legs on an outer (concentric) circle (Fig. 3.11). The leg attachment points on the base plate are fixed and positioned at 120 degrees from each other. So, two variables are used to define the leg attachment points on the base plate and four variables are used to define the leg attachment points on the top plate. In particular, $\rho_{b_{int}}$, $\rho_{b_{ext}}$, ρ_{int}, and ρ_{ext} are the inner and outer circle radius of the base and top plates, respectively; β and γ are the angles that define the deviation of the leg attachment points on the mobile plate from 0 - 120 - 240 degrees. The unknown variable vector is $\boldsymbol{x} = [\beta\ \gamma\ \rho_{b_{ext}}\ \rho_{b_{int}}\ \rho_{ext}\ \rho_{int}]^T$. The leg attachment point positions, with respect to the base plate frame and top plate frame, can be computed in the Cartesian space as a function of the defined parameters as reported below.

$$
\begin{aligned}
a_{x_1} &= \rho_{b_{ext}} \cos(0),\ a_{y_1} = \rho_{b_{ext}} \sin(0) \\
a_{x_2} &= \rho_{b_{ext}} \cos(\frac{2\pi}{3}),\ a_{y_2} = \rho_{b_{ext}} \sin(\frac{2\pi}{3}) \\
a_{x_3} &= \rho_{b_{ext}} \cos(\frac{4\pi}{3}),\ a_{y_3} = \rho_{b_{ext}} \sin(\frac{2\pi}{3}) \\
a_{x_4} &= \rho_{b_{int}} \cos(0),\ a_{y_4} = \rho_{b_{int}} sin(0) \\
a_{x_5} &= \rho_{b_{int}} \cos(\frac{2\pi}{3}),\ a_{y_5} = \rho_{b_{int}} \sin(\frac{2\pi}{3}) \\
a_{x_6} &= \rho_{b_{int}} \cos(\frac{4\pi}{3}),\ a_{y_6} = \rho_{b_{int}} \sin(\frac{4\pi}{3}) \\
b_{x_1} &= \rho_{ext} \cos(-\beta),\ b_{y_1} = \rho_{ext} sin(-\beta) \\
b_{x_2} &= \rho_{ext} \cos(\frac{2\pi}{3} - \beta),\ b_{y_2} = \rho_{ext} \sin(\frac{2\pi}{3} - \beta) \\
b_{x_3} &= \rho_{ext} \cos(\frac{4\pi}{3} - \beta),\ b_{y_3} = \rho_{ext} \sin(\frac{4\pi}{3} - \beta) \\
b_{x_4} &= \rho_{int} \cos(\gamma),\ b_{y_4} = \rho_{int} \sin(\gamma) \\
b_{x_5} &= \rho_{int} \cos(\frac{2\pi}{3} + \gamma),\ b_{y_5} = \rho_{int} \sin(\frac{2\pi}{3} + \gamma) \\
b_{x_6} &= \rho_{int} \cos(\frac{4\pi}{3} + \gamma),\ b_{y_6} = \rho_{int} \sin(\frac{4\pi}{3} + \gamma)
\end{aligned}
\tag{3.14}
$$

3.2.4 The optimization algorithm

An optimization algorithm has been used to combine two different objectives by properly defining a cost function to minimize. In order to minimize the maximum leg force value and to equally distribute among the legs the forces exerted by the linear actuators during a positioning and/or machining task, the maximum RMS value of the forces has been selected as a metric. A second objective has been taken into account to maximize (or do not penalize) the robot workspace volume.

3.2.4.1 Cost Function

The cost function F can be defined as in (3.15), where the parameters k_1 and k_2 determine the weight of each objective in F.

$$F = k_1 W_1 + k_2 W_2 \tag{3.15}$$

The first contribution W_1 takes into account the leg forces necessary to follow a given position and orientation trajectory of the mobile plate, that depends on the specific application, e.g. a part positioning during an assembly process, and withstanding of external forces applied to the top plate, e.g. during the handling of a part subject to a machining process. The contribution W_2 takes into account the workspace volume. A possible choice of the two terms W_1 and W_2 can be as in (3.16), where $\boldsymbol{\tau}(k)$ is the vector of the leg forces at the kth time instant of the task execution and V_W is the volume of the robot workspace for a given design.

$$W_1 = \frac{1}{N}\sum_{k=1}^{N} \|\boldsymbol{\tau}(k)\|\,, \; W_2 = -V_W \tag{3.16}$$

The leg forces are computed by solving the inverse dynamics of the Stewart platform using the dynamic model described in Section 3.2.3.1, using as input the leg positions, velocities and accelerations computed by solving the IK problem given the desired top plate trajectory. The platform workspace volume is computed by considering the geometrical approach proposed by [73]. In particular, the considered workspace is the positional workspace (or fixed-orientation workspace), computed by maintaining the top plate orientation equal to $\varphi_p = \theta_p = \psi_p = 0$.
The choice of such W_1 allows to reduce the maximum value of the forces required by the actuators and it also allows to equalize the mean value of the six forces along the entire considered trajectory. In order to select the most suitable optimization algorithm for the proposed application, the performances of the Genetic Algorithm, the Sequential Quadratic Programming algorithm, the MultiStart algorithm and the GlobalSearch algorithm available in the Matlab Optimization Toolbox, have been compared.

3.2.4.2 Genetic Algorithm

After a number of simulation trials, the optimization parameters have been set to:

- Population Size = Number of variables × 15
- Number of Generations = 50

The optimization process stops if the maximum number of iterations is reached or if in two generations the cumulative change in the fitness function value is less than the termination tolerance value set to be 10^{-6}. The optimization parameters have been chosen so that the optimization algorithm tends to stop for the termination tolerance

Joint positions	x **[*m*]**	y **[*m*]**	z **[*m*]**
$\boldsymbol{a_1}$	-0.0850	-0.1472	0.0400
$\boldsymbol{a_2}$	-0.1700	0	0.0400
$\boldsymbol{a_3}$	-0.0850	0.1472	0.0400
$\boldsymbol{a_4}$	0.0850	0.1472	0.0400
$\boldsymbol{a_5}$	0.1700	0	0.0400
$\boldsymbol{a_6}$	0.0850	-0.1472	0.0400
$\boldsymbol{b_1}$	-0.0418	-0.0498	0.4940
$\boldsymbol{b_2}$	-0.0640	-0.0113	0.4940
$\boldsymbol{b_3}$	-0.0222	0.0611	0.4940
$\boldsymbol{b_4}$	0.0222	0.0611	0.4940
$\boldsymbol{b_5}$	0.0640	-0.0113	0.4940
$\boldsymbol{b_6}$	0.0418	-0.0498	0.4940

Table 3.1: Initial joint positions.

Variable	***Min***	***Max***
ρ_b [m]	0.1500	0.3500
ρ [m]	0.0600	0.2200
θ_{h_b} [deg]	11	49
θ_h [deg]	17	43

Table 3.2: SG4 - variable boundaries.

value criteria ensuring that the GA returns the global minimum (otherwise, it is not guaranteed that the GA founds the best minimum).
For each geometry illustrated in Section 3.2.3.2, in order to avoid collisions between the legs, actuators and fixtures, sets of variable boundaries have been properly defined. Moreover, such boundaries have been defined to satisfy other mechanical design constrains, e.g., maximum dimensions of both base and top plates (0.3500 m and 0.1500 m of radius, respectively), introduced assuming a limited space available for the robot installation. The variable boundaries defined for the considered geometries are reported in Tables 3.2, 3.3, 3.4, 3.5 and 3.6. Moreover, the initial joint positions at the instant time $t = 0$ are reported in Table 3.1.

The proposed GA has been compared to other methods, i.e., the deterministic Sequential Quadratic Programming (SQP) algorithm selected as optimization algorithm in [49] for the optimal design of parallel machines and the stochastic MultiStart and GlobalSearch algorithms available in the Matlab Optimization Toolbox well known for looking for global or multiple minima. The SQP algorithm is a gradient-based

Variable	***Min***	***Max***
ρ_b [m]	0.1500	0.3500
ρ [m]	0.0600	0.2200
θ_{a_1} [deg]	$-90° + \theta_{Bound_b}$	$-30° - \theta_{Bound_b}$
θ_{a_2} [deg]	$-30° + \theta_{Bound_b}$	$30° - \theta_{Bound_b}$
θ_{a_3} [deg]	$90° + \theta_{Bound_b}$	$90° - \theta_{Bound_b}$
θ_{b_1} [deg]	$-90° + \theta_{Bound}$	$-30° - \theta_{Bound}$
θ_{b_2} [deg]	$-30° + \theta_{Bound}$	$30° - \theta_{Bound}$
θ_{b_3} [deg]	$90° + \theta_{Bound}$	$90° - \theta_{Bound}$
θ_{Bound_b} [deg]	11°	
θ_{Bound} [deg]	18°	

Table 3.3: SG8 - variable boundaries.

method for solving constrained nonlinear optimization problems. The MultiStart is an easy and straightforward algorithm that initiates a local solver from a set of starting points and then creates a vector containing found local minima, returning the best of these points as the estimated global minimum. The GlobalSearch works similarly to the MultiStart but the starting points are generated by a scatter-search mechanism in a more complex way. The algorithm then tries to analyze these starting points and discards points that are unlikely to generate a better minimum than the best minimum found so far. The performances of the Multistart algorithm and of the GlobalSearch algorithm have been evaluated by using an Active-Set (AS) algorithm and a SQP algorithm. In other words, from each starting point, an AS algorithm or a SQP algorithm have been executed to find the nearest local minima. Figure 3.12 shows a sketch of the GlobalSearch and MultiStart algorithms [74].

Table 3.7 shows the comparison of the proposed algorithms in the optimization of the SG4 design for the case study I illustrated in Sec. 3.2.5.1. The optimization has been carried out by considering $k_1 = 0.1$ and $k_2 = 100$. The algorithms start from the same $\boldsymbol{x_0}$ and they have been evaluated by considering as stopping criteria the minimum function tolerance value set to 10^{-6}. Moreover, the GA algorithm and the SQP algorithm have been tested setting the maximum number of function evaluation (f_{counts}) to 3000, while, the number of the starting points of the MultiStart algorithm has been set to 30. Finally, the GlobalSearch algorithm has been tested by using the parameters NumTrialPoints[10] and NumStageOnePoints[11] set to 10000

[10]NumTrialPoints is the number of potential start points to examine in addition to $\boldsymbol{x_0}$.

[11]NumStageOnePoints is the number of points in which the cost function is evaluated. Only in the point with the best score the optimization is carried out. The set of NumStageOnePoints trial points is

Variable	***Min***	***Max***
a_{x_1} [m]	0.0050	0.2450
a_{y_1} [m]	-0.3472	-0.1072
a_{x_2} [m]	0.0700	0.3500
a_{y_2} [m]	-0.1000	0.1000
a_{x_3} [m]	0.0050	0.2450
a_{y_3} [m]	0.1072	0.3472
b_{x_1} [m]	0.0018	0.1818
b_{y_1} [m]	-0.2198	-0.0398
b_{x_2} [m]	0.0240	0.2200
b_{y_2} [m]	-0.0363	0.0487
b_{x_3} [m]	0.0022	0.1722
b_{y_3} [m]	0.0511	0.2111

Table 3.4: OA12 - variable boundaries.

and 20, respectively. All the proposed setting parameters have been adjusted in successive simulations and they have been chosen so that the algorithms provided the best result. The presented analysis shows that the GA reaches better results than the other algorithms obtaining a smaller value of the cost function F. In fact, although the convergence times are not comparable because the SQP algorithm converges in a number of function evaluation counts less than the other ones, it sticks in local minima and, so, it provides a worse design in terms of minimum cost function value. The MultiStart algorithm returns values similar to the GA but a great number of starting points and a larger number of cost function evaluation, and then more time, are required compared to the GA to converge to an optimal value. Finally, the GlobalSearch algorithm does not seem to be suitable for the proposed application given the unsatisfactory results obtained during the optimization process because it sticks in local minima. In conclusion, the GA algorithm appears to be the best choice for the proposed application. Moreover, the use of the AS algorithm, in association with the stochastic algorithm MultiStart, is recommended over the SQP algorithm if a sufficient computational power is not available. Note that the Griffis-Duffy geometry requires a double GA process due to the fact that the leg position variable boundaries have to be computed by using the size of the side of both the base and top plates as described in Section 3.2.3.2.4. The optimization time, in this case, increases necessarily. The algorithm scheme is reported in Fig. 3.13. Only one GA is needed for all the other geometries, instead.

removed from the list of points to examine.

Variable	*Min*	*Max*
l_b [m]	0.30	0.70
l [m]	0.20	0.44
a_{x_2} [m]	$-\frac{l_b}{2} + p_{Bound_b}$	$\frac{l_b}{2} - p_{Bound_b}$
a_{x_4} [m]	$-\frac{l_b}{2} + p_{Bound_b} cos(\frac{\pi}{3})$	$-p_{Bound_b} sin(\frac{\pi}{6})$
a_{x_6} [m]	$p_{Bound_b} sin(\frac{\pi}{6})$	$\frac{l_b}{2} - p_{Bound_b} cos(\frac{\pi}{3})$
b_{x_1} [m]	$p_{Bound} sin(\frac{\pi}{6})$	$\frac{l}{2} - p_{Bound} cos(\frac{\pi}{3})$
b_{x_3} [m]	$-\frac{l}{2} + p_{Bound}$	$\frac{l}{2} - p_{Bound}$
b_{x_5} [m]	$\frac{l}{2} + p_{Bound} cos(\frac{\pi}{3})$	$-p_{Bound} sin(\frac{\pi}{6})$
p_{Bound_b} [m]	0.04	
p_{Bound} [m]	0.03	

Table 3.5: GD2+6 - variable boundaries.

Variable	*Min*	*Max*
$\rho_{b_{int}}$ [m]	0.05	0.18
$\rho_{b_{ext}}$ [m]	0.24	0.35
ρ_{int} [m]	0.05	0.12
ρ_{ext} [m]	0.16	0.22
β [deg]	0	60
γ [deg]	0	60

Table 3.6: MSP6 - variable boundaries.

3.2.5 Results and discussion

This section describes the results of the optimization process. The optimization of the Stewart platform has been carried out by considering the geometries illustrated in Section 3.2.3.2 for the case studies described in the Section 3.2.5.1. Moreover, in order to compute the workspace volume of the Stewart platform, the motor strokes, and then, the minimum and maximum leg lengths are required. A motor stroke of 0.2500 m is considered in the proposed simulations.

3.2.5.1 Case study definition

The optimization process has been carried out by considering two different case studies. A positioning task has been considered in the first case study, in which the hexapod moves a work-object along a desired trajectory. The second case study consists of two phases: in the first phase the hexapod performs a positioning task as in the first case study; in the second phase the hexapod holds the work-object during a

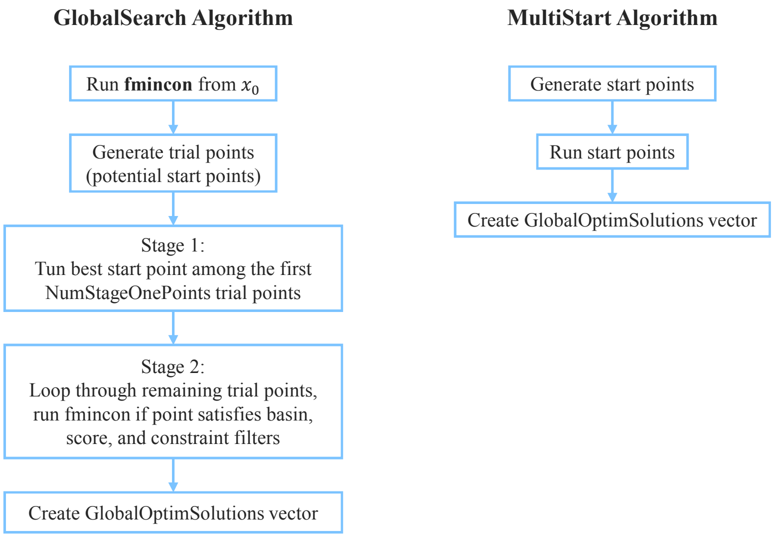


Figure 3.12: MultiStart and GlobalSearch algorithm overview.

Algorithm	F
GA (2280 f_{counts})	-41.2765
SQP (137 f_{counts})	-16.7716
MultiStart AS (3281 f_{counts})	-40.5940
MultiStart SQP (3325 f_{counts})	-38.5824
GlobalSearch AS (147 f_{counts})	-26.2375
GlobalSearch SQP (137 f_{counts})	-16.7716

Table 3.7: Comparison of the optimization algorithms.

manufacturing process, e.g., a drilling process. In detail, in the positioning task the hexapod moves a work-object of 50 kg weight in a desired position along a desired trajectory. The work-object is attached to the mobile plate through a weld joint simulating a tight grasp. The trajectory is planned in the Cartesian space and it is defined by imposing the initial and final pose of the mobile frame (positioned in the COG point of the top plate and oriented to be parallel to the base frame when the top plate is parallel to the base plate). The robot moves from the initial configuration $\boldsymbol{x}_i$ to the final configuration $\boldsymbol{x}_f$ reported in (3.17) (position expressed in meter and orientation in degrees) in a given time $t = 5$ s. The motion timing law is of third order polynomial type and it is implemented using the “spline” command in Matlab. Using the IK algorithm, the leg lengths have been computed and the inverse dynamics of the

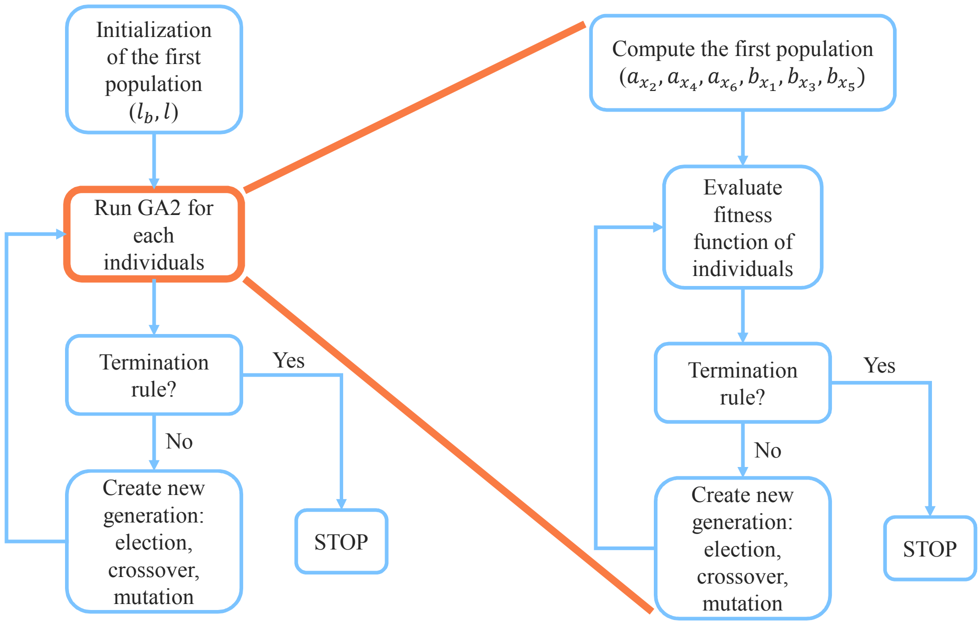


Figure 3.13: Genetic algorithms in Griffis-Duffy geometry.

hexapod has been solved by controlling the position of each linear actuator.

$$\begin{aligned} \boldsymbol{x}_i &= [0,\ 0,\ 0.5125,\ 0,\ 0,\ 0]^T \\ \boldsymbol{x}_f &= [0.0064,\ -0.0043,\ 0.6930,\ 8,\ -3,\ -1]^T \end{aligned} \tag{3.17}$$

In the second case study, when the mobile plate reaches the final pose, the manufacturing task starts. The hexapod holds the work-object fixed in the final position and a force of 500 N is applied along the x direction at $t = 6$ s for 2 seconds on a given point $\boldsymbol{p}_0$ simulating a drilling process. This point is reported in eq. (3.18), where $WorkObj_{COG_f}$ denotes the final position of the work-object (see Fig. 3.14).

$$\boldsymbol{p}_0 = WorkObj_{COG_f} + [0.1000\ 0\ 0]^T \tag{3.18}$$

Moreover, the leg force estimation are also compared by defining a second trajectory:

$$\begin{aligned} \boldsymbol{x}_i &= [0,\ 0,\ 0.5125,\ 0,\ 0,\ 0]^T \\ \boldsymbol{x}_{f_2} &= [-0.0053,\ -0.0137,\ 0.6938,\ -8,\ 5,\ 3]^T. \end{aligned} \tag{3.19}$$

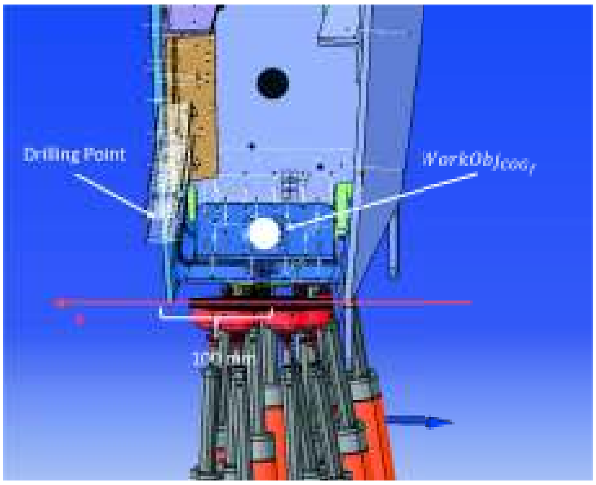


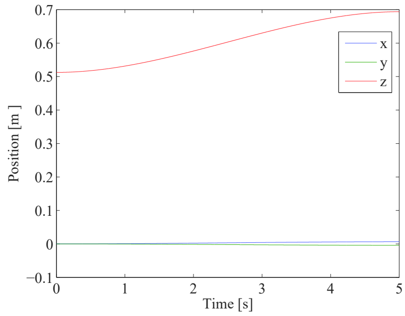
Figure 3.14: Case study 2: drilling point.

In both the trajectories a z displacement of about 181 mm is considered; different orientations are considered, instead. Figure 3.15 shows the trajectories of the hexapod top plate where the position is expressed in meter and the orientation is expressed in degrees.

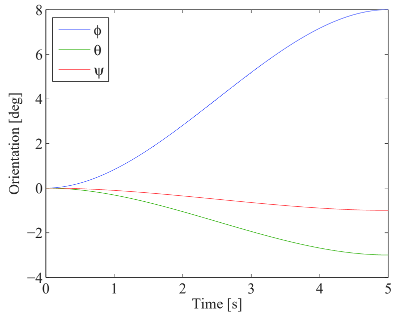
The results of the optimization process are reported below. The analysis has been carried out by changing the values of the weight parameters k_1 and k_2 in the cost function (3.15) and by considering a set of initial joint positions obtained from a purely mechanical design accomplished by reducing the overall dimensions of the Stewart platform and its incumbrance.

3.2.5.2 Results

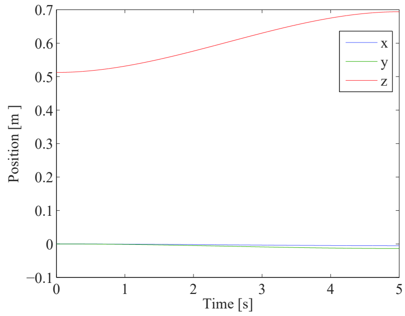
The simulations have been carried out by considering two sets of values of the parameters k_1 and k_2, i.e., $k_1 = 0.1$ - $k_2 = 100$ and $k_1 = 0.1$ - $k_2 = 10$. The first set has been considered to optimize the hexapod design with the aim of increasing the robot workspace; the second one, instead, has been considered in order to obtain an optimized design which decreases the leg forces exerted by the robot during the assigned task, but still taking into account the workspace volume. In fact, a higher emphasis is given to the leg force contribution W_1 in the cost function F during the optimization process by decreasing the weight parameter k_2 and, vice versa, a higher emphasis is given to the leg force contribution W_2 in the cost function F by increasing the same parameter k_2. All the geometries described in Section 3.2.3.2 have been analyzed by



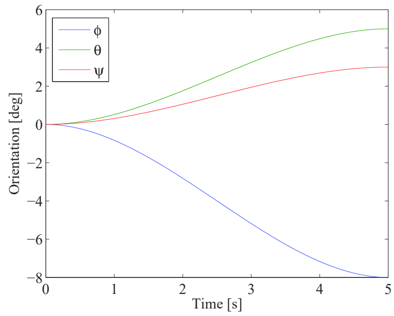
(a) Trajectory 1 - Position



(b) Trajectory 1 - Orientation



(c) Trajectory 2 - Position



(d) Trajectory 2 - Orientation

Figure 3.15: Trajectories.

considering the two sets of parameters. Figure A.1(a) shows the initial workspace of the Stewart platform and Fig. A.1(b)(c) show the leg forces required to execute the tasks described above. The presented analyses are summarized in Table 3.8.

The importance of the trajectory in the optimization phase can be understood by analyzing the evolution of the leg forces considering different trajectories as shown in Figures A.2(d) A.2(e) and Figures A.3(d) A.3(e). In the proposed simulation, the SG4 geometry has been optimized along the first trajectory (Section 3.2.5.1) by considering the case study I. Let consider the obtained design. The leg forces estimated along the second trajectory result to be quite different in terms of magnitude and time evolution.

Similarly, the specification of the task strongly affects the optimization process. Let consider the previous optimized design executing the task specified in the case study II. The estimated leg forces result very high in the manufacturing process although they did not change during the positioning phase as shown in Figure A.2(f) and Figure A.3(f). A more suitable design in terms of leg force values can be obtained

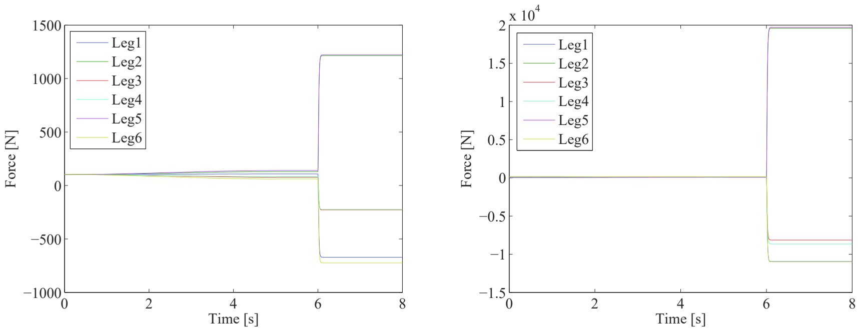
	Max pos. force [N]	Max man. force [N]	V_W [m^3]	Figure
Initial Design	194	2650	0.200860	A.1
SG4 - Case I: k_2 = 100	230	10400	0.52848	A.2
SG4 - Case I: k_2 = 10	143	3965	0.48072	A.3
SG4 - Case II: k_2 = 100	116	1210	0.270140	A.12
SG4 - Case II: k_2 = 10	167	535	0.025509	A.13
SG8 - Case I: k_2 = 100	226	-	0.520960	A.4
SG8 - Case I: k_2 = 10	156	-	0.476040	A.5
SG8 - Case II: k_2 = 100	149	768	0.193180	A.14
SG8 - Case II: k_2 = 10	162	460	0.026268	A.15
OA12 - Case I: k_2 = 100	162	-	0.504220	A.6
OA12 - Case I: k_2 = 10	141	-	0.459170	A.7
OA12 - Case II: k_2 = 100	156	1037	0.326830	A.16
OA12 - Case II: k_2 = 10	144	489	0.032568	A.17
GD2+6 - Case I: k_2 = 100	435	-	0.231880	A.8
GD2+6 - Case I: k_2 = 10	343	-	0.0023545	A.9
GD2+6 - Case II: k_2 = 100	350	748	0.012091	A.18
GD2+6 - Case II: k_2 = 10	360	658	0.0039947	A.19
MSP6 - Case I: k_2 = 100	212	-	0.483340	A.10
MSP6 - Case I: k_2 = 10	157	-	0.435040	A.11
MSP6 - Case II: k_2 = 100	117	1495	0.250080	A.20
MSP6 - Case II: k_2 = 10	141	700	0.030380	A.21

Table 3.8: Simulation results

by considering the manufacturing task in the optimization process. Figure A.12(d) and Figure A.13(d) show that an ad-hoc design provides better results than the other one reducing the maximum force value of about 55% during the manufacturing task. (Note: the figures of the simulations are reported in Appendix A).

3.2.5.3 Discussion

Adjusting the weight parameters k_1 and k_2 the presented tool permits to obtain an optimized solution tailored to the specific application decreasing significatively the forces exerted by the linear actuators during the task and/or increasing the robot workspace volume. In fact, the reported simulations show that by decreasing the k_2 value, the leg forces and the volume workspace decrease accordingly, or, vice versa, a larger workspace can be obtained by choosing a k_2 larger than k_1, e.g. SG4-a, although an higher maximum leg force is achieved. Moreover, given the initial



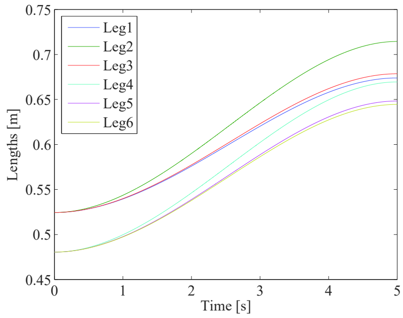
(a) SG4-a design.

(b) OA12-a design.

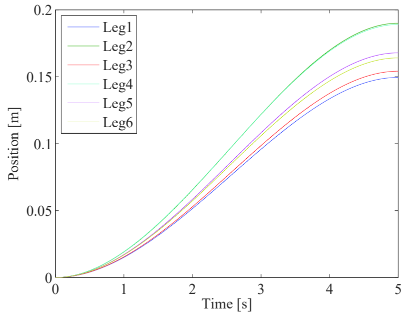
Figure 3.16: Estimated leg forces: external force applied along $-y$ axis.

design in Fig. A.1 that requires a maximum leg force value of about 2650 N and allows to obtain a $V_W = 0.20086\,\mathrm{m}^3$, the presented optimized designs surely provide a smaller maximum leg force (up to 85% reduction of the maximum force in the OA12 geometry). This results into a more accurate and cost-effective choice of the mechanical components of the platform. In other words, the unsymmetric geometries, such as SG8 or OA12, provide a smaller force value than the symmetric ones, but a symmetric geometry could better balance external forces applied along directions not considered in the optimization process. In order to clarify this aspect, let consider the SG4-a and OA12-a optimized designs obtained considering the case study II. By simulating a different task in which an external force of 500 N is applied along $-y$ axis the SG4 geometry requires a smaller force value in the manufacturing process (1222 N compared to the 1210 N estimated by considering the case study II) than the OA12 geometry (1972 N compared to the 1037 N estimated by considering the case study II) as shown in Fig. 3.16. So, a symmetric design may prove more reliable than the unsymmetric one if used in a generic task not considered in the optimization process.

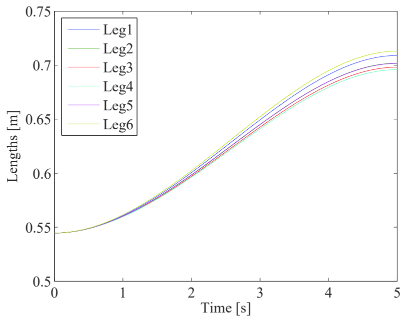
Finally, it is important to ensure that the considered trajectory is inside the workspace of the optimized design. To this aim, it is useful to check that the joint trajectory ranges inside the leg length boundary by computing the inverse kinematics known the Cartesian trajectory. A too small workspace could violate that criteria. Figure 3.17 shows the leg lengths and the actuator positions for the MSP6-b design (a)(b) and for the SG4-b design (c)(d) optimized for the case study II along the trajectory 1. Figure 3.18 shows the leg lengths (a) and the actuator positions (b) for the GD2+6-b design optimized for the case study I, instead. Note that the GD2+6 geometry re-



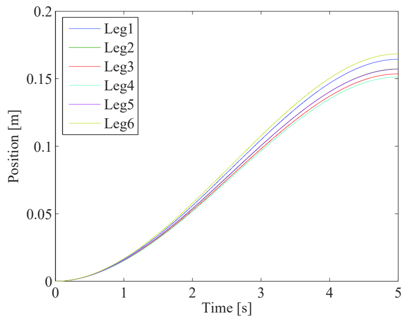
(a) Leg lengths.



(b) Joint positions.



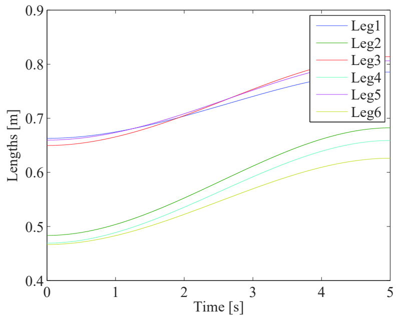
(c) Leg lengths.



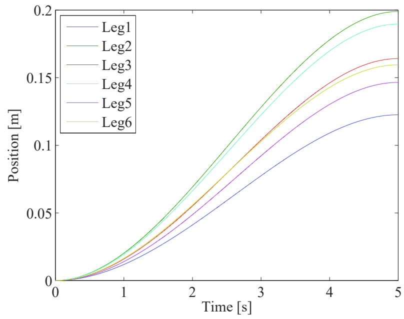
(d) Joint positions.

Figure 3.17: MSP6-b and SG4-b case study II: joint trajectory.

quires longer legs then the other ones although the actuator positions range within the imposed boundaries.



(a) Leg lengths.



(b) Joint positions.

Figure 3.18: GD2+6-b case study I: joint trajectory.

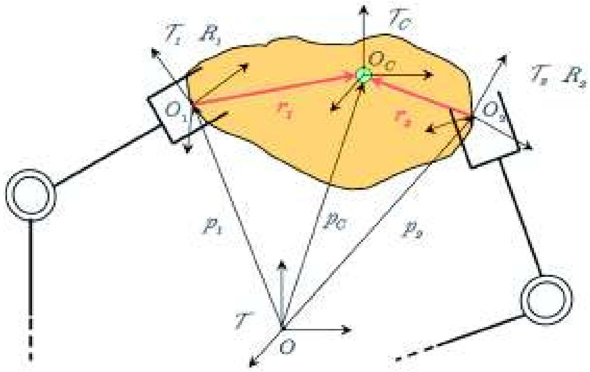


Figure 3.19: Grasp Geometry.

3.3 Coordinated Motion

In the case of heavy and large work-objects, assembly of multiple parts, and handling of large, articulated or flexible objects, e.g., the spar positioning task presented in the previous sections, two or more platforms are required to achieve the common goal. So, given two or more devices rigidly grasping a part, it is needed to coordinate their motion in order to bring the part to the correct location while avoiding, at the same time, internal stresses applied to the object. Internal stresses on the object could damage the object attached to the manipulators and they are due, in the considered applications, to manufacturing tolerances on the components, to the non-coordinated motion of the legs of a single robot or to the non-coordinated motion of the end effector of the robots which keep the same part.

Referring to Fig. 3.19, in which O_c is the object reference frame origin positioned in its center of gravity, O_1 and O_2 are the reference frame system origins positioned in the contact points, O is the global frame origin, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the vectors that connect the contact points O_1 and O_2 to the center of gravity point O_c, respectively, let define the Grasp Matrix $\boldsymbol{W}$ as in eq. (3.29), where $\boldsymbol{S}(\boldsymbol{r})$ is the skew symmetric matrix defined in eq. (3.21).

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_1) & \boldsymbol{I}_3 & \boldsymbol{S}(\boldsymbol{r}_2) & \boldsymbol{I}_3 \end{bmatrix} \tag{3.20}$$

$$\boldsymbol{S}(\boldsymbol{r}) = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{3.21}$$

The mapping between the contact forces $\boldsymbol{h}_e$ and the forces exerted by the manipulators $\boldsymbol{h}_i$ on the object contributing to its motion can be expressed using the grasp matrix as reported in eq. (3.22).

$$\boldsymbol{h}_e = \boldsymbol{W}\boldsymbol{h} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_1) & \boldsymbol{I}_3 & \boldsymbol{S}(\boldsymbol{r}_2) & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{bmatrix} \tag{3.22}$$

Now, defining the internal forces, also called interaction forces or squeeze forces, as the forces lying in the null-space of the grasp matrix $\boldsymbol{W}$ and the equilibrating forces, also called manipulation forces, as the forces that lie in the range-space of $\boldsymbol{W}$, the grasp forces $\boldsymbol{h}$ can be decomposed as in (3.23), where $\boldsymbol{V}$ is the matrix spanning the null-space of $\boldsymbol{W}$ and $\boldsymbol{W}^\dagger$ is defined as in eq. (3.24).

$$\boldsymbol{h} = \boldsymbol{W}^\dagger \boldsymbol{h}_e + \boldsymbol{V}\boldsymbol{h}_i = \boldsymbol{W}^\dagger \boldsymbol{W}\boldsymbol{h} + \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{h} \tag{3.23}$$

$$\boldsymbol{W}^\dagger = \begin{bmatrix} \frac{1}{2}\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\frac{1}{2}\boldsymbol{S}(\boldsymbol{r}_1) & \frac{1}{2}\boldsymbol{I}_3 \\ \frac{1}{2}\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\frac{1}{2}\boldsymbol{S}(\boldsymbol{r}_2) & \frac{1}{2}\boldsymbol{I}_3 \end{bmatrix} \tag{3.24}$$

So, the internal forces or internal stresses $\boldsymbol{h}_I$ can be computed as in (3.25).

$$\boldsymbol{h}_I = \boldsymbol{V}\boldsymbol{V}^\dagger \boldsymbol{h} \tag{3.25}$$

Generalizing to the case of n contact points, the grasp matrix $\boldsymbol{W}$ can be written as in (3.26) and its pseudo-inverse $\boldsymbol{W}^\dagger$ as in (3.27).

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \cdots & \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_1) & \boldsymbol{I}_3 & \cdots & \boldsymbol{S}(\boldsymbol{r}_n) & \boldsymbol{I}_3 \end{bmatrix} \tag{3.26}$$

$$\boldsymbol{W}^\dagger = \begin{bmatrix} \frac{1}{n}\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\frac{1}{n}\boldsymbol{S}(\boldsymbol{r}_1) & \frac{1}{n}\boldsymbol{I}_3 \\ \vdots & \vdots \\ \frac{1}{n}\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\frac{1}{n}\boldsymbol{S}(\boldsymbol{r}_n) & \frac{1}{n}\boldsymbol{I}_3 \end{bmatrix} \tag{3.27}$$

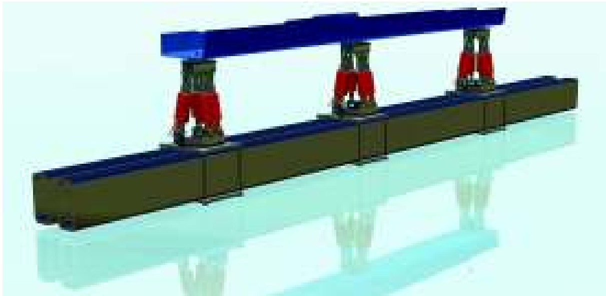


Figure 3.20: Cooperative part positioning.

3.3.1 Simulations

The simulations and the leg force estimation in case of cooperative manipulators and coordinated motion have been carried out in Simulink environment by using the SimMechanics toolbox by computing the forward dynamics of the robots and controlling each linear actuator of the robots in force mode. In this case study, in fact, the ID of the robots has not been calculated directly as in the case of single platform as reported in Section 3.2.3.1 because the SimMechanics toolbox imposes severe limitations in the case of redundant systems (the motion is overdetermined by redundant driver(s) applied to a joint).
Computed the spar trajectory, the trajectory of the end-effector frame of each attached platform has been obtained and, so, through the IK, the trajectory of the robot joints has been computed. The input force value of the joints has been calculated with a PID controller whose reference signals are the joint positions and velocities. Figure 3.20 (picture provided by Protdex) shows the considered system constituted by three hexapods that tightly grasp the front spar of the LOCOMACHS wing-box.

Let consider a task in which the hexapods move a spar of 89 kg weight, from an initial pose $\boldsymbol{x}_{S_i}$ to a desired pose $\boldsymbol{x}_{S_f}$ (position in meters and orientation in degrees). As given in Section 3.3, given the spar trajectory and the grasp frame, the end-effector trajectory of each hexapod can be computed in the Cartesian space. Let define the initial and final pose of the spar as in (3.28) (the frame attached to the spar has been positioned in its COG), the grasp matrix as in (3.29) (data defined with respect to the world frame) and let consider the optimized design of the hexapods in terms of configuration of the hexapod leg attachment points and plate dimensions presented in A.3.

$$
\begin{aligned}
\boldsymbol{x}_{S_i} &= [2.2691,\ -0.0114,\ 1.3443,\ 0,\ -7.1869\, 0.1660]^T \\
\boldsymbol{x}_{S_f} &= [2.2691,\ -0.0114,\ 1.1943,\ 0,\ 0,\ 0]^T
\end{aligned}
\tag{3.28}
$$

$$
\boldsymbol{W} =
$$

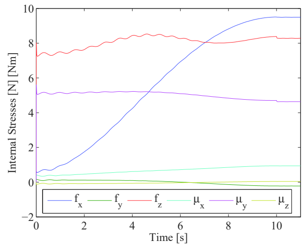
$$
\left[\begin{matrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & -0.1733 & -0.0094 & 1 & 0 & 0 & 0 & -0.03896 & -0.0056 \\
0.1733 & 0 & -1.1160 & 0 & 1 & 0 & 0.03896 & 0 & -0.05136 \\
0.0094 & 1.1160 & 0 & 0 & 0 & 1 & 0.0056 & 0.05136 & 0
\end{matrix}\right.
\begin{matrix} \\ \cdots \\ \\ \\ \cdots \\ \\ \end{matrix}
$$

$$
\begin{matrix} \\ \cdots \\ \\ \\ \cdots \\ \\ \end{matrix}
\left.\begin{matrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0.1019 & -0.0031 & 1 & 0 & 0 \\
0 & 1 & 0 & -0.1019 & 0 & 1.0667 & 0 & 1 & 0 \\
0 & 0 & 1 & 0.0031 & -1.0667 & 0 & 0 & 0 & 1
\end{matrix}\right]
\tag{3.29}
$$

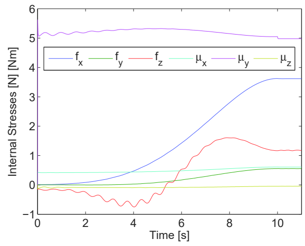
In case of perfect synchronization of the motion of the hexapod top plates, the estimated internal stresses on the attached spar are small as reported in Fig. 3.21 (in the simulations less than 20 N for the forces and 6 N for the torques). If simulated delays are introduced in the transmission of the set point, e.g., for the central hexapod of 8 ms and for the hexapod on the right of 3 ms with respect to the hexapod on the left, the internal stresses significantly increase (up to 265 N for the forces and 10 Nm for the torques), instead, as shown in Fig. 3.22. So, the reported results show show the importance of synchronization of the robots to avoid the internal stress in the coordinated motion tasks.

3.4 Conclusions

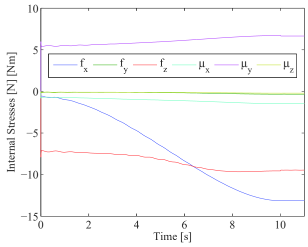
The presented tool has been exploited in the LOCOMACHS project to design and develop two ad-hoc Stewart platforms for the positioning of the ribs and of the lower



(a) Hexapod 1 (left).

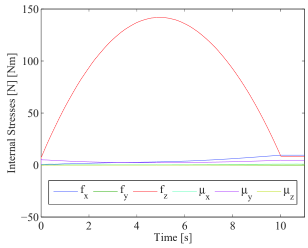


(b) Hexapod 2 (center).

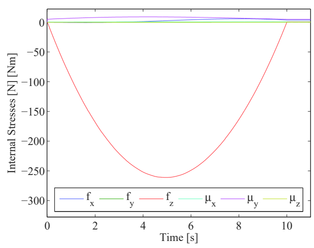


(c) Hexapod 3 (right).

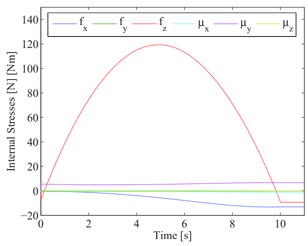
Figure 3.21: Coordinated motion: synchronized trajectories.



(a) Hexapod 1 (left).

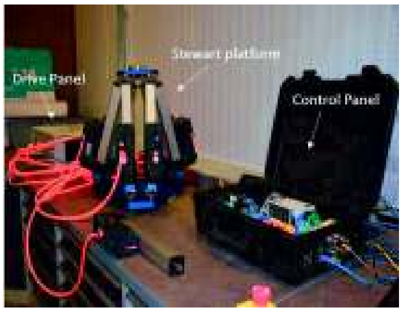


(b) Hexapod 2 (center).



(c) Hexapod 3 (right).

Figure 3.22: Coordinated motion: unsynchronized trajectories.



(a) Spar positioning Stewart platform (picture by MTC/Prodtex).

(b) Spar positioning Stewart platform (picture by MTC/Prodtex).

(c) Rib positioning Stewart platform (picture by Chalmers University).

Figure 3.23: Assembled Stewart platforms.

spar in the wing box assembled in the physical demonstrator. Given the complexity of the fixture and the high number of tools used inside such demonstrator, some mechanical constraints, i.e, plate dimensions, motor strokes and actuator dimensions, have been taken in account during the optimization process. Moreover, the pre-computed trajectory has been considered for each specific application. In Fig. 3.23, the assembled Stewart platforms, designed for the spar positioning and rib positioning, are reported.

In case of coordinated motion, the simulations show the importance of the synchronization of the leg motions and top plate motions of the robots. The lack of synchronization results in a significant increase of internal stresses that could lead to damage the handled object.

CHAPTER 4

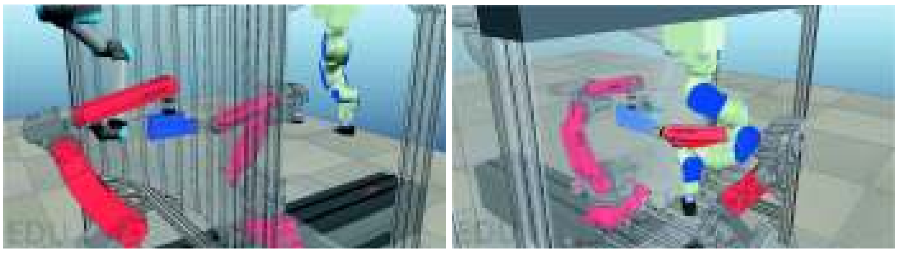
MULTI-TASK CONTROL ALGORITHMS FOR REDUNDANT ROBOTS

Nowadays, robotic systems with a large number of DOFs are commonly used in several applications beyond the original industrial environment. Unstructured and non-repetitive tasks require algorithms that are able to handle multiple objectives such as, the mechanical joint limits, the avoidance of obstacles, the orientation of directional sensors, the arm manipulability, etc.

A classical approach, basically, named multiple tasks priority is widely used to deal with the multiple objectives control, which is formulated in terms of multiple tasks that must be achieved at the same time. This concept, was introduced in [75] tackling the inverse kinematics of manipulators, and then was improved by [76] where obstacle avoidance was involved. Later, in [77] was presented the extension of this approach to an arbitrary number of tasks. In order to surpass the occurrence of algorithmic singularities that the previous methods suffer from, a different methodology which guarantees singularity robustness was proposed by [78], further extended and analyzed to multiple tasks in priority in [79, 80].

Example of possible tasks that can be achieved and handled with a multi-task control algorithm in case of industrial redundant robot are: end-effector position norm, obstacle avoidance, end-effector pose, end-effector field of view, mechanical joint-limit of the robot, robot manipulability, robot nominal configuration.

The multi-task multi-priority control approach can be exploited for the real-time monitoring in the robotized drilling process. The idea is to utilize visual information



(a) (b)

Figure 4.1: UNISA work-cell: new concept.

provided by a vision system and/or thermography information provided by a thermal imaging camera. Firstly, mono-camera or stereo-camera systems may be utilized to increase the positioning accuracy of the low-cost robots during the approach to the panel to drill. The use of visual information, in this case, can reduce or eliminate the positioning errors due to the uncertainties of the model of the part, introduced in the manufacturing process for example. Secondly, the thermography may be exploited to monitor the tool wear or to detect possible damages, e.g., delamination, caused to materials due to the high temperature produced during the machining phase. In this case, the drilling parameters, i.e., speed of rotation of the spindle, feed rate, can be modified in real-time during the drilling process to avoid the delamination and the deformation of the materials around the drilling area, especially in the carbon fiber machining. So, redundant robots can be introduced in the work-cell presented in Section 2 to perform different tasks, i.e., mechanical joint limits, obstacle avoidance, orientation of camera system, positioning of the tool. The new concept is shown in Fig. 4.1, where, in addition to the two Comau SmartSix robots, an Universal Robot UR10 (a 6-dofs robot holding a vision camera) and a Yaskawa SIA5F (a 7-dofs robot holding a thermography system), both mounted on a sliding track, are considered.

In the following, a brief introduction to the problem of the multi-task multi-priority problem is presented, furthermore, the Vision-Based Control and the Thermographic Monitoring topics are treated, focusing on the Image-Based Visual Servoing and on the Camera Calibration and Image Processing issues. Finally, the proposed solution for the thermographic drilling monitoring is presented.

4.1 Multi-Task Multi-Priority Behaviour-Based Algorithms

Let define with $\boldsymbol{\sigma}(t) \in \mathbb{R}^m$ the task variable to control and $\boldsymbol{q}(t) \in \mathbb{R}^n$ the vector of the joint configuration variables. The relation between $\boldsymbol{\sigma}(t)$ and $\boldsymbol{q}(t)$ can be expressed as:

$$\boldsymbol{\sigma}(t) = \boldsymbol{f}(\boldsymbol{q}(t)). \tag{4.1}$$

The corresponding differential relationship is:

$$\dot{\boldsymbol{\sigma}}(t) = \frac{\partial f(\boldsymbol{q}(t))}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}(t) = \boldsymbol{J}(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t), \tag{4.2}$$

where $\boldsymbol{J}(\boldsymbol{q}(t)) \in \mathbb{R}^{m \times n}$ is the analytical task Jacobian matrix, $\dot{\boldsymbol{q}}(t) \in \mathbb{R}^n$ is the joint velocity vector and n represents the number of DOFs of the considered robotic system. Assuming a single m-dimensional task for which a desired values $\boldsymbol{\sigma}_{des}(t) \in \mathbb{R}^m$ is assigned, the motion references $\boldsymbol{q}_{des}(t) \in \mathbb{R}^n$ for the robot can be computed by integrating the locally inverse mapping in (4.2) as reported in (4.3), where $J^\dagger$ is the right pseudoinverse matrix of J.

$$\dot{\boldsymbol{q}}_{des} = \boldsymbol{J}^\dagger \dot{\boldsymbol{\sigma}}_{des} = \boldsymbol{J}^\dagger (\boldsymbol{J}\boldsymbol{J}^\dagger)^{-1} \dot{\boldsymbol{\sigma}}_{des} \tag{4.3}$$

The solution achieved by integrating (4.3) suffers from drift, so, used back in (4.1) it does not result in $\boldsymbol{\sigma}_{des}$. Then, a CLIK version of the algorithm is usually implemented [78]:

$$\dot{\boldsymbol{q}}_{des} = \boldsymbol{J}^\dagger (\dot{\boldsymbol{\sigma}}_{des} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}) = \boldsymbol{J}^\dagger \dot{\boldsymbol{\sigma}}_{ref}, \tag{4.4}$$

where $\tilde{\boldsymbol{\sigma}} \in \mathbb{R}^m$ is the task error $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{des} - \boldsymbol{\sigma}$, and $\Lambda \in \mathbb{R}^{m \times m}$ is a positive defined matrix of gains.

In case of system redundancy ($n \gg m$) the classic general solutions contains a null projector operator as explained in Section 2.5 (eq. (4.5)). This allows to generate a motion of the robotic system that does not affect that of the given task.

$$\dot{\boldsymbol{q}}_{des} = \boldsymbol{J}^\dagger \dot{\boldsymbol{\sigma}}_{ref} + (\boldsymbol{I}_n - \boldsymbol{J}^\dagger \boldsymbol{J})\dot{\boldsymbol{q}}_{null} \tag{4.5}$$

For highly redundant systems, multiple tasks can be arranged in priority. Let consider three tasks that will be denoted with the subscript a, b and c, respectively:

$$\begin{aligned} \boldsymbol{\sigma}_a &= \boldsymbol{f}_a(\boldsymbol{q}) \in \mathbb{R}^{m_a} \\ \boldsymbol{\sigma}_b &= \boldsymbol{f}_b(\boldsymbol{q}) \in \mathbb{R}^{m_b}. \\ \boldsymbol{\sigma}_c &= \boldsymbol{f}_c(\boldsymbol{q}) \in \mathbb{R}^{m_c} \end{aligned} \tag{4.6}$$

For each task a corresponding Jacobian can be defined: $\boldsymbol{J}_a \in \mathbb{R}^{m_a \times n}$, $\boldsymbol{J}_b \in \mathbb{R}^{m_b \times n}$ and $\boldsymbol{J}_c \in \mathbb{R}^{m_c \times n}$. Analogously, let define the null-space projectors for the first and second tasks as $\boldsymbol{N}_a = (\boldsymbol{I}_n - \boldsymbol{J}_a^{\dagger}\boldsymbol{J}_a)$ and $\boldsymbol{N}_b = (\boldsymbol{I}_n - \boldsymbol{J}_b^{\dagger}\boldsymbol{J}_b)$. As proposed in [80], the generalization of the singularity-robust task priority inverse kinematic solution can be achieved by defining the Jacobian matrix $\boldsymbol{J}_{ab} \in \mathbb{R}^{(m_a+m_b)\times n}$ as:

$$\boldsymbol{J}_{ab} = \begin{bmatrix} \boldsymbol{J}_a \\ \boldsymbol{J}_b \end{bmatrix}, \ \boldsymbol{N}_{ab} = (\boldsymbol{I}_n - \boldsymbol{J}_{ab}^{\dagger}\boldsymbol{J}_{ab}). \tag{4.7}$$

The solution, then, can be computed as:

$$\dot{\boldsymbol{q}}_{des} = \underbrace{\boldsymbol{J}_a^{\dagger}\dot{\boldsymbol{\sigma}}_{a,ref}}_{\dot{\boldsymbol{q}}_{a,des}} + \underbrace{\boldsymbol{N}_a\boldsymbol{J}_b^{\dagger}\dot{\boldsymbol{\sigma}}_{b,ref}}_{\dot{\boldsymbol{q}}_{b,des}} + \underbrace{\boldsymbol{N}_{ab}\boldsymbol{J}_c^{\dagger}\dot{\boldsymbol{\sigma}}_{c,ref}}_{\dot{\boldsymbol{q}}_{c,des}} \tag{4.8}$$

The generalization to N tasks is straightforward. Let assume that $\boldsymbol{\sigma}_y \in \mathbb{R}^{m_y}$ is the N th task, and that the task x ($\boldsymbol{\sigma}_x \in \mathbb{R}^{m_x}$) precedes the task y in priority. Equation (4.8) become:

$$\dot{\boldsymbol{q}}_{des} = \boldsymbol{J}_a^{\dagger}\dot{\boldsymbol{\sigma}}_{a,ref} + \boldsymbol{N}_a\boldsymbol{J}_b^{\dagger}\dot{\boldsymbol{\sigma}}_{b,ref} + \ldots + \boldsymbol{N}_{ab\ldots x}\boldsymbol{J}_y^{\dagger}\dot{\boldsymbol{\sigma}}_{y,ref} \tag{4.9}$$

in which $\boldsymbol{N}_{ab\ldots x}$ is the null space of the Jacobian matrix

$$\boldsymbol{J}_{ab\ldots x} = \begin{bmatrix} \boldsymbol{J}_a \\ \boldsymbol{J}_b \\ \vdots \\ \boldsymbol{J}_x \end{bmatrix}. \tag{4.10}$$

4.2 Vision-Based Control

Vision allows a robotic system to obtain geometrical and qualitative information on the surrounding environment to be used both for motion planning and control. In particular, control based on feedback of visual measurements is termed visual servoing. Then, the task in visual servoing is to control the pose of the robot end effector with respect to the target using visual features extracted from the images acquired thought vision sensors. The vision sensor (usually referred to as camera) can be mounted on the robot end effector or can be fixed in the world. The first case is referred to as end-point closed-loop or *eye-in-hand*; the second case is referred to as end-point open-loop. In this work, only the eye-in-hand configuration is discussed.

In the following, a brief introduction to the image processing is presented and the two main approaches to visual servoing are introduced, namely Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS).

4.2.1 Image Processing

The image processing is a computational process that transforms one or more input images into an output image or in a little set of data to be used in robotics applications. Images are simply large arrays of pixel values but for robotic applications images have too much data and not enough information, so, the image processing is the fundamental operation of the feature extraction from a complex image. Features are typically scalars, e.g., the area or aspect ratio of a region, or short vectors that represent the coordinate of an object or the parameters of a line or of an ellipse.

In general, an image function is defined as a vector function whose components represent the values of one or more physical quantities related to the pixels. In the case of color images, the image function defined on a pixel of coordinates (X_I, Y_I) has three components $I_r(X_I, Y_I)$, $I_g(X_I, Y_I)$ and $I_b(X_I, Y_I)$, corresponding to the light intensity in the red, green and blue wavelengths. For a monochrome black-and-white image, the image function is scalar and coincides with the light intensity in shades of gray $I(X_I, Y_I)$, also referred to as gray level. The gray-scale function, and, especially the *gray-level histogram*, is particularly important in the analysis of the images. The histogram provides the frequency of occurrence of each gray level in the image. Usually, the gray levels are quantized from 0 to 255. The value $h(p)$ of the histogram at a particular gray level p represents the number of image pixels with that gray level. If this value is divided by the total number of pixels, the histogram is termed *normalized histogram* (see Fig. 4.2).

4.2.1.1 Image Segmentation

Image segmentation is the process of partitioning an image into application meaningful regions, referred to as segments. The aim is to segment or separate those pixels that represent objects of interest from all other pixels in the scene. Usually, distinct

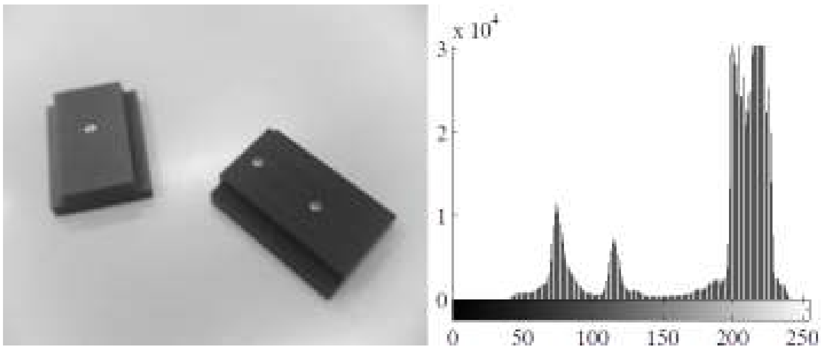


Figure 4.2: Black and white image and corresponding gray-level histogram.

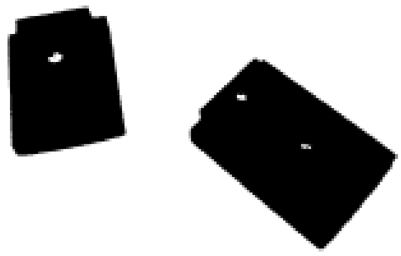


Figure 4.3: Binary image.

segments of the image correspond to distinct objects of the environment or homogeneous object parts. Two complementary approaches are used in the problem of the image segmentation: the first is based on finding connected regions of pixels in the image, the second is focused on the detection of the edges or boundaries of a region. The complementarity of the two approaches relies on the fact that a boundary can be achieved by isolating the contours of a region and, conversely, a region can be achieved simply by considering the set of pixels contained within a closed boundary.

4.2.1.1.1 Region-Based Segmentation The idea underlying the region-based segmentation techniques is to obtain connected regions by merging of initially small groups of adjacent pixels into larger ones. Two adjacent regions can be merged only if the pixels belonging to these regions satisfy a common property, termed *uniformity predicate*. An example of uniformity predicate is a given interval in which the gray level of the pixels of a region has to belong. In many practical application, a thresholding approach is adopted and a light intensity scale composed of only two values (0 and 1) is considered. This operation is referred to as *binary segmentation* or image

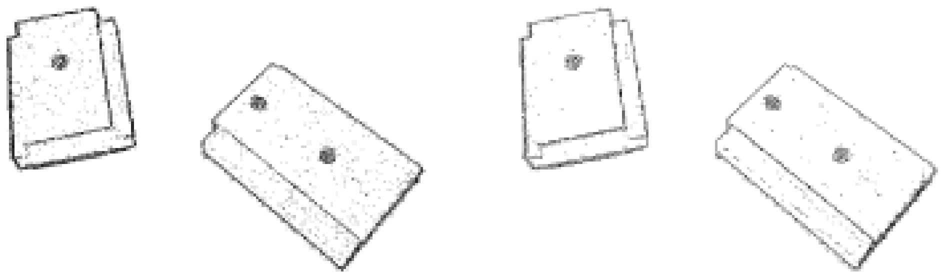


Figure 4.4: Example of edge detection.

binarization, and corresponds to separating one or more objects present in the image from the background by comparing the gray level of each pixel with a threshold l. For light objects against a dark background, all the pixels whose gray level is greater than the threshold are considered to belong to a set S_o, corresponding to the objects, while all the other pixels are considered to belong to a set S_b corresponding to the background. Similarly, the reversed operation can be considered for dark objects against a light background. The choice of the threshold value is crucial in the binary segmentation and in the proper recognition of objects belonging to the scene. A widely adopted method for selecting the threshold is based on the gray-level histogram, under the assumption that it contains clearly distinguishable minimum and maximum values, corresponding to the gray levels of the objects and of the background.

4.2.1.1.2 Boundary-Based Segmentation or Edge Detection Boundary-based segmentation techniques obtain a boundary by grouping many single local edges, corresponding to local discontinuities of image gray level. In other words, local edges are sets of pixels where the light intensity changes abruptly. The algorithms for boundary detection first derive an intermediate image based on local edges from the original gray-scale image, then they construct short-curve segments by edge linking, and finally obtain the boundaries by joining these curve segments through geometric primitives often known in advance. Several edge detection techniques exist. Most of them require the calculation of the gradient or of the Laplacian of function $I(X_I, Y_I)$. The most common operators are the Roberts operator, the Sobel operator, the Prewitt operator and the Canny operator. Figure 4.4 shows a contours of an image obtained by using the Robert operator on the left and the Sobel operator on the right.

4.2.1.2 Image Feature Extraction

Image feature extraction is the first step in using image data to control a robot. It is an operation that reduces the data rate from $10^6 - 10^8$ bytes per frame at the output of a camera to something of the order of tens of features per frame that can be used as input to a visual control system.

4.2.1.2.1 Moments The feature parameters used in visual servoing applications sometimes require the computation of the so-called moments. Moments are a rich and computationally cheap class of image features which can describe region size and location as well as shape. The general definition of moment $m_{i,j}$ of a region $\mathcal{R}$ of a frame, with $i, j = 0,\ 1,\ 2,\ ...,$ is:

$$m_{i,j} = \sum_{X_I, Y_I \in \mathcal{R}} I(X_I, Y_I) X_I^i Y_I^j. \tag{4.11}$$

From the definition (4.11), some notable quantities are obtained as in (4.13).

$$m_{i,j} = \sum_{X_I, Y_I \in \mathcal{R}} X_I^i Y_I^j \tag{4.12}$$

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}},\ \bar{y} = \frac{m_{0,1}}{m_{0,0}} \tag{4.13}$$

In particular, the moment $m_{0,0}$ in (4.12) coincides with the area of the region in case of binary images; the quantities $\bar{x}$ and $\bar{y}$ the *centroid* of the region. These coordinates can be used to detect uniquely the position of region $\mathcal{R}$ on the image plane.
The moment in (4.12) depends on the position of the region $\mathcal{R}$ in the image plane. The *central moments*, invariant values with respect to the translation, therefore, can be considered as in (4.14).

$$\mu_{i,j} = \sum_{X_I, Y_I \in \mathcal{R}} (X_I - \bar{x})^i (Y_I - \bar{y})^j \tag{4.14}$$

Moreover, by using the second order central moments $\mu_{2,0}$ and $\mu_{0,2}$, the inertia moments of the considered region, with respect to the axes X_I and Y_I, can be computed. The moments $\mu_{1,1}$ has the meaning of inertia product and the matrix in (4.15) has the meaning of the inertia tensor relative to the centre of mass.

$$\mathcal{I} = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{bmatrix} \tag{4.15}$$

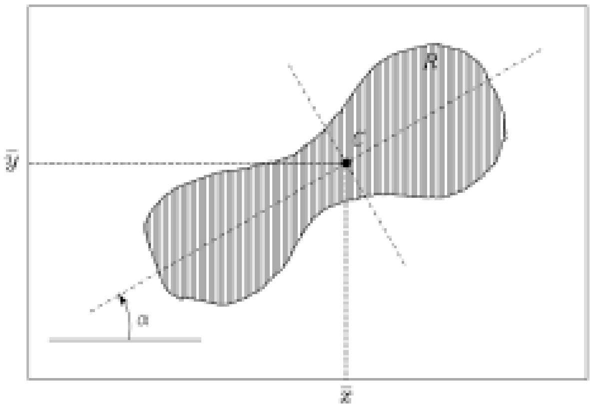


Figure 4.5: Region of a binary image and some features.

The eigenvalues of matrix $\mathcal{I}$ define the *principal moments* of inertia, and the corresponding eigenvectors define the *principal axes of inertia*. If region $\mathcal{R}$ is asymmetric, the principal moments of $\mathcal{I}$ are different. In this case, the orientation of $\mathcal{R}$ in terms of the angle α between the principal axis corresponding to the maximum moment and axis X can be computed as:

$$\alpha = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right). \tag{4.16}$$

4.2.1.2.2 Interaction Matrix Let consider a generic feature vector $\boldsymbol{s}$:

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_n \end{bmatrix}. \tag{4.17}$$

If the object is in motion with respect to the camera, the feature vector $\boldsymbol{s}$ is time-varying and, in general, a $(k \times 1)$ velocity vector $\dot{\boldsymbol{s}}$ can be defined in the image plane. The relative velocity of the object with respect to the camera can be defined as

$$\boldsymbol{v}^c_{c,o} = \begin{bmatrix} \dot{\boldsymbol{o}}^c_{c,o} \\ \boldsymbol{R}^T_c(\boldsymbol{\omega}_o - \boldsymbol{\omega}_c) \end{bmatrix}, \tag{4.18}$$

where $\dot{\boldsymbol{o}}^c_{c,o}$ is the time derivative of vector $\boldsymbol{o}^c_{c,o} = \boldsymbol{R}^T_c(\boldsymbol{o}_o - \boldsymbol{o}_c)$, representing the relative position of the origin of the object frame with respect to the origin of the camera frame, while $\boldsymbol{\omega}_o$ and $\boldsymbol{\omega}_c$ are the angular velocities of the object frame and camera

frame, respectively.
Therefore, the $(k \times 6)$ matrix $\boldsymbol{J}_s$ relating the feature velocity vector $\dot{\boldsymbol{s}}$ to the object-camera relative velocity vector $\boldsymbol{v}_{c,o}^c$ can be defined as in (4.19) and it is termed *image Jacobian*.

$$\dot{\boldsymbol{s}} = \boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{T}_o^c)\boldsymbol{v}_{c,o}^c \tag{4.19}$$

The image Jacobian $\boldsymbol{J}_s$m in general, depends on the current value of the feature vector $\boldsymbol{s}$ and on the relative pose of the object with respect to the camera $\boldsymbol{T}_o^c$.

By defining the mapping between the image plane velocity $\dot{\boldsymbol{s}}$, the absolute velocity of the camera frame $\boldsymbol{v}_c^c$ and the absolute velocity if the object frame $\boldsymbol{v}_o^c$ as in (4.20),

$$\boldsymbol{v}_c^c = \begin{bmatrix} \boldsymbol{R}_c^T \dot{\boldsymbol{o}}_c \\ \boldsymbol{R}_c^T \omega_c \end{bmatrix}, \ \boldsymbol{v}_o^c = \begin{bmatrix} \boldsymbol{R}_c^T \dot{\boldsymbol{o}}_o \\ \boldsymbol{R}_c^T \omega_o \end{bmatrix} \tag{4.20}$$

the vector $\dot{\boldsymbol{o}}_{c,o}^c$ can be rewritten as:

$$\dot{\boldsymbol{o}}_{c,o}^c = \boldsymbol{R}_c^T(\dot{\boldsymbol{o}}_o - \dot{\boldsymbol{o}}_c) + \boldsymbol{S}(\boldsymbol{o}_{c,o}^c)\boldsymbol{R}_c^T \omega_c. \tag{4.21}$$

From the (4.20) and (4.21), the (4.18) can be expressed in the compact form

$$\boldsymbol{v}_{c,o}^c = \boldsymbol{v}_o^c + \boldsymbol{\Gamma}(\boldsymbol{o}_{c,o}^c)\boldsymbol{v}_c^c, \ \boldsymbol{\Gamma}(\cdot) = \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{S}(\cdot) \\ \boldsymbol{O} & -\boldsymbol{I} \end{bmatrix}. \tag{4.22}$$

Therefore, eq. (4.19) can be rewritten as

$$\dot{\boldsymbol{s}} = \boldsymbol{J}_s \boldsymbol{v}_o^c + \boldsymbol{L}_s \boldsymbol{v}_c^c, \tag{4.23}$$

where the $(k \times 6)$ matrix $\boldsymbol{L}_s$ is the *interaction matrix* and it is defined as

$$\boldsymbol{L}_s = \boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{T}_o^c)\boldsymbol{\Gamma}(\boldsymbol{o}_{c,o}^c). \tag{4.24}$$

The interaction matrix defines the linear mapping between the absolute velocity of the camera $\boldsymbol{v}_c^c$ and the corresponding image plane velocity $\dot{\boldsymbol{s}}$, in the case that the object is fixed with respect to the base frame ($\boldsymbol{v}_o^c = \boldsymbol{0}$). The inverse relationship allows to compute the image Jacobian from the interaction matrix:

$$\boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{T}_o^c) = \boldsymbol{L}_s\boldsymbol{\Gamma}(-\boldsymbol{o}_{c,o}^c). \tag{4.25}$$

In the following, examples of computation of interaction matrix and image Jacobian for points, lines and ellipses features, are reported.

4.2.1.2.3 Point Feature Let consider a point P identified with respect to the camera frame by the vector

$$\boldsymbol{r}_c^c = \boldsymbol{R}_c^T(\boldsymbol{p} - \boldsymbol{o}_c), \tag{4.26}$$

where $\boldsymbol{p}$ is the position of the point P with respect to the base frame. Choosing the vector of the feature $\boldsymbol{s}$ to be equal to the normalized coordinates of the points, from eq. (B.11) yields:

$$\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{r}_c^c), \;\; \boldsymbol{s}(\boldsymbol{r}_c^c) = \frac{1}{z_c}\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{4.27}$$

with $\boldsymbol{r}_c^c = [x_c\ y_c\ z_c]^T$. Computing the time derivative of eq. (4.27) yields

$$\dot{\boldsymbol{s}} = \frac{\partial \boldsymbol{s}(\boldsymbol{r}_c^c)}{\partial \boldsymbol{r}_c^c}(\dot{\boldsymbol{r}}_c^c), \tag{4.28}$$

with

$$\frac{\partial \boldsymbol{s}(\boldsymbol{r}_c^c)}{\partial \boldsymbol{r}_c^c} = \frac{1}{z_c}\begin{bmatrix} 1 & 0 & -x_c/z_c \\ 0 & 1 & -y_c/z_c \end{bmatrix} = \frac{1}{z_c}\begin{bmatrix} 1 & 0 & -X \\ 0 & 1 & -Y \end{bmatrix}. \tag{4.29}$$

Under the assumption of $\boldsymbol{p}$ constant, from the time derivative of eq. (4.26) yields:

$$\dot{\boldsymbol{r}}_c^c = -\boldsymbol{R}_c^T\dot{\boldsymbol{o}}_c + \boldsymbol{S}(\boldsymbol{r}_c^c)\boldsymbol{R}_c^T\omega_c = [-\boldsymbol{I}\ \boldsymbol{S}(\boldsymbol{r}_c^c)]\boldsymbol{v}_c^c. \tag{4.30}$$

Combining (4.28) and (4.30), the following expression of interaction matrix of a point can be obtained:

$$\boldsymbol{L}_s(\boldsymbol{s}, z_c) = \begin{bmatrix} -\frac{1}{z_c} & 0 & \frac{X}{z_c} & XY & -(1+X^2) & Y \\ 0 & -1\frac{1}{z_c} & \frac{Y}{z_c} & 1+Y^2 & -XY & -X \end{bmatrix}. \tag{4.31}$$

The image Jacobian of a point can be computed from eq. (4.31) by using eq. (4.25):

$$\boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{T}_o^c) = \frac{1}{z_c}\begin{bmatrix} 1 & 0 & -X & -r_{o,y}^c X & r_{o,z}^c + r_{o,x}^c X & -r_{o,y}^c \\ 0 & 1 & -Y & -(r_{o,z}^c + r_{o,y}^c Y) & r_{o,x}^c Y & r_{o,x}^c \end{bmatrix}, \tag{4.32}$$

where $r_{o,x}^c,\ r_{o,y}^c,\ r_{o,z}^c$ are the components of the vector $\boldsymbol{r}_o^c = \boldsymbol{r}_c^c - \boldsymbol{o}_c^c = \boldsymbol{R}_o^c\boldsymbol{r}_o^o$, with $\boldsymbol{r}_o^o$ is the constant vector expressing the position of point P with respect to the object frame.

The presented results can be generalized in case of a set of n points $P_1, ..., P_n$. The $(2n \times 6)$ interaction matrix can be computed by considering the $(2n \times 1)$ feature vector

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_n \end{bmatrix}, \tag{4.33}$$

as in eq. (4.34).

$$\boldsymbol{L}_s(\boldsymbol{s}, \boldsymbol{z}_c) = \begin{bmatrix} \boldsymbol{L}_{s_1}(\boldsymbol{s}_1, z_{c,1}) \\ \vdots \\ \boldsymbol{L}_{s_n}(\boldsymbol{s}_n, z_{c,n}) \end{bmatrix} \tag{4.34}$$

with $\boldsymbol{z}_c = [z_{c,1}, ..., z_{c,n}]^T$. The image Jacobian of a set of points can be easily computed from the interaction matrix using (4.25).

4.2.1.2.4 Line Feature Let consider a part of a line connecting two point P_1 and P_2. The projection on the image plane is still a line segment that can be represented in terms of the middle point coordinates $\bar{x}$, $\bar{y}$, the length L and the angle α formed by the line with respect to X axis. Thus, the feature vector can be defined as:

$$\boldsymbol{s} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ L \\ \alpha \end{bmatrix} = \begin{bmatrix} (X_1 + X_2)/2 \\ (Y_1 + Y_2)/2 \\ \sqrt{\Delta X^2 + \Delta Y^2} \\ \tan^{-1}(\Delta Y/\Delta X) \end{bmatrix} = \boldsymbol{s}(\boldsymbol{s}_1, \boldsymbol{s}_2) \tag{4.35}$$

with $\Delta X = X_2 - X_1$, $\Delta Y = Y_2 - Y_1$ and $\boldsymbol{s}_i = [X_i \; Y_i]^T$, $i = 1, 2$. Computing the time derivative of the reported relation yields

$$\begin{aligned} \dot{\boldsymbol{s}} &= \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_1}\dot{\boldsymbol{s}}_1 + \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_2}\dot{\boldsymbol{s}}_2 \\ &= \left(\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_1}\boldsymbol{L}_{s_1}(\boldsymbol{s}_1, z_{c,1}) + \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_2}\boldsymbol{L}_{s_2}(\boldsymbol{s}_2, z_{c,2})\right)\boldsymbol{v}_c^c, \\ &= \boldsymbol{L}_s(\boldsymbol{s}, \boldsymbol{z}_c)\boldsymbol{v}_c^c \end{aligned} \tag{4.36}$$

where $\boldsymbol{L}_{s_i}$ is the interaction matrix of point P_i with

$$\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \\ -\Delta X/L & -\Delta Y/L \\ \Delta Y/L^2 & -\Delta X/L^2 \end{bmatrix} \quad \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{s}_2} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \\ \Delta X/L & \Delta Y/L \\ -\Delta Y/L^2 & \Delta X/L^2 \end{bmatrix}. \tag{4.37}$$

The presented interaction matrix is valid under the assumption that the line segment is fixed with respect to the base frame and the corresponding image Jacobian can be computed by using eq. (4.25).

4.2.1.2.5 Ellipse Feature A circle in the world is projected, in the general case, to an ellipse in the image which is described by

$$X^2 + s_1 Y^2 - 2s_2 XY + 2s_3 X + 2s_4 Y + s_5 = 0 \tag{4.38}$$

where $\boldsymbol{s} = [s_1, ..., s_5]^T$ is the feature vector of the ellipse. The image Jacobian of the ellipse $\boldsymbol{J}_s$ is reported in eq. (4.39) [81], where $\boldsymbol{\rho} = (\alpha, \beta, \gamma)$ defines the plane $ax + by + cZ + d = 0$ in the Cartesian space, in which the ellipse lies and $\alpha = -a/d$, $\beta = -b/d$, $\gamma = -c/d$.

$$\boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{\rho}) =$$
$$\begin{bmatrix} 2bs_2 - 2as_1 & 2s_1(b - as_2) & 2bs_4 - 2as_1s_3 & \\ b - as_2 & bs_2 - a(2s_2^2 - s_1) & a(s_4 - 2s_2s_3) + bs_3 & \cdots \\ c - as_3 & a(s_4 - 2s_2s_3) + cs_2 & cs_3 - a(2s_3^2 - s_5) & \\ s_3b + s_2c - 2as_4 & s_4b + s_1c - 2as_2s_4 & bs_5 + cs_4 - 2as_3s_4 & \cdots \\ 2cs_3 - 2as_5 & 2cs_4 - 2as_2s_5 & 2cs_5 - 2as_3s_5 & \end{matrix}$$
$$\begin{matrix} & 2s_4 & 2s_1s_3 & -2s_2(s_1 + 1) \\ \cdots & s_3 & 2s_2s_3 - s_4 & s_1 - 2s_2^2 - 1 \\ & -s_2 & 1 + 2s_3^2 - s_5 & s_4 - 2s_2s_3 \\ \cdots & s_5 - s_1 & 2s_3s_4 + s_2 & -2s_2s_4 - s_3 \\ & -2s_4 & 2s_3s_5 + 2s_3 & -2s_2s_5 \end{bmatrix} \tag{4.39}$$

Note that, like the cases of point and line Jacobian, also for the ellipse Jacobian a depth information about the target is required. The Jacobian normally has a rank of five, but this drops to three when the projection is of a circle centred in the image plane, and a rank of two if the circle is a point.
An advantage of the ellipse feature is that the ellipse can computed from the set of all boundary points without needing to solve the correspondence problem.

The (5×6) Jacobian has a maximum rank of only 5, so, the camera velocity $\boldsymbol{v}_c^c$ cannot uniquely obtained from eq. (4.28). Then, two solutions can be pursued. Firstly, if the final view is a circle, then the rotation about the axis passing through the centre of the circle is irrelevant, and in this case the sixth column of the Jacobian can be deleted to make it square (in this case $\omega_z = 0$). The second approach is to

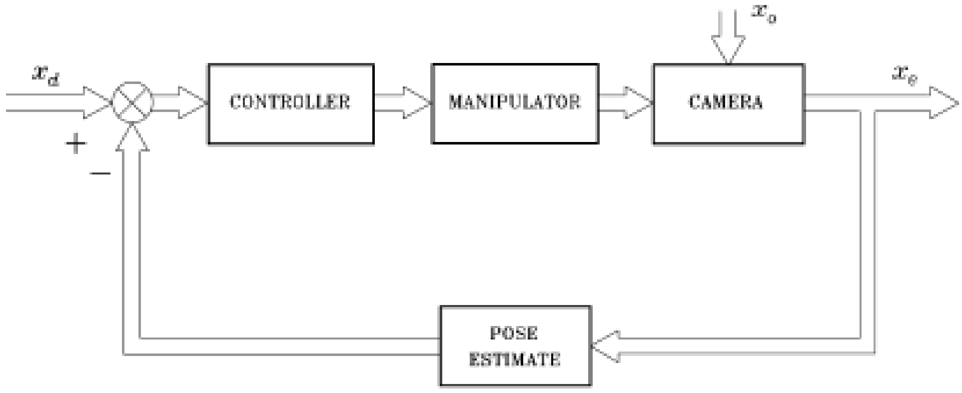


Figure 4.6: General block scheme of PBVS.

combine the features for the ellipse with a point feature as reported in (4.40), where $\boldsymbol{J}_e$ is the image Jacobian of the ellipse and $\boldsymbol{J}_p$ is the image Jacobian of the point.

$$\begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \vdots \\ \dot{s}_5 \\ \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_e(\boldsymbol{s}) \\ \boldsymbol{J}_p(\boldsymbol{p}_1) \end{bmatrix} \boldsymbol{v}_c^c \tag{4.40}$$

The stacked Jacobian is now (7×6) and we can solve for camera velocity.

4.2.2 Position-Based Visual Servoing

In the position-based visual servoing approach the feedback is based on the real-time estimation of the pose of the observed object with respect to the camera using visual measurements. The main drawback of this approach is that the object may exit from the camera field of view during the transient or as a consequence of planning errors. So, the feedback loop turns out to be open due to lack of visual measurements and instability may occur. This approach, moreover, suffers from the errors committed in the camera calibration process. In fact, the presence of uncertainties on calibration parameters, both intrinsic and extrinsic, produces errors on the estimate of operational space variables that may be seen as an external disturbance acting on the feedback path of the control loop, where disturbance rejection capability is low. Furthermore, the object geometry must be known if only one camera is used, because it is necessary for pose estimation, while it may be unknown when a stereo camera system is used.

Figure 4.6 shows the general block scheme of the PBVS control algorithm. Assuming that the object is fixed with respect to the base frame, the PBVS can be an be formulated by imposing the desired value to the relative pose of the object frame with respect to the camera frame in terms of the homogeneous transformation matrix $\boldsymbol{T}_o^d$, where superscript d denotes the desired pose of the camera frame. The homogeneous transformation matrix $\boldsymbol{T}_o^c$, representing the relative pose of the object frame with respect to the camera frame, and the matrix $\boldsymbol{T}_o^d$ can be used to obtain the matrix $\boldsymbol{T}_c^d$ representing the pose displacement of the camera frame in the current pose with respect to the desired pose as in (4.41).

$$\boldsymbol{T}_c^d = \boldsymbol{T}_o^d (\boldsymbol{T}_o^c)^{-1} = \begin{bmatrix} \boldsymbol{R}_c^d & \boldsymbol{o}_{d,c}^d \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{4.41}$$

From the (4.41) the error vector in the operational space in (4.42) can be derived, where $\boldsymbol{\phi}_{d,c}$ is the vector of Euler angles extracted from the rotation matrix $\boldsymbol{R}_c^d$.

$$\tilde{\boldsymbol{x}} = -\begin{bmatrix} \boldsymbol{o}_{d,c}^d \\ \boldsymbol{\phi}_{d,c} \end{bmatrix} \tag{4.42}$$

The control law has to be designed so that the operational space error $\tilde{\boldsymbol{x}}$ tends to zero asymptotically. Below, the *resolved-velocity control* scheme is illustrated. The presented approach is based on the computation of the imposed joint velocity $\dot{\boldsymbol{q}}_r$ from the operational space error (4.42) and the imposed reference trajectory for the joint variables $\boldsymbol{q}_r(t)$ is computed, consequently, from via a simple integration.

Let assume that the manipulator is equipped with a high-gain motion controller in the joint space or in the operational space; in other words, the controlled manipulator can be considered as an ideal positioning device (4.43).

$$\boldsymbol{q}(t) \approx \boldsymbol{q}_r(t) \tag{4.43}$$

Noting that when the end-effector frame and the camera frame coincide, the following equality is valid

$$\dot{\tilde{\boldsymbol{x}}} = -\boldsymbol{J}_{A_d}(\boldsymbol{q}, \tilde{\boldsymbol{x}})\dot{\boldsymbol{q}} \tag{4.44}$$

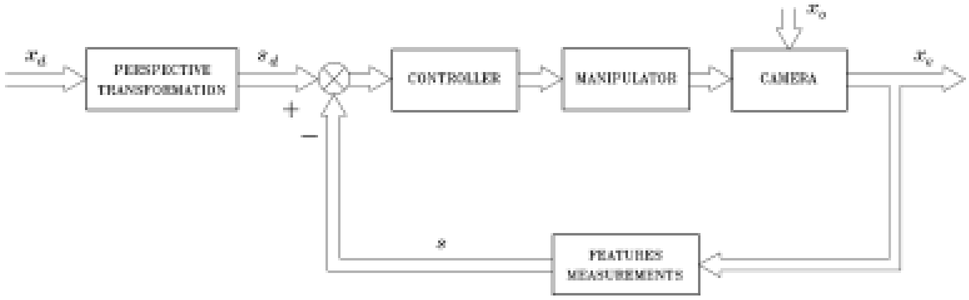


Figure 4.7: General block scheme of IBVS.

with

$$J_{A_d}(q, \tilde{x}) = T_A^{-1}(\phi_{d,c}) \begin{bmatrix} R_d^T & 0 \\ 0 & R_d^T \end{bmatrix} J(q), \tag{4.45}$$

the choice for the joint space reference velocity in eq. (4.46) is suggested.

$$\dot{q}_r = J_{A_d}^{-1}(q_r, \tilde{x}) K \tilde{x} \tag{4.46}$$

Substituting the (4.46) in (4.44), and using the (4.43) yields the linear equation

$$\dot{\tilde{x}} + K\tilde{x} = 0. \tag{4.47}$$

This equality, for a positive definite matrix K, implies that the operational space error tends to zero asymptotically with a convergence of exponential type and speed depending on the eigenvalues of matrix K; the larger the eigenvalues, the faster the convergence.

For a complete analysis and discussion of the PBVS, refer to [10].

4.2.3 Image-Based Visual Servoing

In the image-space visual servoing approach, the control action is computed on the basis of the error defined as the difference between the value of the image feature parameters in the desired configuration, computed using perspective transformation or directly measured with the camera in the desired pose, and the value of the parameters measured with the camera in the current pose. The conceptual advantage of this solution regards the fact that the real-time estimate of the pose of the object with respect to the camera is not required. Moreover, since the control acts directly in the image feature parameters, it is possible to keep the object within the camera field of view during the motion. However, due to the nonlinearity of the mapping between the image feature parameters and the operational space variables, singular configurations may occur, which cause instability or saturation of the control action. Also, the end-effector trajectories cannot be easily predicted in advance and may produce collisions with obstacles or joint limits violation. With respect to the PBVS, in the IBVS the quantities used for the computation of the control action are directly defined in the image plane and measured in pixel units and the desired value of the feature parameters is measured using the camera. This implies that the uncertainty affecting calibration parameters can be seen as a disturbance acting on the forward path of the control loop, where disturbance rejection capability is high. Furthermore, the IBVS does not require knowledge of the object geometry, even for mono-camera systems.

Figure 4.7 shows the general block scheme of the PBVS control algorithm. In general, if the object is fixed with respect to the base frame, image-based visual servoing can be formulated by stipulating that the vector of the object feature parameters has a desired constant value $\boldsymbol{s}_d$ corresponding to the desired pose of the camera. It is worth noticing that the task is assigned directly in terms of feature vector $\boldsymbol{s}_d$, while pose $\boldsymbol{x}_{d,o}$ does not need to be known.
The control law must be designed so as to guarantee that the image space error

$$\boldsymbol{e}_s = \boldsymbol{s}_d - \boldsymbol{s} \tag{4.48}$$

tends asymptotically to zero.

Since $\dot{\boldsymbol{s}}_d = \boldsymbol{0}$ and the object is fixed with respect to the base frame, by time derivative of (4.48) yields

$$\dot{\boldsymbol{s}}_s = -\dot{\boldsymbol{s}} = -\boldsymbol{J}_L(\boldsymbol{s}, z_c, \boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4.49}$$

where

$$\boldsymbol{J}_L(\boldsymbol{s}, z_c, \boldsymbol{q}) = \boldsymbol{L}_s(\boldsymbol{s}, z_c)\begin{bmatrix} \boldsymbol{R}_d^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_d^T \end{bmatrix}\boldsymbol{J}(\boldsymbol{q}). \tag{4.50}$$

The concept of resolved-velocity control suggests the choice of the reference velocity in joint space as

$$\dot{\boldsymbol{q}}_r = \boldsymbol{J}_L^{-1}(\boldsymbol{s}, z_c, \boldsymbol{q}_r)\boldsymbol{K}_s\boldsymbol{e}_s. \tag{4.51}$$

The expression in (4.51) assumes that $\boldsymbol{J}_L$ is invertible. Therefore, replacing the control law in (4.49) yields the linear equation

$$\dot{\boldsymbol{e}}_s + \boldsymbol{K}_s \boldsymbol{e}_s = \boldsymbol{0} \tag{4.52}$$

that is asymptotically stable by choosing a positive definite matrix $\boldsymbol{K}_s$, and the error $\boldsymbol{e}_s$ tends asymptotically to zero with convergence of exponential type and speed depending on the eigenvalues of matrix $\boldsymbol{K}_s$.
Notice that this control scheme requires the computation of the inverse of matrix $\boldsymbol{J}_L$. Therefore any problems related to the singularities of this matrix which are both those of the geometric Jacobian and those of the interaction matrix could be taken into account in the computation of the matrix.

For a complete discussion on the IBVS, refer to [10].

4.3 Thermographic Monitoring of the Robotized Drilling Process

High temperatures can provoke changes in the microstructure of materials during the machining process, resulting in form errors that may cause loss of the machined material, delamination and a decrease of tool life. As a result, high temperatures can be responsible of an increase in production costs.
Infrared Thermography (IRT) is a non-contact, non-invasive, fast and whole field method for Non Destructive Evaluation (NDE) applications that in the last decades is finding space in the monitoring of industrial manufacturing processes. Two approaches to the IRT are known in literature, namely Active and Passive IRT modes. In particular, Active Thermography involves an external stimulus to impart heat on the test object, whereas in Passive Thermography, the test object has its own internal source of heat.

In this work, a passive thermography system for monitoring the drilling process, is proposed. In particular, the activities have been focused on the development of a multi-task multi-priority algorithm and a visual control system to monitor the drilling process of composite parts to implement in ROS (Robot Operating System) environment[12] and on a Yaskawa SIA5F equipped with an Optris PI450 thermal imager.

[12] ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more.

(a) ROS Industrial logo.

(b) OpenCV logo.

Figure 4.8: ROS and OpenCV logo.

Moreover, OpenCV[13] has been chosen to properly perform the image processing operations.

4.3.1 State of the Art

The infrared thermography finds its use in a wide range of applications ranging from NDE to preventive maintenance, heat exchangers, transmission lines, missile tracking, night vision, electrical inspection, mechanical inspection, etc. But, only few studies have been carried out on the analysis of the effects of temperature on the tool wear and the damage of the carbon material in case of a drilling process. More works on the indirect monitoring methods that make use of force, vibration and current measurements have been carried out [82].

Davies et al. [83] reviewed several used temperature measurement methods and showed how they can be applied to temperature monitoring during material removal. Rares [84] proposed a method to identify the possible future damages at the automatic technological installations caused by the influence of the heating. In [85] Lauro et al. presented a methodology to select the machining parameters in order to define the best parameters of influence for decreasing temperature during the milling of aluminum alloys. Bagavathiappan [86] and Shindou [87] presented a complete analysis on the monitoring of the cutting tool temperature during the milling process. They used infrared thermography for online monitoring of the cutting tool temperature during the micro-end milling, with the aim to study the effects of milling parameters such as spindle speed, feed rate and depth of cut on the tool temperature. In the field of drilling, analysis on the distribution of the temperature on the drilled

[13]OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception.

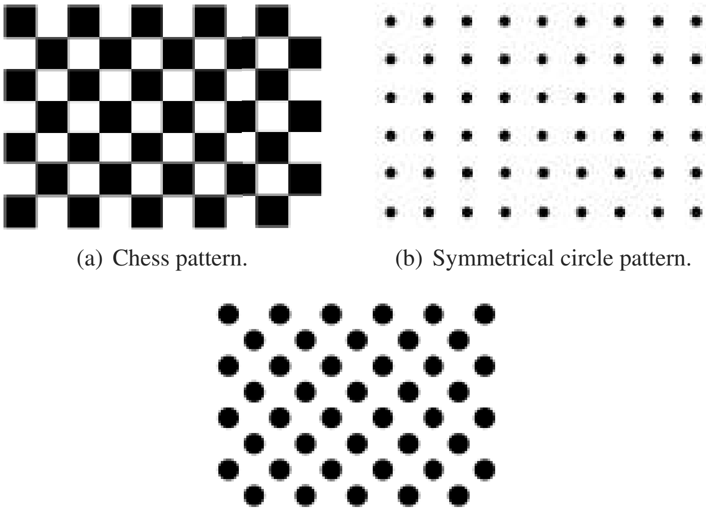
materials have been performed. Roseiro et al. [88] and Augustin [89] showed the importance of the thermography in the orthopedic surgery, and, in particular, in the drilling procedure of the bones in which the increase in temperature may lead to the bone necrosis. In [90] Honner proposed a technique for the thermography analyses of the hole-drilling residual stress measuring. Meola et al. [91] used infrared thermography, coupled with phased array ultrasonic, to detect defects and impact damage in carbon fibre reinforced composites.

In the presented work a thermographic system to perform drilling analysis during the machining of carbon fiber or aluminium parts, has been developed. In particular, a 7-dofs industrial robot equipped with an Optris PI450 thermal imaging camera has been used to monitor the operations. An image-based visual servoing algorithm has been developed to keep the features in the field-of-view (FOV) of the camera and to keep the optical axis perpendicular to the panel while the drilling robot performs the drilling task. Furthermore, in order to avoid collisions between the monitoring robot and the drilling robot, an obstacle avoidance algorithm has been implemented. Hereinafter, the camera calibration procedure, the performed simulations and the experiments are reported.

4.3.2 Camera Calibration

An important problem for visual servoing applications is the calibration of the camera. Calibration consists of the estimation of the *intrinsic parameters*, characterizing matrix $\mathbf{\Omega}$ defined in (B.9), and of the *extrinsic parameters*, characterizing the pose of the camera frame with respect to the end-effector frame (for eye-in-hand cameras).

The calibration of standard vision cameras requires the use of a standard pattern. The OpenCV library provides a tested and easy-to-use calibration procedure that allows to estimate the intrinsic and extrinsic camera parameters, and the distortion matrix of the camera. Moreover, it can be used with three different patterns: the classical black-white chessboard, the symmetrical circle pattern and the asymmetrical circle pattern (see Fig. 4.9). Acquired images depicting the chosen pattern from different positions are used as input to the calibration procedure. The number of images is higher for the chessboard pattern and less for the circle ones. In theory the chessboard pattern requires at least two snapshots. However, in practice a certain amount of noise is present in the input images, then, for a good estimation, at least 10 good snapshots are required. It is obvious that, it is not possible to use a printed calibration sheet for a thermal imager but a thermal calibration plate is required. The



(a) Chess pattern. (b) Symmetrical circle pattern.

(c) Unsymmetrical circle pattern.

Figure 4.9: Camera calibration patterns.

proposed solution makes use of a calibrated plate made of PVC material, in which a symmetrical circle pattern has been drilled. Each circle has a diameter of 8 mm and it is 28.8 mm away from the next hole. The idea is to use a heat source positioned behind the calibration plate and, so, to exploit the phenomenon of conduction of the infrared rays emitted by the source itself toward the thermal imager. In particular, the plate permits the passage of the rays only through the holes. Figure 4.10 shows the considered system constituted by the 7-dofs robot and the thermal image camera and the developed calibration plate (reporting a 6×9 symmetrical circle pattern) attached to a heat source (an oven). The camera calibration procedure has been performed in three steps. In the first step, the intrinsic parameters of the camera have been estimated. The camera, mounted on the robot end effector thought an ad-hoc designed support developed in ABS material by using a 3D printer, has been positioned in 15 different positions with respect to the calibration plate, and, then, 15 snapshots have been acquired. Simultaneously, the corresponding robot joint positions have been acquired. In Fig. 4.11(a), the designed camera tool, the camera frame and the object frame are shown. By using the 15 images, the camera calibration procedure available in the OpenCV has been executed obtaining the intrinsic parameters of the camera. Moreover, the calibration procedure provides the estimated pose of the camera frame with respect to the object frame for each images used in the calibration

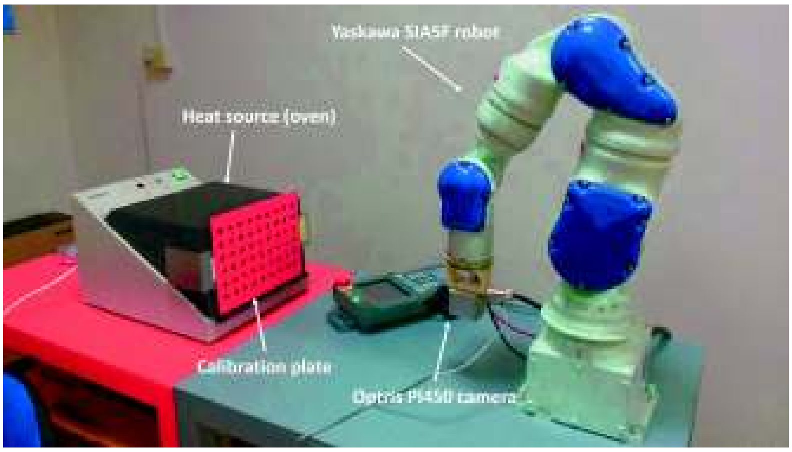
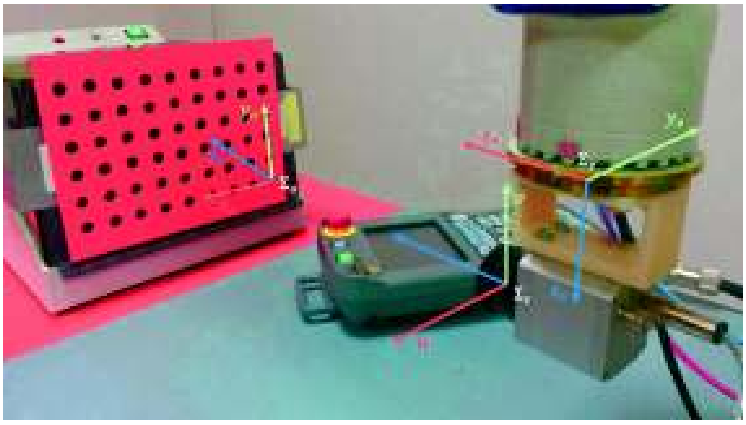


Figure 4.10: Thermography system.

process (this kind of information will be used in the third step to estimate the extrinsic parameters of the camera). The acquired thermal images, before to be used for the calibration procedure, have been processed, obtaining from the RGB images the corresponding binary images through a thresholding operation. In particular, by using the properly OpenCV function, an inverted binary thresholding has been performed (see Section 4.3.3.2). In Figure 4.12 an example of thermal image and the corresponding binary image are reported. Figure 4.13 reports two pictures taken during the calibration procedure. In particular, Fig. 4.13(a) shows the recognition phase of the input images in which the circle centers, and, then, the input circle patterns, have been recognized. Figure 4.13(b) shows the resulting image resulting from the distortion removal procedure. The estimated camera matrix is

$$\boldsymbol{\theta} = \begin{bmatrix} 619.0495 & 0.0000 & 191.5000 \\ 0.0000 & 619.0495 & 143.5000 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.53}$$

In the second phase, the pose of the frame attached to the calibration plate (object frame) with respect to the robot base frame has been estimated. To this aim, a second tool has been designed and developed with the 3D printer. Such a tool allows the robot to reach the holes on the calibration plate. So, chosen a subset of 7 holes and positioned the tool in the holes, the joint positions of the robot have been acquired.



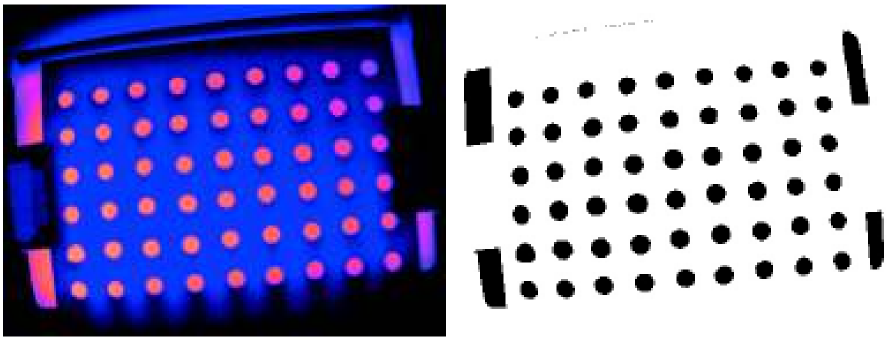
(a) Camera tool.

(b) Object calibration tool.

Figure 4.11: Designed tools.

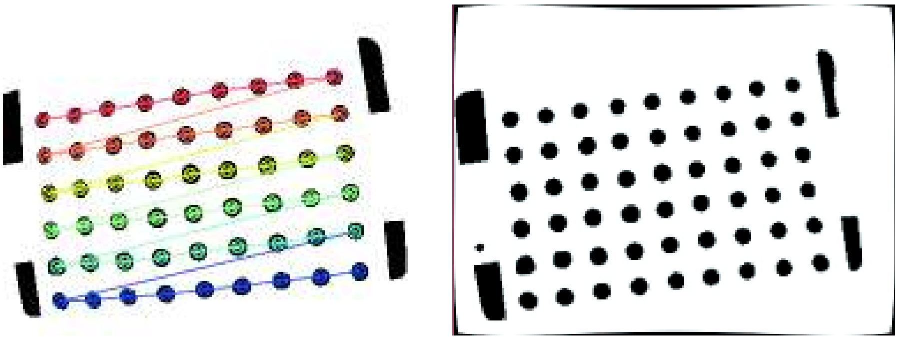
Thus, known the calibration point positions with respect to the object frame and the joint positions of the robot corresponding to the considered holes, the calibration algorithm reported in Section 4.3.2.1 has been used to estimate the pose of the object frame with respect to the base frame of the robot. Figure 4.11(b) reports a picture of the object calibration phase. The resulting transformation matrix $\boldsymbol{T}_o^b$ representing the pose of the object frame with respect to the base frame is reported in (4.54).

$$\boldsymbol{T}_o^b = \begin{bmatrix} 0.9996 & -0.0207 & -0.0179 & -0.1220 \\ -0.0185 & -0.0069 & -0.9741 & -0.7203 \\ 0.0201 & 0.9741 & -0.0073 & -0.2232 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.54}$$



(a) Thermal image. (b) Binary image.

Figure 4.12: Example of thermal image binarization.



(a) Pattern recognition. (b) Distortion removal procedure.

Figure 4.13: Camera calibration procedure in OpenCV.

The third step consisted of estimating the extrinsic parameters of the camera and, thus, estimating the pose of the camera frame with respect to the end-effector frame. Given the pose of the object frame with respect to the base frame estimated in the second phase, and given the pose of the camera with respect to the object frame for each snapshot used in the first phase, the extrinsic parameters of the camera can be estimated as reported in Section 4.3.2.2 by solving a least square problem. The resulting transformation matrix $\boldsymbol{T}_c^e$ representing the pose of the camera frame with respect to the end-effector frame is reported in (4.55).

$$\boldsymbol{T}_c^e = \begin{bmatrix} 0.9998 & -0.0215 & 0.0016 & 0.0088 \\ -0.0021 & -0.0205 & 0.9998 & 0.0203 \\ -0.0214 & -0.9996 & -0.0206 & 0.0915 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.55}$$

4.3.2.1 Object Calibration

Under the *hypothesis of robots perfectly calibrated*, the calibration procedure assumes that:

- The calibrator is geometrically perfect and the position of the calibration points $\boldsymbol{p}_i$ is perfectly known with respect to the object frame
- It is mechanically possible to bring the end effector in the points $\boldsymbol{p}_i$ with infinite precision
- The tool is geometrically perfect and its dimensions are known

Indicating with:

- $\Sigma_b = (o_b, x_b, y_b, z_b)$ the robot base frame
- $\Sigma_o = (o_o, x_o, y_o, z_o)$ the object frame
- $\bar{\boldsymbol{R}}_o^b$ the unknown rotation matrix
- $\bar{\boldsymbol{o}}_{o,b}^b$ the unknown position $\boldsymbol{o}_o$ with respect to Σ_b
- $\boldsymbol{p}_i^o$ the position of the calibration point $\boldsymbol{p}_i$ with respect to Σ_o
- $\boldsymbol{p}_i^b$ the position of the calibration point $\boldsymbol{p}_i$ with respect to Σ_b

the (4.56) can be considered to solve the problem.

$$\boldsymbol{p}_i^b = \bar{\boldsymbol{o}}_{o,b}^b + \bar{\boldsymbol{R}}_o^b \boldsymbol{p}_i^o, \; i = 1, ..., N \tag{4.56}$$

$$\tilde{\boldsymbol{p}}_i^b = \boldsymbol{A}_o^b \tilde{\boldsymbol{p}}_i^o \tag{4.57}$$

Thus, in order to obtain both the unknown variables, $\bar{\boldsymbol{o}}_{o,b}^b$ and $\bar{\boldsymbol{R}}_o^b$, an optimization problem can be solved by considering the objective function in (4.58), where $\boldsymbol{p}_i^b$ are directly read on the Teach Pendant of the robot when the tool is in the calibration point $\boldsymbol{p}_i$ or computed from the joint positions using the direct kinematics of the robot, and $\boldsymbol{p}_i^o$ are taken from the CAD of the calibration plate.

$$V(\bar{\boldsymbol{o}}_{o,b}^b, \bar{\boldsymbol{R}}_o^b) = \sum_{i=1}^{N} \left\| \boldsymbol{p}_i^b - (\bar{\boldsymbol{o}}_{o,b}^b + \bar{\boldsymbol{R}}_o^b \boldsymbol{p}_i^o) \right\|^2 , \tag{4.58}$$

where N is the number of the considered calibration points. The estimated parameters can be rewritten in compact form as in (4.59).

$$\bar{\boldsymbol{T}}_o^b = \begin{bmatrix} \bar{\boldsymbol{R}}_o^b & \bar{\boldsymbol{o}}_{o,b}^b \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{4.59}$$

The objective function has been implemented as a Matlab function and minimized by calling the *fminunc*[14] function. In order to avoid the Euler angles representation singularities, and to guarantee that the orthonormality constraint on the rotation matrix is satisfied, the rotation matrix $\bar{\boldsymbol{R}}_o^b$ has been replaced with a unit quaternion rotation. This is necessary, since using an unconstrained minimization algorithm, a not orthonormal matrix could be achieved. Furthermore, the number of unknown variables decrease from 12 to 7, so, only four calibration points, each constituted by three position coordinates, are required.

4.3.2.2 Extrinsic Parameter Calibration

The proposed calibration procedure assumes that the pose of the object frame with respect to the base frame is perfectly known.
Let indicate with:

- $\Sigma_b = (o_b, x_b, y_b, z_b)$ the robot base frame
- $\Sigma_o = (o_o, x_o, y_o, z_o)$ the object frame
- $\Sigma_e = (o_e, x_e, y_e, z_e)$ the robot end-effector frame
- $\Sigma_c = (o_c, x_c, y_c, z_c)$ the camera frame
- $\bar{\boldsymbol{T}}_c^e$ the unknown and constant transformation matrix representing the pose of Σ_c with respect to Σ_e

[14]Fminunc attempts to find a minimum of a scalar function of several variables, starting at an initial estimate (it is generally referred to as unconstrained nonlinear optimization).

- $\bar{\boldsymbol{T}}_o^b$ the constant transformation matrix representing the pose of Σ_o with respect to Σ_b

- $\boldsymbol{T}_{c_i}^o$ the transformation matrices representing the pose of the camera frame Σ_c with respect to the object frame Σ_o, with $i = 1, ..., N$, acquired during the calibration procedure of the intrinsic parameters of the camera.

The transformation matrix obtained by computing the direct kinematics of the robot can be written as:

$$\boldsymbol{T}_c^b(\boldsymbol{q}_i) = \boldsymbol{T}_e^o(\boldsymbol{q}_i)\bar{\boldsymbol{T}}_c^e = \bar{\boldsymbol{T}}_o^b\boldsymbol{T}_{c_i}^o(\boldsymbol{q}_i). \tag{4.60}$$

From the (4.60), the unknown transformation matrix $\bar{\boldsymbol{T}}_c^e$ can be derived as in (4.61) which reformulated in a least square problem yields the solution reported in eq. (4.62).

$$\bar{\boldsymbol{T}}_c^e = \boldsymbol{T}_e^o(\boldsymbol{q}_i)^{-1}\bar{\boldsymbol{T}}_o^b\boldsymbol{T}_{c_i}^o(\boldsymbol{q}_i). \tag{4.61}$$

$$\bar{\boldsymbol{T}}_c^e = \begin{bmatrix} \boldsymbol{T}_e^o(\boldsymbol{q}_1) \\ \vdots \\ \boldsymbol{T}_e^o(\boldsymbol{q}_N) \end{bmatrix}^{\dagger} \begin{bmatrix} \bar{\boldsymbol{T}}_o^b\boldsymbol{T}_{c_1}^o(\boldsymbol{q}_1) \\ \vdots \\ \bar{\boldsymbol{T}}_o^b\boldsymbol{T}_{c_N}^o(\boldsymbol{q}_N) \end{bmatrix} \tag{4.62}$$

4.3.3 Implemented Control Scheme

The proposed thermographic monitoring system has been simulated in Matlab/Simulink with the support of the Robotics Toolbox and Machine Vision Toolbox developed by Peter Corke. The work has been focused on the development of a multi-task multi-priority behaviour-based algorithm in which obstacle avoidance, orientation of camera, positioning of the tool tasks have been exploited. In particular, the redundant robot Yaskawa SIA5F, equipped with an Optris PI450 thermal imaging camera, has been used to monitor the drilling operations of carbon fiber or aluminium materials. An image-based visual servoing algorithm has been developed to keep the features in the FOV of the camera and to keep the optical axis perpendicular to the panel while the drilling robot performs the drilling task. Furthermore, in order to avoid collisions between the monitoring robot and the drilling robot, an obstacle avoidance algorithm has been implemented.

The idea behind the work is based on the fact that when a drilling operation is running, and when the tool is drilling the panel, it creates a warmer zone around the tool similar to a ring. Based on the maximum temperature measured in the area, the

tool wear can be estimated, and/or the material cracking and delamination can be detected. So, the proposed control scheme takes into account two functional phases. When the drilling process is not running no feature can be extracted from the thermal images, therefore, a position control algorithm, coupled with an obstacle avoidance algorithm, has been implemented. The first phase allows the robot to roughly position the camera within the region where the drilling will take place. In the second phase, when the drilling is running, the feature can be properly extracted from the thermographic data, and, thus, the monitoring robot can be exploited to accurately position the camera satisfying the defined objectives. To this aim, in order to properly extract the features from the images, an image-based visual servoing algorithm has been implemented to keep the features in the FOV of the camera and to keep the optical axis perpendicular to the panel while the drilling robot performs the drilling task. Moreover, the obstacle avoidance algorithm has been associated also with the IBVS algorithm to avoid collisions between the monitoring robot and the drilling robot during the drilling process.
The switch between the two functional phases takes place through the measurement of the maximum value of the temperature in the neighborhood of the drilling point. When the maximum measured temperature is less than a defined threshold, the joint velocities computed by using the positional control law are considered, while, when the maximum temperature is greater than the threshold, the joint velocities provided by the IBVS control law are considered. To ensure the continuity of the joint variables computed by the control scheme, and therefore, to ensure the smooth transition between the two functional phases, the outputs of the two considered control laws have been combined through a sigmoidal function. Moreover, from the above, it is clear that in the first control law (position control law) consists of two different tasks, while, the second control law (IBVS control law) consists of three tasks properly prioritized as illustrated in detail Section 4.1.

In the following, the considered kinematic model of the Yaskawa SIA5F is reported, the image processing operations carried out to process the thermographic images are explained, and the two control laws are discussed in details.

4.3.3.1 Yaskawa SIA5F Modeling

In this Section, the D-H table of the Yaskawa SIA5F is reported. With reference to Fig. 4.14, the D-H parameters are reported in Tab. 4.1. The relations between the

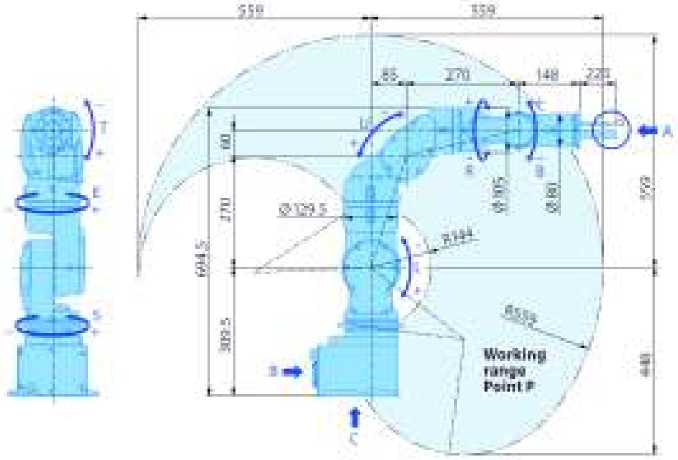


Figure 4.14: Yaskawa SIA5F mechanical scheme.

Joint	α_{i}	$\mathbf{a}_i$ [mm]	$\mathbf{d}_i$ [mm]	$\boldsymbol{\theta}_{\mathrm{i}}$
1	$-\pi/2$	0	0	q_1
2	$\pi/2$	0	0	q_2
3	$\pi/2$	85	270	q_3
4	$\pi/2$	60	0	q_4
5	$-\pi/2$	0	270	q_5
6	$\pi/2$	0	0	q_6
7	0	0	148	q_7

Table 4.1: Yaskawa SIA5F D-H table.

Yaskawa convention and the D-H convention are:

$q_1^{D-H} = q_1^{Yaskawa}$
$q_2^{D-H} = q_2^{Yaskawa}$
$q_3^{D-H} = q_3^{Yaskawa}$
$q_4^{D-H} = q_4^{Yaskawa} - \pi/2$
$q_5^{D-H} = -q_5^{Yaskawa}$
$q_6^{D-H} = q_6^{Yaskawa}$
$q_7^{D-H} = -q_7^{Yaskawa}$

Similarly, the joint velocities and accelerations are:

$\dot{q}_1^{D-H} = \dot{q}_1^{Yaskawa}$
$\dot{q}_2^{D-H} = \dot{q}_2^{Yaskawa}$
$\dot{q}_3^{D-H} = \dot{q}_3^{Yaskawa}$

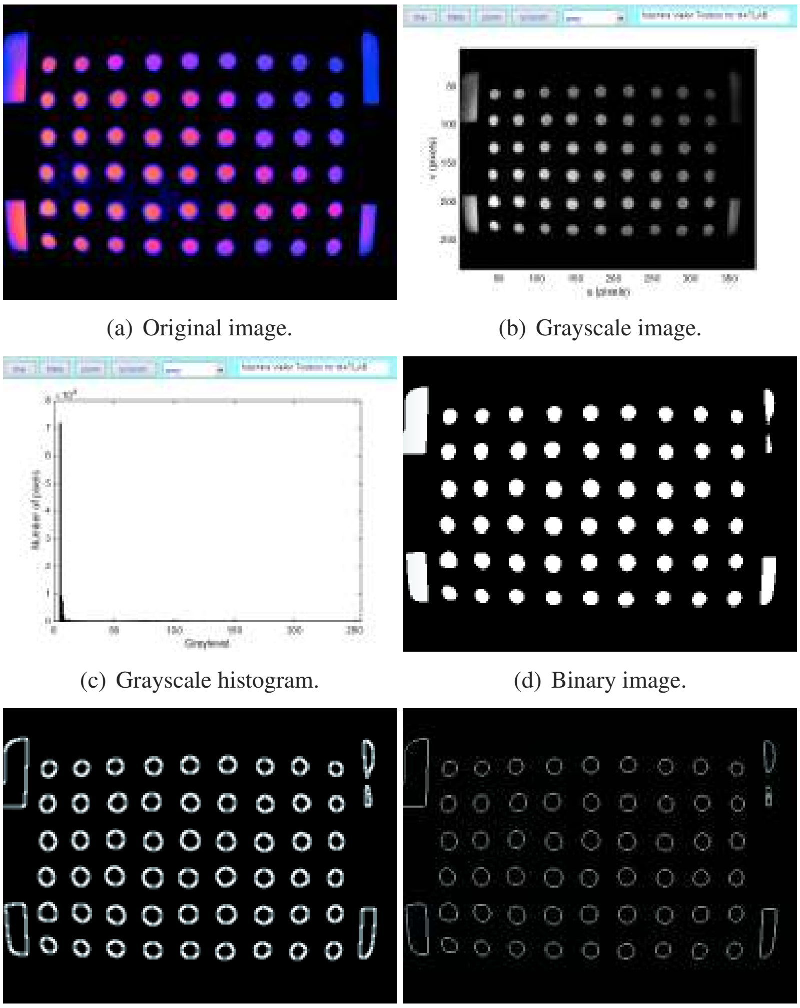
$\dot{q}_4^{D-H} = \dot{q}_4^{Yaskawa}$
$\dot{q}_5^{D-H} = -\dot{q}_5^{Yaskawa}$
$\dot{q}_6^{D-H} = \dot{q}_6^{Yaskawa}$
$\dot{q}_7^{D-H} = -\dot{q}_7^{Yaskawa}$

$\ddot{q}_1^{D-H} = \ddot{q}_1^{Yaskawa}$
$\ddot{q}_2^{D-H} = \ddot{q}_2^{Yaskawa}$
$\ddot{q}_3^{D-H} = \ddot{q}_3^{Yaskawa}$
$\ddot{q}_4^{D-H} = \ddot{q}_4^{Yaskawa}$
$\ddot{q}_5^{D-H} = -\ddot{q}_5^{Yaskawa}$
$\ddot{q}_6^{D-H} = \ddot{q}_6^{Yaskawa}$
$\ddot{q}_7^{D-H} = -\ddot{q}_7^{Yaskawa}$

Moreover, the joint position in D-H convention can be obtained from the joint values reported on the teach pendant of the robot by using the following relations:
$q_1^{D-H} = q_1^{Yaskawa}$
$q_2^{D-H} = -q_2^{Yaskawa}$
$q_3^{D-H} = q_3^{Yaskawa}$
$q_4^{D-H} = q_4^{Yaskawa} + \pi/2$
$q_5^{D-H} = q_5^{Yaskawa}$
$q_6^{D-H} = q_6^{Yaskawa}$
$q_7^{D-H} = q_7^{Yaskawa}$

4.3.3.2 Image Processing

The images acquired from the thermal imaging camera, before being used in the feature extraction process, have to be processed in order to reduce the amount of significant pixels. Starting from the acquired image containing the thermographic data, a binary image can be obtained passing though a thresholding process, a Sobel filter and an erosion process. In particular, starting from the original image (Fig. 4.15(a)), a gray-scale image (Fig. 4.15(b)) can be obtained. The gray-scale image can be used to compute the gray-scale level histogram (Fig. 4.15(c)), required for the calculation of a suitable threshold to be used in the thresholding process. The thresholding process is, then, executed obtaining a binary image (Fig. 4.15(d)). To compute the edge of the region of interest, a Sobel filter can be applied (Fig. 4.15(e)). Finally, in order to reduce the number of pixels to process in the extraction of the features, an erosion



(a) Original image. (b) Grayscale image.

(c) Grayscale histogram. (d) Binary image.

(e) Sobel filter. (f) Erosion process.

Figure 4.15: Image post processing.

filter can be applied on the binary image processed with the Sobel filter (Fig. 4.15(f)). Note that despite the number of pixels is considerably reduced, the continuity of the contours has not been lost. In the images, an example which depicts one of the images used for the camera calibration process, is reported.

All the details on the OpenCV operations, are available in the official OpenCV documentation [92].

4.3.3.3 Position Control Law

The position control block allows to roughly position the thermal imaging camera in proximity of the panel to drill. The position control block, moreover, has been used to avoid the singularity problem that may affect the IBVS control law due to the absence of the feature in the FOV of the camera.
The input reference to the position control block is a pre-computed Cartesian trajectory. Through the CLIK algorithm of the SIA5F, the joint velocities $\dot{\boldsymbol{q}}_{PC}$ are computed. The joint velocities $\dot{\boldsymbol{q}}_{PC}$ are merged with the joint velocity obtained from the obstacle avoidance algorithm by considering the sigmoidal function $\lambda(d)$, whose independent variable d represents the minimum distance between the monitoring robot and the drilling robot (see Section 4.3.3.5). The output of the position control block, then, is the joint velocities $\dot{\boldsymbol{q}}_{PC-OA}$ computed as in (4.63), where $\boldsymbol{J}_{OA}$ is the partial Jacobian of the robot computed until the point of minimum distance, and $\lambda(d)$ is defined as in (4.64) with $\alpha = 50$ and $c = 0.4$ (see Fig. 4.16).

$$\begin{aligned} \dot{\boldsymbol{q}}_{PC-OA} &= \dot{\boldsymbol{q}}_1 + \dot{\boldsymbol{q}}_2 \\ &= (1 - \lambda(d))\, \dot{\boldsymbol{q}}_{PC} + \lambda(d) \left[\dot{\boldsymbol{q}}_{OA} + \left(\boldsymbol{I} - \boldsymbol{J}_{OA}^{\dagger} \boldsymbol{J}_{OA} \right) \dot{\boldsymbol{q}}_{PC} \right] \end{aligned} \tag{4.63}$$

$$\lambda(d) = \frac{1}{1 + e^{-\alpha(c-d)}} \tag{4.64}$$

From (4.63) and (4.64), it results that when the minimum distance d between the monitoring robot and the drilling robot is less than about 0.3 m, only the velocities $\dot{\boldsymbol{q}}_{OA}$ are considered, while, when the distance is greater then 0.5 m about only the velocities $\dot{\boldsymbol{q}}_{PC}$ are taken into account. When the distance is between 0.3 m and 0.5 m, the output of the position control block is a combination of the two joint velocity vectors. Note that the joint velocities $\dot{\boldsymbol{q}}_2$ is a combination of the two considered tasks (see Section 4.1): the obstacle avoidance task has the highest priority and the position control task is projected in the null space of the first task.

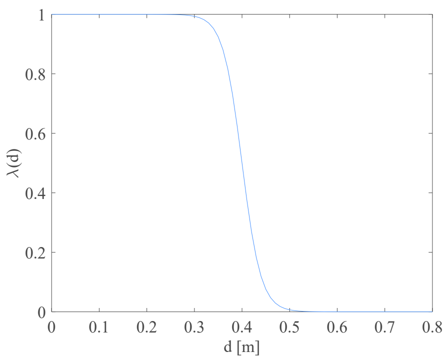


Figure 4.16: Sigmoidal function for position control block.

4.3.3.4 IBVS Control Law

The IBVS control block provides the joint velocities required to achieve the thermographic monitoring activities. In particular, two tasks have been implemented. The first task assures that the thermographic feature is in the FOV of the camera: any reason would bring the camera to lose the features would cause problems of singularity in computing the image Jacobian. The second task ensures that the optical axis is perpendicular to the panel during the drilling process. In fact, to obtain accurate results from the thermographic analysis, it is required that the angle formed by the optical axis and the normal to the panel is as small as possible. A too big angle may affect the detection of deformation and delamination of the material around the hole. The described tasks have been coupled with the obstacle avoidance algorithm already introduced in the previous section. So, a three-task multi-priority algorithm has been implemented for the IBVS control law. In particular, the obstacle avoidance task has the highest priority, while the optical axis task has the lowest priority. The output of the IBVS control law has been computed as in (4.65), where $\boldsymbol{J}_{OA-FOV}$ is the combined Jacobian computed as in (4.66), $\dot{\boldsymbol{q}}_{OA}$ are the joint velocities computed with the obstacle avoidance algorithm, $\dot{\boldsymbol{q}}_{FOV}$ are the output velocities of the FOV task, $\dot{\boldsymbol{q}}_{OPT}$ are the joint velocities provided by the optical axis task, and $\lambda(d)$ is the sigmoidal function described in the previous section. So, the (4.65) yields that, when the minimum distance d is less than 0.3 m about, the velocities $\dot{\boldsymbol{q}}_{3-tasks}$ resulting from the combination of the three proposed tasks are considered, while, when the distance is greater then 0.5 m about the velocities $\dot{\boldsymbol{q}}_{2-tasks}$ obtained by the combination of the FOV task and optical axis task are taken into account. When the distance is between 0.3 m and 0.5 m, the output of the IBVS control law is a combination of both the joint velocity vectors.

$$
\begin{aligned}
\dot{\boldsymbol{q}}_{IBVS-OA} &= \dot{\boldsymbol{q}}_1 + \dot{\boldsymbol{q}}_2 \\
&= \lambda(d)\dot{\boldsymbol{q}}_{3-tasks} + (1-\lambda(d))\,\dot{\boldsymbol{q}}_{2-tasks} \\
&= \lambda(d)\left[\dot{\boldsymbol{q}}_{OA} + \left(\boldsymbol{I} - \boldsymbol{J}_{OA}^{\dagger}\boldsymbol{J}_{OA}\right)\dot{\boldsymbol{q}}_{FOV}\right] \\
&+ (1-\lambda(d))\left[\left(\boldsymbol{I} - \boldsymbol{J}_{OA-FOV}^{\dagger}\boldsymbol{J}_{OA-FOV}\right)\dot{\boldsymbol{q}}_{OPT}\right]
\end{aligned} \tag{4.65}
$$

$$
\boldsymbol{J}_{OA-FOV} = \begin{bmatrix} \boldsymbol{J}_{OA} \\ \boldsymbol{J}_{FOV} \end{bmatrix} \tag{4.66}
$$

The FOV task and the optical axis task are described in details in the following sections.

4.3.3.4.1 FOV Task The FOV task [93] depends on suitable function or moments that can be extracted from the camera images, and it aims at keeping the features in the camera field of view. As described above, the features can be conceived as a ring around the drilling hole, from which a set of points constituting an ellipse can be extrapolated. The mean and the variance of such a set of points may be used to keep the feature points in the neighborhood of the center of the image plane and to constrain the distance of the camera form the object.
Let define the mean s_m of feature points as

$$
\boldsymbol{s}_m = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{s}_i = [x_m\ y_m]^T. \tag{4.67}
$$

Its time derivative is given by

$$
\dot{\boldsymbol{s}}_m = \boldsymbol{J}_m\dot{\boldsymbol{s}} \tag{4.68}
$$

where $\boldsymbol{J}_m$ is a $(2 \times 2k)$ matrix computed as

$$
\boldsymbol{J}_m = \frac{1}{k}\left[\boldsymbol{I}_2, \cdots, \boldsymbol{I}_2\right]. \tag{4.69}
$$

Analogously, the variance of the feature points can be defined as in (4.70), where the corresponding Jacobian $\boldsymbol{J}_v$ is a $(2 \times 2k)$ matrix that relates the time derivative of $\boldsymbol{s}_v$ to $\dot{\boldsymbol{s}}$ is given by eq. (4.71).

$$
\boldsymbol{s}_v = \frac{1}{k}\sum_{i=1}^{k}\begin{bmatrix}(x_i - x_m)^2 \\ (y_i - y_m)^2\end{bmatrix} \tag{4.70}
$$

$$
\begin{aligned}
\dot{\boldsymbol{s}}_v &= \boldsymbol{J}_v \dot{\boldsymbol{s}} \\
&= \frac{2}{k} \begin{bmatrix} x_1 - x_m & 0 & \cdots & x_k - x_m & 0 \\ 0 & y_1 - y_m & \cdots & 0 & y_k - y_m \end{bmatrix} \dot{\boldsymbol{s}}
\end{aligned} \tag{4.71}
$$

Note that $\boldsymbol{J}_v$ is singular when all the points are collinear.

The complete task has been defined as:

$$
\begin{aligned}
\dot{\boldsymbol{s}}_{mv} &= \boldsymbol{J}_{mv} \dot{\boldsymbol{s}}, \\
\boldsymbol{J}_{mv} &= \begin{bmatrix} \boldsymbol{J}_m \\ \boldsymbol{J}_v \end{bmatrix},
\end{aligned} \tag{4.72}
$$

from which the joint velocities of the robot can be computed as in (4.73), where $\boldsymbol{L}$ is the image Jacobian of a set of points (4.34), and $\boldsymbol{J}^c$ is the robot Jacobian expressed in camera frame (4.74).

$$
\dot{\boldsymbol{q}}_{FOV} = (\boldsymbol{J}_{mv} \boldsymbol{L} \boldsymbol{J}^c)^{\dagger} \begin{bmatrix} \dot{\boldsymbol{s}}_{m-com} \\ \dot{\boldsymbol{s}}_{v-com} \end{bmatrix} \tag{4.73}
$$

$$
\boldsymbol{J}^c = \begin{bmatrix} \boldsymbol{R}_c^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_c^T \end{bmatrix} \boldsymbol{J} \tag{4.74}
$$

A possible choice of the command velocities $\dot{\boldsymbol{s}}_{m-com}$ and $\dot{\boldsymbol{s}}_{v-com}$ for the mean and variance tasks, can be given by

$$
\begin{aligned}
\dot{\boldsymbol{s}}_{m-com} &= \dot{\boldsymbol{s}}_{m-des} + \gamma_1 (\boldsymbol{s}_{m-des} - \boldsymbol{s}_m), \\
\dot{\boldsymbol{s}}_{v-com} &= \dot{\boldsymbol{s}}_{v-des} + \gamma_2 (\boldsymbol{s}_{v-des} - \boldsymbol{s}_v),
\end{aligned} \tag{4.75}
$$

with γ_1 and γ_2 positive gains.

4.3.3.4.2 Optical Axis Task The optical axis task allows regulating to zero an error vector based on the current and goal images. The extracted features are the ellipse features (see Section 4.2.1.2.5) due to the fact that the heating region around the drilling points can be considered to be a ring. In order to have a uniform view of the drilling area, the aim is to keep the optical axis perpendicular to the part of the panel in which the drilling operation occurs.

Let consider the following definition of the feature error, where the vector $\dot{\boldsymbol{s}}_e$ is the ellipse feature vector defined in Section 4.2.1.2.5:

$$
\boldsymbol{e} = \dot{\boldsymbol{s}}_e + \boldsymbol{K} (\boldsymbol{s}_{e-des} - \boldsymbol{s}_e), \tag{4.76}
$$

with $\boldsymbol{K}$ a constant positive definite matrix. The feature error vector $\boldsymbol{e}$ can be related to the camera velocity $\boldsymbol{v}_r^c$ through the ellipse image Jacobian (4.39) $\boldsymbol{L}_s$:

$$\boldsymbol{v}_r^c = \boldsymbol{L}_s^{\dagger}\boldsymbol{e}. \tag{4.77}$$

Taking into account the equations introduced above, the joint velocities of the robot can be obtained from (4.78).

$$\dot{\boldsymbol{q}}_{OPT} = \boldsymbol{J}^{c^{\dagger}}\boldsymbol{v}_r^c = \boldsymbol{J}^{\dagger}\begin{bmatrix}\boldsymbol{R}_c & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}_c\end{bmatrix}\boldsymbol{L}_s^{\dagger}\boldsymbol{e} \tag{4.78}$$

4.3.3.5 Obstacle Avoidance Algorithm

The distance vector between any point on the monitoring robot $\boldsymbol{P}$ and any point $\boldsymbol{O}$ on the drilling robot $\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})$ is assumed to be known. When the distance $d = \|\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})\|$ is measured to be less than $d_{min} = \|\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})\|_{min}$, a repulsive joint velocity vector is produced and used to modify on-line the current trajectory of the monitoring robot. A simple repulsive vector can be defined as:

$$\boldsymbol{V}_{rep} = V_{max}\frac{\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})}{\|\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})\|}. \tag{4.79}$$

The defined repulsive vector has the same direction as $\boldsymbol{D}(\boldsymbol{P},\boldsymbol{O})$, and V_{max} is the maximum admissible speed in the Cartesian space. The corresponding joint velocities can be computed by considering the transpose of the partial Jacobian associated with the point $\boldsymbol{P}$:

$$\dot{\boldsymbol{q}}_{OA} = \boldsymbol{J}_P^T\boldsymbol{V}_{rep}. \tag{4.80}$$

4.3.3.6 Simulations

The joint velocities obtained from the positional control block and the IBVS control block have been merged by considering a sigmoidal function $\lambda(T)$ as in (4.81), whose independent variable T is the maximum temperature measured with the thermal imaging camera near the drilling point. The parameters α and c have been chosen equal to 0.2 and 45, respectively (see Fig. 4.17).

$$\lambda(T) = \frac{1}{1+e^{-\alpha(c-T)}} \tag{4.81}$$

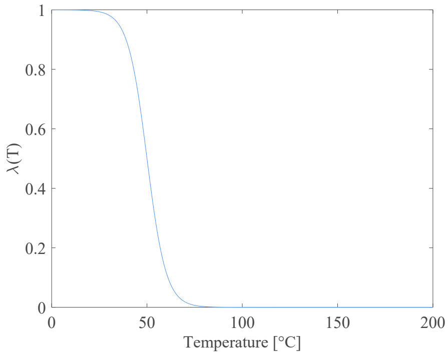


Figure 4.17: Sigmoidal function for velocities merging.

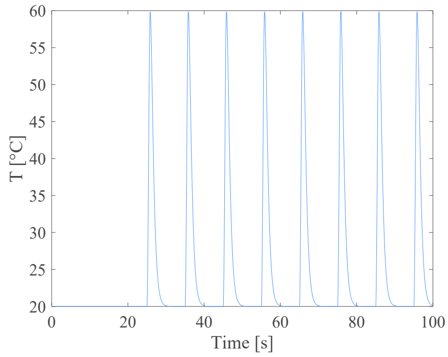


Figure 4.18: Temperature signal.

The simulated work-cell consisted by the 7-dofs Yaskawa SIA5F robot only where a sphere moves near the robot simulating an obstacle. The drilling process has been simulated by considering a periodic signal, a delayed square wave with a duty cycle of 8%, period 10 s, amplitude of 42.5 °C and bias of 20 °C filtered with a second order low-pass filter (see Fig. 4.18). The control parameters have been set as: $\gamma_1 = 3$, $\gamma_2 = 3$, $\boldsymbol{K} = \mathrm{diag}(0.5, ..., 0.5)$ and $V_{max} = 0.2$ m/s. In order to reduce the occurrences of the singularities of the image Jacobian in the optical axis task, an augmented system has been considered: in addition to the five ellipse features, three point features extracted by three points evenly spaced on the ellipse have been used. Moreover, in the presented simulations, the camera frame has been considered to be coincident with the end-effector frame and the calibration matrix of the camera is chosen to be:

$$\mathbf{\Theta} = \begin{bmatrix} 800 & 0 & 512 \\ 0 & 800 & 512 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.82}$$

Two case studies have been analyzed. In the first case study, the obstacle moves near to the third link of the robot, while, in the second case study, the obstacle moves near to the sixth link of the robot. For both the cases, the desired features have been computed by considering the manipulator in the configuration

$$\boldsymbol{q}_d = [0,\ \pi/4,\ 0,\ \pi/4,\ 0,\ \pi/2,\ \pi/2]^T\ . \tag{4.83}$$

An ellipse constituted by ten points has been positioned in front of the robot to simulate the ring that appears on the panel during the drilling phase. The constants that parameterize such ellipse are reported in (4.84), where $x_{0_{e-e}}$ is the x coordinate of the end-effector when the robot is in the desired pose $\boldsymbol{q}_d$.

$$a = 1,\ b = 0,\ c = 0,\ d = -(x_{0_{e-e}} + 0.7) \tag{4.84}$$

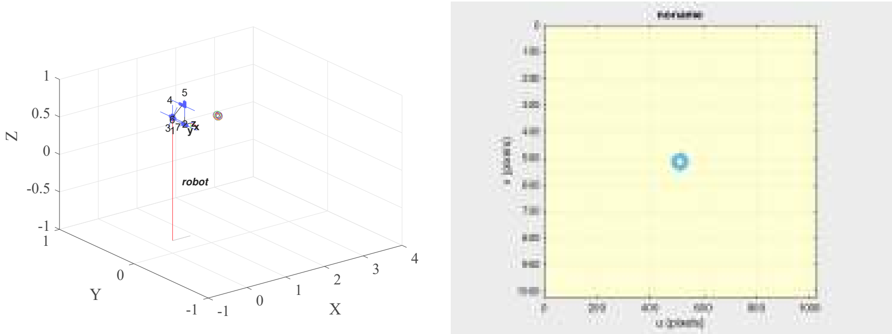
As described in the previous sections, the IBVS algorithm has been used to adjust the pose of the end effector during the drilling phase, too. So, only a rough positioning that allows the camera to keep the features in the FOV is required. To simulate this behaviour, the reference pose in input to the position control block was different to the desired pose and it has been chosen as (position in meters and orientation in degrees):

$$\boldsymbol{x}_f = [0.4046,\ 0.0560,\ 0.6882,\ 4.5089,\ 86.4111,\ -90.2828]^T\ , \tag{4.85}$$

$$\boldsymbol{x}_d = [0.4590,\ 0.0000,\ 0.6708,\ 0.0000,\ 90.0000,\ -90.0000]^T\ . \tag{4.86}$$

Moreover, an initial trajectory of three seconds has been considered to bring the robot from the initial configuration $\boldsymbol{q}_i$, corresponding to the camera pose $\boldsymbol{x}_i$ (4.87), to the final configuration $\boldsymbol{q}_f$, corresponding to the final camera pose $\boldsymbol{x}_f$ (4.85).

$$\boldsymbol{x}_i = [0.2197,\ 0.0971,\ 0.6281,\ 12.6276,\ 79.2127,\ -85.7664]^T \tag{4.87}$$



(a) Robotics toolbox view. (b) Desired features on the image plane.

Figure 4.19: Desired configuration for the feature extraction.

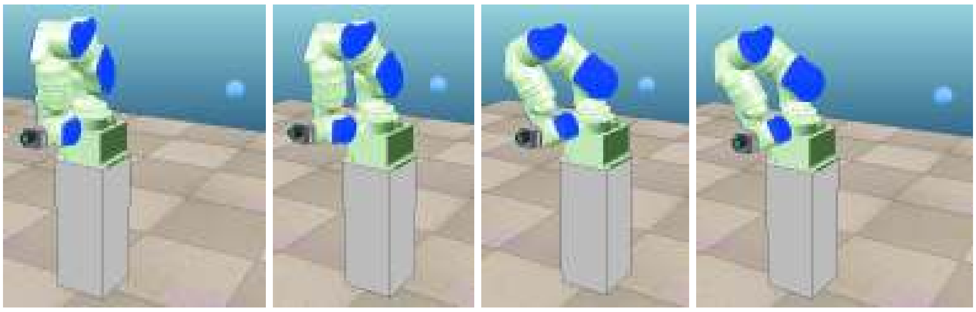
In eq. (4.88) and in eq. (4.89), the initial features and the desired features are reported.

$$
\begin{aligned}
\boldsymbol{s}_{e_i} &= [1.0000,\ 0.0000,\ 0.0000,\ 0.0000,\ -0.0008,\ 512.0000\ \cdots \\
&\cdots\ 534.8571,\ 490.2616,\ 519.0632,\ 525.4351,\ 493.5082]^T\ , \\
\boldsymbol{s}_{m_i} &= [790.1770,\ 605.0272]^T \\
\dot{\boldsymbol{s}}_{m_i} &= [0.0000,\ 0.0000]^T \\
\boldsymbol{s}_{v_i} &= [200.4919,\ 182.4392]^T \\
\dot{\boldsymbol{s}}_{v_i} &= [0.0000,\ 0.0000]^T
\end{aligned}
\tag{4.88}
$$

$$
\begin{aligned}
\boldsymbol{s}_{e_d} &= [1.0000,\ 0.0000,\ 0.0000,\ 0.0000,\ -0.0008,\ 512.0000\ \cdots \\
&\cdots\ 534.8571,\ 490.2616,\ 519.0632,\ 525.4351,\ 493.5082]^T\ , \\
\boldsymbol{s}_{m_d} &= [512.0000,\ 512.0000]^T \\
\dot{\boldsymbol{s}}_{m_d} &= [0.0000,\ 0.0000]^T \\
\boldsymbol{s}_{v_d} &= [290.2494,\ 290.2494]^T \\
\dot{\boldsymbol{s}}_{v_d} &= [0.0000,\ 0.0000]^T
\end{aligned}
\tag{4.89}
$$

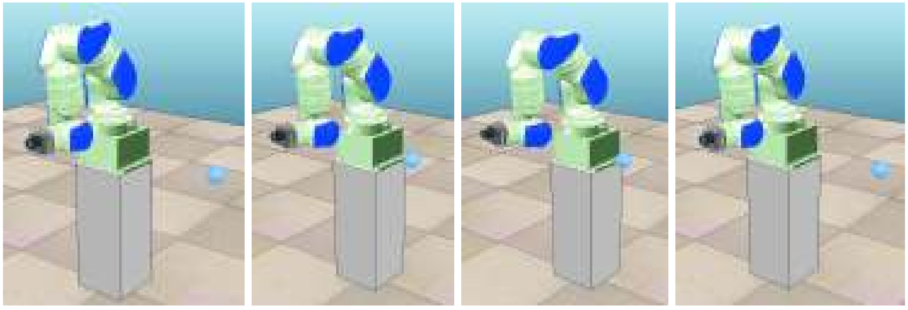
Figure 4.19 shows a robotics toolbox view of the robot in the desired pose and the ellipse points on the left and the ellipse points projected on the image plane on the right, while, in Fig. 4.22 the implemented simulink block diagram is reported.

The results of the proposed simulations are reported below. Figure 4.20 and Figure 4.21 show four frames portraying the robot during the obstacle avoidance action for the first and the second case study, respectively. The figures show that the end-effector pose changes (in the second case it changes significantly) due to the fact



(a) $d > \rho_{min}$. (b) $d < \rho_{min}$. (c) $d << \rho_{min}$. (d) $d > \rho_{min}$.

Figure 4.20: Visual servoing: case 1 - obstacle avoidance.



(a) $d > \rho_{min}$. (b) $d < \rho_{min}$. (c) $d << \rho_{min}$. (d) $d > \rho_{min}$.

Figure 4.21: Visual servoing: case 2 - obstacle avoidance.

that the obstacle avoidance task has the highest priority. When the obstacle moves away from the robot, the position control error converges to zero and the camera frame returns to the desired value.

The joint velocities generated by each task, the feature errors, the commanded joint velocities and positions are reported in Figures 4.23, 4.24, 4.25 and 4.26 for both the case studies. It is clear that the obstacle introduces a disturbance in both the case studies so that the feature errors (Fig. 4.24(d)(e)(f), Fig. 4.26(d)(e)(f)) increase accordingly. Contrarily, the feature errors decrease when the drilling actions are acting, demonstrating the effectiveness of the implemented control algorithm. Moreover, the trend of the minimum distance d (Fig. 4.23(a), Fig. 4.25(a)) between the robot and the obstacle, shows that the distance is always grater than the minimum value ρ_{min} set to 0.4 m, while, simultaneously, the other tasks are assured exploiting the redundant joint of the robot as the feature error trends show. Finally, although the joint velocities generated by each task exceed the maximum admissible velocity

(200 °/s for the SIA5F), the commanded joint velocities are in the nominal range.

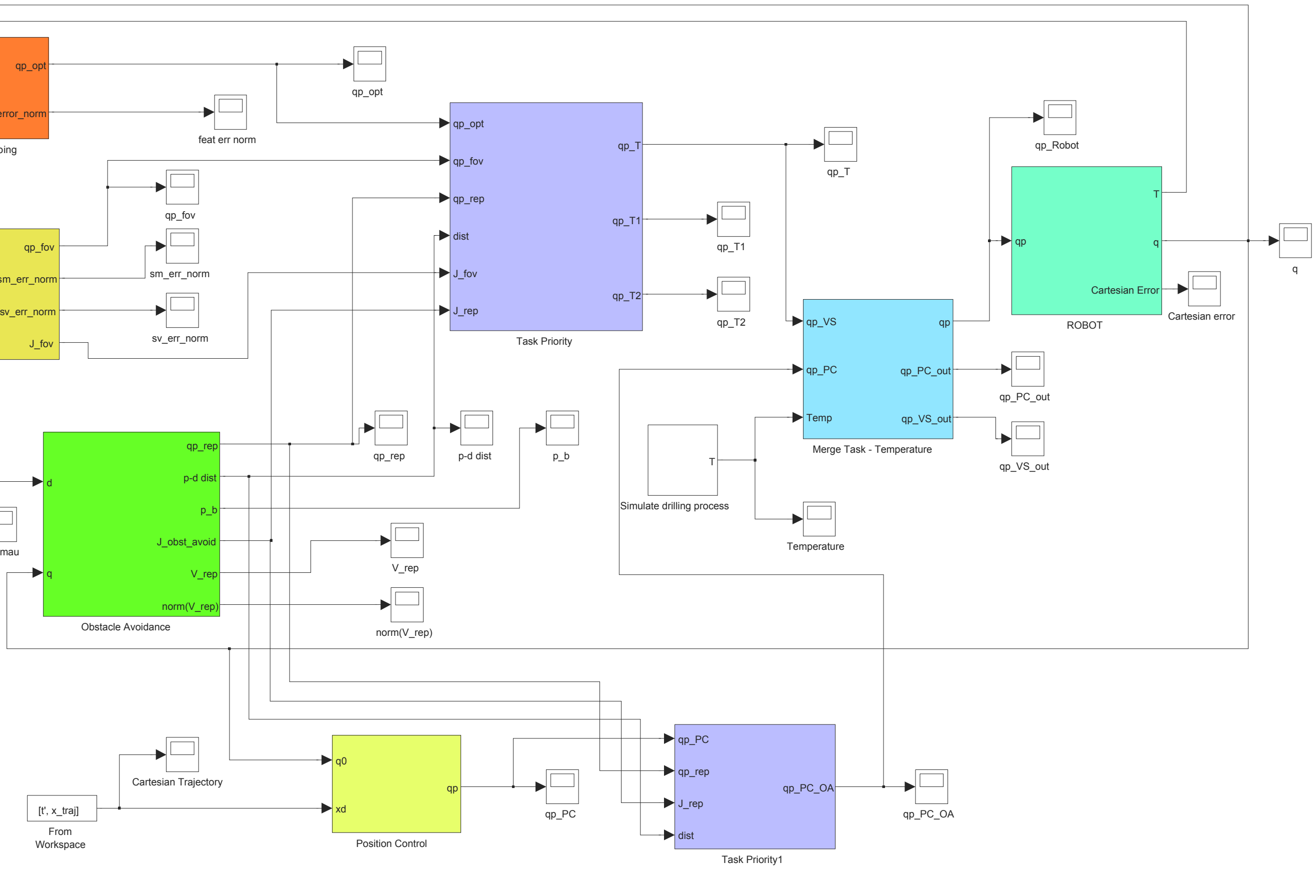
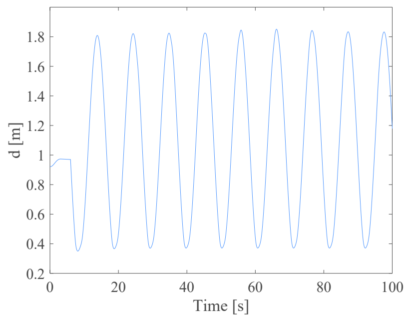
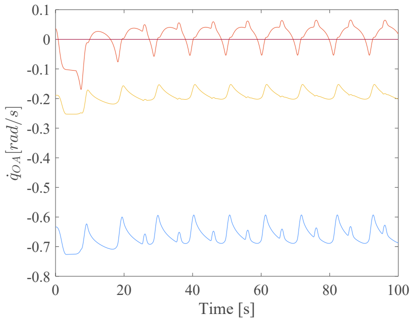


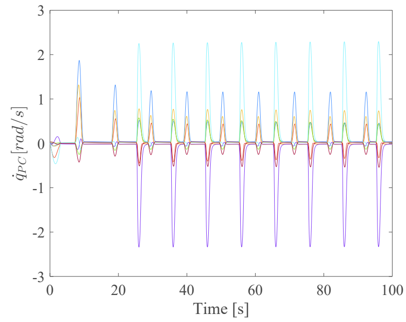
Figure 4.22: Implemented block diagram.



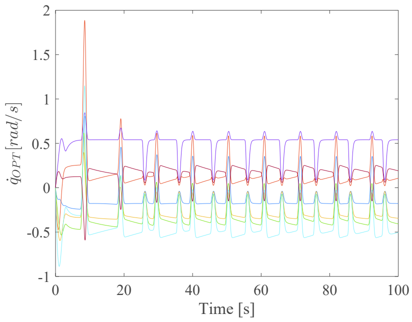
(a) d.



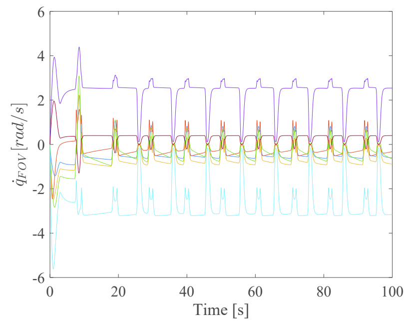
(b) Joint velocities - OA task.



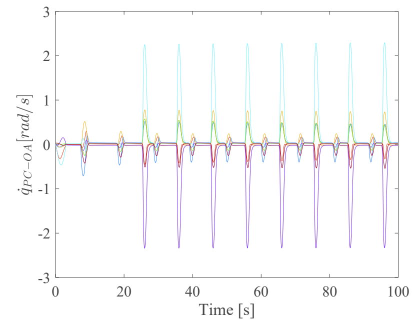
(c) Joint velocities - PC task.



(d) Joint velocities - OPT task.

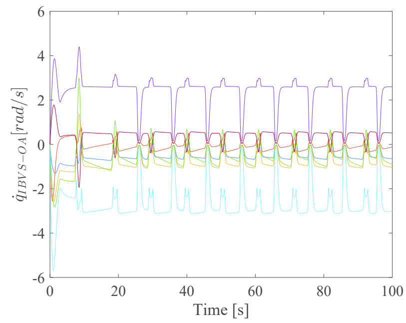


(e) Joint velocities - FOV task.

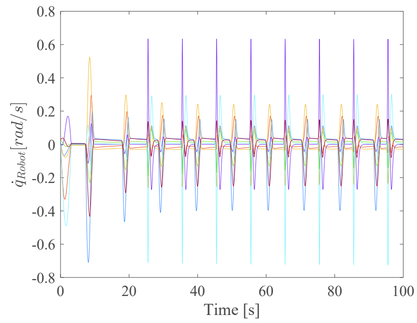


(f) Joint velocities - PC control block.

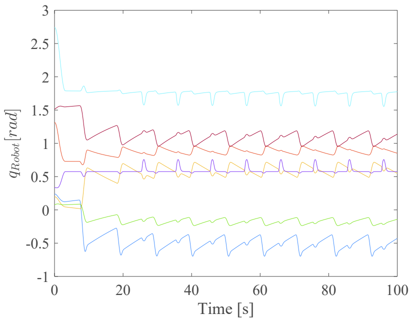
Figure 4.23: Visual servoing: case 1 - simulation results.



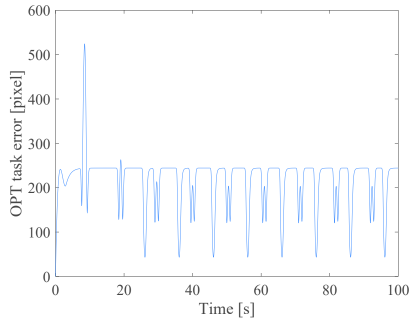
(a) Joint velocities - IBVS control block.



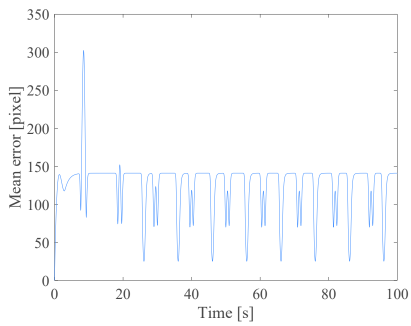
(b) Commanded joint velocities.



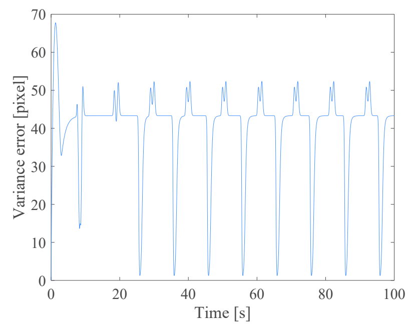
(c) Commanded joint positions.



(d) OPT task error.

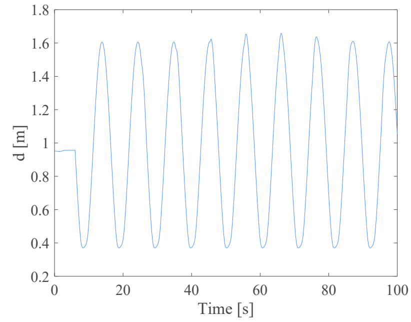


(e) FOV task - mean error.

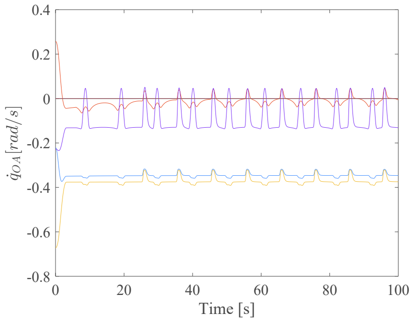


(f) FOV task - variance error.

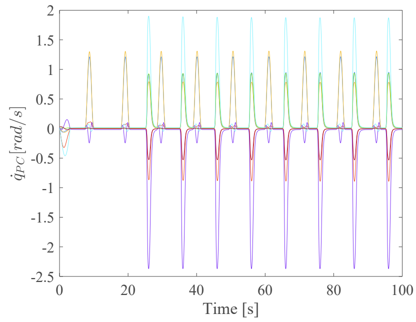
Figure 4.24: Visual servoing: case 1 - simulation results (continue).



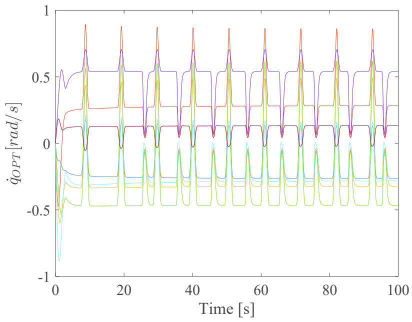
(a) d.



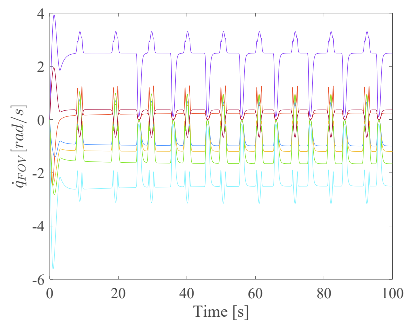
(b) Joint velocities - OA task.



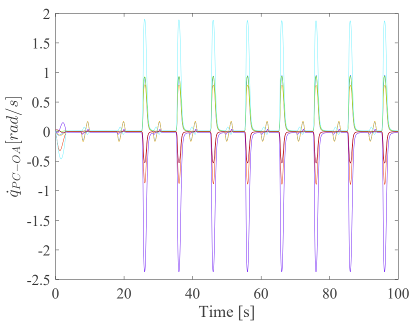
(c) Joint velocities - PC task.



(d) Joint velocities - OPT task.

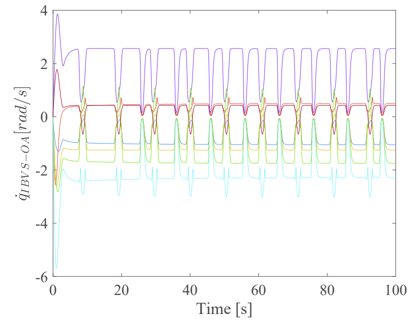


(e) Joint velocities - FOV task.

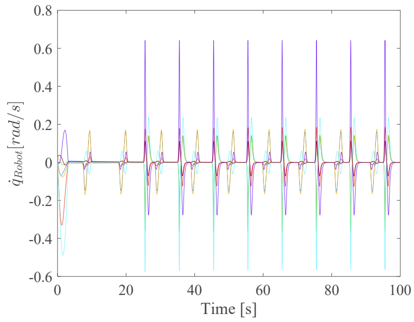


(f) Joint velocities - PC control block.

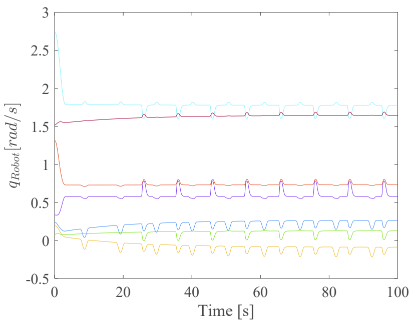
Figure 4.25: Visual servoing: case 2 - simulation results.



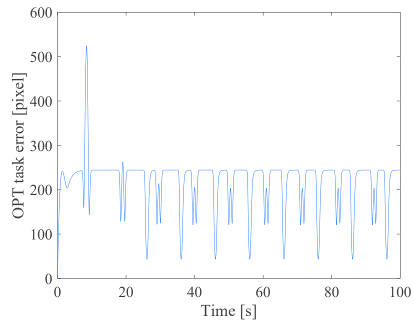
(a) Joint velocities - IBVS control block.



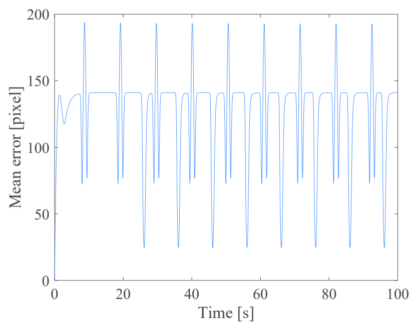
(b) Commanded joint velocities.



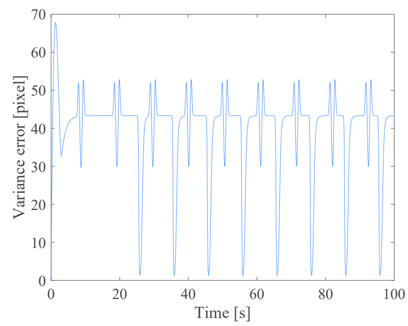
(c) Commanded joint positions.



(d) OPT task error.



(e) FOV task - mean error.



(f) FOV task - variance error.

Figure 4.26: Visual servoing: case 2 - simulation results (continue).

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

This thesis presented the development of new methodologies and the integration of existing ones addressed to achieve a “lean” process for the assembly operations in the aeronautics industry. The activities have been focused on development of a robotized drilling solution that makes use of a force/moment controller in a cooperative dual arm robotic cell, on the development of a new methodology to support the design of parallel robots to be used into a flexible fixture for the positioning of the aeronautic parts (such as ribs and spars), and on the development of a multi-task multi-priority control system exploited for the real-time thermographic monitoring of the robotized drilling process.

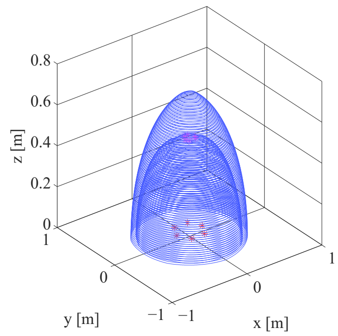
In the first part, the efforts regarded the development of a 14 degrees of freedom cooperative dual arm robotic cell. The use of cooperative robotics solution, coupled with the use of a force sensor, has been useful for both the analyzed drilling methods: the drilling with jigs and without jigs. Simulations and experiments have been presented to validate the effectiveness of the proposed solution. Future works will concern the validation of the simulations in case of drilling without jigs. The experiments will be carried out as soon as the required instrumentation will be ready.

In the automatic part positioning topic, the activities focused on the development of a simulation environment and an optimization tool to support the design of ad-hoc Stewart platforms for specific applications. In particular, in order to maximize the payload and improve the rejection of external forces exerted on the mobile platform during positioning or manufacturing applications, e.g., drilling process, a dynamic optimization has been carried out. Moreover, in order to avoid reduction of the robot workspace, also a kinematic optimality criterion has been considered in the optimization process as well. A new two-stage Genetic Algorithm has been used to combine the two different optimum objectives by properly defining a cost function to minimize. The proposed solution has been exploited for the development of two different Stewart platforms used by the LOCOMACHS consortium into the physical demonstrator to achieve the accurate positioning of the front spar and rib 3 in the LOCOMACHS wing-box.

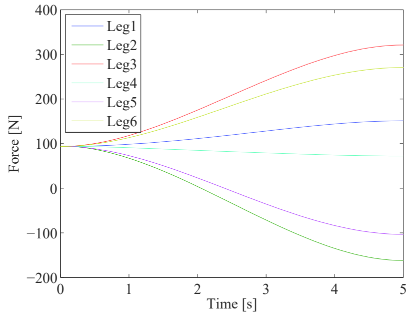
The third part of the thesis concerned the development of a multi-task multi-priority control algorithm exploited for the real-time thermographic monitoring of the robotized drilling process. The idea was to utilize thermography information provided by a thermal imaging camera to monitor the tool wear or to detect possible damages caused to materials due to the high temperature produced during the machining phase and, then, to modify in real-time the process parameters to avoid the delamination and the deformation. The presented solution will be used within the STEP FAR project for the analysis of the drilling process of carbon fiber panels.

APPENDIX A

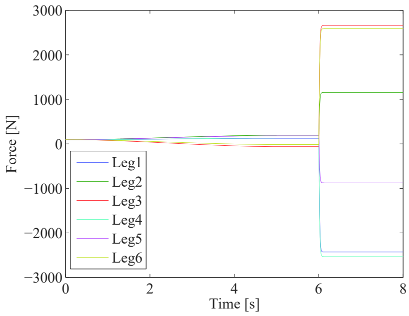
STEWART PLATFORM OPTIMAL DESIGNS



(a) Workspace. $V_W = 0.20086\,\mathrm{m}^3$.

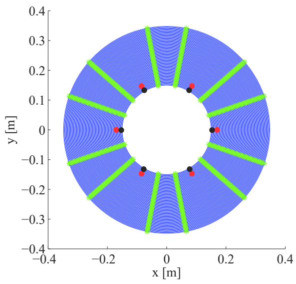


(b) Leg forces - case study I.

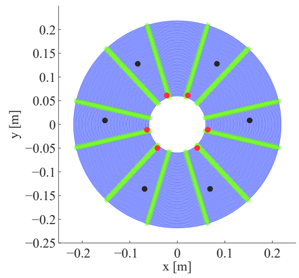


(c) Leg forces - case study II.

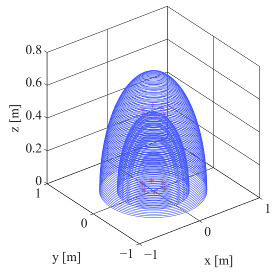
Figure A.1: Initial design.



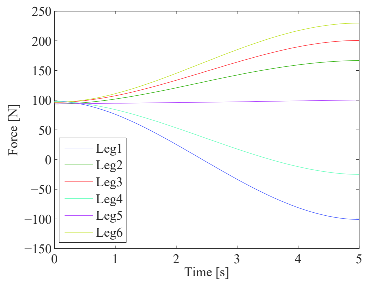
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



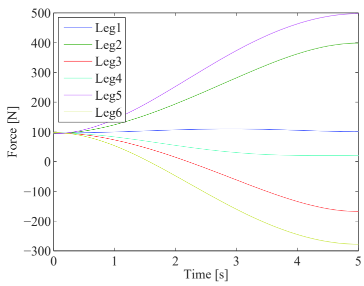
(b) Leg attachment points on top plate. In red initial design; in black optimized design.



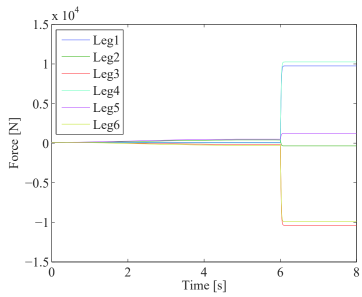
(c) Workspace. $V_W = 0.52848\,\mathrm{m}^3$.



(d) Actuation leg forces - trajectory 1.

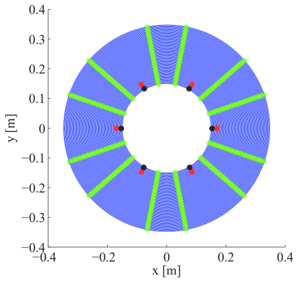


(e) Actuation leg forces - trajectory 2.

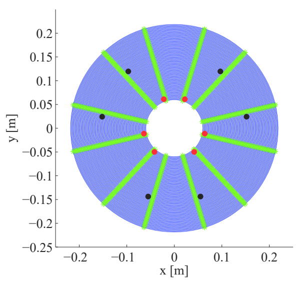


(f) Actuation leg forces - trajectory 1 with external force.

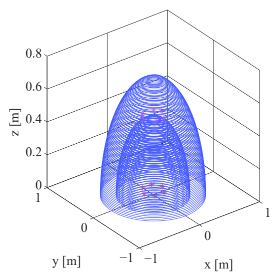
Figure A.2: Stewart-Gough Geometry (4 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 100$.



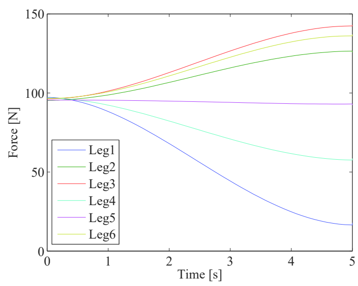
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



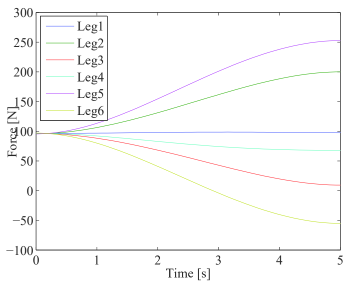
(b) Leg attachment points on top plate. In red initial design; in black optimized design.



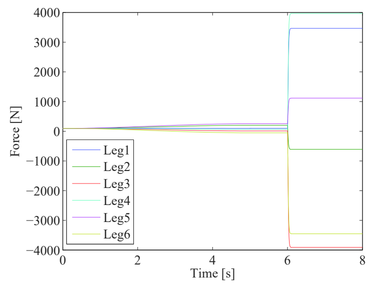
(c) Workspace. $V_W = 0.48072\,\mathrm{m}^3$.



(d) Actuation leg forces - trajectory 1.

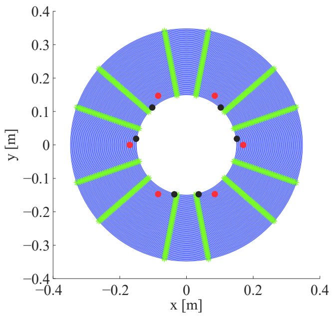


(e) Actuation leg forces - trajectory 2.

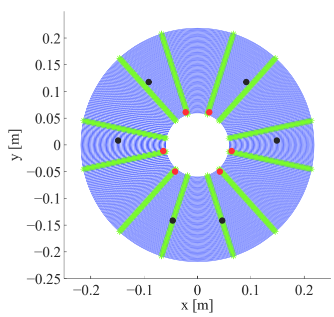


(f) Actuation leg forces - trajectory 1 with external force.

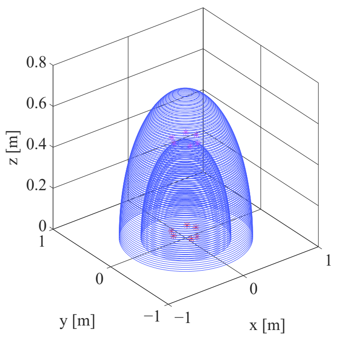
Figure A.3: Stewart-Gough Geometry (4 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 10$.



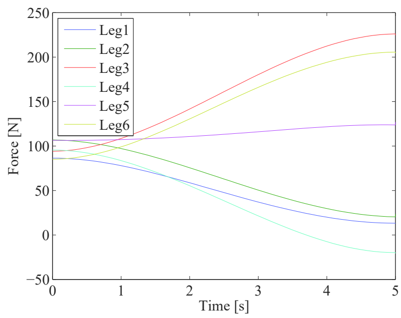
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

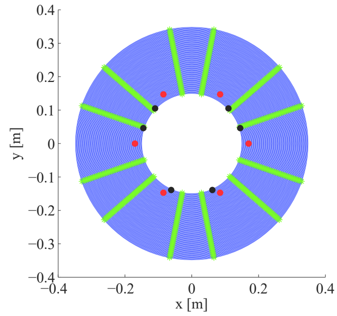


(c) Workspace. $V_W = 0.52096\,\mathrm{m}^3$.

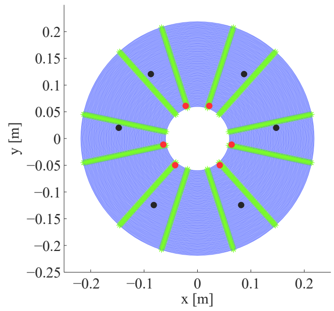


(d) Actuation leg forces - trajectory 1.

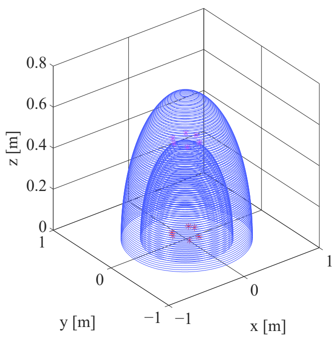
Figure A.4: Stewart-Gough Geometry (8 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 100$.



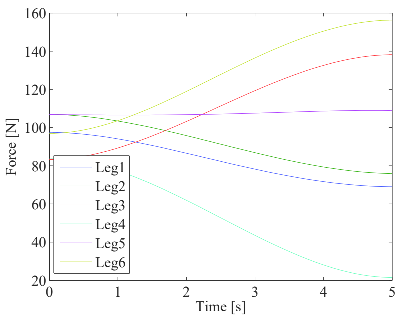
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

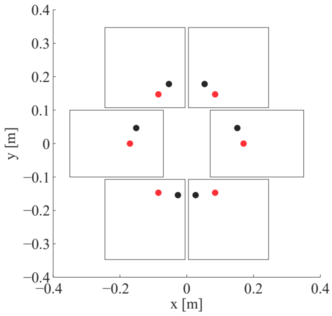


(c) Workspace. $V_W = 0.47604\,\mathrm{m}^3$.

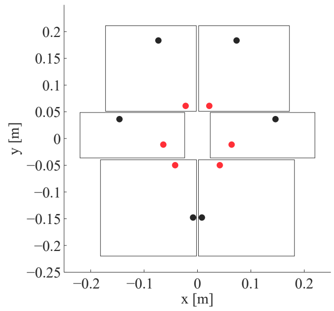


(d) Actuation leg forces - trajectory 1.

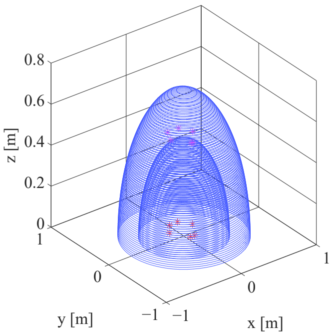
Figure A.5: Stewart-Gough Geometry (8 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 10$.



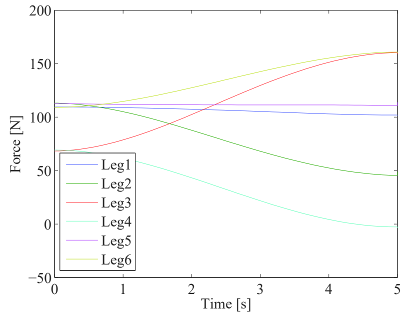
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

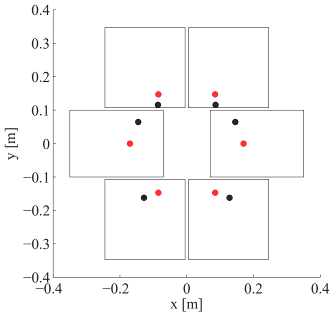


(c) Workspace. $V_W = 0.50422\,\mathrm{m}^3$.

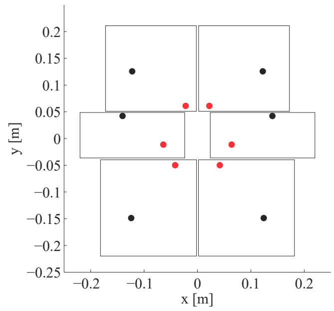


(d) Actuation leg forces - trajectory 1.

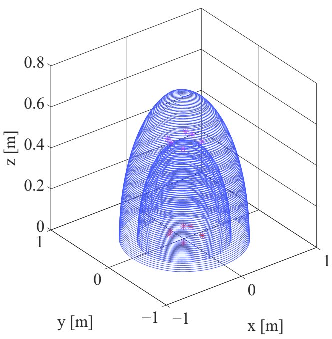
Figure A.6: One Axis Geometry (12 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 100$.



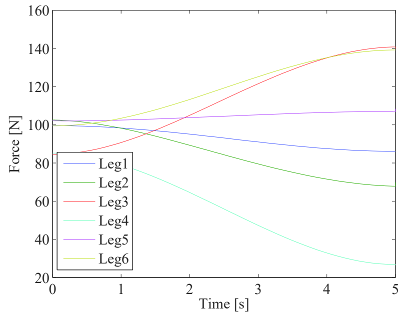
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

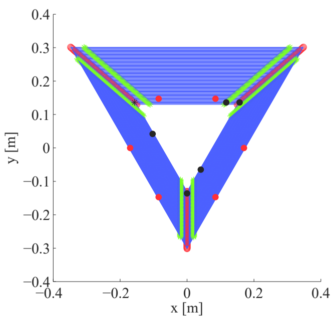


(c) Workspace. $V_W = 0.45917\,\mathrm{m}^3$.

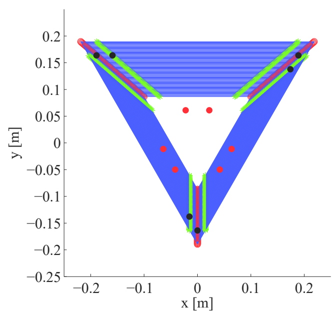


(d) Actuation leg forces - trajectory 1.

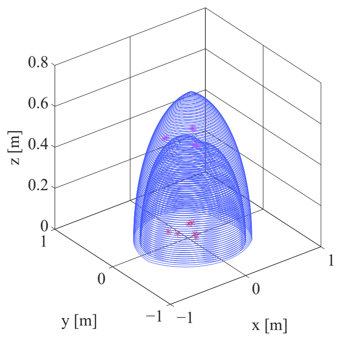
Figure A.7: One Axis Geometry (12 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 10$.



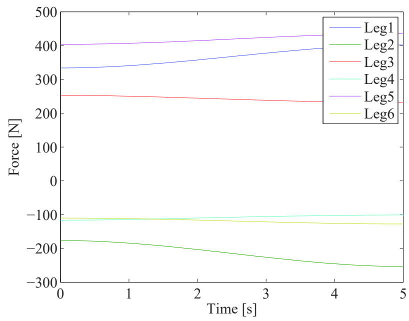
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

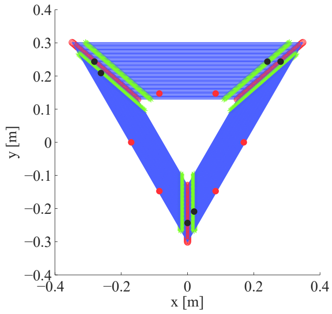


(c) Workspace. $V_W = 0.231880\,\mathrm{m}^3$.

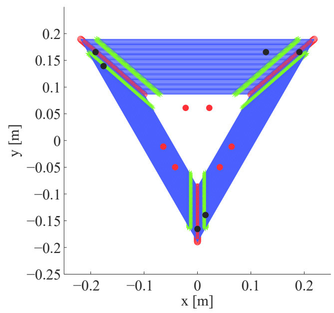


(d) Actuation leg forces - trajectory 1.

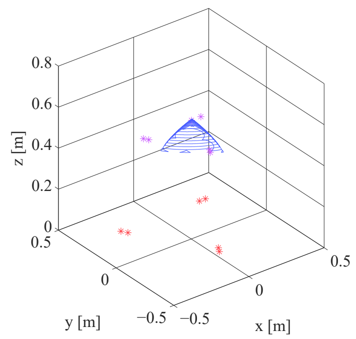
Figure A.8: Griffis-Duffy Geometry (2+6 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 100$.



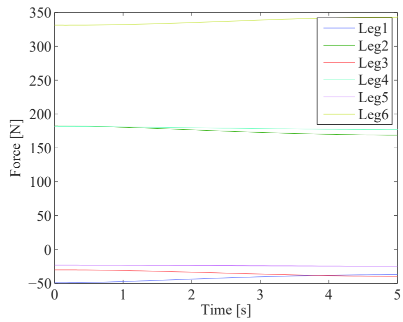
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

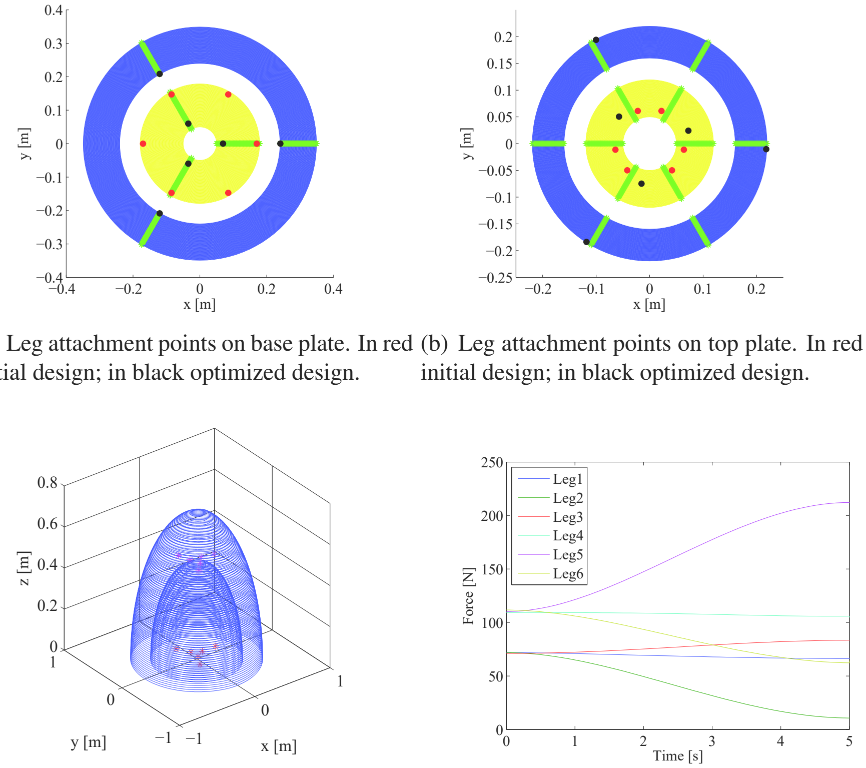


(c) Workspace. $V_W = 0.0023545\,\mathrm{m}^3$.



(d) Actuation leg forces - trajectory 1.

Figure A.9: Griffis-Duffy Geometry (2+6 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 10$.



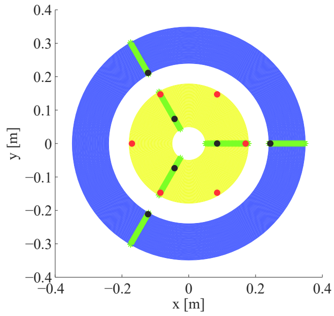
(a) Leg attachment points on base plate. In red initial design; in black optimized design.

(b) Leg attachment points on top plate. In red initial design; in black optimized design.

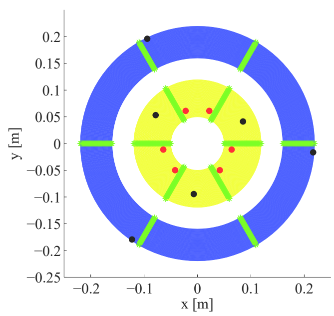
(c) Workspace. $V_W = 0.48334\,\mathrm{m}^3$.

(d) Actuation leg forces - trajectory 1.

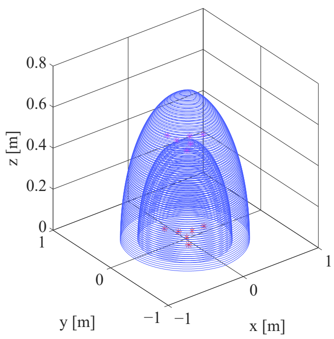
Figure A.10: MSP Geometry (6 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 100$.



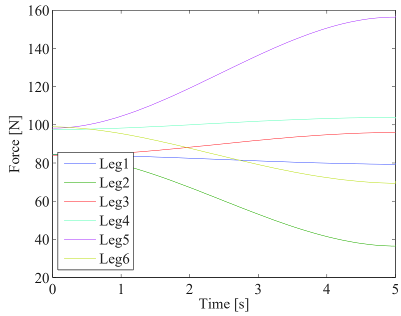
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

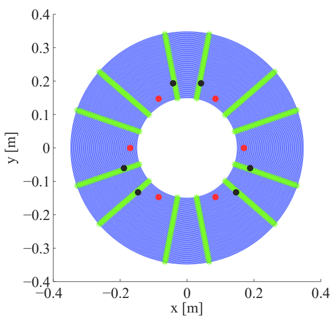


(c) Workspace. $V_W = 0.43504\,\mathrm{m}^3$.

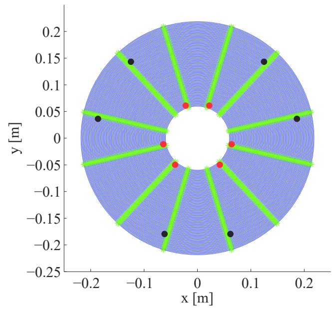


(d) Actuation leg forces - trajectory 1.

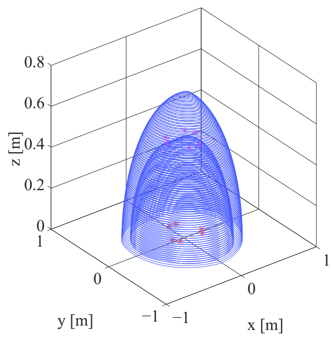
Figure A.11: MSP Geometry (6 Variables) - Case study I: $k_1 = 0.1$, $k_2 = 10$.



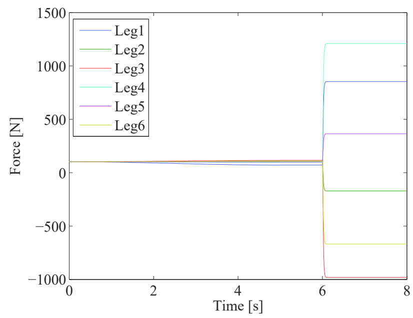
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

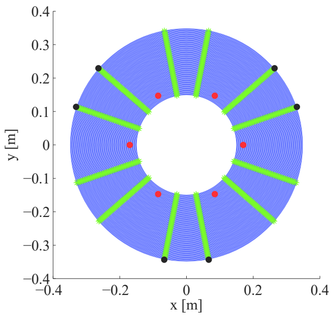


(c) Workspace. $V_W = 0.27014\,\text{m}^3$.

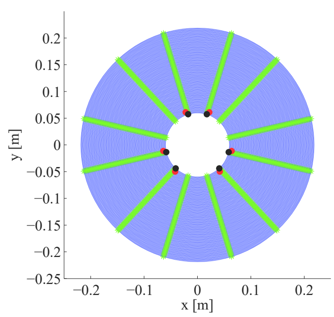


(d) Actuation leg forces - trajectory 1.

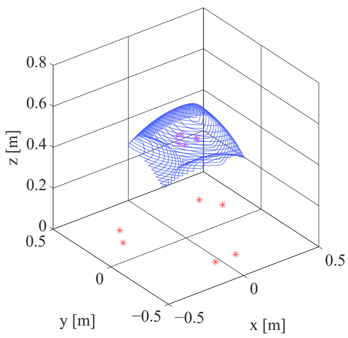
Figure A.12: Stewart-Gough Geometry (4 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 100$.



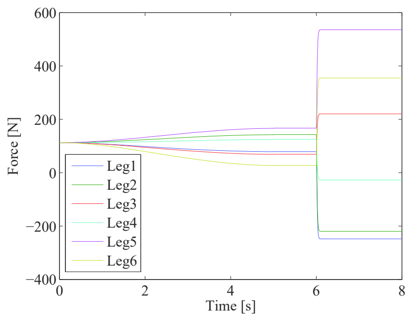
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

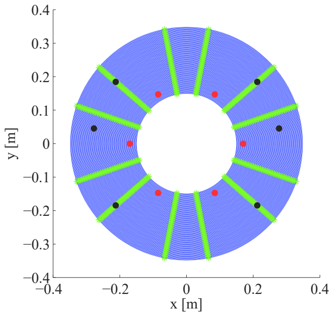


(c) Workspace. $V_W = 0.025509\,\mathrm{m}^3$.

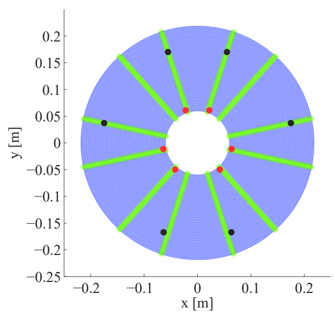


(d) Actuation leg forces - trajectory 1.

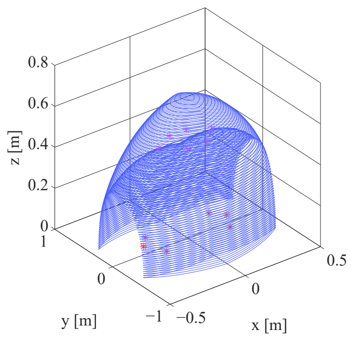
Figure A.13: Stewart-Gough Geometry (4 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 10$.



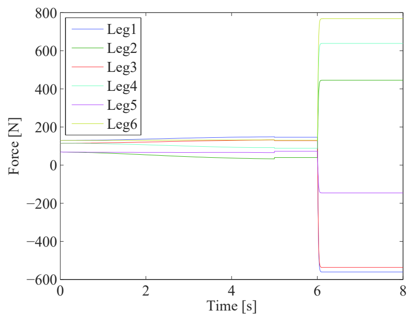
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

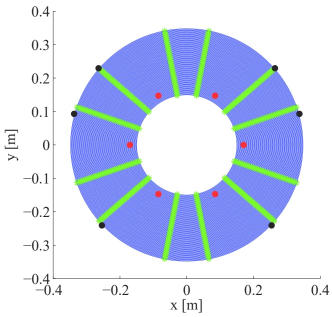


(c) Workspace. $V_W = 0.19318\,\mathrm{m}^3$.

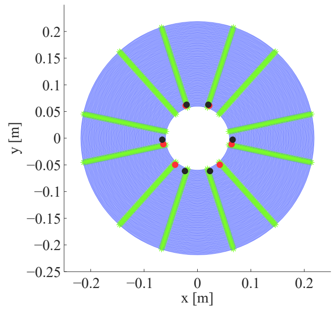


(d) Actuation leg forces - trajectory 1.

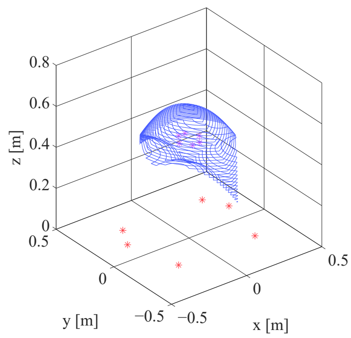
Figure A.14: Stewart-Gough Geometry (8 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 100$.



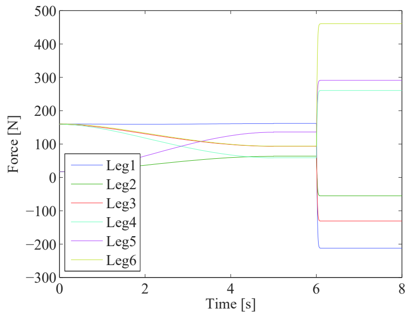
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

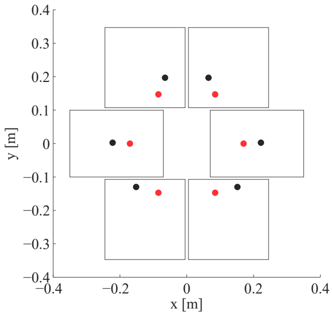


(c) Workspace. $V_W = 0.026006\,\mathrm{m}^3$.

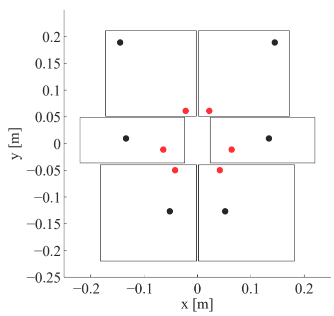


(d) Actuation leg forces - trajectory 1.

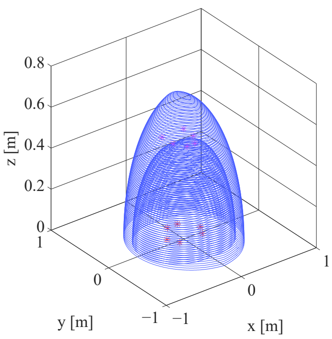
Figure A.15: Stewart-Gough Geometry (8 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 10$.



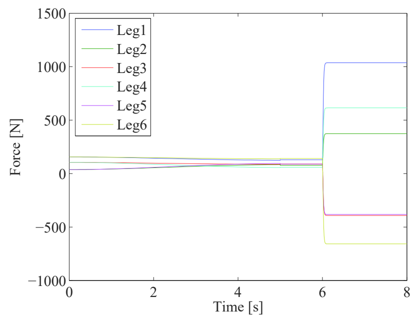
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

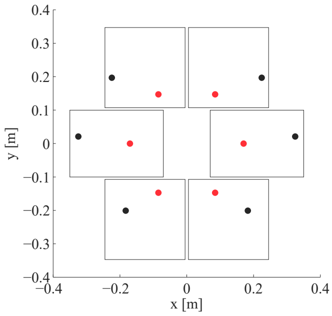


(c) Workspace. $V_W = 0.32683\,\mathrm{m}^3$.

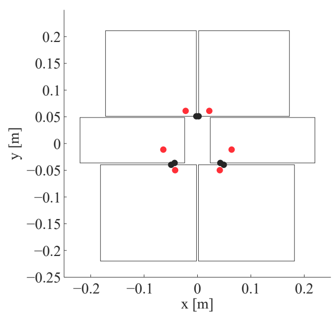


(d) Actuation leg forces - trajectory 1.

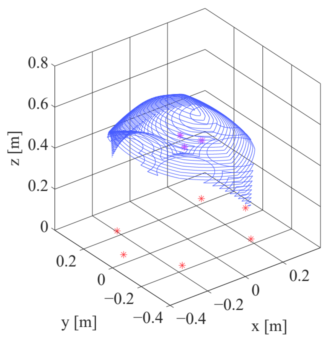
Figure A.16: One Axis Geometry (12 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 100$.



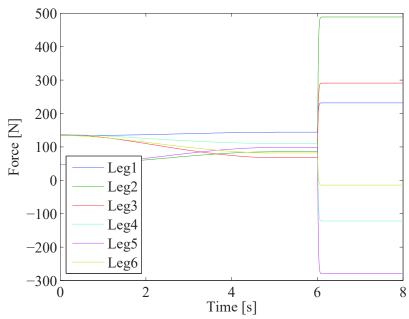
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

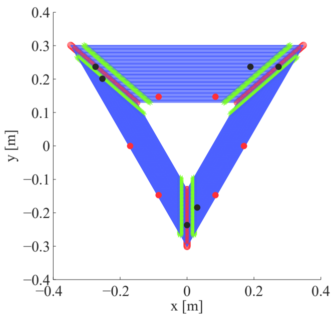


(c) Workspace. $V_W = 0.032568\,\mathrm{m}^3$.

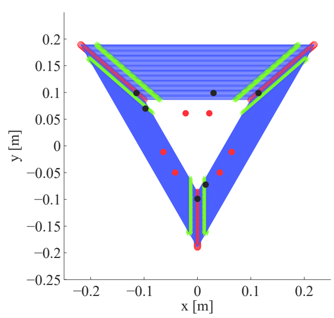


(d) Actuation leg forces - trajectory 1.

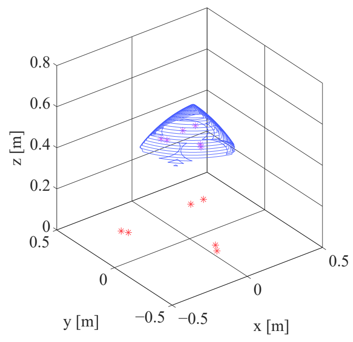
Figure A.17: One Axis Geometry (12 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 10$.



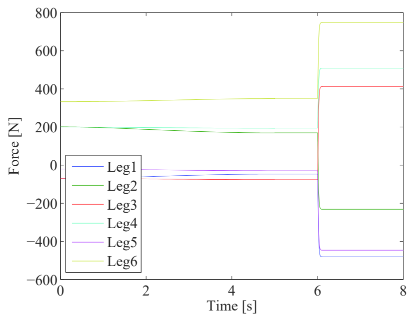
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

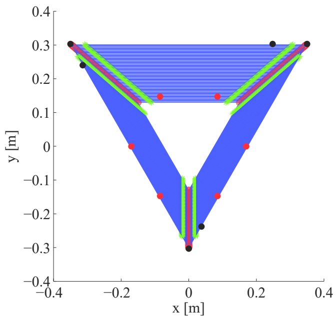


(c) Workspace. $V_W = 0.012091\ \mathrm{m}^3$.

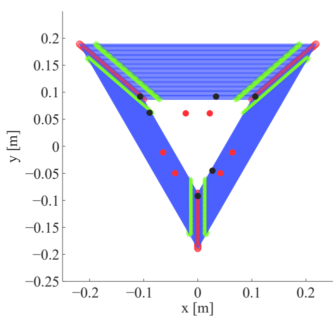


(d) Actuation leg forces - trajectory 1.

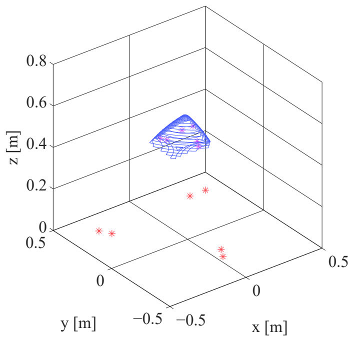
Figure A.18: Griffis-Duffy Geometry (2+6 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 100$.



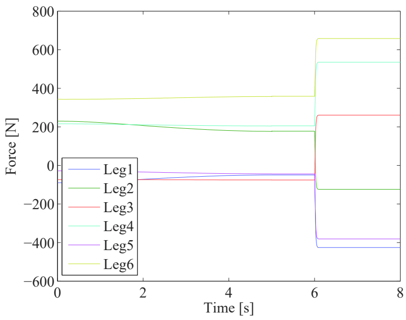
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

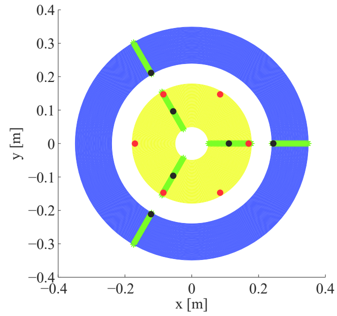


(c) Workspace. $V_W = 0.0039947\,\mathrm{m}^3$.

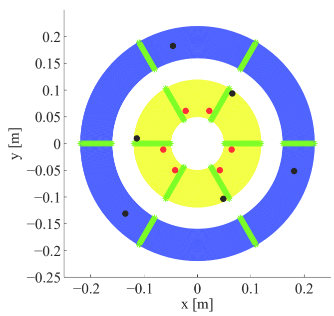


(d) Actuation leg forces - trajectory 1.

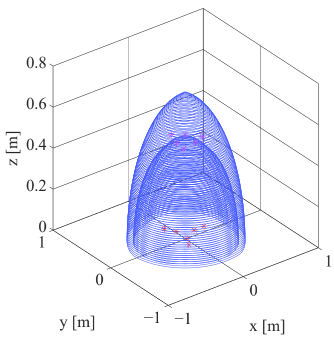
Figure A.19: Griffis-Duffy Geometry (2+6 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 10$.



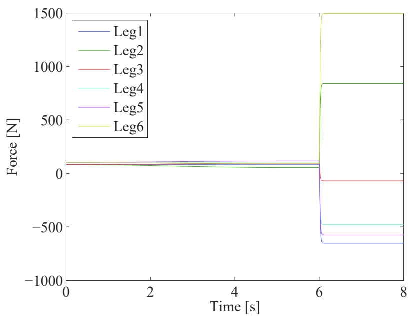
(a) Leg attachment points on base plate. In red initial design; in black optimized design.



(b) Leg attachment points on top plate. In red initial design; in black optimized design.

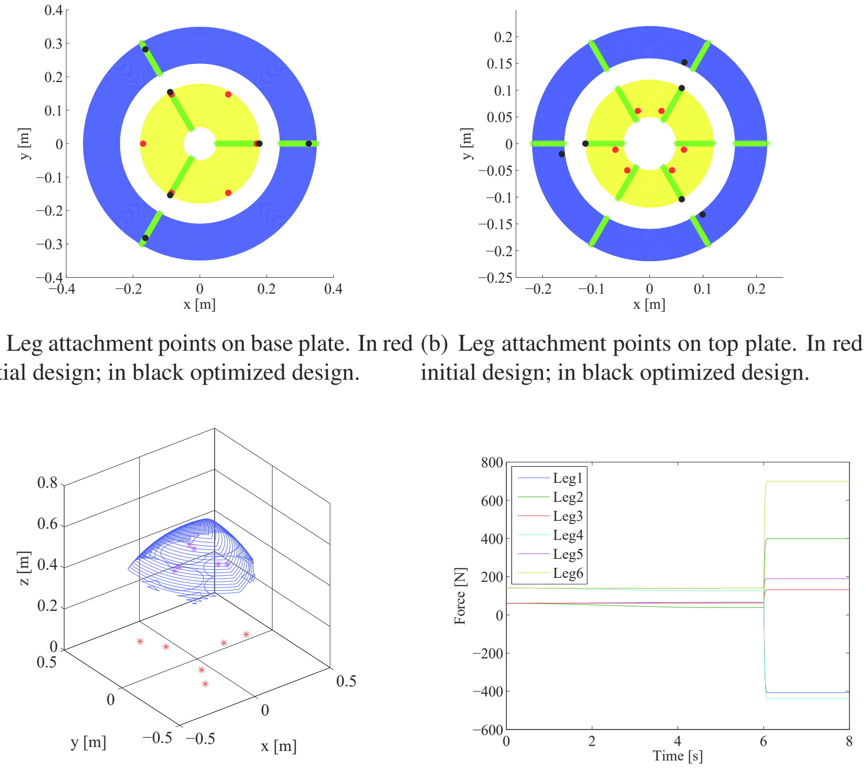


(c) Workspace. $V_W = 0.25008\,\mathrm{m}^3$.



(d) Actuation leg forces - trajectory 1.

Figure A.20: MSP Geometry (6 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 100$.



(a) Leg attachment points on base plate. In red initial design; in black optimized design.

(b) Leg attachment points on top plate. In red initial design; in black optimized design.

(c) Workspace. $V_W = 0.030380\,\mathrm{m}^3$.

(d) Actuation leg forces - trajectory 1.

Figure A.21: MSP Geometry (6 Variables) - Case study II: $k_1 = 0.1$, $k_2 = 10$.

APPENDIX B

CAMERA: BASICS

A camera is a complex system comprising several devices other than the photosensitive sensor, a lens and analog preprocessing electronics. The lens is responsible for focusing the light reflected by the object on the plane where the photosensitive sensor lies, called the *image plane*.

Let consider a frame $O_c - x_c y_c z_c$ attached to the camera, whose location with respect to the base frame is identified by the homogeneous transformation matrix $\boldsymbol{T}_c^b$, and a point on the object whose coordinates are $\boldsymbol{p}^c = [p_x^c \ p_y^c \ p_z^c]^T$. The coordinate transformation from the base frame to the camera frame is

$$\boldsymbol{p}^c = \boldsymbol{T}_b^c \boldsymbol{p}, \tag{B.1}$$

where $\boldsymbol{p}$ denotes the object position with respect to the base frame.

Let consider now a frame attached to the image plane, whose axes X and Y are parallel to the axes x_c and y_c of the camera frame, positioned at the intersection of the optical axis with the image plane (*principal point*). The point in the camera frame can be transformed into a point in the image plane via the *perspective transformation* (see Fig. B.1):

$$\begin{aligned} X_f &= -\frac{f p_x^c}{p_z^c} \\ Y_f &= -\frac{f p_y^c}{p_z^c} \end{aligned}, \tag{B.2}$$

where $(X_f,\ Y_f)$ are the new coordinates in the frame defined on the image plane, and

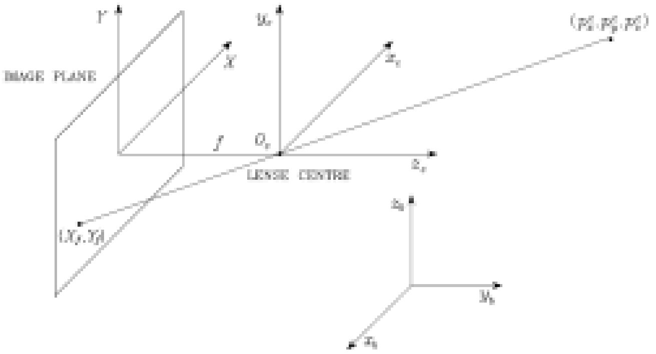


Figure B.1: Perspective transformation.

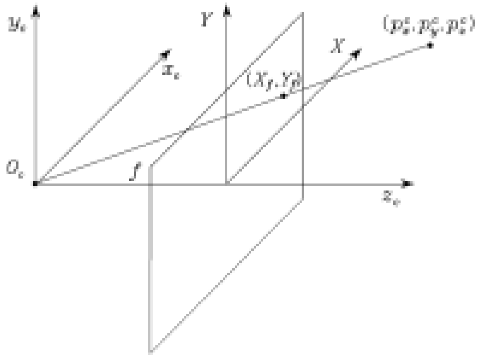


Figure B.2: Frontal perspective transformation.

f is the focal length of the lens, all expressed in metric units.
The presence of the minus sign in the equations of the perspective transformation is consistent with the fact that the image of an object appears upside down on the image plane of the camera. Such an effect can be avoided, for computational ease, by considering a virtual image plane positioned before the lens, in correspondence of the plane $z_c = f$ of the camera frame (see Fig B.2). In this way, the *frontal perspective transformation* is obtained (B.3).

$$\begin{aligned} X_f &= \frac{f p_x^c}{p_z^c} \\ Y_f &= \frac{f p_y^c}{p_z^c} \end{aligned} \tag{B.3}$$

Note that the introduced relationships are valid only in theory, since the real lenses are always affected by imperfections, which cause image quality degradation. Two types of distortions can be recognized, namely, *aberrations* and *geometric distortion*.

Given an image, a spatial sampling is needed since an infinite number of points in the image plane exist. The CCD or CMOS sensors play the role of spatial samplers. they split the image into basic elements or spatial sampling unit called pixels. So, the coordinates (X, Y) of a point in the image plane can be expressed in pixels, i.e., (X_I, Y_I). Moreover, the pixel coordinates of the point are related to the coordinates in metric units through two scale factors α_x and α_y:

$$\begin{aligned} X_I &= \frac{\alpha_x f p_x^c}{p_z^c} + X_0 \\ Y_I &= \frac{\alpha_y f p_y^c}{p_z^c} + Y_0 \end{aligned}, \tag{B.4}$$

where X_0 and Y_0 are the offsets which take into account the position of the origin of the pixel coordinate system with respect to the optical axis. The transformation in (B.4) can be write in linear form by considering the homogeneous representation of the point (x_I, y_I, z_I) as in (B.5), with $\lambda > 0$.

$$\begin{aligned} X_I &= \frac{x_I}{\lambda} \\ Y_I &= \frac{y_I}{\lambda} \end{aligned} \tag{B.5}$$

From the aforesaid, the (B.5) can be rewritten as

$$\begin{bmatrix} x_I \\ y_I \\ \lambda \end{bmatrix} = \lambda \begin{bmatrix} X_I \\ Y_I \\ 1 \end{bmatrix} = \mathbf{\Omega \Pi} \begin{bmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{bmatrix} \tag{B.6}$$

where

$$\mathbf{\Omega} = \begin{bmatrix} f\alpha_x & 0 & X_0 \\ 0 & f\alpha_y & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.7}$$

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{B.8}$$

At this point, the overall transformation from the Cartesian space of the observed object to the image space of its image in pixels is characterized by composing the transformations in (B.1) and in (B.6) as:

$$\boldsymbol{\Theta} = \boldsymbol{\Omega}\boldsymbol{\Pi}\boldsymbol{T}_b^c \tag{B.9}$$

which represents the *camera calibration matrix*. Note that such a matrix contains intrinsic parameters ($\alpha_x,\ \alpha_y,\ X_0,\ Y_0,\ f$) in $\boldsymbol{\Omega}$ depending on the sensor and lens characteristics as well as *extrinsic parameters* in $\boldsymbol{T}_c^b$ depending on the relative position and orientation of the camera with respect to the base frame.

If the intrinsic parameters of a camera are known, from a computationally viewpoint, it is convenient to refer to the *normalized coordinates* (X, Y), defined by the normalized perspective transformation defined in metrical units

$$\lambda \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \boldsymbol{\Pi} \begin{bmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{bmatrix}. \tag{B.10}$$

Comparing (B.9) with (B.11) yields the invertible transformation

$$\begin{bmatrix} X_I \\ Y_I \\ 1 \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{B.11}$$

relating the normalized coordinates to those expressed in pixels through the matrix of intrinsic parameters.

BIBLIOGRAPHY

[1] B. Hempstead, R. DeVlieg, R. Mistry, and M. Sheridan. Drill and drive end effector. *SAE Technical Paper 2001-01-2576*, 2001.

[2] R. DeVlieg, K. Sitton, E. Feikert, and J. Inman. Once (one-sided cell end effector) robotic drilling system. *SAE Technical Paper 2002-01-2626*, 2002.

[3] J. Atkinson, J. Hartmann, S. Jones, and P. Gleeson. Robotic drilling system for 737 aileron. *SAE Technical Paper 2007-01-3821*, 2007.

[4] R. DeVlieg. Robotic trailing edge flap drilling system. *SAE Technical Paper 2009-01-3244*, 2009.

[5] R. DeVlieg and E. Feikert. One-up assembly with robots. *SAE Technical Paper 2008-01-2297*, 2008.

[6] E. Whinnem, G. Lipczynski, and I. Eriksson. Development of orbital drilling for the boeing 787. *SAE Int. J. Aerosp. 1*, 1:811–816, 2009.

[7] Shusheng Bi and Jie Liang. Robotic drilling system for titanium structures. *The International Journal of Advanced Manufacturing Technology*, 54(5-8):767–774, 2011.

[8] T. Olsson, A. Robertsson, and R. Johansson. Flexible Force Control for Accurate Low-Cost Robot Drilling. In *IEEE International Conference on Robotics and Automation*, pages 4770–4775, Brijuni, Croatia, 2007.

[9] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Bjö rkman, R. Isaksson, G. Ossbahr, and T. Brogårdh. Cost-efficient drilling using industrial robots with high-bandwidth force feedback. 26:24–38, 2010.

[10] Bruno Siciliano, Lorenzo Sciavicco, and Luigi Villani Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2008.

[11] Bruno Siciliano and Luigi Villani. *Robot Force Control*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition, 2000.

[12] Ciro Natale, Ralf Koeppe, and Gerd Hirzinger. A systematic design procedure of force controllers for industrial robots. *Mechatronics, IEEE/ASME Transactions on*, 5(2):122–131, 2000.

[13] Fabrizio Caccavale. Cooperative manipulators. In John Baillieul and Tariq Samad, editors, *Encyclopedia of Systems and Control*, pages 1–7. Springer London, 2014.

[14] Henrik Kihlman. Affordable automation for airframe assembly: developing of key enabling technologies. 2005.

[15] Henrik Kihlman, Gilbert Ossbahr, Magnus Engström, and John Anderson. Low-cost automation for aircraft assembly. Technical report, SAE Technical Paper, 2004.

[16] Doug Stewart. A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386, 1965.

[17] Ilian Bonev. The true origins of parallel robots. 2011.

[18] Ahmad Naveed Ismail. *The Design and Construction of a 6 Degree-of-Freedom Parallel-Link Platform Type Manipulator*. PhD thesis, MIT Dept. of Mechanical Engineering, 1988.

[19] Gene L. Waltman. Black magic and gremlins: Analog flight simulations at nasa's flight research center. *Monographs In Aerospace History*, (20), 2000.

[20] Zhijiang Du, Ruochong Shi, and Wei Dong. A piezo-actuated high-precision flexible parallel pointing mechanism: Conceptual design, development, and experiments. *Robotics, IEEE Transactions on*, 30(1):131–137, 2014.

[21] YX Su, BY Duan, and CH Zheng. Genetic design of kinematically optimal fine tuning stewart platform for large spherical radio telescope. *Mechatronics*, 11(7):821–835, 2001.

[22] M Girone, Grigore Burdea, Mourad Bouzit, V Popescu, and Judith E Deutsch. A stewart platform-based system for ankle telerehabilitation. *Autonomous robots*, 10(2):203–212, 2001.

[23] Kenji Hashimoto, Yusuke Sugahara, Hun-ok Lim, and Atsuo Takanishi. Optimization design of a stewart platform type leg mechanism for biped walking vehicle. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 169–178. Springer Berlin Heidelberg, 2011.

[24] Joseph P Conti, Charles M Clinton, Guangming Zhang, and Albert J Wavering. *Workspace variation of a hexapod machine tool*. Citeseer, 1998.

[25] Srinivas M Satya, Placid M Ferreira, and Mark W Spong. Hybrid control of a planar 3-dof parallel manipulator for machining operations. *TRANSACTIONS-NORTH AMERICAN MANUFACTURING RESEARCH INSTITUTION OF SME*, pages 273–280, 1995.

[26] Robert S Stoughton and Tatsuo Arai. A modified stewart platform manipulator with improved dexterity. *Robotics and Automation, IEEE Transactions on*, 9(2):166–173, 1993.

[27] R Neugebauer, W-G Drossel, C Harzbecker, S Ihlenfeldt, and S Hensel. Method for the optimization of kinematic and dynamic properties of parallel kinematic machines. *CIRP Annals-Manufacturing Technology*, 55(1):403–406, 2006.

[28] Marco Ceccarelli, Giuseppe Carbone, and Erika Ottaviano. Multi criteria optimum design of manipulators. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 53, 2005.

[29] F Hao and J-P Merlet. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and machine theory*, 40(2):157–171, 2005.

[30] Bang Jun Lv, SJ Zhu, and JF Xing. Multi-criteria optimal design of parallel manipulators based on natural frequency. *Applied Mechanics and Materials*, 29:2435–2442, 2010.

[31] JA Cabrera, F Nadal, JP Munoz, and A Simon. Multiobjective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm. *Mechanism and machine theory*, 42(7):791–806, 2007.

[32] A.B.K. Rao, P.V.M. Rao, and S.K. Saha. Dimensional design of hexaslides for optimal workspace and dexterity. *Robotics, IEEE Transactions on*, 21(3):444–449, June 2005.

[33] MA Laribi, L Romdhane, and S Zeghloul. Analysis and dimensional synthesis of the delta robot for a prescribed workspace. *Mechanism and Machine Theory*, 42(7):859–870, 2007.

[34] Anna Kosinska, Miroslaw Galicki, and Krzysztof Kedzior. Designing and optimization of parameters of delta-4 parallel manipulator for a given workspace. *Journal of Robotic Systems*, 20(9):539–548, 2003.

[35] Karol Miller. Maximization of workspace volume of 3-dof spatial parallel manipulators. *Journal of Mechanical Design*, 124(2):347–350, 2002.

[36] Yawei Yang and John F O'Brien. A sequential method for the singularity-free workspace design of a three legged parallel robot. *Mechanism and Machine Theory*, 45(11):1694–1706, 2010.

[37] Yawei Yang and John F O'Brien. A sequential method for the singularity free workspace design of a planar 3-arm parallel robot. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1977–1982. IEEE, 2008.

[38] Haidong Li, Clément M Gosselin, and Marc J Richard. Determination of the maximal singularity-free zones in the six-dimensional workspace of the general gough–stewart platform. *Mechanism and machine theory*, 42(4):497–511, 2007.

[39] AM Hay and JA Snyman. Methodologies for the optimal design of parallel manipulators. *International Journal for Numerical Methods in Engineering*, 59(1):131–152, 2004.

[40] Shaoping Bai. Optimum design of spherical parallel manipulators for a prescribed workspace. *Mechanism and Machine Theory*, 45(2):200–211, 2010.

[41] Xin-Jun Liu, Jinsong Wang, Kun-Ku Oh, and Jongwon Kim. A new approach to the design of a delta robot with a desired workspace. *Journal of Intelligent and Robotic Systems*, 39(2):209–225, 2004.

[42] Dan Zhang, Zhengyi Xu, Chris M. Mechefske, and Fengfeng Xi. Optimum design of parallel kinematic toolheads with genetic algorithms. *Robotica*, 22(1):77–84, January 2004.

[43] Dan Zhang and Clement M Gosselin. Kinetostatic analysis and design optimization of the tricept machine tool family. *Journal of manufacturing science and engineering*, 124(3):725–733, 2002.

[44] Zhen Gao, Dan Zhang, Xiaolin Hu, and Yunjian Ge. Design, analysis, and stiffness optimization of a three degree of freedom parallel manipulator. *Robotica*, 28(03):349–357, 2010.

[45] Xin-Jun Liu, Zhen-Lin Jin, and Feng Gao. Optimum design of 3-dof spherical parallel manipulators with respect to the conditioning and stiffness indices. *Mechanism and Machine Theory*, 35(9):1257–1267, 2000.

[46] Clement Gosselin and Jorge Angeles. The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator. *Journal of Mechanical Design*, 110(1):35–41, 1988.

[47] Kourosh E Zanganeh and Jorge Angeles. Kinematic isotropy and the optimum design of parallel manipulators. *The International Journal of Robotics Research*, 16(2):185–197, 1997.

[48] Hua Chen, Weishan Chen, and Junkao Liu. Optimal design of stewart platform safety mechanism. *Chinese Journal of Aeronautics*, 20(4):370–377, 2007.

[49] Yongjie Zhao. Dimensional synthesis of a three translational degrees of freedom parallel robot while considering kinematic anisotropic property. *Robotics and Computer-Integrated Manufacturing*, 29(1):169–179, 2013.

[50] Imed Mansouri and Mohammed Ouali. The power manipulability - a new homogeneous performance index of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, 27(2):434 – 449, 2011. Translational Research - Where Engineering Meets Medicine.

[51] Alexandre Lecours and Clément Gosselin. Reactionless two-degree-of-freedom planar parallel mechanism with variable payload. *Journal of Mechanisms and Robotics*, 2(4):041010, 2010.

[52] Gursel Alici and Bijan Shirinzadeh. Optimum dynamic balancing of planar parallel manipulators based on sensitivity analysis. *Mechanism and Machine Theory*, 41(12):1520–1532, 2006.

[53] Djana Ilia and Rosario Sinatra. A novel formulation of the dynamic balancing of five-bar linkages with applications to link optimization. *Multibody System Dynamics*, 21(2):193–211, 2009.

[54] Yongjie Zhao, Feng Gao, Xingjian Dong, and Xianchao Zhao. Dynamics analysis and characteristics of the 8-pss flexible redundant parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 27(5):918–928, 2011.

[55] Yongjie Zhao. Dynamic optimum design of a three translational degrees of freedom parallel robot while considering anisotropic property. *Robotics and Computer-Integrated Manufacturing*, 29(4):100–112, 2013.

[56] Karl Gotlih, Denis Kovac, Tomaz Vuherer, Simon Brezovnik, Miran Brezocnik, and Andrejka Zver. Velocity anisotropy of an industrial robot. *Robotics and Computer-Integrated Manufacturing*, 27(1):205–211, 2011.

[57] Yongjie Zhao and Feng Gao. Dynamic formulation and performance evaluation of the redundant parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 25(4):770–781, 2009.

[58] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[59] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[60] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.

[61] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, 1981.

[62] W Hock and Klaus Schittkowski. A comparative performance evaluation of 27 nonlinear programming codes. *Computing*, 30(4):335–358, 1983.

[63] M.J.D. Powell. Variable metric methods for constrained optimization. In R. Glowinski, J.L. Lions, and Iria Laboria, editors, *Computing Methods in Applied Sciences and Engineering, 1977, I*, volume 704 of *Lecture Notes in Mathematics*, pages 62–72. Springer Berlin Heidelberg, 1979.

[64] Zsolt Ugray, Leon Lasdon, John Plummer, Fred Glover, James Kelly, and Rafael Martí. Scatter search and local nlp solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.

[65] F Glover. A template for scatter search and path relinking. *Lecture Notes in Computer Science*, 1363:13–54, 1998.

[66] L. Dixon and G. P. Szegö. The global optimization problem: an introduction. *Towards Global Optimisation 2*, 1978.

[67] Domagoj Jakobovic and Leonardo Jelenkovic. The forward and inverse kinematics problems for stewart parallel mechanisms. *8th International Scientific Conference on Production Engineering: Computer Integrated Manufacturing and High Speed Machining*, pages 1–12, 2002.

[68] Zafer Bingul and Oguzhan Karahan. Dynamic modeling and simulation of stewart platform, serial and parallel robot manipulators - kinematics, dynamics, control and optimization. *Dr. Serdar Kucuk (Ed.)*, 2012.

[69] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. In *Robotics and Automation, 2004. Proceedings. ICRA 04. 2004 IEEE International Conference on*, volume 4, pages 3665–3670 Vol.4, April 2004.

[70] The MathWorks Inc., http://it.mathworks.com/help/physmod/sm/mech/ug/configuring-methods-of-solution.html.

[71] The MathWorks Inc., http://it.mathworks.com/help/physmod/sm/mech/ug/finding-forces-from-motions.html.

[72] Júlia Borras, Federico Thomas, and Carme Torras. New geometric approaches to the analysis and design of stewart–gough platforms. *Mechatronics, IEEE/ASME Transactions on*, 19(2):445–455, 2014.

[73] Clement Gosselin. Determination of the workspace of 6-dof parallel manipulators. *Journal of mechanical design*, 112(3):331–336, 1990.

[74] The MathWorks Inc., http://it.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html.

[75] Anthony A Maciejewski and Charles A Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The international journal of robotics research*, 4(3):109–117, 1985.

[76] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.

[77] Bruno Siciliano and Jean-Jacques E Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216. IEEE, 1991.

[78] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *Robotics and Automation, IEEE Transactions on*, 13(3):398–410, 1997.

[79] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*, 1(1):27–39, 2008.

[80] Gianluca Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *Robotics, IEEE Transactions on*, 25(5):985–994, 2009.

[81] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2011.

[82] Erkki Jantunen. A summary of methods applied to tool condition monitoring in drilling. *International Journal of Machine Tools and Manufacture*, 42(9):997–1010, 2002.

[83] M.A. Davies, T. Ueda, R. M'Saoubi, B. Mullany, and A.L. Cooke. On the measurement of temperature in material removal processes. *{CIRP} Annals - Manufacturing Technology*, 56(2):581 – 604, 2007.

[84] Pancu Rares, Bungau Constantin, and Ganea Macedon. Thermographic diagnosis at machine tools. *Nonconventional Technologies Review*, pages 62 – 66, 2013.

[85] Carlos Henrique Lauro, Lincoln Cardoso Brandão, and Sergio Luiz Moni Ribeiro Filho. Monitoring the temperature of the milling process using infrared camera. *Scientific Research and Essays*, 8(23):1112–1120, 2013.

[86] S Bagavathiappan, BB Lahiri, S Suresh, J Philip, and T Jayakumar. On-line monitoring of cutting tool temperature during micro-end milling using infrared thermography. *Insight-Non-Destructive Testing and Condition Monitoring*, 57(1):9–17, 2015.

[87] Masatoshi Shindou, Hiroyuki Kodama, Toshiki Hirogaki, and Eiichi Aoyama. Monitoring of end-mill process based on infrared imagery with a high speed thermography. In *Key Engineering Materials*, volume 625, pages 213–218. Trans Tech Publ, 2014.

[88] L. Roseiro, C. Veiga, V. Maranha, A.Neto, N. Laraqi, A. Baïri, and N. Alilat. *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, 8(11):816 – 819, 2014.

[89] Goran Augustin, Slavko Davila, Toma Udiljak, Denis Stjepan Vedrina, and Dinko Bagatin. Determination of spatial distribution of increase in bone temperature during drilling by infrared thermography: preliminary report. *Archives of orthopaedic and trauma surgery*, 129(5):703–709, 2009.

[90] Milan Honner, Pavel Litoš, and Michal Švantner. Thermography analyses of the hole-drilling residual stress measuring technique. *Infrared physics & technology*, 45(2):131–142, 2004.

[91] C Meola, S Boccardi, GM Carlomagno, ND Boffa, E Monaco, and F Ricci. Nondestructive evaluation of carbon fibre reinforced composites with infrared thermography and ultrasonics. *Composite Structures*, 134:845–853, 2015.

[92] OpenCV, http://www.opencv.org.

[93] Sadeghian H., Villani L., Kamranian Z., and Karami A. Visual servoing with safe interaction using image moments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Piscataway*, pages 5479–5485. IEEE, 2015.