UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO IN SCIENZE DELL'INGEGNERIA
CORSO DI DOTTORATO IN INGEGNERIA INDUSTRIALE

Task Optimization of Functionally Redundant Parallel Kinematics Machines

Ph.D. Dissertation of:
David Corinaldi

Advisor:
Prof. Massimo Callegari

Ph.D. Course Coordinator:
Prof. Ferruccio Mandorli

XV edition - new series

UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO IN SCIENZE DELL'INGEGNERIA
CORSO DI DOTTORATO IN INGEGNERIA INDUSTRIALE

Task Optimization of Functionally Redundant Parallel Kinematics Machines

Ph.D. Dissertation of:
David Corinaldi

Advisor:
Prof. Massimo Callegari

Ph.D. Course Coordinator:
Prof. Ferruccio Mandorli

XV edition - new series

UNIVERSITÀ POLITECNICA DELLE MARCHE
SCUOLA DI DOTTORATO IN SCIENZE DELL'INGEGNERIA
FACOLTÀ DI INGEGNERIA
Via Brecce Bianche – 60131 Ancona (AN), Italy

Alla mia famiglia
e a chi ne farà parte.

To my family
and to those who will join it.

Acknowledgments

This thesis was written under the supervision of Prof. Massimo Callegari. He devoted his time, resources and knowledge to my growth, supporting me with his willingness and understanding. A sincere thank goes to him. Moreover, I would like to thank Prof. Ferruccio Mandorli, coordinator of the Ph.D Course, who carefully followed my training course.
The hospitality of Prof. Jorge Angeles allowed me to make a valuable research experience abroad at McGill University. In addition, I would like to thank him for having shared his research branch and his deep knowledge with me. His expertise and his successful career are inspiring me greatly.
I also thank all the members of the Machine Mechanics Group. A constant comparison with them was very helpful. The group was very friendly and its contagious sympathy made time at UnivPM really enjoyable.
Finally, I would like to express my deep gratitude to the reviewers Prof. Jorge Angeles and Prof. Marco Carricato for their helpful suggestions and comments that have improved this thesis.

Ancona, Feb. 2017

David Corinaldi

Ringraziamenti

Questa tesi è stata redatta sotto la supervisione del Prof. Massimo Callegari. Egli ha contribuito con tempo, risorse e conoscenze alla mia crescita, sostenendomi con la sua disponibilità e comprensione. A lui va un sentito ringraziamento. Inoltre, vorrei ringraziare il Prof. Ferruccio Mandorli, coordinatore del Corso di Dottorado, che ha attentamente seguito il mio percorso educativo.
L'ospitalità del Prof. Jorge Angeles mi ha permesso di vivere una esperienza di studio all'estero presso la McGill University davvero interessante. Oltre a questo, lo vorrei ringraziare per aver condiviso il presente filone di ricerca e le sue profonde conoscenze. La sua competenza e la sua carriera di successo sono una grande fonte di ispirazione.
Ringrazio anche tutti i membri del gruppo di Meccanica delle Macchine. Un confronto continuo con loro è stato molto ulile. Il gruppo si è dimostrato davvero amichevole e la sua contagiosa simpatia ha reso le giornate in ufficio piacevoli e divertenti.
Infine, vorrei esprimere la mia profonda gratitudine ai revisori della tesi Prof. Jorge Angeles e Prof. Marco Carricato per i loro prezioni consigli e commenti che mi hanno permesso di migliorare questa tesi.

Ancona, Feb. 2017

David Corinaldi

Abstract

The work developed in this thesis results from a research project carried out during my PhD Program that aimed at the development of tools and techniques to support the execution of industrial tasks in the automation field. Nowadays, the use of industrial robots in new Flexible Manufacturing Systems (FMS) is increasingly popular because of their versatility. Frequently, these manipulators have a full ability to position and to orient their moving platform so as to cover a more varied range of tasks to be performed. However, there is a large class of industrial tasks in which the orientation of the robot terminal in one direction has no influence to the accomplishment of a task; for example, in a milling process, a further rotation along the tool axis in addition to that of the spindle is not required. Clearly this implies a redundancy between the manipulator and the task at hand, leading to the following question: what orientation along this axis should the moving platform take during the task? What posture, corresponding to this moving platform pose, should the manipulator assume?
Redundancy provides alternatives, but often the problem is neglected in practical applications and the choice of the convenient orientation is casual. The need of better performance has recently driven research toward the formulation of questions about optimization problems and their subsequent solution by means of specific algorithms. In the definition of the problem, we entrust to an objective function, often related to indices that quantify the robot kinematic/static or dynamic performance, that depend on the posture assumed by the manipulator. The main topic of the present thesis is thus redundancy, called specifically *functional redundancy* that has been recently unearthed but that has involved serial manipulators so far; although the extension of some theories to the parallel robot class is often deducted, a comprehensive study about parallel manipulators is missing. Parallel kinematics machines are known to have strengths like higher rigidity, accuracy and load capacity if compared with serial robots, features that make them more attractive for machining purposes. Drawbacks for their implementation reside in the complexity of their kinematics and in the high number of singular postures they usually have, inherent in the multi-limb architecture. This thesis addresses the optimization problem of finding the best posture of parallel manipulators when they perform tasks in functional redundancy conditions. The optimization problem is dealt with in terms of formulation and numerical solution. The work is focused on a particular class of parallel robots, namely robots that provide motions of pure rotation of their mobile platform. Such choice allowed us to highlight some advantages related to a particular architecture that they share. The mentioned

study of finding the best posture is then extended to an optimal planning of pointing trajectories, with special emphasis on manufacturing tasks.
The approach followed in this research project required the integration of various disciplines: the theory of kinematic chains, trajectory planning and dynamics of multi-body systems. The research methodology combines theoretical study, assisted by mathematical programming software, and virtual simulations to verify results, performed by dynamic multi-body software. Unfortunately, the validation of results on physical prototypes is postponed to future works.
This thesis is organized as follows. Chapter 1 presents the research project and provides an up-to-date review of six-axis parallel kinematics machines used in manufacturing, as well as common industrial tasks that employ axially symmetric tools. A unified redundancy analysis for both serial and parallel manipulators is developed in Chapter 2. In order to reduce the complexity of the kinematics of these machines, the convenient idea of dividing a six-axis robot in two separate machines with lower mobility is introduced. Thereby, the positioning problem is neglected, since the functional redundancy that arises is associated with a rotational degree of freedom, and the study in Chapter 3 focuses on a class of spherical parallel machines. In Chapters 4 and 5 kineto-static and dynamic analyses of parallel robots are respectively addressed with the purpose of evaluating their performance by means of suitable indices. The study of singular configurations is reported in Chapter 6 with emphasis on the singularity-surface representation in the space of Euler-Rodrigues parameters. Then, the posture optimization is used for redundancy solution in Chapter 7. Robot dexterity, dynamic manipulability and swiftness for a prescribed pointing direction of the tool were maximized. The optimal postures are then found for the overall machine workspace and comments about the analogies of the results for the class of manipulators were made. A further optimization problem, with a level more complex than the previous one, uses an average value of the indices to draw a pointing trajectory by means of recent techniques for creating Bézier curves on a sphere. In this way the trajectory follows attractive regions around points of kinematic and dynamic isotropy (Chapter 8). The last Chapter 9 gathers the obtained results and leaves room for comments and future insights.
Finally, the topic of this dissertation is the optimization of tasks of functionally redundant parallel manipulators. The functional redundancy of these robots is exploited with respect to the task to be performed. this is the case of a task that requires a fewer number of degrees of freedom than those available by the manipulator. The objective function of the optimization problem refers to indices aimed at the improvement either the accuracy of the mobile platform orientation or the dynamic performance that plays an important role when the tool accelerations are not negligible. Numeric results show the same isotropy directions for both kinematic and dynamic optimization, even among all the considered classes of robots. They reveal a bond with the geometry of the manipulators, as it can be deduced from the dependency of the indices from

the posture. Moreover, the proposed methods for posture optimization and for optimum trajectory planning ensure computational efficiency while avoiding singularity representation.

Contents

List of Figures

List of Tables

0.1 List of Notation and Acronyms

In this dissertation vectors will be indicated with lower cases, while matrices with upper cases; both vectors and matrices have a bold font. $\mathbf{e}$ corresponds to a unit vector and the hat over a vector $\hat{\mathbf{a}}$ indicates its versor. The foregoing acronyms are often used:

KC: kinematic chain

PKM: parallel kinematics machine

SPM: spherical parallel manipulator

BP: base platform

MP: mobile platform

EE: end-effector

dof: degrees of freedom

IKP: Inverse Kinematics Problem

Chapter 1

Introduction

This work is focused on the functional kinematic redundancy of parallel kinematics machines in the execution of industrial tasks. After a brief overview of the research project, this chapter addresses two subjects: the first is about parallel manipulators currently used in industrial processes and the second is about common manufacturing tasks that do not require a full mobility of the robot moving platform.

1.1 Overview of the Research Project

The evolution of production systems generates changing demands that are increasingly satisfied by the industrial robotics. Looking with a future perspective the major challenges that begin to take shape, they are aimed towards the improvement of performance of robotic devices which, depending on the need, may be facing either the increase of productivity by the reduction of time and energy consumption, or the accuracy improvement for the assigned task. To achieve these goals, some key enablers have been defined within the robotics research in favor of innovative design principles of manipulators and combined with advanced control concepts. In conjunction with these, the guidelines of the new cells for mechanical machining require versatile manipulators that are used in the most recent flexible manufacturing systems (FMS) which are capable to automatically change the product productions. To meet the requirements of flexibility and versatility in the manufacturing processes, the choice of the robot employed often falls in manipulators of the more generic type, i.e. robots capable to position their moving platform with six degrees of freedom (dof).
Most of the tasks of high industrial relevance such as milling, welding, additive manufacturing and plasma, water and laser cutting are characterized by five degrees of freedom because they require the use of an axially symmetric device whose orientation with respect to the axis of symmetry is irrelevant to the task to be performed. Therefore, with respect to the full-mobility manipulator used, i.e. a manipulator with six-dof, there is a redundant degree of freedom. This redundancy takes more specifically the name of functional redundancy. It constitutes a potentiality that is exploited for optimization purposes in the research project carried out.
Although these situations of functional redundancy are very common in manufactur-

ing processes, they have been ignored in industrial applications over the years and only recent studies have devised algorithms for its exploitation, highlighting the advantages of such optimization. The most widespread robots employed in manufacturing are serial manipulators due to their superior versatility and large workspace. Serial robots are currently used to perform tasks in the most recent manufacturing cells, mainly because they possess greater versatility and affordability when compared with five-axis numerical control machines at the expense of a lower stiffness. In machining operations the stiffness is often a key factor due to the stringent demands on the dimensional tolerances of the machined pieces therefore parallel kinematics machines (PKM) carry the potentiality to satisfy the stated needs, because they usually offer a greater rigidity than serial robots.
The reason behind the lower stiffness of serial robots is inherent in the supporting architecture of the end-effector, as their outboard links are supported by their proximal counterparts. The serial-robot stiffness is, thus, equivalent to a serial array of springs, whose compliance is additive: the larger the number of springs connected in series, the more compliant the array becomes. On the contrary, the moving plate or the mobile platform of a PKM is supported in parallel by all its limbs, thus its stiffness is additive: the more limbs the PKM carries, the larger its stiffness is.
The optimization problems associated to functional redundancy situations have been already solved for serial robots [1], while this work is focused on parallel manipulators. Parallel kinematics machines are not widespread at the industrial level due to their smaller workspace. However, compared to serial robots, they offer better dynamic performance due to actuators fixed to the frame instead of the movable arms and, as already said, increased stiffness of the structure. Therefore, when they are used for mechanical machining, they allow the products to achieve more stringent accuracy specifications. Since the accuracy of such manipulators fits in well with the functional redundancy optimization, the algorithms developed so far for serial manipulators will be reviewed in this thesis for applications to parallel manipulators.
The functionally redundant motions, i.e. the movements that a robot can perform without affecting the assigned task, are often exploited for optimization purposes of parameters such as the dexterity or the swiftness of the manipulator. More in details, the robots are considered functionally redundant because the dimension of the operational space (Cartesian space accessible by the moving platform) is larger than the dimension of the task space (Cartesian space of the task). The application of this concept for the optimization of parallel manipulators has not been exhaustively addressed in the literature yet, therefore it represents a fascinating topic of research.
One of the issues in the analysis of parallel manipulators with six-degrees of freedom is the complexity of their kinematic model, which can adversely affect the path planning. What is sometimes done in conventional processes is to separate the full-mobility task in sub-elementary tasks performed by separate machines with lower mobility. The mobility decomposition allows even the control algorithms to be split

and the most common division is made up of two three-dof machines where one has 3 degrees of freedom for the translation of its moving platform (EE) while the other executes motions of pure rotations (three-dof). Obviously, in this case, the functional redundancy is owned by the spherical robot which can perform rotational redundant motions around the symmetry axis of the tool. In this thesis a class of spherical parallel robots (3-dof of spherical motion about a fixed point) is taken into account. All the considered SPMs are isostatic and have three identical legs: the machines of this class have different topology, i.e. different sequence of the joints in the leg. This study aims at the formalization of the optimization problem and at the partial generalization of the problem. In fact, the presented approach is applicable to every spherical robotic manipulator with a single degree of functional redundancy and no degree of intrinsic redundancy. In other words, to parallel kinematics machines characterized by a single dof difference between the operational and the task spaces, while the joint and operational space dimensions are equal.

1.1.1 Initial objective

At Machine Mechanics Laboratory, a novel architecture for an assembly cell has been developed decomposing the full mobility of a single manipulator in two PKMs with lower mobility: a spherical PKM and a translational one. Although the manufacturing cell has been widely used to assemble various mechanical parts, other manufacturing operations are not excluded. Acknowledging that the majority of the industrial tasks require a low number of degrees of freedom, the problem of exploiting the degree of functional redundancy of the spherical manipulator has emerged. Although functional redundancy can be used to increase the accuracy of the manipulator above what is currently available, as reported by Léger and Angeles [2] for serial robots, a comprehensive study has not been yet provided for parallel manipulators. Hence, the aim of the thesis is the optimization of tasks performed by functionally redundant parallel robots. This work, in the framework of maximizing robot performance, consists of devising optimum five-dof machining operations with the PKMs mentioned above. A motivation of this work is that the exploitation of functional redundancy to improve performance of PKMs may be a key factor for their wider use in manufacturing applications and beyond.

1.1.2 Methodology and tools

This work has made extensive use of *screw algebra* to describe the behavior of robots. The mathematical framework of this theory was firstly developed by Sir Robert Stawell Ball [3] and then recovered by Hunt [4], who applied screw theory to study the kinematics of spatial mechanisms, with special emphasis on the analysis of singularities. Although PKMs are known to posses a higher complexity of the kinematics and dynamic relationships, the application of this tool brings advantages in terms of analy-

sis and synthesis and a more compact representation of equations. The mathematical relationships have been verified and implemented with computer algebra both for symbolic computation and the numerical solution of optimization problems. Virtual models of the mechanical systems have been build inside a CAD environment, while a CAE software has been used for mobility verification and for animation purposes. The project methodology is directed towards finding the optimum posture at an incompletely specified posture, i.e. out of the six scalar quantities (three Cartesian point coordinates and three independent orientation variables) defining a rigid-body pose, only five are specified in machining operations using a turning tool (the orientation of the tool about its axis of rotation is left unspecified). This unused dof is exploited to maximize robot performance.

1.1.3 Publications arising from the research

Part of the work on the posture optimization exploiting functional redundancy, which will be developed in Chapter 5, has been applied to a practical case study of a spherical PKM performing an assembly task. This project was presented at the *Advances in Robot Kinematics* (ARK) conference and it is available in a open source version. The reference of the printed version of the paper is

- D. Corinaldi, J. Angeles and M. Callegari, "Posture Optimization of a Functionally Redundant Parallel Robot", in *Springer Proceedings on Advanced Robotics* (SPAR). Springer, 2017.

1.2 Industrial Parallel Robots

Among the wide range of programmable robotic mechanical systems, this thesis deals with the industrial types of robots called manipulators. A manipulator, in general, is a mechanical system aimed at object manipulation. They deserve special attention for various reasons: the arm is the simplest robotic form and hence, appears as building blocks of other, more complex robotic mechanical systems. Two types of industrial manipulators can be distinguished: serial and parallel. The serial manipulators have an open chain architecture like an human arm, while, from Merlet's book [5] *a parallel mechanism is defined as a multi-degree-of-freedom (multi-DOF) mechanism composed of one moving platform (MP) and one base platform (BP)connected by at least two serial KCs in parallel. These serial KCs are called legs (or limbs).* Because of the closed-loop architecture, not all of the joints can be independently actuated and usually the number of actuated joints is selected to be equal to the number of degrees of freedom of the manipulator. Parallel manipulators whose number of chains is strictly equal to the number of d.o.f. of the moving platform are called full-parallel kinematics machines. Similarly, they are called full-mobility parallel manipulator when they can position and orient its mobile platform with six-dof. From a constructive point

of view there are many choices regarding the type of legs, joints and numbers of attached legs to the platform which can be significant factors when the workspace and actuation requirements are determined. When the legs of the parallel manipulator are identical the robot is called symmetrical. The discussion will focus on symmetrical parallel manipulators.

1.2.1 Characteristics of PKMs

The first prototypes of parallel kinematics machines were enthusiastically welcomed in 1994 as the new generation of machine tools thanks to their specific characteristics that guarantee better performance [6], namely:

- simpler solution of the inverse kinematics problem IKP;
- noncumulative joint error;
- higher structural rigidity: the load is carried by the parallel links, and in some structures there are only compression-tension modes;
- modularity: each kinematic chain is composed of the same physical modules.
- Locating the actuated joint adjacent to the fixed base, rather than attaching it midway in an articulated leg like the traditional Stewart-Gough platform 2.2, is considered advantageous [7].Such architectures proved to be beneficial based on the following factors [8]:
  - absorption of major portion of reaction forces by the ground resulting in almost vibration-free operation with light-weight mobile components;
  - reduced effect of inertia due to the elimination of actuator's weight;
  - absence of interference of actuators and routing cables due to base location of actuators;
  - further, by selecting the base actuated joint to be prismatic, the proximal links are not subjected to the bending moments and the corresponding stresses.

Contrary to serial robots, where all joints are actuated, parallel robots are supplied with unactuated joints, which brings about a substantial difference between the two types. The presence of unactuated joints makes the analysis of parallel manipulators, in general, more complex when compared to serial robots. However, PKMs suffer from

- singular configurations, a well-known problem in the robotics field;
- a low workspace-to-footprint ratio;
- a complicated solution of the direct or forward kinematics problem (DKP);

- more difficult control techniques.

The kinematics and dynamics behavior of the physical machine is strongly influenced by manufacturing tolerances and assembly errors, and so, especially for machining applications, calibration strategies must be defined and consolidated. Functional optimization may further increase the performance of these manipulators. To this aim, this research could be a stimulus to their spread in the manufacturing field. In fact, once PKM performance as a machine tool reaches high level, economy of scale will solve any cost issues due to their production.

1.2.2 PKM machining centers (1960-2000)

Despite a wide range of PKM architectures have been proposed, nowadays the most popular implementation on a commercial scale is the Stewart-Gough platform born from the early days of the parallel robotics [9].
Jaime Gallardo-Alvarado [10] reveals that the first attempts in parallel machine design have been made for cinematography industry in 1931 by Gwinnet. He proposed a spherical parallel manipulator as a cinema motion simulator. Several years later, a more practical application of a parallel manipulator was introduced by Willard I.V. Pollard (1940): a spray painting machine consisting on a five-dof three-branched parallel robot. Unfortunately, both those two parallel robots were never built. A couple of years later, in 1947, the nowadays called *Stewart-Gough platform* was invented, i.e. the variable-length-strut octahedral hexapod. This parallel robot became the most popular, because it changed industry, and was replicated over a thousand times. Dr. Eric Gough was the person who built this machine in the early 1950s, as a universal tire-testing machine. However, it was Klaus Cappel who later designed independently the very similar hexapod. He patented it, licensed it to the first flight simulator companies, and made the first commercial octahedral hexapod motion simulators. Yet, it was Stewart in 1965 who, unintentionally, made Gough's concept popular and proposed the idea for flight simulators, this time to academia.
In 1966 Tindales [11] proposed the use of parallel kinematics machining for manufacturing processes; however, a great interest grown for the application of these mechanisms in the metalworking field only in the 90s [6]. The first CNC-type hexapod machine tool prototypes (Variax from Giddings & Lewis Fig. 1.1 a) and the Octahedral Hexapod from Ingersoll Fig. 1.1b and c) were presented at the 1994 International Machine Tool Show (IMTS) in Chicago. These prototypes were enthusiastically welcomed as the new generation of machine tools due to their specific characteristics that should guarantee better performance, as described in the previous subsection. All of these advantages induced machine tool builders and researchers to investigate the applications of parallel kinematics machines for six-axis machining, a field where traditional machine tools had not gained the hoped-for success yet. Since the 1994 debut, several other prototypes of parallel kinematics machines have been built and

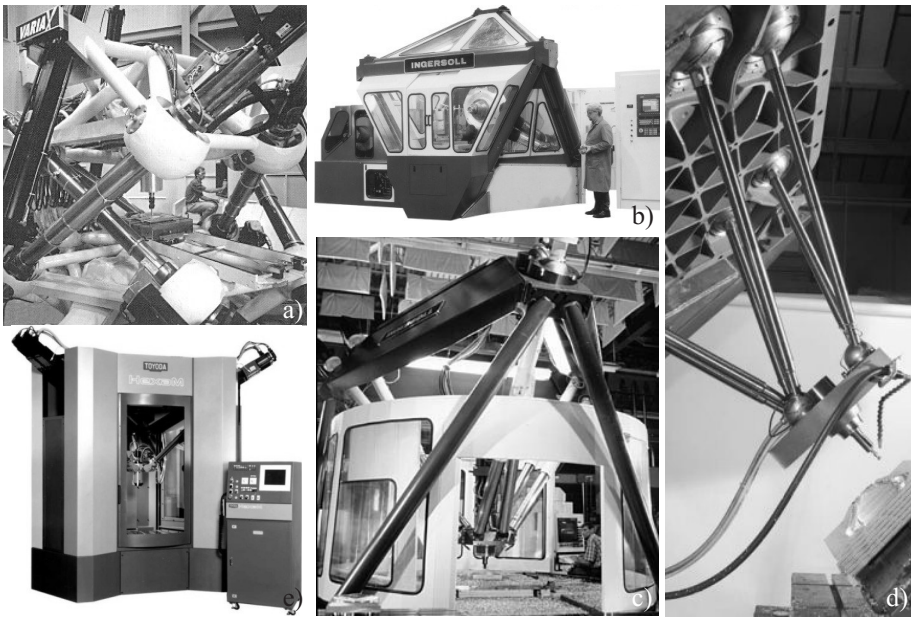


Figure 1.1: six-axis PKM machining centers. a) Variax (Giddings & Lewis), b) Octahedral Hexapod HOH-600 and c) VOH-1000 (Ingersoll), d) Tornado (Hexel), e) HexaM (Toyoda). Video by clicking on the figure: G&L Variax advertisement.

commercialized: Fig.s 1.1 and 1.2 show an a non-exhaustive list of six-axis PKM machining centers between the years 1960 and 2000. A chronological list of European key patents was presented by Pritschow [12] during the 1^{st} *European-American Forum on Parallel Kinematics Machines* held in Milan in 1998. Other prototypes of PKMs for machining operations were presented during the *EM0* in Paris, and it was claimed that this exhibition might mark the rebirth of PKMs. According to Tonhoff, more than 90% of existing PKM prototypes in the 2000s and employed in the machine tool sector where high stiffness is required, are fully parallel. Moreover, most of them belong to the Stewart-Gough type even though the technology of joint manufacturing allows rotations of the mobile platform to be limited to 30 degrees. Nevertheless, as highlighted at the 1999 *EM*0 in Paris, many new prototypes with different architectures from the Stewart platform are appearing.

1.2.3 Advances on PKM

Today, the research stream on PKM is still active and requested, as evidenced for example in Europe by the Robotic Roadmap for financial support of the H2020 program. According to these guidelines, the goals for mechanical systems design can be summed up as smaller, lighter, faster, stronger. In this framework, a relevant technol-

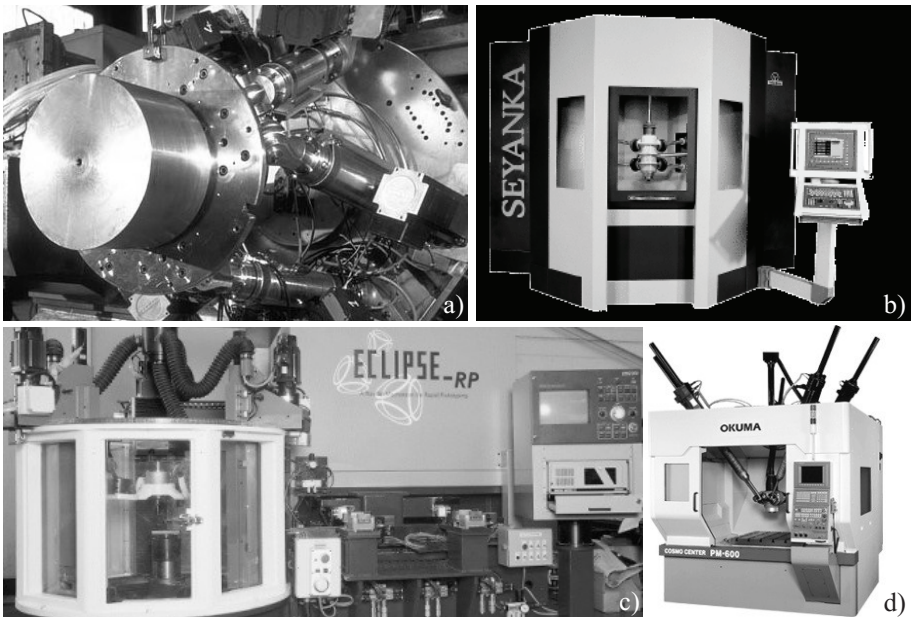


Figure 1.2: Other six-axis PKM machining centers. a) HEXAPODE CMW 300 (CMW), b) SEYANKA (Tekniker), c) Eclipse-RP (Daeyoung Machinery), d) Cosmo Center PM-600 (Okuma). Video by clicking on the figure: Okuma PM-600 six-axis milling machine.

ogy target is redundancy optimization: new control paradigms with constraint-based optimisation and the use of task redundancy for best trade-off among different objectives (e.g. productivity, manipulability, safety, ergonomics, etc.). By way of example, a very active research center for Research and Innovation in the field of PKM manufacturing is the *Nuclear Advanced Manufacturing Research Centre (Nuclear AMRC)* at the University of Sheffield. The ARMC receives significant investments by the European Commission to innovate manufacturing processes in particular in civil nuclear field. A Research involving the Nuclear AMRC's innovative robotic machining cell with a PKM has been presented at a high-profile engineering conference in Toulouse, France (2011). The project aims to develop a single automated system which can carry out a range of processes such as machining, welding, dressing and inspection over a large area to very high precisions. The robot cell is based around a hexapod PKM from Fanuc Robotics which can carry a variety of tool heads, e.g. in Fig. 1.3 the PKM carries a milling spindle. The FANUC F200iB parallel robot which has been implemented is engineered for applications requiring extreme rigidity and exceptional repeatability. The robot's position is tracked by an indoor GPS system and in trials, accuracies of $0.2\,mm$ have been achieved. However, further improvements could be done with the solution of the optimization problem to find the best posture due to

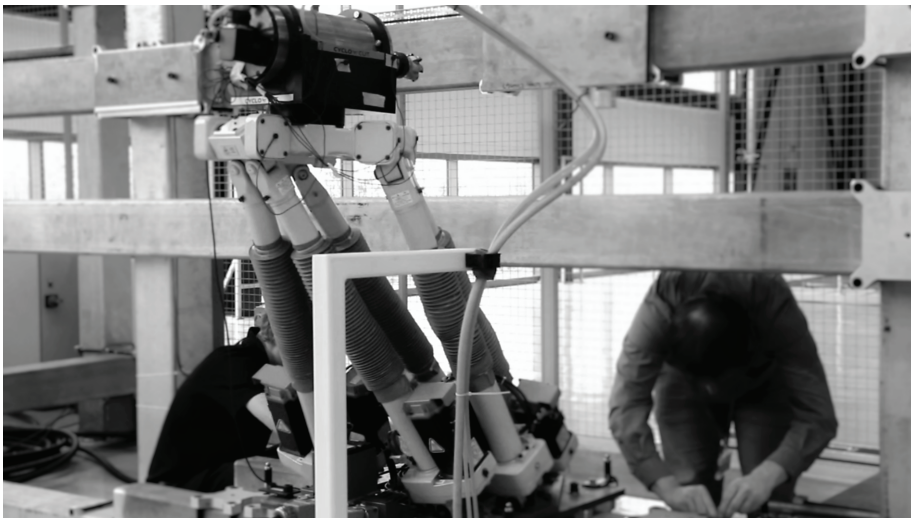


Figure 1.3: AMRC's innovative robotic machining with FANUC F200iB PKM. The video by clicking the figure shows the flexibility and accuracy of movement of the robotic system.

functional redundancy, as evidenced by recent research developments. In fact, their research is still active in the field of advanced machining and surfacing for the New Nuclear Manufacturing (NNUMAN) project [13, 14]. NNUMAN researchers report that future designs for large nuclear vessels and components will require more efficient machining techniques for both existing and future reactor materials. The research center is leading research to develop and characterize optimised cutting techniques, which include key aspects of machine dynamics. This research explores highly innovative approaches for machining very large components, for example using deep-hole drilling and using machining robots with indoor positioning systems, together with assisted machining techniques.

1.3 Industrial relevance of five-dof tasks

Parallel robots have left academic laboratories and have found their way in an increasingly larger number of application fields, such as telescopes, fine positioning devices, fast packaging, medical and machine tool. A wide range of industrial applications do not require to fully position the robot EE in the Cartesian space, i.e usually the dimensions of the task space have a dimension lower than six, i.e. lower than the complete dimension of Cartesian space. This is because the moving platform holds a tool for the manufacturing task, e.g. a spindle for milling, that has an axial-symmetry. The rotation of the MP along this axis is irrelevant for the task. Only five dof are specified in machining operations using a turning tool: the orientation of the tool about its

axis of rotation is left unspecified and optimization strategies should be used to find the best robot posture to maximize robot performance. In this section, a research was tackled on industrial processes that requires five dof: milling, tape laying, additive manufacturing and plasma, water jet and laser cutting are five-dof task common in manufacturing operations. A non-exhaustive list of these task is proposed, offering to the reader reflections on possible advantages provided by redundancy exploitation.

1.3.1 Machining with PKMs

Compared to a milling machine or a lathe, serial robots possess much less stiffness (by a factor of 20–50 times), but much greater dexterity. A widely accepted definition states that the dexterity of a robotic moving platform is a measure of its capability to change the manipulated object configuration from an initial configuration to a final one, arbitrarily chosen within the device workspace. A serial robot's stiffness is usually very anisotropic throughout its workspace and may vary for a typical heavy duty model in the range $200 - 700\text{N}/\text{mm}$. Therefore, robots can machine workpieces (grinding, fettling, polishing etc.) provided that tool forces can be reduced to acceptable values for a given robot manipulator. This incremental approach to machining, particularly for cutting and forming, can produce good results. The choice of mechanism, its kinematic properties, the computation methods used to determine joint motions, and the intended application of a robot manipulator are all closely related. With advances in the state of the art in kinematic algorithms and computer hardware processing capabilities, computation is much less of a constraint on mechanism choice than it was for early robot designers. For this reasons, the use of PKM can be a great compromise ensuring high forces but generally reducing the work space. The choice of mechanical structure of the PKM depends mostly on fundamental mechanical requirements such as payload and workspace size. Considering a given level of cost, there is usually a tradeoff between workspace size and stiffness.

Milling

The milling machine is one of the most important tool used in today's manufacturing industry, mostly used to cut special mechanical parts. The diffusion of parallel robots, instead of conventional gantry CNC machines for machining operation, arises from the need to obviate the low rigidity of the cantilever-link structure and the positioning errors of the cutting tool due to the actuator error accumulations. Despite the low workspace and dexterity, a PKM possesses high stiffness, high accuracy, higher load-carrying and lower inertia. To this aim six-dof hexapod PKMs are generally used, because they support the tool from six different directions. The stiffness of the structure is well conciliated with the choice of leaving the motors to the base to have a relatively light structure and thereby low inertia, features that improve the performance for the high-speed machining.

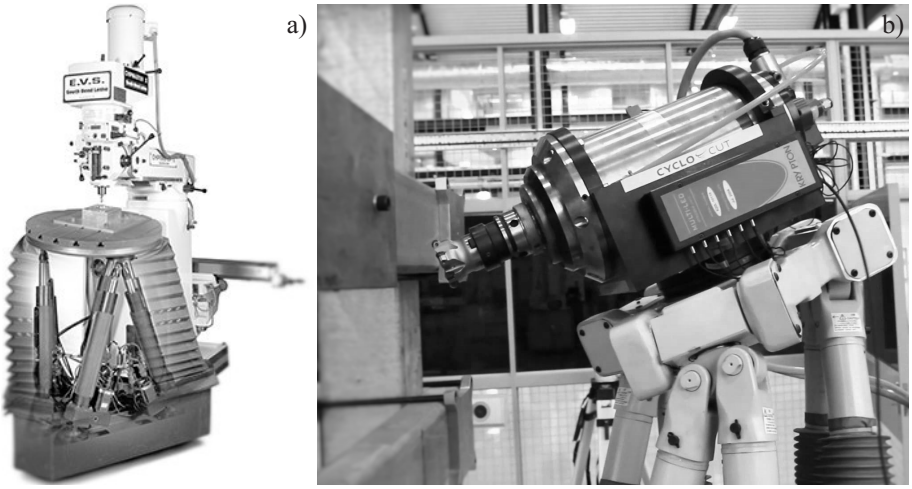


Figure 1.4: a) P2000 OKM designed by Mikrolar for milling application and b) Fanuc 200iB with a milling spindle.

Functional redundancy optimization could outperform the machine competition, feeding the demand on high quantities of dimensionally accurate and inexpensive parts that has increased as a result of technological progress. The optimal motion could offer better cutting efficiency and improve surface finish as well as machinability, pursuing the current research trend driven by the increasing requirements for cost competitions, process chain reduction and for developing economical and efficient manufacturing processes.

Welding: Friction Stir Welding (FSW)

Friction Stir Welding (FSW) represents one of the latest innovative techniques in the area of welding for its advantages over conventional fusion joining technology. As a matter of fact, typical drawbacks associated with conventional fusion welding, such as shrinkage, porosity and distortions, are eliminated to a great extent by FSW since it is a solid state process. Therefore, materials considered difficult to be fusion welded, such as aluminium and magnesium alloys, can be successfully friction stir welded [15, 16]. Friction stir welding can be applied in many industrial field, such as automotive, shipbuilding, aerospace and railway industries [17].
However, in some of these applications, FSW could be scarcely competitive since the high welding forces require expensive customised machines that, in turn, also cause relatively high productivity losses due to the inability to achieve high duty cycles. The large axial force to be maintained between the welding tool and workpiece is the primary requirement of FSW process, which has also been a great obstacle to the design and application of FSW in manufacturing. Further complicating the issue is the need

to perform FSW over three-dimensional contours, which requires a mechanism dexterous enough to set the stir pin used in welding to track a predefined trajectory with prescribed poses. Unlike common computer numerical control machining centers, often provided with only three motion axes, industrial robots have a high degree of mobility, which gives them an intrinsic versatility in accomplishing welding of complex geometries, also with tight radii of curvature.
Such drawback can be overcome by developing robotic FSW [18]. However a force feedback control, which is mandatory for compensating the reduced machine structural stiffness, often it is not sufficient for a serial robotic arm to ensure a high axial force. Therefore, in order to overcome the low-stiffness issues of serial robots, researchers at Polytechnic University of Marche proposes to employ a six-axis robot with a hybrid structure [19], characterised by an arm with parallel kinematics and a roll-pitch-roll wrist with serial kinematics, as illustrated in Fig. 1.5a). Shi et al. [20]

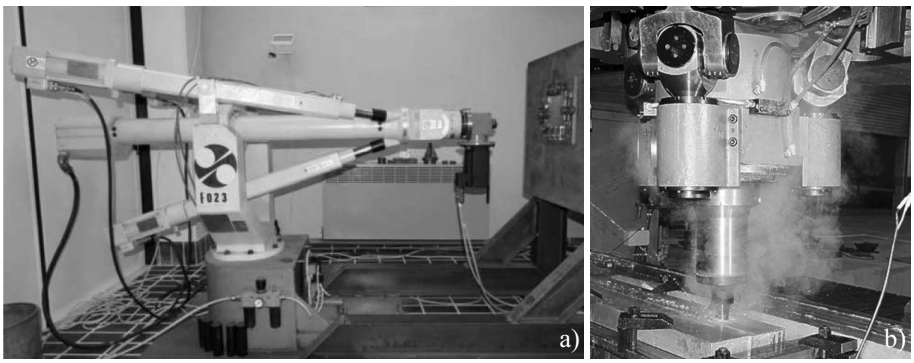


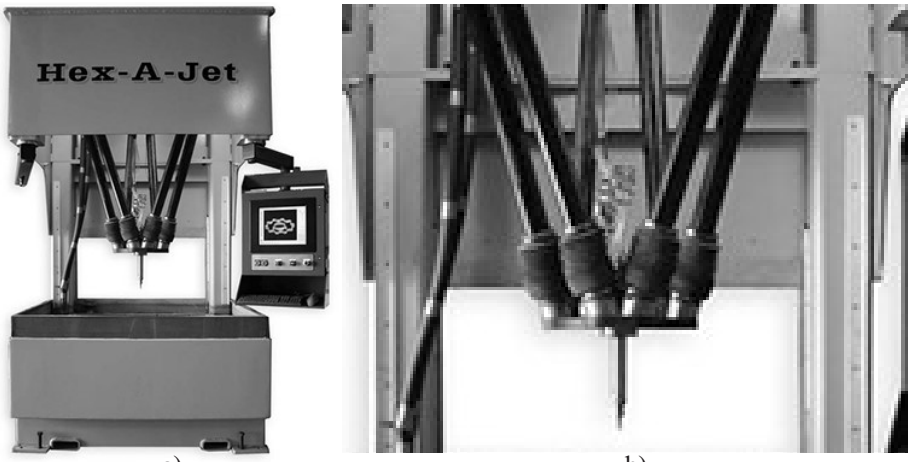
Figure 1.5: Picture and video of COMAU Tricept HP1 robot used for FSW at UnivPM a) and b) Shi's prototype of the PKM tool head for friction stir welding.

proposed the application of a 3-PRS parallel mechanism (tripod PKM with a prismatic, a revolute and a spherical joint in each limb) as a welding tool head and employed it to form a five-axis welding machine tool for FSW. The optimal design of the PKM was carried out by the authors, and the kinematic dexterity is estimated over the whole orientation workspace for optimization purposes. In the author's opinion, the full potentiality of parallel robotic for FSW has not been sufficiently investigated notwithstanding this system provides a very high flexibility and requires relatively low investments. Even for this application the functional redundancy that emerges from the use of a six-axis parallel robot, could further increase performance parameters.

Water jet cutting

Water jets are applied in nearly all areas of modern industry, such as automotive industry, aerospace industry, construction engineering, environmental technology, chemical process engineering, and industrial maintenance [21]. In the past, water jet applica-

tions have mostly been three-axis tasks: this fact has limited the number of parts that can be cut with a water jet, even if sometimes this technology would be better, faster and cheaper. Multi-axis surfaces, tapered edges or accurate non-tapered edges, casting flash removal and many other parts can be completed with the use of a six-axis PKM as a waterjet machine. An interesting commercial application of PKM for water jet cutting comes from *Mikrolar* company, that designed and produced high precision positioning systems. Mikrolar has also addressed significant issues in the manufacturing area and as a result has developed and patented a six-dof PKM system for rapid measuring, programming and water jet cutting of complicated or irregular parts called *Hexa-A-Jet* and showed in Fig. 1.6. The device is purpose-built to allow an operator



a) b)

Figure 1.6: Hex-A-Jet (Mikrolar) for water-jet cutting: a) machining center and b) detail of the hexapod PKM. Video by clicking the figure of a water-cut task.

to measure each individual part and then transfer that part directly to the water jet system for rapid material removal. The inherent design of the hexapod contributes to a high mechanical stiffness while the exploiting of functional redundancy could allow the workpiece to achieve the most stringent dimensional specifications.

Chapter 2

Kinematic Redundancy

The core concept of this dissertation is the functional redundancy, which represents a particular class of kinematic redundancy of a robot with respect to a particular task. In the past, such kind of redundancy found several appellations so that literature can lead to ambiguous definitions or even misunderstandings. In this chapter, the definitions of redundancy are reported as they were established for serial robots, and are extended to parallel manipulators. In the remainder of the section, the redundancy analysis of a class of parallel manipulators is performed by means of *screw theory* with the aim of quantifying their degree of kinematic redundancy.

2.1 Definitions

The *kinematic redundancy* is not an intrinsic feature of a machine, but rather a concept related to both manipulator and the task to be performed. In other words, if a manipulator is redundant for a specific task, it may not be redundant for another one. To properly define the redundancies, three spaces have to be identified:

- the *joint-space* $\mathscr{J}$ is the space of the independent actively controlled joint variables or actuated joint variables;
- the *operational-space* $\mathscr{O}$ is the Cartesian space reachable by the moving platform, whether a serial robot or a parallel robot is considered;
- the *task-space* $\mathscr{T}$ is the Cartesian space of the task.

Obviously, each space, that more precisely is a submanifold of $\mathrm{SE}(3)$, is characterized by a dimension. For the robot to be able to accomplish a task, the relations below should always be satisfied, even if redundancy is not available.

- $\dim(\mathscr{T}) \leq 6$, task space dimension cannot exceed the maximum number of degrees of freedom of a rigid body in the space (3 translational dof and 3 rotational dof);
- $\mathscr{T} \subseteq \mathscr{O}$, the task-space is totally included into the operational space of the manipulator, otherwise the task cannot be performed by the robot. Hence, $\dim(\mathscr{O}) \geq \dim(\mathscr{T})$.

- $\dim(\mathscr{T}) \leq \dim(\mathscr{O}) \leq \dim(\mathscr{J})$, the dimension of the operational-space is between the two other space dimensions. $\dim(\mathscr{J})$ has the greatest value and represents the *global mobility* of the robot.

At first sight, a simple comparison of the number of actuated joint variables and the number of degrees of freedom by the task indicates whether a given kinematic structure can be considered redundant or not.

Definition 2.1.1. *A pair made of a manipulator and a task is said to be kinematically redundant when the dimension of the joint-space $\mathscr{J}$, is greater than the dimension of the task-space $\mathscr{T}$.*

$$\dim(\mathscr{J}) > \dim(\mathscr{T}) \Leftrightarrow \text{kinematic redundancy of robot} - \text{task} \tag{2.1}$$

In other words, a kinematic redundant mechanism is a mechanism with more actuators than the number of controlled degrees of freedom required by the task. The degree of kinematic redundancy of a pair manipulator-task, namely r_k, is computed as

$$r_k = \dim(\mathscr{J}) - \dim(\mathscr{T}) \tag{2.2}$$

However, definition 2.1.1 does not take into account the operational space. A more rigorous definition distinguishes among *functional* and *intrinsic* redundancy.

Definition 2.1.2. *A manipulator is said to be intrinsically redundant when the dimension of the joint-space is greater that the dimension of the resulting operational space.*

$$\dim(\mathscr{J}) > \dim(\mathscr{O}) \Leftrightarrow \text{intrinsically redundancy robot} \tag{2.3}$$

Definition 2.1.3. *A pair made of a manipulator and a task is said to be functionally redundant when the operational-space dimension is greater than the dimension of the task-space.*

$$\dim(\mathscr{O}) > \dim(\mathscr{T}) \Leftrightarrow \text{functional redundancy of robot} - \text{task} \tag{2.4}$$

From these two definitions 2.1.2 and 2.1.3, the degree of intrinsic and functional redundancy for a manipulator and a task, named respectively r_I and r_F are computed as

$$r_i = \dim(\mathscr{J}) - \dim(\mathscr{O}) \tag{2.5}$$

$$r_f = \dim(\mathscr{O}) - \dim(\mathscr{T}) \tag{2.6}$$

Clearly, from Eq.s (2.5) and (2.6) the total degree of kinematic redundancy of Eq. (2.2) can be rewritten as

$$r_k = r_i + r_f \tag{2.7}$$

which makes clear that kinematic redundancy comes from two different sources: the functional and the intrinsic redundancy.

2.2 Redundancy Analysis

Redundancy analysis of robots and tasks aims to determine their degree of kinematic redundancy. Since the dimensions of the various spaces are required, the analysis is usually based on the mobility criterion of a mechanism which can take different forms. The *general* or *full-cycle mobility* of the manipulator of degree $m = \dim(\mathscr{J})$, i.e. the number of independent variables needed to specify any configuration of the manipulator, may be determined from the well known *Chebychev-Grübler-Kutzbach' formula*:

$$m = 6(l - n - 1) + \sum_{i=1}^{n} d_i \tag{2.8}$$

where l is the total number of members (including the base), n the total number of joints and d_i the number of degrees of freedom of the joints of the i^{th} chain. It is well known that the classical Chebychev-Grübler-Kutzbach mobility criterion 2.8, which is based solely on topology, fails to provide the correct mobility in many instances. In order to clarify such issue, the analysis of the kinematic chains mobility is carried out by employing the concept of *screw*, which allows linear/angular velocities or torques/forces to be represented in six-dimensional vectors, namely twists and wrenches. In kinetostatics, a body's instantaneous motion is represented by a twist, while a system of forces acting on the body is given by a wrench. Screws are classes of twists (or wrenches) that are scalar multiples of each other. In classical geometrical terms, the finite-pitch screw of a twist is given by a line, i.e. the screw axis, and a metric quantity, i.e. the pitch. Infinite-pitch screws are pure directions (free vectors corresponding to instantaneous translations or force couples), obtained as a limit case when either the screw axis or the pitch goes to infinity [22]. The mobility analysis developed by Kong and Gosselin [23] is taken up and adapted for the determination of the kinematic redundancy. The approach, based on screw theory, allows the instantaneous mobility of the machine to be detected, however, it can be extended to the full-cycle mobility.

The instantaneous motion of the moving platform, whether it is held by a single or multiple kinematic chains, is represented by its *twist system*. A system that requires a minimum of t linearly independent screws for its description is called a t-system. The constraints on the EE provided by the links of a serial KCs, by means of the KC joints, can be expressed as the screw system reciprocal to the twist system, which is usually called the *wrench system* of the KC. The wrench system of a KC is a w-system, and for the reciprocity bond $w = 6 - t$. Twist and wrench systems, whose order is respectively t and w, are screw systems obtained by associating a scalar, i.e. the twist

amplitude (or the wrench intensity), to a screw. A screw system of order n ($0 \leq n \leq 6$) comprises all the screws that are linearly dependent on n given linearly independent screws. According to the definitions of the spaces of the previous subsection, if the twist system does not change order when the moving platform undergoes a small displacement from a general configuration, the order of twist system of the EE t is equal to the dimension of the operational space, i.e. $t = \dim(\mathscr{O})$. The mobility m or global mobility of a manipulator is the sum of

- the number of independent parameters required to determine the relative configuration of the moving platform and
- the number of independent parameters required to determine the configuration of all the links in all the legs with the relative configuration of the moving platform specified.

For an m-dof mechanism, m is equal to the dimension of the joint space, $m = \dim(\mathscr{J})$ and it is also the number of the actuated joints whenever the robot is not intrinsically redundant.

2.2.1 Serial Kinematic Chains

In a serial manipulator the moving platform is connected to the ground by a single kinematic chain. The algorithm for the determination of the degree of functional redundancy is divided into the following steps:

Step 1. The mobility m of a serial KC is equal to the sum of the degrees of freedom of all its joints. Revolute joints (R), prismatic joints (P) are one-dof joints, universal joints (U) and cylindrical joints (C) leave two-dof, while, the spherical joints (S) three-dof. Joints with more than one-dof can be substituted in the kinematic chain with a series of one-dof joints that are kinematically equivalent.

Step 2. The twist system $\mathscr{T}$ or the wrench system $\mathscr{W}$ of the KC must be determined. In a serial manipulator these systems coincide for the MP and for the KC. The output twist of the moving platform $\mathscr{T}$ is the summation of the twists of the j joints $\mathscr{T}_c$ of the KC.

$$\mathscr{T} = \sum_{j=1}^{m} \mathscr{T}_c \tag{2.9}$$

Due to the reciprocal bond between twists and wrenches, it is possible to assert that the wrench system of the KC $\mathscr{W}$ coincides with the intersection of the

wrench spaces $\mathscr{W}_c$ that are reciprocal to each joint twist:

$$\mathscr{W} = \bigcap_{j=1}^{m} \mathscr{W}_c \tag{2.10}$$

Identifying a wrench space reciprocal to a given twist system is a simple linear operation. A non-exhaustive list of rules aimed at identifying wrench systems composed by zero-pitch and infinite-pitch screws was proposed by Kong and Gosselin. As already said, every joint with more than one dof in the kinematic chain is kinematically equivalent to a series of revolute (R) and prismatic joints (P) allowing the KC to be decompose as a series of R and P joints. In a KC with R and P joints, the following rules can be followed based on the reciprocity condition of screws:

- There is a 0-pitch wrench in the wrench system, which represents a force constraint exerted by the chain on the terminal, if its direction is coplanar to the axes of any R joint in the KC and it is perpendicular to the direction of any P joint in the KC.
- There is ∞-pitch wrench in the wrench system, which represents a torque exerted by the KC on the terminal, if its axis is perpendicular to the axis of any R joints in the KC.

The mobility obtained using twists and wrenches is an instantaneous characteristic of the chain. When t and w are the same in different general configurations, the mobility m is said to be *full-cycle*.

Step 3. With this assumption, the degree of *intrinsic redundancy* r_i of the serial chain, is the difference between the dof of the serial chain m and the order of the twist system of the KC t. By using Eq. (2.9) to determine the twist system of the EE, the Eq. (2.5) with a different notation becomes:

$$r_i = m - t \tag{2.11}$$

Given n t-system, there is a unique reciprocal screw system of order $(6-n)$ which comprises all the screws reciprocal to the original screw system. Therefore, we have

$$t = 6 - w \tag{2.12}$$

Eq. (2.11) can be rewritten in term of the wrench system order w instead of t

$$r_f = m - 6 + w \tag{2.13}$$

which is more useful if the wrench system is obtained using the rules defined.

Step 4. The number of the dof requires by the task allows the degree of func-

tional redundancy r_f to be calculated through the formula Eq. (2.20):

$$r_f = t - \dim(\mathscr{K}) \tag{2.14}$$

Step 5. Finally the degree of kinematic redundancy r_k could be calculated with Eq. (2.7).

Following this line of reasoning, to create an intrinsically redundant dof there will be a linear dependence between the joint twists in the KC. The conditions for the twists of the joints within the same leg to be linearly dependent are not difficult to address if the KC is composed of only R and P joints. For brevity, such conditions are listed below without explanation [23]:

1. there are no coaxial R joints;
2. there are no P joints along the same direction;
3. the direction of at most one P joint is parallel to the axis of an R joints;
4. at most three R joints have parallel axes;
5. the axes of at most three R joints pass through the same point;
6. the directions of at most two P joints are parallel to the same plane;
7. the sum of the number of R joints with parallel axes and the number of P joints is not greater than four;
8. if the directions of nP P joints are perpendicular to the axes of nR R joints with parallel axes, then $n\mathrm{P} + n\mathrm{R} \leq 3$;
9. There are at most six R joints in the KC.

Example: 7R Canadarm2

The Canadarm2 used on the International Space Station is a 7R serial manipulator. The algorithm to determine its degree of kinematic redundancy is applied:

Step 1. The total number of dof of the joints is $m = 7$, because there are seven R joints in the KC with one-dof.

Step 2. The wrench system of the EE is a 0-system ($w = 0$) because no screw can be found that respect the rules 2.2.1, i.e. it is coplanar with any R joint axes or perpendicular to any of them. Therefore, $t = 6 - w = 6$, i.e. the twist system attains its maximum order and the EE can be relatively positioned with six-dof.

Step 3. The degree of intrinsic redundancy is equal to one $r_i = m - t = 1$, in fact, the limit imposed by point 9 of the previous list 2.2.1 is exceeded.

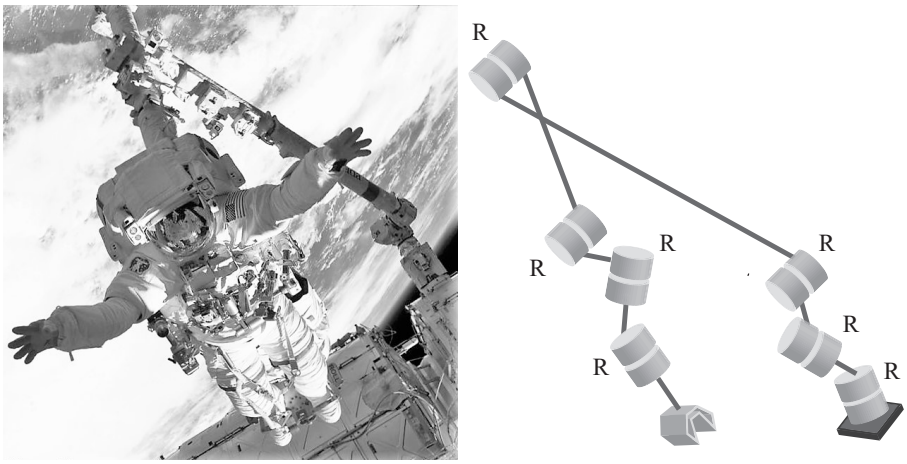


Figure 2.1: Picture of Canadarm2 on the left and the scheme of its architecture on the right. Video by clicking the picture shows the mobility of the Canadarm2.

Step 4. If the task is to position and orient the astronaut in the entrance to the space station, it is a six-dof task, therefore the degree of functional redundancy is $r_f = 0$.

Step 5. The total degree of kinematic redundancy is $r_k = 1$.

2.2.2 Parallel Kinematics Machines

A parallel manipulator is constituted by more than one serial kinematic chains which connect the *base platform* (BP) to the *moving platform* (MP). The redundancy analysis resumes the analysis for the serial chain because steps 1-3 are first performed to each kinematic chains of the parallel kinematics machine PKM, disconnected from the platforms. Their contribution is then extended to the mobile platform. The method is here proposed:

Step 1. First of all, each leg of the parallel mechanism should be considered and treated as a serial kinematic chain. Due to this, steps from 1 to 3 of the previous method have to be performed for each kinematic chain indicated with the subscript c. As a consequence, r_{ic} degrees will be found of intrinsic redundancy, with $c = 1, ..., p$.

Step 2. The number of independent parameters needed to determine the configuration of all the links in all legs with the relative configuration of the moving platform specified is the intrinsic redundancy degrees of the PKM and it is the

sum of the intrinsic redundant dof of all the legs:

$$r_i = \sum_{c=1}^{p} r_{ic} \tag{2.15}$$

where r_i is the *degree of intrinsic redundancy* of the PKM.

Step 3. Once the twist-system of each leg of order t_c is found, the mobility of the MP, i.e. the number of independent parameters to determine the configuration of the moving platform is equal to order t of the twist system of the PKM. Therefore, t is equal to the order of the operational space of the PKM and it is also called the *connectivity* of the moving platform. The twist system $\mathscr{K}$ of the MP is the intersection of the twist systems $\mathscr{K}^i$ of all its legs

$$\mathscr{K} = \bigcap_{c=1}^{p} \mathscr{K}_c \tag{2.16}$$

Contrary to what happens for the serial chains, the output wrench-system $\mathscr{W}$ of order w of the moving platform is the linear combination of the wrench systems $\mathscr{W}^i$ of all its legs:

$$\mathscr{W} = \sum_{c=1}^{p} \mathscr{W}_c \tag{2.17}$$

Even in this case, Eq. (2.12) is valid . The instantaneous analysis is extended to the full-cycle mobility if the order of the system screws involved are the same for others general configurations of the PKM.
In addition to the order of the twist system of the MP, another important index of a PKM is defined as

$$\Delta = \sum_{c=1}^{p} w_c - w \tag{2.18}$$

where Δ is called the *number of over-constraints* (also *passive constraints*) if $\Delta > 0$.

Step 4. The mobility m of the PKM, i.e. the number of independent parameters to determine the configuration of the manipulator is the sum of the mobility of the MP t and the degree of intrinsic redundancy r_i.

$$m = t + r_i = 6 - w + \sum_{c=1}^{p} r_{ic} \tag{2.19}$$

m is also the number of the actuated joints or *active joints* of the parallel robot and it is equal to the dimension of the joint space. Clearly, unlike a serial robot, in a parallel robot there are usually *passive joints*.

Step 5. The number of dof required by the task allows the degree of functional redundancy r_f to be calculated through the formula:

$$r_f = t - \dim(\mathscr{K}) \tag{2.20}$$

Step 6. Finally the degree of kinematic redundancy r_k could be calculated with Eq. (2.7).

This redundancy analysis will be illustrated using an example of a PKM performing a task.

Example: 6-UPS PKM Stewart-Gough platform

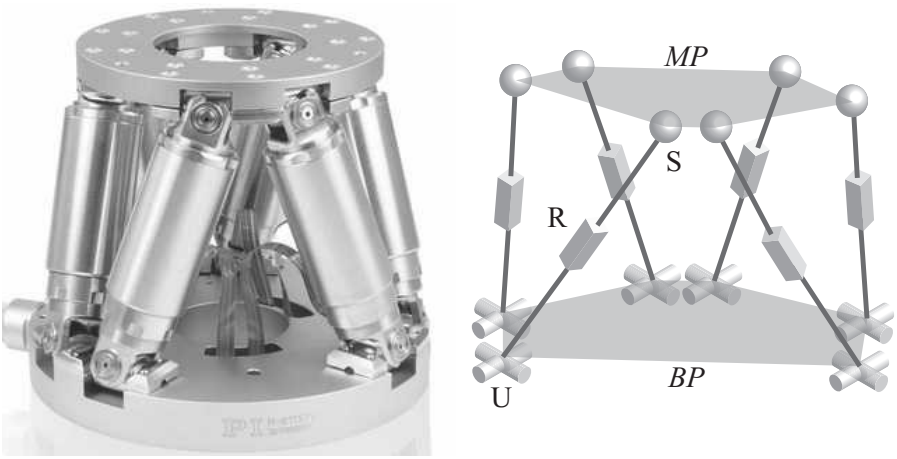


Figure 2.2: A Stewart-Gough commercial platform: PI H-811.S11 six-axis motion hexapod a) and its advertisement on the video by clicking the figure. The architecture scheme of a classical Stewart-Gough platform is illustrated in b).

Consider a Stewart-Gough platform, a six-legged PKM, shown in Fig. 2.2a. From the architecture scheme Fig. 2.2b, the direction of the P joint of each leg intersects the center of both the universal and the spherical joints, located at the BP and at the MP respectively. The parallel robot has six identical legs and so it is called *symmetrical parallel manipulator* with URS topology.

Step 1. For each KC $m_c = 6$ (two dof for the universal joint, one dof for the prismatic joint and three dof for the spherical joint), while the order of the twist system is $t_c = 6$, no one screw can be found that respects the previous conditions. Thus, $r_{ic} = 6-6 = 0$ for $c = 1,...,6$. These parameters do not change when the moving platform undergoes a small displacement from a general configuration, therefore the mobility is full-cycle.

Step 2. The degree of intrinsic redundancy of the PKM is $r_i = 0$.

Step 3. Since no constraint is imposed by the legs on the MP, it has full mobility $t = 6$: the number of independent parameters to determine the relative configuration of the moving platform is six. The number of over-constraints is $\Delta = \sum_{c=1}^{6} 0 - 0 = 0$.

Step 4. The mobility m of the PKM is $m = t + r_i = 6 + 0 = 6$ and it is also the number of the actuated joints. Usually they are the six prismatic joints.

Step 5. The task to be performed requires to position and point the MP: the rotation around the axis of pointing does not affect the purpose of the task, thus the required degrees of freedom are 5. $r_f = t - \dim(\mathscr{K}) = 6 - 5 = 1$.

Step 6. The degree of kinematic redundancy r_k could be calculated as $r_k = r_i + r_f = 1$.

2.2.3 Global mobility inspection

The redundancy analysis of a PKM obtained using the foregoing method is instantaneous. A systematic approach for the determination whether the robot has global mobility or not is provided by Kong and Gosselin in Appendix B of [23]. For terseness the methodology is not given here, however a sufficient condition to prove that the instantaneous mobility may be extended to the global mobility is that $\Delta = 0$ i.e. the PKM is isostatic. Then, the instantaneous mobility analysis of the isostatic PKM can be extended to the global mobility and inspection ends, otherwise other controls have to be made to extend this concept.

Chapter 3

A class of tripod SPMs

One of the issues in the analysis of parallel kinematics machines with six-dof is the complexity of their kinematics, which can adversely affect the path-planning. In conventional processes, full-mobility operations are commonly decomposed into elementary sub-tasks, to be performed by separate machines with lower mobility. A case study is described by Carbonari et al. [24] and it has been developed until the prototypal stage, decomposing the architecture of a mechatronic system with six-dof, in two parallel robots cooperating while performing a five-dof task. This idea is described in this chapter and the PKM that belongs to the functional redundancy is identified and investigated. Since its MP undergoes a three-dof spherical motion (or rotates about a fixed point), it is called spherical parallel manipulator (SPM). After a classification of this type of manipulators, the chapter examines a particular class of isostatic SPMs. The class under study is interesting because the manipulators may work even when precise geometrical conditions are not accurately satisfied. A convenient architecture for the class of robots that maximizes the workspace and grants optimal manipulability is also proposed. Kinematics relations and reciprocal screws systems are found by means of screw theory. The study allows us to expand the scope of investigation and to attempt a generalization of the theory to be developed. However, despite only a manipulators class is considered, the application to a generic parallel manipulator with six-dof is still possible. The functional redundancy is always related to the rotational behavior when an axially symmetric tool is used and it can be separately studied from the positioning problem of the moving platform.

3.1 Decomposition in minor-mobility PKMs

Parallel kinematics machines are known to be characterized by many advantages like a lightweight construction and high stiffness but also present some drawbacks, like a limited workspace, a great number of joints of the mechanical structure and a complex kinematics, especially for six-dof machines. To obviate the last problem, Tsai and Joshi [25] first introduced the idea of cooperation which derives from the concept of hybrid machines. They proposed to decompose full-mobility operations into elementary sub-tasks to be performed by separate minor mobility machines, as it is

commonly done in conventional machining operations. Although this solution requires a more sophisticated controller, it would lead to the design of simpler machines that could be used also in stand-alone applications for three-dof tasks and would also increase the modularity and reconfigurability of the robotized industrial process. It is remarked that this solution allows us to exploit the well-known benefits of parallel kinematics machines, but at the same time it overcomes many typical drawbacks.
To this aim the researchers at the Machine Mechanics Laboratory at the Polytechnic University of Marche, Ancona, envisaged the architecture of a mechatronic system where two parallel robots cooperate in order to perform a complex task, see Fig. 3.1. The kinematics of both machines is based upon the same 3-CPU topology but the

Figure 3.1: The minor-mobility PKM prototypes developed at the Machine Mechanics Laboratory: the upper is I.Ca.Ro while the lower, Sphe.I.Ro.

joints are differently assembled so as to obtain a translating parallel machine (TPM) with one mechanism and a spherical parallel machine (SPM) with the other. The two parallel robots have been developed till the prototypal stage: Fig. 3.2 shows the translating manipulator, called I.Ca.Ro. [26, 27] and the kinematics of the orienting manipulator, which has been called Sphe.I.Ro. [28]. By looking at the two machines, it appears clear that both are composed of three identical legs with the same joint sequence: cylindrical (C), prismatic (P) and then universal (U) joints, usually called 3-CPU topology. In Sphe.I.Ro, however, the axes of the cylindrical joints and those of the outer revolute pairs in the universal joints all intersect at a common point, which is

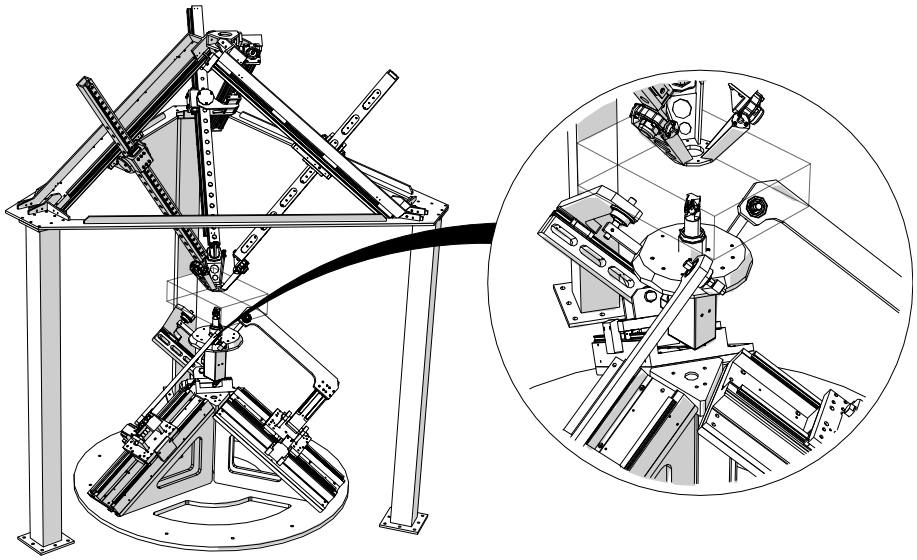


Figure 3.2: Virtual system architecture of a machining cell based on two cooperating three-dof parallel robots conceptualized at UnivPM.

the centre of the spherical motion of the mobile platform. By taking into consideration that both machines share the same 3-CPU topology, a more recent research looked for the possibility to obtain at a common structure that would be able to yield both kinds of motions [29], the kinematotripic 3-CPU.
When this decomposition is performed for a task that has a tool with a rotational axis, e.g. a milling operation, the functionally redundant manipulator is the spherical one as shown in Fig. 3.2. Then, this redundancy is exploited to orient the axis of the spindle while the manipulator is assuming a posture that maximizes performance. Since

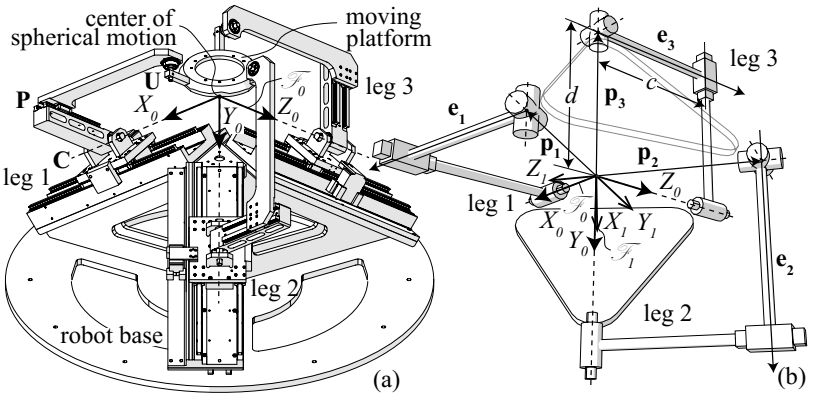


Figure 3.3: Virtual model (a) and prototype (b) of the spherical 3-CPU manipulator Sphe.I.Ro.

spherical parallel manipulators are of great industrial interest, their classification is proposed in the next section.

3.2 A classification of SPMs

According to Kong and Gosselin [23], a three-dof spherical parallel manipulator (SPM) is a parallel manipulator where the moving platform undergoes a three-dof spherical motion or rotates about a fixed point. It is also called three-dof orientational parallel manipulator (Karouia and Hervé [30]) or rotational parallel manipulator (Li and Huang [31]). SPMs have a wide range of applications including orienting devices and wrists. The research on SPMs has received much attention from many researchers, and many types of SPMs have been proposed.
Following the classification made by Kong and Gosselin [23] SPMs fall into two classes: *overconstrained* or *hyperstatic* SPMs and *non-overconstrained* or *isostatic* SPMs. According to the definitions given in Chapter 2, the condition of overconstrained PKM is reached when $\Delta > 0$, otherwise the PKM is isostatic when $\Delta = 0$. Three-legged overconstrained (hyperstatic) SPMs include:

O.1) 3-RRR SPMs in which each leg is composed of three R joints whose axes pass through the center of spherical motion ([32], [33], [34]);

O.2) SPMs in which each leg is composed of three R joints whose axes pass through the center of spherical motion as well as one inactive joint ([23]).

Another grouping may be done for the three-legged isostatic SPMs:

NO.1) SPMs in which each leg is composed of an R joint, a planar parallelogram joint, and an S joint ([35]);

NO.2) SPMs in which each leg is composed of two R joints whose axes pass through the center of spherical motion and three successive joints which are equivalent to a planar joint or a spherical joint ([30], [36], [37], [23], [38]);

NO.3) SPMs in which each leg is composed of three R joints whose axes pass through the center of spherical motion and two inactive joints ([39]; [23]; [40]; [31]).

The main advantage of the isostatic architecture is that it avoids the strict dimensional and geometric tolerances needed by overconstrained machines during manufacturing and assembly phases. Moreover, modular solutions characterised by three identical legs (symmetrical PKM) are preferred for economic reasons. It must be said that these advantages are usually paid with a more complex structure and the possible presence of singular configurations (translation singularities) in which the spherical constraint between platform and base fails [41]. The 3-CPU topology of Sphe.I.Ro falls into the NO.2) class that is studied in the next section.

3.3 An isostatic SPM class

Sphe.I.Ro. is a minor mobility parallel robot based on the 3-CPU architecture which provides the mobile platform with a spherical motion, driven by 3 linear induction motors. The center of the moving platform spherical motion coincides with the intersection point of the three axes of the cylindrical joints, as shown in figure 3.3. According to the classification of the previous section Sphe.I.Ro has an isostatic limb mechanism of the N.O2) type. In this section this class is studied in details to obtain the relevant screws of its leg joint. The planar motion of the three joint in the middle of the two revolute joints can be generated by a planar pair and it is kinematically equivalent to following sequences of prismatic joint and revolute pairs: PRR, RPR, PPR, PRP and RRR. However, the prismatic and revolute joints must respect the following rules to produce planar motion: P directions must be parallel to the plane of motion while R axes remain perpendicular to this plane. The combinations of these sequences with the other two revolute joints give the five leg topologies: (RP)R(RR)=CRU, (RR)P(RR)=UPU, (RP)P(RR)=CPU, (RP)R(PR)=CRC and (RR)R(RR)=URU, as illustrated in Fig. 3.4.

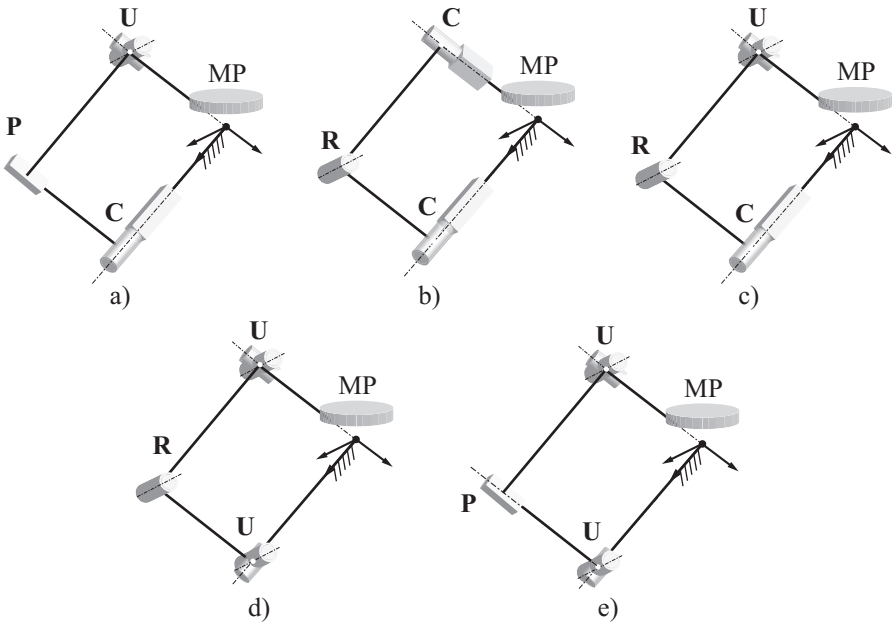


Figure 3.4: Limb architectures of the three legged isostatic SPMs class: a) CPU, b) CRC, c) CRU, d) URU and e) UPU SPMs.

3.3.1 Redundancy Analysis

The redundancy analysis is here applied for the class of manipulator under study.

Step 1. Each limb consists of four links and five revolute (R) or prismatic (P) joints. Since these joints leaves one-dof of relative motion between the links that they connect, the three limbs have connectivity equal to five $m_c = 5$.
Looking at Fig. 3.4, the reader could verify that there is a zero-pitch screw in the wrench system of each leg that is perpendicular to the plane of the leg and passes through the center of the spherical motion. The minimum order of the wrench system for each leg is therefore $w_c = 1$, thus, $r_{ic} = 5-5 = 0$ for $c = 1,2,3$. These parameters do not change when the moving platform undergoes a small displacement from a general configuration so the mobility obtained is full-cycle.

Step 2. The degree of intrinsic redundancy is $r_i = 5-5 = 0$.

Step 3. The architecture of the manipulator should be chosen in such a way that the three wrenches compose a three-system, allowing the mobile platform to translate. This could be achieved if the three screw are linearly independent: to this aim, in the next section a particular arrangement of the limb planes will be defined. In any general configuration, the wrench system of a SPM is a three-system whose center is at the center of rotation of the moving platform. The mobile platform has lower-mobility $t = 3$, i.e. the number of independent parameters necessary to determine the relative configuration of the moving platform is equal to three. The number of over-constraints is $\Delta = \sum_{c=1}^{3} 0-0 = 0$ in fact the class is called isostatic.

Step 4. The mobility m of the PKM is $m = t + r_i = 3$ and three is also the number of the actuated joints: in order to reduce inertia, they are usually the joints at the base.

Step 5. If the task to be performed is a pointing task, two dof are required and the degree of functional redundancy is $r_f = t - \dim(\mathscr{K}) = 3-2 = 1$.

Step 6. The degree of kinematic redundancy r_k could be calculated as $r_k = r_i + r_f = 1$.

3.3.2 Relevant joints screws

The screws associated to each joint and the reciprocal screws to a set of joints are required to define the indices which quantify the performance of the manipulator.
First of all, the screws associated to a revolute and to a prismatic joint are defined. A zero-pitch screw $\mathbf{s}^0$ is associate to a revolute joint and it is defined by a unit vector **e**, which defines the direction of the axis of rotation, and a vector **n**, that quantifies the moment about the origin of the reference frame. This moment **n** determines its

location by means of a vector $\mathbf{p}$ that is the position vector of any point of the screw. Thus, the screw coordinates of the zero-pitch screws $\mathbf{s}^0$ can be expressed as

$$\mathbf{s}^0 = \begin{bmatrix} \mathbf{e} \\ \mathbf{n} \end{bmatrix} = \begin{bmatrix} \mathbf{e} \\ \mathbf{p} \times \mathbf{e} \end{bmatrix} \tag{3.1}$$

A screw with infinite pitch $\mathbf{s}^\infty$ is described only by its axis $\mathbf{e}$ and it is associated with a prismatic joint

$$\mathbf{s}^\infty = \begin{bmatrix} \mathbf{0} \\ \mathbf{e} \end{bmatrix} \tag{3.2}$$

A 3-UPU PKM is taken into consideration. In the previous section, the topology of the limb has been defined; however, the three leg planes should be built in such a way that the three wrenches compose a three-system, thus they have to be linearly independent. The simplest possible setting of the limbs is that they all lie within vertical planes, while the three screws result linearly dependent, allowing the platform to translate along the vertical direction. Among all the possible setting of these normal axes in space that grant them to be linearly independent, it has been chosen to tilt the limbs' planes so that they are mutually orthogonal in the initial configuration, as already done by for the 3-CPU PKM by Callegari [41]. This arrangement greatly simplifies the kinematics relations that will be worked out later on; moreover, even if this arrangement changes during the machine operation, this configuration is the farthest from the singular setting, granting a better kinematic manipulability of the wrist. Fig. 3.5 shows this setting in a general configuration of the manipulator and the video confirms its mobility. Using the nomenclature indicated in the figure, the screws associated with each joints are written and collected in the first row of Table 3.1. The reader should pay attention that only for these tables the subscript on the right corner of the screws indicates its pitch, otherwise, it always indicates the reference system. Moreover, a feasible choice for the screws reciprocal to all the joints of the leg except for the joint considered is represented in the second row. These screws are useful because they allow the influence on the velocity relationships of the joint rates reciprocal to them to be eliminated, as we will see in the next chapter.
These screws can be written as a function of the parameters describing the pose of the MP through the rotation matrix. To this end, the vectors in Table 3.1 are expressed in terms of the rotation matrix $\mathbf{Q}$, represented in the reference system $\mathscr{F}_0$ that maps a vector from the mobile frame $\mathscr{F}_1$ fixed to the MP to the frame $\mathscr{F}_0$ fixed to the ground. The versors of the reference frame fixed to the ground are indicated with $\mathbf{i}_0$, $\mathbf{j}_0$, $\mathbf{k}_0$ while the versors of the reference frame fixed to the MP are indicated with the subscript 1. For the leg highlighted in Fig. 3.5 the relevance vectors could be expressed as function of the rotation matrix using Table 3.2. These relationships express the articular coordinates for a given pose of the moving platform pose as the results of the *inverse kinematics problem*. These relationships can be easily obtained from the

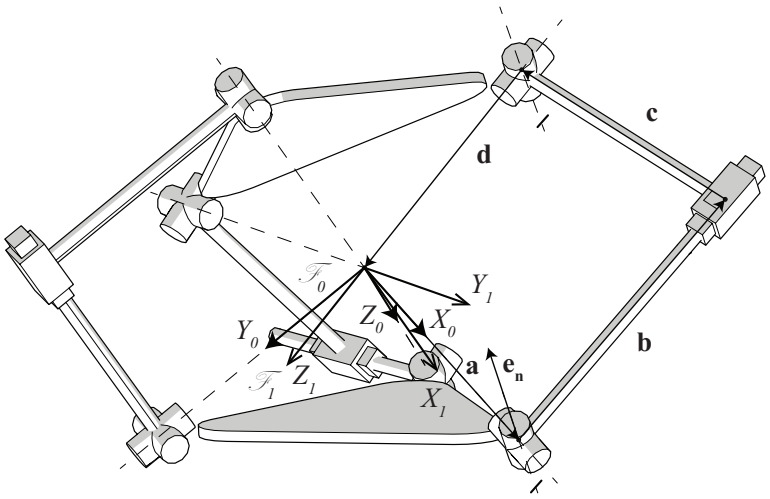


Figure 3.5: 3-UPU SPM in a general configuration. Video by clicking the figure: mobility verification through a spiral pointing trajectory of the MP.

	R	R	P	R	R
$\mathbf{s}$	$\begin{bmatrix}\hat{\mathbf{a}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{c}}\end{bmatrix}_\infty$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ \mathbf{0}\end{bmatrix}_\infty$
$\mathbf{s}^\perp$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{c}}\\ -\mathbf{d}\times\hat{\mathbf{c}}\end{bmatrix}_0$	$\begin{bmatrix}\dfrac{\mathbf{d}+\mathbf{a}}{\|\mathbf{d}+\mathbf{a}\|}\\ -\mathbf{d}\times\dfrac{\mathbf{d}+\mathbf{a}}{\|\mathbf{d}+\mathbf{a}\|}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{c}\\ \mathbf{a}\times\mathbf{c}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$

Table 3.1: 3-UPU joint screws of the first limb: the screw $\mathbf{s}$ of each joint and the screw $\mathbf{s}^\perp$ reciprocal to all the joint in the chain except for the considered joint.

study of the robot in some generic configurations. For example if we want to express these vectors in the reference frame $\mathscr{F}_0$, $[\mathbf{i}_0]_0 = [1\,0\,0]^T$ and $[\mathbf{k}_1]_0 = \mathbf{Q}[\mathbf{k}_1]_1$. Therefore $\mathbf{a} = [a\,0\,0]^T$ and $\mathbf{d} = -d\mathbf{Q}[0\,1\,0]^T$, where the parameters a and d are fixed geometrical parameters chosen during the design of the machine. These data of the others SPMs of the class are collected in the following tables.

	a	**d**	$\mathbf{e}_n$	**b**	**c**
$\hat{\bullet}$	$\mathbf{i}_0$	$-\mathbf{k}_1$	$\frac{-\mathbf{d}\times\mathbf{a}}{\|-\mathbf{d}\times\mathbf{a}\|}$	$\frac{\hat{\mathbf{a}}\times\mathbf{e}_n}{\|\hat{\mathbf{a}}\times\mathbf{e}_n\|}$	$\frac{\hat{\mathbf{b}}\times\mathbf{e}_n}{\|\hat{\mathbf{b}}\times\mathbf{e}_n\|}$
$\|\bullet\|$	a	d	1	$\sqrt{a^2+d^2-c^2+2\mathbf{i}_0\mathbf{i}_1}$	c

Table 3.2: 3-UPU IKP: unit vectors and magnitudes of the relevant vectors in the first leg.

3.3.3 Global Mobility Verification

The global mobility of the SPM class under study can be verified by using different approaches. It has been already used the screw theory for the instantaneous mobility analysis. The class of PKM is isotatic and this is a sufficient condition to extend the redundancy analysis to the global mobility, according to subsection 2.2.3. Also Hervé has already proven the mobility of this class of SPM in [42] by the means of Lie algebra. However, this subsection is dedicated at formally demonstrating the mobility of the SPM class using algebraic geometry and the Study's parameters to describe the MP pose. It allows us to write the polynomials of the bound equation of the single leg in a generic configuration.

In order to fully represent the attitude of the mobile platform, the Study's parameters are chosen for the parametrization of its kinematics. This well known notation [43, 44] was introduced by Study in (3.4)-(3.6) who used a superabundant set of eigh parameters in the projective space of dimension 7 ($\mathbb{P}^7$) to describe the Euclidean transformations. The 8 parameters $[x_0 : x_1 : x_2 : x_3 : y_0 : y_1 : y_2 : y_3]$ are related through a quadric polynomial equation (namely the Study's quadric) and through a metric equation which ensures that the transformations actually represent rigid body motions. Thus, the key idea is to map the Euclidean space through a one to one representation χ in the projective space $\mathbb{P}^7$:

$$\chi : {}^0\mathbf{T}_1 \in SE(3) \mapsto \mathbf{x} \in \mathbb{P}^7 \tag{3.3}$$

If the homogeneous notation is used, the transformation ${}^0\mathbf{T}_1$ describing the pose of the mobile platform with respect to the base platform looks like:

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \mathbf{p} \\ \mathbf{0} & x_0^2+x_1^2+x_2^2+x_3^2 \end{bmatrix} \tag{3.4}$$

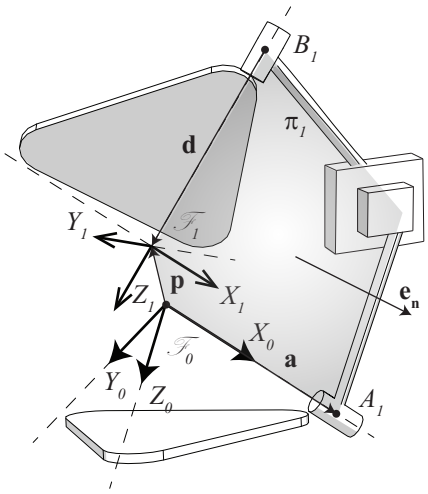


Figure 3.6: Kinematics of the i-th leg for the class architectures.

where $\mathbf{Q}$ describes the orientation between the two frames and it is

$$\mathbf{Q} = \begin{bmatrix} x_0^2+x_1^2-x_2^2-x_3^2 & 2(x_1x_2-x_0x_3) & 2(x_1x_3+x_0x_2) \\ 2(x_1x_2+x_0x_3) & x_0^2-x_1^2+x_2^2-x_3^2 & 2(x_2x_3-x_0x_1) \\ 2(x_1x_3-x_0x_2) & 2(x_2x_3+x_0x_1) & x_0^2-x_1^2-x_2^2+x_3^2 \end{bmatrix} \tag{3.5}$$

while $\mathbf{p}$ is the position vector between the two origins as shown in Fig. 3.6

$$\mathbf{p} = \begin{bmatrix} 2(-x_0y_1+x_1y_0-x_2y_3+x_3y_2) \\ 2(-x_0y_2+x_1y_3+x_2y_0-x_3y_1) \\ 2(-x_0y_3-x_1y_2+x_2y_1+x_3y_0) \end{bmatrix} \tag{3.6}$$

A set of such parameters corresponds to only one transformation in the Euclidean space and allows to uniquely define the configuration of a rigid body, e.g. the mobile platform, when the Study's quadric equations are satisfied:

$$\begin{aligned} x_0y_0+x_1y_1+x_2y_2+x_3y_3 &= 0 \\ x_0^2+x_1^2+x_2^2+x_3^2 &\neq 0 \end{aligned} \tag{3.7}$$

If the inequality in (3.7) is assumed equal to an arbitrary constant, the expression becomes a normalizing equation that ensures that matrix (3.4) is effectively a non singular transformation matrix; typically it is assumed $x_0^2+x_1^2+x_2^2+x_3^2=1$. Thus, equations (3.7) yields the two polynomials:

$$\begin{aligned} &\sigma_1 : x_0y_0+x_1y_1+x_2y_2+x_3y_3, && \sigma_1 = 0 \\ &\sigma_2 : x_0^2+x_1^2+x_2^2+x_3^2-1, && \sigma_2 = 0 \end{aligned} \tag{3.8}$$

The vanishing sets of polynomials σ_1 and σ_2 relate the eight Study's parameters and represent an intrinsic characteristic that the parameters set must fulfill. It is then necessary to provide a complete algebraic description of the kinematics of the parallel platform taking advantage of the constraints imposed by the architecture of each joints arrangement. As shown in the following, a distinction can be made among the constraint equations that compose the kinematic model of the moving platform based on their dependency on the robot actuation parameters. Some equations, in fact, can be directly gathered by legs kinematic architecture. Other ones, instead, take into consideration the effect of the motion of actuated joints on the whole robot.

In order to obtain a complete description of robot kinematics, Study's quadric equations (3.7) must be put together to the algebraic relations which are characteristic of the specific platform. Firstly, the relations which are peculiar of the legs architecture are introduced. This procedure is also useful to formally demonstrate the mobility of the platform within its workspace.

As visible in Fig. 3.6, the limb is built so as to constrain the axes of the first and the last revolute joint on the plane π_i. Equivalently, one can say that the line ℓ_i, identified by the axis of such joint, has an intersection point with all the lines that lie on the plane π_i, for example the axis of the first revolute here called m_i. The two revolute joints are connected through a planar joint that allows them to move with a relative planar motion. This joint bound the two axes to lie in the same plane for all the configurations. The two lines can be parametrized through their direction plus a passage point, as:

$$\ell_i : \alpha_i \hat{\mathbf{d}}_i + \mathbf{p} \qquad m_i : \beta_i \hat{\mathbf{a}}_i \tag{3.9}$$

with $\alpha_i, \beta_i \in \mathbb{R}$. For the leg constraint to be fulfilled it might be:

$$\ell_i = m_i \qquad \rightarrow \qquad \alpha_i \hat{\mathbf{d}}_i + \mathbf{p} = \beta_i \hat{\mathbf{a}}_i \tag{3.10}$$

or, in matrix form and expressing all the terms with respect the frame $\mathscr{F}_0$

$$\begin{bmatrix} [\mathbf{Q}]_0 [\hat{\mathbf{d}}_i]_1 & -[\hat{\mathbf{a}}_i]_0 & [\mathbf{p}]_0 \end{bmatrix} \begin{bmatrix} \alpha_i & \beta_i & 1 \end{bmatrix}^T = \mathbf{0} \tag{3.11}$$

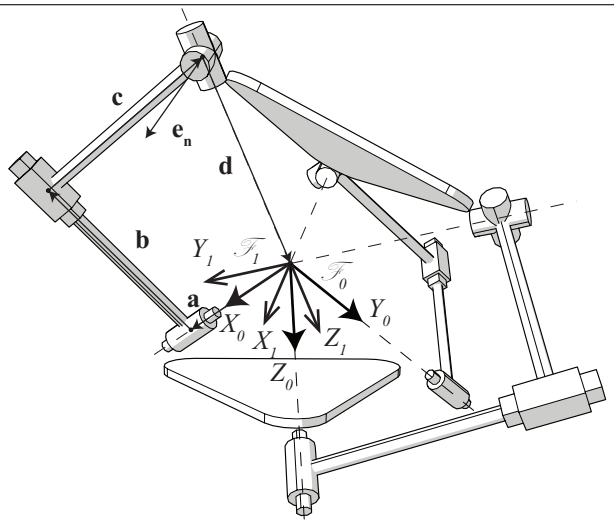
A non trivial solution of this last homogeneous equation can be found only letting the determinant of the respective matrix vanish. Substituting (3.5) and (3.6) for $\mathbf{Q}$ and $\mathbf{p}$ in (3.11), three polynomials in the Study's parameters can be found, one for each leg. After simplification [45], the three constraints can be expressed by the varieties of the following polynomials:

$$\begin{aligned} g_1 &: x_0 y_2 + x_1 y_3 + x_2 y_0 + x_3 y_1 \qquad g_1 = 0 \\ g_2 &: x_0 y_3 + x_1 y_2 + x_2 y_1 + x_3 y_0 \qquad g_2 = 0 \\ g_3 &: x_0 y_1 + x_1 y_0 + x_2 y_3 + x_3 y_2 \qquad g_3 = 0 \end{aligned} \tag{3.12}$$

The vanishing set of polynomials g_1, g_2, and g_3, together with Study's quadric equations (3.8) are still not sufficient to fully describe the kinematics of the PKM class since a set of actuation dependent equations can still be written. Nonetheless, such relations provide all the needed information about the mobility of the MP [44, 46]. With the aim of performing a complete kinematics analysis of the robot it is possible to decompose the polynomial ideal in its primary components. Each primary component, then, is peculiar of a particular branch of robot mobility [29]. In any case, the main target of this section is to demonstrate the ability of the class to perform motions of pure rotations and, to this aim, it is sufficient to ensure that the conditions provided by such mobility are compatible with the constraints provided by the three legs. In particular, it can be easily verified that the general transformation matrix $\mathbf{T}$ describes a pure rotation when the four Study's parameters y_0, y_1, y_2, and y_3 are null. In this case the translation vector $\mathbf{p}$ vanishes, denoting a pure rotational behaviour of the end-effector. Also, it is as much easy to verify that such condition fulfils both the constraints expressed by equations (3.12), and the Study's quadric $\sigma_1 = 0$. The homogeneous transformation matrix can be simplified consequently as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{3.13}$$

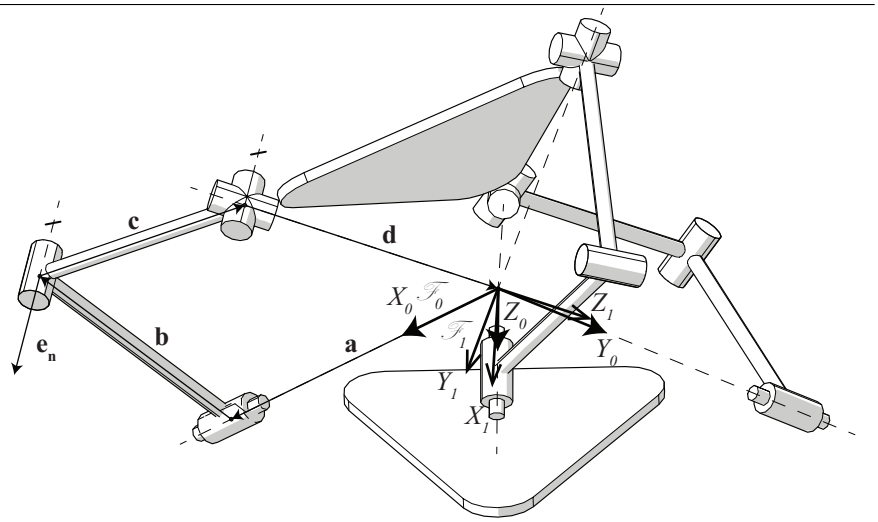
3-CPU



	P	**R**	**P**	**R**	**R**
s	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{a}}\end{bmatrix}_\infty$	$\begin{bmatrix}\hat{\mathbf{a}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{b}}\end{bmatrix}_\infty$	$\begin{bmatrix}\mathbf{e}_4\\ -\mathbf{d}\times\mathbf{e}_4\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ \mathbf{0}\end{bmatrix}_0$
$\mathbf{s}^\perp$	$\begin{bmatrix}\hat{\mathbf{c}}\\ -\mathbf{d}\times\hat{\mathbf{c}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{b}}\\ -\mathbf{d}\times\hat{\mathbf{b}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{0}\\ \mathbf{e}_4\end{bmatrix}_\infty$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$
	a	**d**	**n**	**b**	**c**
$\hat{\bullet}$	$\mathbf{i}_0$	$-\mathbf{k}_1$	$\dfrac{\hat{\mathbf{a}}\times\hat{\mathbf{d}}}{\|\hat{\mathbf{a}}\times\hat{\mathbf{d}}\|}$	$\dfrac{\hat{\mathbf{a}}\times\mathbf{e}_n}{\|\hat{\mathbf{a}}\times\mathbf{e}_n\|}$	$\dfrac{\hat{\mathbf{b}}\times\mathbf{e}_n}{\|\hat{\mathbf{b}}\times\mathbf{e}_n\|}$
$\|\bullet\|$	$c-d\mathbf{i}_0^T\mathbf{k}_1$	d	1	$d\|\mathbf{i}_0\times\mathbf{k}_1\|$	c

Table 3.3: 3-CPU SPM screw joints, reciprocal screw joints and IKP of the first limb.

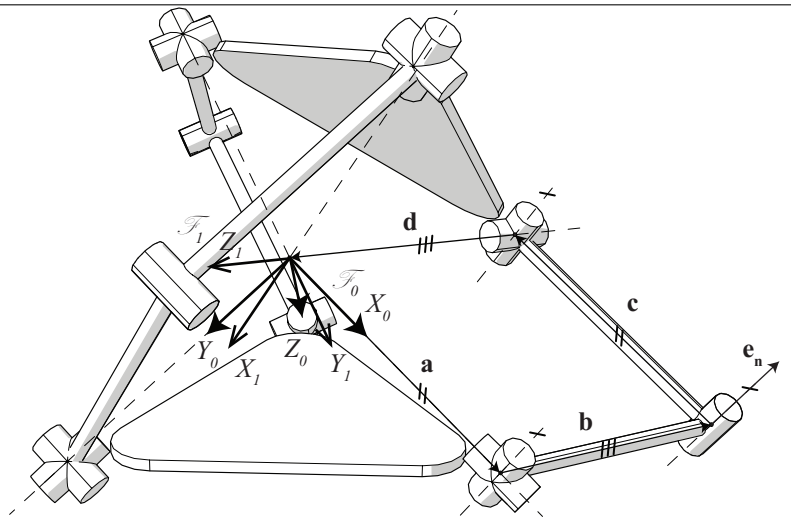
3-CRU



	P	**R**	**R**	**R**	**R**
s	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{a}}\end{bmatrix}_\infty$	$\begin{bmatrix}\hat{\mathbf{a}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ (\mathbf{a}+\mathbf{b})\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ \mathbf{0}\end{bmatrix}_0$
$\mathbf{s}^\perp$	$\begin{bmatrix}\hat{\mathbf{c}}\\ -\mathbf{d}\times\hat{\mathbf{c}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{b}}\\ -\mathbf{d}\times\hat{\mathbf{b}}\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{b}}\\ \mathbf{a}\times\hat{\mathbf{b}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$
	a	**d**	**n**	**b**	**c**
$\hat{\bullet}$	$\mathbf{i}_0$	$-\mathbf{k}_1$	$\dfrac{\hat{\mathbf{a}}\times\hat{\mathbf{d}}}{\|\hat{\mathbf{a}}\times\hat{\mathbf{d}}\|}$	$\dfrac{\hat{\mathbf{a}}\times\mathbf{e}_n}{\|\hat{\mathbf{a}}\times\mathbf{e}_n\|}$	$-\mathbf{d}-\mathbf{a}-\mathbf{b}$
$\|\bullet\|$	a	d	1	b	c
	$a=\sqrt{c^2-(b-d\sqrt{1-\mathbf{i}_0^T\mathbf{i}_1})^2}-d\mathbf{i}_0^T\mathbf{i}_1$				

Table 3.4: 3-CRU SPM screw joints, reciprocal screw joints and IKP of the first limb.

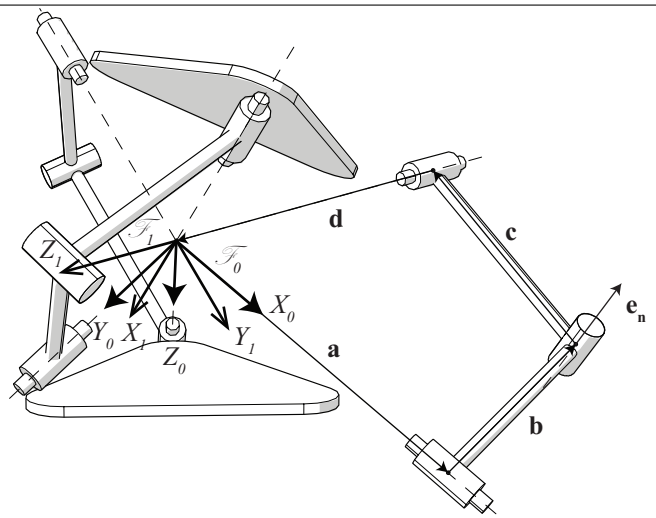
3-URU



	R	R	R	R	R
s	$\begin{bmatrix}\hat{\mathbf{a}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ (\mathbf{a}+\mathbf{b})\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ \mathbf{0}\end{bmatrix}_0$
$\mathbf{s}^\perp$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{a}}\\ -\mathbf{d}\times\hat{\mathbf{a}}\end{bmatrix}_0$	$\begin{bmatrix}\frac{\mathbf{d}+\mathbf{a}}{\|\mathbf{d}+\mathbf{a}\|}\\ -\mathbf{d}\times\frac{\mathbf{d}+\mathbf{a}}{\|\mathbf{d}+\mathbf{a}\|}\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{b}}\\ \mathbf{a}\times\hat{\mathbf{b}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$
	a	**d**	**n**	**b**	**c**
$\hat{\bullet}$	$\mathbf{i}_0$	$-\mathbf{k}_1$	$\frac{\hat{\mathbf{a}}\times\hat{\mathbf{d}}}{\|\hat{\mathbf{a}}\times\hat{\mathbf{d}}\|}$	$-\mathbf{i}_1$	$-\mathbf{i}_0$
$\|\bullet\|$	a	d	1	b	c

Table 3.5: 3-URU SPM screw joints, reciprocal screw joints and IKP of the first limb.

3-CRC



	P	**R**	**R**	**R**	**P**
s	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{a}}\end{bmatrix}_\infty$	$\begin{bmatrix}\hat{\mathbf{a}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ (\mathbf{a}+\mathbf{b})\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ \mathbf{0}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{d}}\end{bmatrix}_\infty$
$\mathbf{s}^\perp$	$\begin{bmatrix}\hat{\mathbf{a}}\\ (\mathbf{a}+\mathbf{b})\times\hat{\mathbf{a}}\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{e}_n\\ -\mathbf{d}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\mathbf{0}\\ \hat{\mathbf{e}_n}\end{bmatrix}_\infty$	$\begin{bmatrix}\mathbf{e}_n\\ \mathbf{a}\times\mathbf{e}_n\end{bmatrix}_0$	$\begin{bmatrix}\hat{\mathbf{d}}\\ (\mathbf{a}+\mathbf{b})\times\hat{\mathbf{d}}\end{bmatrix}_0$
	a	**d**	**n**	**b**	**c**
$\hat{\bullet}$	$\mathbf{i}_0$	$-\mathbf{k}_1$	$\dfrac{\hat{\mathbf{a}}\times\hat{\mathbf{d}}}{\|\hat{\mathbf{a}}\times\hat{\mathbf{d}}\|}$	$\dfrac{\hat{\mathbf{a}}\times\mathbf{e}_n}{\|\hat{\mathbf{a}}\times\mathbf{e}_n\|}$	$\dfrac{\hat{\mathbf{d}}\times\mathbf{e}_n}{\|\hat{\mathbf{d}}\times\mathbf{e}_n\|}$
$\|\bullet\|$	$\dfrac{b\mathbf{i}_0^T\mathbf{i}_1}{\|\mathbf{i}_0\times\mathbf{i}_1\|}$	$\dfrac{c\mathbf{i}_0^T\mathbf{i}_1}{\|\mathbf{i}_0\times\mathbf{i}_1\|}$	1	b	c

Table 3.6: 3-CRC SPM screw joints, reciprocal screw joints and IKP of the first limb.

Chapter 4

Kinetostatic Performance Index

When the pair robot-task is functionally redundant more dof are available than needed; therefore robots can be exploited to accomplish a secondary task. Functional redundancy can be used to increase the accuracy of the manipulator above what is currently available, as reported by Léger and Angeles [2] for serial robots. A strategy to increase the accuracy of the task is to minimize the condition number of the Jacobian matrix which depends on the robot posture. In this chapter the same concept is applied to parallel manipulators. The relationship which relates the velocities of the actuators with the kinematic velocity of the mobile platform are obtained and then used to extract the condition number. This local dexterity index will be an objective function of the optimization problem.

4.1 General Velocity Analysis

To calculate the condition number we need to determine the Jacobian matrix of the PKM. The Jacobian matrix formulation can be easily obtained by means of the relevant screws of the kinematic chains, which have been already found in the foregoing chapter for a SPM class. Following the procedure developed by [47] and then by Tsai [48], the manipulator velocity vector can be written for all robot limbs as a linear combination of the velocities that each joint in the kinematic chain provides. Thus, for the c^{th} leg, the hexa-dimensional vector $\mathbf{t} = [\boldsymbol{\omega}^T \ \mathbf{v}^T]^T$ representing the moving platform twist is:

$$\mathbf{t} = \begin{bmatrix} \mathbf{s}_{1c} & \mathbf{s}_{2c} & \dots & \mathbf{s}_{mc} \end{bmatrix} \dot{\mathbf{q}}_c \tag{4.1}$$

The MP twist does not depend on any point of the body, since it is an instantaneous property of the rigid body (identified by Mozzi's axis, the linear velocity along this axis, and the angular velocity magnitude). However, the twist representation is point-dependent because it depends on the (arbitrary) reference point that is chosen to compute moments (namely, the point with respect to which the location of Mozzi's axis in space is specified). Moreover, $\mathbf{s}_{jc}$ denotes the unit screw relative to the j^{th} joint of c^{th} limb and $\dot{\mathbf{q}}_c$ is the vector collecting both actuated and non actuated joints rates of the c^{th} limb. For terseness, it is considered that each limb only owns one actuated joint,

placed at the first position of the vector appearing at the right side of Eq. (4.1). The manipulator velocity expression can be usefully simplified by introducing the screws reciprocal to all non actuated joints screws, as derived in Section 3.3.2 for the studied SPMs: the corresponding table shows the reciprocal screw to the first joint $\mathbf{s}_{1c}^{\perp}$. A unit screw is said to be reciprocal to another if it happens that $(\mathbf{\Gamma}\mathbf{s}^{\perp})^T\mathbf{s} = (\mathbf{s}^{\perp})^T\mathbf{\Gamma}\mathbf{s} = 0$ having introduced the 6×6 symmetrical array to reverse the first three rows of one of them with the last three of a screw.

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{O} & \mathbf{1} \\ \mathbf{1} & \mathbf{O} \end{bmatrix} \tag{4.2}$$

where $\mathbf{O}$ and $\mathbf{1}$ denote, respectively, the 3×3 zero and identity matrices.
If the unit screw reciprocal to all non actuated joints $\mathbf{s}_{1c}^{\perp}$ of the c^{th} leg is detected, Eq. (4.1) can be simplified by multiplication of both left and right sides by $(\mathbf{\Gamma}\mathbf{s}_{1c}^{\perp})^T$. By doing so, the non actuated joints rates influence on the moving platform is eliminated:

$$(\mathbf{\Gamma}\mathbf{s}_{1c}^{\perp})^T\mathbf{t} = \begin{bmatrix} (\mathbf{\Gamma}\mathbf{s}_{1c}^{\perp})^T\mathbf{s}_{1c} & 0 & \dots & 0 \end{bmatrix}\dot{\mathbf{q}}_{1c} = (\mathbf{\Gamma}\mathbf{s}_{1c}^{\perp})^T\mathbf{s}_{1c}\dot{q}_{1c} \tag{4.3}$$

where $\dot{q}_{1c}$ is the velocity of joint 1 of the c^{th} leg, thus representing the actuation rate of the respective leg. This approach really simplifies the expressions however, by doing so, it is lost the possibility to spot singular behaviors that involve non-actuated variables and reaction forces transmitted within the kinematic chain [49, 50] . By collecting into a single expression the p equations arising from (4.3), having indicated with p the total number of limbs, it is obtained:

$$\begin{bmatrix} (\mathbf{s}_{11}^{\perp})^T \\ (\mathbf{s}_{12}^{\perp})^T \\ \dots \\ (\mathbf{s}_{1p}^{\perp})^T \end{bmatrix}\mathbf{\Gamma}\mathbf{t} = \begin{bmatrix} (\mathbf{\Gamma}\mathbf{s}_{11}^{\perp})^T\mathbf{s}_{11} & 0 & 0 & 0 \\ 0 & (\mathbf{\Gamma}\mathbf{s}_{12}^{\perp})^T\mathbf{s}_{12} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & (\mathbf{\Gamma}\mathbf{s}_{1p}^{\perp})^T\mathbf{s}_{1p} \end{bmatrix}\begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{12} \\ \dots \\ \dot{q}_{1p} \end{bmatrix} \tag{4.4}$$

which is the usual velocity mapping in the form

$$\mathbf{J}_d\mathbf{\Gamma}\mathbf{t} = \mathbf{J}_i\dot{\mathbf{q}}_1 \tag{4.5}$$

where $\mathbf{J}_d$ and $\mathbf{J}_i$ are respectively the forward and inverse geometrical Jacobian matrix and $\dot{\mathbf{q}}_1$ is the time rate vector of actuated joints. The subscript 1 indicates that the PKM has three actuated joints placed at the first position from the BP of each limb. However this procedure is general and can be applied to any parallel robot.

4.1.1 SPM Jacobian Matrices

Due to moving platform reduced mobility of the SPM, the twist $\mathbf{t}$ of the moving platform with respect to the center of spherical motion is

$$\mathbf{t} = [\boldsymbol{\omega}^T \ \mathbf{0}^T]^T \tag{4.6}$$

Due to this fact, the influence of $\mathbf{J}_d$, i.e. the forward geometrical Jacobian matrix, on Eq. (4.5) is limited to a 3×3 sub-matrix that is constituted, in case of spherical motions, by its last three columns. Moreover, three actuated joint fixed to the base are chosen, since non-intrinsically redundant SPMs have three-dof, and their inverse kinematics Jacobian matrix $\mathbf{J}_i$ becomes a 3×3 matrix.
The Jacobian matrices of the considered SPM class can be derived using the screws in the foregoing tables. The relationships of the inverse kinematics are also provided, allowing the Jacobian matrices to be directly expressed in terms of the rotation matrix $\mathbf{Q}$ between the fixed and mobile reference frame.
Two examples of spherical parallel robot are considered: they are the 3-CPU and the 3-UPU SPMs.

3-CPU

By replacing the screws of the 3-CPU SPM of Table 3.3 inside Eq. (4.10), the diagonal entries of the inverse kinematics Jacobian matrix $\mathbf{J}_i$ become

$$\mathbf{s}_{1c}^{\perp} \boldsymbol{\Gamma} \mathbf{s}_{1c} = \begin{bmatrix} \hat{\mathbf{c}}_c \\ -\mathbf{d}_c \times \hat{\mathbf{c}}_c \end{bmatrix}^T \begin{bmatrix} \mathbf{O} & \mathbf{1} \\ \mathbf{1} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{a}}_c \end{bmatrix} = \hat{\mathbf{c}}_c^T \hat{\mathbf{a}}_c \tag{4.7}$$

The mobility analysis of the 3-CPU reveals that versors $\mathbf{c}_c$ and $\mathbf{a}_c$ on the same c leg are anti-parallel for every posture of the manipulator. This implies that

$$\hat{\mathbf{c}}_c^T \hat{\mathbf{a}}_c = -1 \tag{4.8}$$

and that the inverse kinematics Jacobian matrix $\mathbf{J}_i$ is

$$\mathbf{J}_i = -\mathbf{1} \tag{4.9}$$

where $\mathbf{1}_{(3\times3)}$ is the 3 by 3 identity matrix.
On the left side of Eq. (4.10)

$$\begin{bmatrix} (\mathbf{s}_{11}^{\perp})^T \\ (\mathbf{s}_{12}^{\perp})^T \\ (\mathbf{s}_{13}^{\perp})^T \end{bmatrix} \boldsymbol{\Gamma} \mathbf{t} = \begin{bmatrix} \hat{\mathbf{c}}_1^T & (-\mathbf{d}_1 \times \hat{\mathbf{c}}_1)^T \\ \hat{\mathbf{c}}_2^T & (-\mathbf{d}_2 \times \hat{\mathbf{c}}_2)^T \\ \hat{\mathbf{c}}_3^T & (-\mathbf{d}_3 \times \hat{\mathbf{c}}_3)^T \end{bmatrix} \begin{bmatrix} \mathbf{O} & \mathbf{1} \\ \mathbf{1} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} (-\mathbf{d}_1 \times \hat{\mathbf{c}}_1)^T \\ (-\mathbf{d}_2 \times \hat{\mathbf{c}}_2)^T \\ (-\mathbf{d}_3 \times \hat{\mathbf{c}}_3)^T \end{bmatrix} \boldsymbol{\omega} \tag{4.10}$$

The velocity analysis gives the following equation relating the velocities of the actuators to the kinematic velocity of the mobile platform .

$$\begin{bmatrix} (\mathbf{d}_1 \times \hat{\mathbf{c}}_1)^T \\ (\mathbf{d}_2 \times \hat{\mathbf{c}}_2)^T \\ (\mathbf{d}_3 \times \hat{\mathbf{c}}_3)^T \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \dot{q}_{1,1} \\ \dot{q}_{1,2} \\ \dot{q}_{1,3} \end{bmatrix} \rightarrow \mathbf{J}_d \, \boldsymbol{\omega} = \dot{\mathbf{q}}_1 \tag{4.11}$$

The 3-CPU manipulator shows a particular property of the velocity relationship consisting in the independence of the inverse kinematics Jacobian matrix from the posture of the manipulator. A more general case is provided by the 3-UPU parallel manipulator and, for this reason, it is studied as next example.
The direct Jacobian matrix can be written in terms of the rotation matrix between the two frames using the inverse kinematic relationships. In this way $\mathbf{J}_d$ becomes

$$\mathbf{J}_d = d \begin{bmatrix} ([\mathbf{i}_0]_0 \times \mathbf{Q}[\mathbf{k}_1]_1)^T \\ ([\mathbf{j}_0]_0 \times \mathbf{Q}[\mathbf{i}_1]_1)^T \\ ([\mathbf{k}_0]_0 \times \mathbf{Q}[\mathbf{j}_1]_1)^T \end{bmatrix} \tag{4.12}$$

where $[\mathbf{i}_0]_0 = [\mathbf{i}_1]_1 = [1\,0\,0]^T$, $[\mathbf{j}_0]_0 = [\mathbf{k}_1]_1 = [0\,1\,0]^T$ and $[\mathbf{k}_0]_0 = [\mathbf{k}_1]_1 = [0\,0\,1]^T$. This equation points out that computation of moving platform Jacobian matrix requires an explicit knowledge of the pose of the manipulator which, for PKMs with no intrinsic redundancy, corresponds to a posture of the manipulator.

3-UPU

A more general case is given by the 3-UPU architecture because both the direct and the inverse kinematic Jacobian matrices are not constant. In fact,

$$\mathbf{s}_{1c}^{\perp} \boldsymbol{\Gamma} \mathbf{s}_{1c} = \begin{bmatrix} \mathbf{e}_{nc} \\ -\mathbf{d}_c \times \mathbf{e}_{nc} \end{bmatrix}^T \begin{bmatrix} \mathbf{O} & \mathbf{1} \\ \mathbf{1} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}_c} \\ \mathbf{0} \end{bmatrix} = (-\mathbf{d}_c \times \mathbf{e}_{nc})^T \, \hat{\mathbf{a}_c} \tag{4.13}$$

and the inverse kinematics Jacobian matrix is equal to

$$\mathbf{J}_i = \begin{pmatrix} \text{diag} & (-\mathbf{d}_1 \times \mathbf{e}_{n1})^T \, \hat{\mathbf{a}_1}, & (-\mathbf{d}_2 \times \mathbf{e}_{n2})^T \, \hat{\mathbf{a}_2}, & (-\mathbf{d}_3 \times \mathbf{e}_{n3})^T \, \hat{\mathbf{a}_3} \end{pmatrix} \tag{4.14}$$

while the direct Jacobian matrix becomes

$$\mathbf{J}_d = \begin{bmatrix} (-\mathbf{d}_1 \times \mathbf{e}_{n1})^T \\ (-\mathbf{d}_2 \times \mathbf{e}_{n2})^T \\ (-\mathbf{d}_3 \times \mathbf{e}_{n3})^T \end{bmatrix} \tag{4.15}$$

The condition number is commonly calculated multiplying both sides of the velocity equation by the inverse of the matrix $\mathbf{J}_i$.

$$\mathbf{J}_d\boldsymbol{\omega} = \mathbf{J}_i\,\dot{\mathbf{q}}_1 \quad \Rightarrow \quad \mathbf{J}_i^{-1}\mathbf{J}_d\boldsymbol{\omega} = \dot{\mathbf{q}}_1 \tag{4.16}$$

$\mathbf{J}_i$ can be easily inverted because it is a diagonal matrix but this step can be done only if $\mathbf{J}_i$ is not singular. It leads to this result

$$\mathbf{J}_i^{-1}\mathbf{J}_d = \mathbf{J} = \begin{bmatrix} \dfrac{(-\mathbf{d}_1 \times \mathbf{e}_{n1})^T}{(-\mathbf{d}_1 \times \mathbf{e}_{n1})^T \hat{\mathbf{a}}_1} \\ \dfrac{(-\mathbf{d}_2 \times \mathbf{e}_{n2})^T}{(-\mathbf{d}_2 \times \mathbf{e}_{n2})^T \hat{\mathbf{a}}_2} \\ \dfrac{(-\mathbf{d}_3 \times \mathbf{e}_{n3})^T}{(-\mathbf{d}_3 \times \mathbf{e}_{n3})^T \hat{\mathbf{a}}_3} \end{bmatrix} \tag{4.17}$$

Once again we can write the Jacobian matrix as a function of the rotation matrix $\mathbf{Q}$ using Table 3.2, however for terseness this is not reported.

4.2 Condition Number

The condition number is a local dexterity index which quantifies the error amplification between the joints and the EE relative errors [51]: by lowering the condition number value, the propagation of joint errors to the EE pose is reduced, thus increasing accuracy. This index will be used as the objective function of the optimization problem to further improve the accuracy of the manipulator above what is currently available.
The discussion to obtain an explicit expression of the condition number in terms of the EE pose parameters is presented considering a SPM; however an extension to a general case is easily deductible. The velocity analysis of an SPM gives the following general relationship by doing the inversion of the inverse kinematic Jacobian matrix

$$\underbrace{\begin{bmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \\ \mathbf{n}_3^T \end{bmatrix}}_{\mathbf{J}} \underbrace{\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}}_{\boldsymbol{\omega}} = \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}}_{\dot{\mathbf{q}}} \tag{4.18}$$

where $\mathbf{n}_c^T$ indicates the row vectors of the Jacobian matrix $\mathbf{J}$ for $c = 1,2,3$.
The general formula to calculate the condition number of the Jacobian matrix $\mathbf{J}$ is

$$\kappa(\mathbf{J}) = \|\mathbf{J}\|\|\mathbf{J}^{-1}\| \tag{4.19}$$

where it is required the calculation of the matrix norm $\|\bullet\|$. The choice falls on the Frobenius norm because it is the only matrix norm that gives an analytic function of the condition number everywhere, i.e. for any real value of its scalar arguments. Here, we recall that a function is analytic at a point if the function admits a series expansion at that point. This requires that the function, first and foremost, have all its derivatives with respect to the arguments continuous. Compared to the classical 2-norm condition number, which is the rate between the largest and the smallest singular values of the matrix, the Frobenius based condition number is the product of the rms of the singular values of the matrix by the rms value of their reciprocal, namely

$$\kappa_F(\mathbf{J}) = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2\right)\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right)} \tag{4.20}$$

In particular in the 2-norm condition number, discontinuities are due to the switch in its formula of the intermediate singular values with one of the extreme values: this point is well discussed by Khan and Angeles [52]. This does not happen for the Frobenius based norm since it is the only matrix norm that is infinitely differentiable with respect to its arguments and it allows the minimum search to be straightforwardly achieved by virtue of its smoothness properties.
Based on the Frobenius norm of the 3×3 Jacobian $\mathbf{J}$, the square of the condition number $\kappa_F(\mathbf{J})$ becomes

$$\kappa_F^2(\mathbf{J}) = \|\mathbf{J}\|_F^2\|\mathbf{J}^{-1}\|_F^2 = \frac{1}{n^2}\mathrm{tr}(\mathbf{J}\mathbf{J}^T)\mathrm{tr}(\mathbf{J}^{-1}\mathbf{J}^{-T}) \tag{4.21}$$

where $\|\mathbf{J}\|$ is obtained as the positive square root of $\|\mathbf{J}\|_F^2 = \mathrm{tr}(\mathbf{J}\mathbf{J}^T)/n$. The factor $1/n$ accounts for a *weighted norm*, which equals the rms of the Jacobian matrix singular values. Each trace of this expression is then expanded, to express it as function of the row vectors of the Jacobian matrix, namely,

$$\mathrm{tr}(\mathbf{J}\mathbf{J}^T) = \mathrm{tr}\left(\begin{bmatrix}\mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3\end{bmatrix}^T\begin{bmatrix}\mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3\end{bmatrix}\right) = \sum_{i=1}^{3}\|\mathbf{n}_i\|^2 \tag{4.22}$$

Regarding the other term $\mathrm{tr}(\mathbf{J}^{-1}\mathbf{J}^{-T})$, the inverse of $\mathbf{J}^T$ can be expressed in terms of its columns explicitly, without introducing any components, if the concept of *reciprocal bases* is recalled [53].

$$\mathbf{J}^{-T} = (\mathbf{J}^T)^{-1} = \frac{1}{\Delta}\begin{bmatrix}\mathbf{n}_2\times\mathbf{n}_3 & \mathbf{n}_3\times\mathbf{n}_1 & \mathbf{n}_1\times\mathbf{n}_2\end{bmatrix}^T, \quad \Delta \equiv \mathbf{n}_1\times\mathbf{n}_2\cdot\mathbf{n}_3 \tag{4.23}$$

By transposing this matrix $\mathbf{J}^{-1}$ is obtained and the trace can be written as before. Hence, the square of the condition number of a 3×3 Jacobian matrix is

$$\kappa_F^2(\mathbf{J}) = \frac{\left(\|\mathbf{n}_1\|^2 + \|\mathbf{n}_2\|^2 + \|\mathbf{n}_3\|^2\right)\left(\|\mathbf{n}_2 \times \mathbf{n}_3\|^2 + \|\mathbf{n}_3 \times \mathbf{n}_1\|^2 + \|\mathbf{n}_1 \times \mathbf{n}_2\|^2\right)}{9(\mathbf{n}_1 \times \mathbf{n}_2 \cdot \mathbf{n}_3)^2} \quad (4.24)$$

The condition number can attain values from unity to infinity. It attains its minimum value of unity for matrices with identical singular values and in this case the matrix is called *isotropic*. On the other hand, a singular matrix condition number is infinite due to the vanishing of its smallest singular value; therefore, the matrix is said singular.

4.2.1 Global Condition Number

The condition number is a local indication for the dexterity of a manipulator. To evaluate the dexterity of a robot over a given workspace W, Gosselin [54] has introduced the *global conditioning index* (GCI) as

$$\mathrm{GCI} = \frac{\displaystyle\int_W \frac{1}{\kappa} dW}{\displaystyle\int_W dW} \quad (4.25)$$

which corresponds to the average value of $1/\kappa$. This concept is mainly used for the optimal design of robots for which the extremal and average value of any performance are important design factors. In this thesis, the GCI will be used to evaluate the improvement of performance due to the exploitation of functional redundancy throughout the workspace. The reader should note that the GCI must rely on a numerical evaluation consisting of the discretization of the workspace using regular grid, the computation of $1/\kappa_i$ at every node, and the approximation of the GCI: the sum of the $1/\kappa_i$ is divided by the number of nodes and by the workspace size as handled by Merlet [51].

4.2.2 Example: 3-CPU Condition Number

The condition number can be expressed as a function of the rotation matrix $\mathbf{Q}$ describing the orientation of the MP with respect to the BP. From Eq. (4.12) the row vector associated to the first leg of the Jacobian matrix is

$$\mathbf{n}_1 = [\mathbf{i}_0]_0 \times \mathbf{Q}[\mathbf{i}_1]_1 \quad (4.26)$$

and it is used to calculate the square of the condition number using Eq. (4.24).
A condition number can be obtained in terms of the rotation matrix $\mathbf{Q}$ which needs a proper parametrization to describe the orientation of the mobile platform: among the various representations that describe the rotation, we choose the one based on the

rotation invariants, and in particular, the Euler-Rodrigues Parameters (ERPs), i.e., the four scalars $\mathbf{r}$ and r_0 [55]:

$$\mathbf{Q} = (r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R}, \quad \mathbf{R} = \mathrm{CPM}(\mathbf{r}) \tag{4.27}$$

where $\mathrm{CPM}(\mathbf{r})$ denotes the cross-product matrix[1] of $\mathbf{r} \equiv \mathbf{e}\sin(\phi/2)$. Let $\mathbf{e}$ be the unit vector of the rotation axis and θ its axis of rotation, r_0 is defined as $r_0 = \cos(\theta/2)$. This representation is more robust than the others because it does not entail any singularity; therefore, smoother trajectories can be obtained. Moreover the reader should note that these four parameters correspond to the first four Study's parameters. The array of unknowns becomes $\mathbf{x} = [\mathbf{r}^T \; r_0]^T$, which are not independent, for they obey the constraint $\|\mathbf{r}\|^2 + r_0^2 = 1$.
A symbolic computation with computer algebra leads to an expression for $\kappa_F^2(\mathbf{J})$ on the form

$$\kappa_F^2(\mathbf{J}) = \frac{1}{9}\frac{N_1 N_2}{D} \tag{4.28}$$

where

$$N_1 = \|\mathbf{n}_1\|^2 + \|\mathbf{n}_2\|^2 + \|\mathbf{n}_3\|^2 = d^2 \\ (3r_0^4 + 2r_0^2r_1^2 + 2r_0^2r_2^2 + 2r_0^2r_3^2 + 24r_0r_1r_2r_3 + 3r_1^4 + 2r_1^2r_2^2 + 2r_1^2r_3^2 + 3r_2^4 + 2r_2^2r_3^2 + 3r_3^4) \tag{4.29}$$

$$\begin{aligned} N_2 = {} & \|\mathbf{n}_2 \times \mathbf{n}_3\|^2 + \|\mathbf{n}_3 \times \mathbf{n}_1\|^2 + \|\mathbf{n}_1 \times \mathbf{n}_2\|^2 = \\ & d^4(4(r_0r_1 + r_2r_3)^2(r_0^2 - r_1^2 - r_2^2 + r_3^2)^2 + 4(r_0r_2 + r_1r_3)^2(r_0^2 + r_1^2 - r_2^2 - r_3^2)^2 + \\ & 4(r_0r_3 + r_1r_2)^2(r_0^2 - r_1^2 + r_2^2 - r_3^2)^2 + (r_0^2 - r_1^2 - r_2^2 + r_3^2)^2(r_0^2 - r_1^2 + r_2^2 - r_3^2)^2 + \\ & (r_0^2 - r_1^2 - r_2^2 + r_3^2)^2(r_0^2 + r_1^2 - r_2^2 - r_3^2)^2 + (r_0^2 - r_1^2 + r_2^2 - r_3^2)^2(r_0^2 + r_1^2 - r_2^2 - r_3^2)^2 \\ & + 16(r_0r_1 + r_2r_3)^2(r_0r_2 + r_1r_3)^2 + 16(r_0r_1 + r_2r_3)^2(r_0r_3 + r_1r_2)^2 \\ & + 16(r_0r_2 + r_1r_3)^2(r_0r_3 + r_1r_2)^2) \end{aligned} \tag{4.30}$$

$$D = (\mathbf{n}_1 \times \mathbf{n}_2 \cdot \mathbf{n}_3)^2 = \\ (d^3(r_1 + r_2 + r_3 - r_0)(r_1 - r_2 - r_3 - r_0)(-r_1 + r_2 - r_3 - r_0)(-r_1 - r_2 + r_3 - r_0))^2 \tag{4.31}$$

[1] That is, $\mathrm{CPM}(\mathbf{r}) = \partial(\mathbf{r} \times \mathbf{v})/\partial\mathbf{v}$, $\forall \mathbf{v}$, $\mathbf{x} \in \mathbb{R}^3$

Chapter 5

Dynamic Performance Indices

In order to increase the efficiency of industrial processes, i.e. to decrease the cycle time of manufacturing operations, the dynamic behavior of the machine should be taken into consideration. In this chapter, the optimization of the functional redundancy will also consider the dynamic performance of the manipulator for fast motion and high-precision manipulator tasks. To such aim, three indices, related to the inertia matrix, that quantify the dynamic performance of three-dof parallel manipulators are considered [56]. Two of them represent the dynamic manipulability, i.e. the manipulating ability of the robot in positioning and orienting the mobile platform, while the third one identifies the robot's swiftness, i.e. its skill to produce EE accelerations due to the same variation of the actuation forces. All of them can be expressed as functions of the parameters that describe the pose of the MP, the inertial properties and the geometrical parameters of the links. They are used as objective functions for the optimization along each pointing direction, once the properties of the parallel manipulator are given. When the accelerations are not negligible the dynamic behavior of the manipulator depends on its generalized inertia matrix and this matrix can be derived from the expression of the kinetic energy of the manipulator reduced to the mobile platform. The *screw theory* is used to write the equations in a compact form and derive the eigenvalues of the inertia matrix. These eigenvalues are then used to quantify the dynamic manipulability through the concept of dynamic manipulability ellipsoids, whose semi-axes own a length equal to the eigenvalues of the matrix. An ellipsoid with equal eigenvalues is obviously a sphere and it indicates a dynamically isotropic configuration of the robot. The swiftness of the mobile platform is quantified by the arithmetic mean of the eigenvalues: the higher their average value is, the smaller the acceleration that the MP can reach from the considered configuration becomes.

5.1 Manipulator Inertia Matrix

The dynamic performance indices are computed from the inertia matrix reduced to the end-effector. This matrix can be derived from the total kinetic energy of the robot, that is the summation of the kinetic energies of all its r bodies. Thus, for a parallel manipulator, it is the sum of the end-effector kinetic energy T_{ee} plus the kinetic energies of

all the links. T_{lc} indicates the kinetic energy of the link l on the leg c: the total number of links in a limb is h, while the total number of legs is p. It is:

$$T = \sum_{b=1}^{r} T_b = \frac{1}{2}\mathbf{t}_{ee}^T \mathbf{M}_{ee}\mathbf{t}_{ee} + \sum_{c=1}^{p}\sum_{l=1}^{h} \frac{1}{2}\mathbf{t}_{lc}^T \mathbf{M}_{lc}\mathbf{t}_{lc} \tag{5.1}$$

In previous equation, we denoted by $\mathbf{M}_{lc}$ the inertia dyad of the body lc [55]:

$$\mathbf{M}_{lc} = \begin{bmatrix} \mathbf{I}_{lc} & \mathbf{O} \\ \mathbf{O} & m_{lc}\mathbf{1} \end{bmatrix} \tag{5.2}$$

where $\mathbf{1}$ and $\mathbf{O}$ are the 3×3 identity and zero matrices respectively, while $\mathbf{I}_{lc}$ is the inertia matrix of the body which is defined with respect to the center of mass C_{lc} of that body. Moreover, its mass is denoted by m_{lc}, whereas $\boldsymbol{\rho}_{lc}$ and $\dot{\boldsymbol{\rho}}_{lc}$ indicate the position and the velocity vectors of C_{lc} in an inertial frame. Furthermore, $\mathbf{t}_{lc}$ denotes the twist of the same body, defined in terms of the angular velocity vector $\boldsymbol{\omega}_{lc}$ and the velocity of C_i. The inertia tensor of the link lc referred to its center of gravity is indicated as $[\mathbf{I}_{lc}]^{lc}$ and it is expressed in the reference system fixed to the link lc, thus with constant entries. In order to express $\mathbf{M}_{lc}$ in a reference frame system fixed to the ground, the following relation is used:

$$[\mathbf{I}_{lc}]^0 = \mathbf{Q}_{lc}[\mathbf{I}_{lc}]^{lc}\mathbf{Q}_{lc}^T \tag{5.3}$$

where $\mathbf{Q}_{lc}$ is the rotation matrix between reference frames $\mathscr{F}_{lc}$ and $\mathscr{F}_0$. Provided that singular configurations are excluded, the inverse velocity analysis of the manipulator allows the following relationships to be determined in an explicit form:

$$\mathbf{t}_{lc} = \mathbf{Z}_{lc}\boldsymbol{\omega} \qquad \mathbf{t}_{ee} = \mathbf{Z}_{ee}\boldsymbol{\omega} \tag{5.4}$$

Since a spherical parallel manipulator is studied, $\boldsymbol{\omega}$ is the angular velocity of the mobile platform, while $\mathbf{Z}_{lc}$ and $\mathbf{Z}_{ee}$ are 6×3 matrices depending on parameters $\mathbf{x}$ describing the pose of the MP and on the manipulator link geometry. In general, these expressions are linear in the manipulator generalized coordinates rates. Taking into account Eq. 5.4, Eq. 5.1 becomes:

$$T = \sum_{1}^{r} T_i = \frac{1}{2}\boldsymbol{\omega}^T \mathbf{Z}_{ee}^T \mathbf{M}_{ee}\mathbf{Z}_{ee}\boldsymbol{\omega} + \sum_{c=1}^{p}\sum_{l=1}^{h} \frac{1}{2}\boldsymbol{\omega}^T \mathbf{Z}_{lc}^T \mathbf{M}_{lc}\mathbf{Z}_{lc}\boldsymbol{\omega} \tag{5.5}$$

This expression could be rewritten as:

$$T = \frac{1}{2}\boldsymbol{\omega}^T \mathbf{N}\boldsymbol{\omega}, \quad \mathbf{N}(\mathbf{x}) = \mathbf{Z}_{ee}^T \mathbf{M}_{ee}\mathbf{Z}_{ee} + \sum_{c=1}^{p}\sum_{l}^{h} \mathbf{Z}_{lc}^T \mathbf{M}_{lc}\mathbf{Z}_{lc} \tag{5.6}$$

The matrix **N** is the inertia matrix referred to the end-effector of the manipulator. It is a symmetric and positive-definite matrix depending both on **x**, the parameters used to describe the pose of the manipulator, and on the inertial and geometric parameters of the links. The problem to determine the **Z** matrices of this equation is addressed in the next section.

5.2 Determination of the Z matrices

A procedure for the determination of the **Z** matrices of Eq. (5.6) is applied to the 3-CPU SPM; however, a generalization to other 3-dof parallel kinematic machines can be straightforwardly achieved. All the data needed to the reader to implement the simplified model of the robot are given in Tables 5.1 and 5.2.
Let $\mathscr{F}_0$ be a reference system fixed to the ground whose unit axes $\mathbf{i}_0$, $\mathbf{j}_0$ and $\mathbf{k}_0$ are coaxial with the cylindrical joints indicated with $\mathbf{a}_c$, and let $\mathscr{F}_1$ be a reference system fixed to the mobile platform, whose axes are directed along the vectors $-\mathbf{d}_c$. Moreover, $\mathbf{e}_c$ represents the unit vector perpendicular to the c^{th} leg plane. Using the expressions indicated in Table 3.3, it is possible to calculate the vectors $\mathbf{a}_c$, $\mathbf{b}_c$, $\mathbf{c}_c$, $\mathbf{d}_c$, $\mathbf{e}_c$ of the leg c as a function of the rotation matrix **Q** in terms of the pose of the MP. After that, the screw joints $\mathbf{s}_{jc}$ can be computed, where j indicates again the joint in the c^{th} limb. Such screws are expressed with respect to the center of the spherical motion and, after substitutions, they will be written in terms of the end-effector pose. The Jacobian matrices can be obtained through such screw vectors as:

$$\mathbf{K}_{0c} = \begin{bmatrix} \mathbf{s}_{1c} & \mathbf{s}_{2c} & \mathbf{s}_{3c} & \mathbf{s}_{4c} & \mathbf{s}_{5c} \end{bmatrix} \quad c = 1, ..., p \tag{5.7}$$

Since screws are defined with respect to the center of the spherical motion, it is the same also for the Jacobian matrices $\mathbf{K}_{0jc}$ and the subscript 0 points up this fact. However, these Jacobian matrices have to be expressed with respect to the center of gravity of the mobile platform, better than the center of spherical motion. To this end, the *twist-transfer formula*, which relates the twists of the same rigid body at two different points, is defined. Being A and P two arbitrary points of a rigid body, the tansfer formula is:

$$\mathbf{t}_P = \mathbf{U}(\mathbf{a}, \mathbf{p})\mathbf{t}_A, \quad \mathbf{U}(\mathbf{a}, \mathbf{p}) = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{A} - \mathbf{P} & \mathbf{1} \end{bmatrix}, \quad \mathbf{t}_A = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_A \end{bmatrix}, \quad \mathbf{t}_P = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_P \end{bmatrix} \tag{5.8}$$

where **a** and **p** are the position vectors of the two points with respect to the origin of the reference system and $\mathbf{A} = \mathrm{CPM}(\mathbf{a})$ and $\mathbf{P} = \mathrm{CPM}(\mathbf{p})$ [1]. Therefore, the Jacobian matrices with respect to the center of gravity of the EE are:

$$\mathbf{K}_c = \mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_5)\mathbf{K}_{0c} \tag{5.9}$$

[1] CPM denotes the cross-product matrix, that is, $\mathrm{CPM}(\mathbf{x}) = \partial(\mathbf{x} \times \mathbf{v})/\partial\mathbf{v}$, $\forall \mathbf{v}$, $\mathbf{v} \in \mathbb{R}^3$

with $\mathbf{0} = [0\,0\,0]^T$ and $\boldsymbol{\rho}_5$ the position vector of the end-effector center of gravity. When a SPM is taken into account, the mobile platform can only perform angular velocities with respect to the center of spherical motion, so its twist with respect to this point becomes $\mathbf{t}_0 = [\boldsymbol{\omega}^T\ \mathbf{0}^T]^T$. The twist tranfer formula is applied to it:

$$\mathbf{t} = \mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_5)\mathbf{t}_0 \tag{5.10}$$

The three contributions of each kinematic chain to the mobile platform are

$$\mathbf{t} = \mathbf{K}_c\dot{\mathbf{q}}_c \tag{5.11}$$

where $\dot{\mathbf{q}}_c$ is the vector of both active and passive joint rates of the c^{th} limb.
According to Table 3.3, screws $\mathbf{s}_{jc}^{\perp}$ are calculated. The screw $\mathbf{s}_{jc}^{\perp}$ is reciprocal to any joint of the c leg except for the considered j^{th} joint. Such screws may be usefully employed to eliminate the contribution of the joints screws reciprocal to it on the previous expression. The goal is to find the explicit relations of the joint rates as function of the pose parameters $\mathbf{x}$. Multiplying Eq.s 5.11 by each reciprocal screw written with respect to the EE center of gravity, any term associated to a joint screw reciprocal to that is eliminated. This is done for each joint on the kinematic chain and for any leg, i.e. we multiply Eq.s 5.11 by $(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{jc}^{\perp})^T$ (where $\mathbf{U}$ replaces $\mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_5)$ for the sake of conciseness):

$$(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{jc}^{\perp})^T\mathbf{t} = (\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{jc}^{\perp})^T\mathbf{K}_c\dot{\mathbf{q}}_c \tag{5.12}$$

The $\mathbf{s}_{jc}^{\perp}$ screw leaves only the contribution of $\mathbf{s}_{jc}$ in Eq. 5.12:

$$(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{jc}^{\perp})^T\mathbf{t} = \begin{bmatrix}((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{jc}^{\perp})^T\mathbf{U}\mathbf{s}_{jc} & 0 & \dots & 0\end{bmatrix}\dot{\mathbf{q}}_c \tag{5.13}$$

By grouping the expressions for all the legs ($c = 1,2,3$) and considering each joint j, we will obtain m expressions, where m is the total number of joints in a limb:

$$\begin{bmatrix}(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j1}^{\perp})^T\\(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j2}^{\perp})^T\\(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j3}^{\perp})^T\end{bmatrix}\boldsymbol{\Gamma}\mathbf{t} = \begin{bmatrix}((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j1}^{\perp})^T\mathbf{U}\mathbf{s}_{j1} & 0 & 0\\0 & ((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j2}^{\perp})^T\mathbf{U}\mathbf{s}_{j2} & 0\\0 & 0 & ((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j3}^{\perp})^T\mathbf{U}\mathbf{s}_{j3}\end{bmatrix}\begin{bmatrix}\dot{q}_{j1}\\\dot{q}_{j2}\\\dot{q}_{j3}\end{bmatrix} \tag{5.14}$$

In order to obtain the joint rates relationships, the diagonal matrix can be easily inverted and multiplied by both sides of the equation:

$$\dot{\mathbf{q}}_j = \begin{bmatrix}(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j1}^{\perp})^T/((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j1}^{\perp})^T\mathbf{U}\mathbf{s}_{j1})\\(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j2}^{\perp})^T/((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j2}^{\perp})^T\mathbf{U}\mathbf{s}_{j2})\\(\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j3}^{\perp})^T/((\boldsymbol{\Gamma}\mathbf{U}\mathbf{s}_{j3}^{\perp})^T\mathbf{U}\mathbf{s}_{j3})\end{bmatrix}\boldsymbol{\Gamma}\mathbf{t} \quad j = 1,...,m \tag{5.15}$$

The joint rates vectors $\dot{\mathbf{q}}_j$ are expressed in terms of the pose parameters $\mathbf{x}$ and of the angular velocity of the end effector $\boldsymbol{\omega}$.
The twist of each link l of the limb c is obtained from $c = 1$ to p as:

$$\begin{aligned}
&\mathbf{t}_{1c} = \mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_{1c})\mathbf{s}_{1c}\dot{q}_{1c} \\
&\mathbf{t}_{2c} = \mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_{2c}) \begin{bmatrix} \mathbf{s}_{1c} & \mathbf{s}_{2c} \end{bmatrix} \begin{bmatrix} \dot{q}_{1c} & \dot{q}_{2c} \end{bmatrix}^T \\
&\ldots \\
&\mathbf{t}_{hc} = \mathbf{U}(\mathbf{0}, \boldsymbol{\rho}_{hc}) \begin{bmatrix} \mathbf{s}_{1c} & \mathbf{s}_{2c} & \mathbf{s}_{3c} & \mathbf{s}_{4c} & \mathbf{s}_{5c} \end{bmatrix} \begin{bmatrix} \dot{q}_{1c} & \dot{q}_{2c} & \dot{q}_{3c} & \dot{q}_{4c} & \dot{q}_{5c} \end{bmatrix}^T
\end{aligned} \tag{5.16}$$

The substitution of the Eq.s 5.15 into Eq.s 5.16 allows to write each twist $\mathbf{t}_{lc}$ in the form of a linear system of equations by collecting the components of $\boldsymbol{\omega}$ in a vector: in this way the form of Eq. 5.4 is reached. This step can be readily done by means of computer algebra.
The derived $\mathbf{Z}$ matrices can then be used for the determination of the inertia matrix $\mathbf{N}$ in a symbolic form and in terms of the pose of the mobile platform. The previous analysis has neglected the contribution of the Coriolis and centrifugal effects that are non-linear in pose parameters rates, the terms related to gravity and dissipation forces. These effects are relevant for the dynamic behavior of the manipulator and, in general, they must be considered. However, Callegari [41] showed that Coriolis and centrifugal effects and the terms related to gravity and dissipation forces do not affect the dynamic indices based on the inertia matrix eigenvalues, because they merely cause a translation of the manipulability ellipsoid. Hence, the proposed dynamic indices that only depend from robot pose can be used without neglecting these effects; by virtue of this fact they greatly simplify the optimization problem, therefore, they should be preferred when a reduced number of variables is desired.

5.3 Local Dynamic Indices

Some dynamic indices have been proposed in the literature for parallel manipulators. In particular, referring to an inertia matrix of the EE obtained in the Cartesian space, Yoshikawa [57] presented the dynamic manipulability ellipsoid and an index for the dynamic manipulability measurement. These indices could lead the optimum design of parallel manipulator architecture as addressed by Ma and Angeles in [58], however they assumed that only the mobile platform has a significant inertia with respect to the other links. Di Gregorio and Parenti-Castelli in [56] proposed to obtain the generalized inertia matrix reduced to the end-effector by means of the kinetic energy, as derived in the previous section, and they used its eigenvalues to define dynamic indices.

5.3.1 Index of Dynamic Manipulability

The *dynamic manipulability* represents the capacity of the robot to perform accelerations in any direction of the Cartesian workspace. It is quantified through the concept of dynamic manipulability ellipsoids, whose semi-axes own a length equal to the eigenvalues of the inertia matrix. An ellipsoid with equal eigenvalues is obviously a sphere and indicates a dynamically isotropic configuration of the robot. A three-dof manipulator configuration is dynamically isotropic when the eigenvalues of the inertia matrix $\lambda_i(\mathbf{x})$ for $i = 1,2,3$ in that posture are all equal, which implies that the following index assumes the unit value:

$$\kappa_d = \frac{27\lambda_1\lambda_2\lambda_3}{(\lambda_1+\lambda_2+\lambda_3)^3} \tag{5.17}$$

This is true because $\mathbf{N}$ is a symmetric and positive-semidefinite matrix: its singular values and eigenvalues coincide. The index κ_d quantifies the dynamic manipulability so it is called *index of dynamic manipulability* of a robot because the product of three scalar quantities, whose sum is assigned, is maximum only if they are all equal. It is equal to one only for dynamically isotropic configurations and it is less than one for all the other configurations ($0 \leq \kappa_d \leq 0$).

5.3.2 Condition number based on the Frobenius norm

Another index that is studied in this paper is the condition number of the inertia matrix. The condition number is based on the norm of the matrix and in order to obtain an analytic function, as it was for the previous index, the Frobenius norm is chosen:

$$\kappa_F(\mathbf{N}) = \|\mathbf{N}\|_F\|\mathbf{N}^{-1}\|_F = \frac{1}{n}\sqrt{\mathrm{tr}\left(\mathbf{N}\mathbf{N}^T\right)\mathrm{tr}\left(\mathbf{N}^{-1}\mathbf{N}^{-T}\right)} \tag{5.18}$$

The Frobenius norm has the peculiarity to be infinitely differentiable with respect to its arguments. This choice allows a straightforward minimum search by virtue of its smoothness properties. A simple formula of the condition number is based on singular values of the considered matrix; being the inertia matrix symmetric and positive-semidefinite, its singular values and eigenvalues coincide. So the condition number based on the Frobenius norm takes the form:

$$\kappa_F = \frac{1}{3}\sqrt{(\lambda_1^2+\lambda_2^2+\lambda_3^2)\left(\frac{1}{\lambda_1^2}+\frac{1}{\lambda_2^2}+\frac{1}{\lambda_3^2}\right)} \tag{5.19}$$

It is often used as a dexterity index when applied to the Jacobian matrix; the application to the matrix of inertia keeps the features of assuming the unity value when its eigenvalues are all equal, hence, it can be used to indicate of dynamic manipulability as in the previous case.

5.3.3 Manipulator swiftness

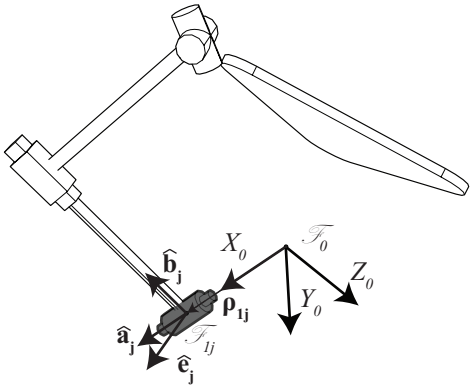
The *manipulator swiftness* is the attitude to cause end-effector acceleration for the same variation of the active forces. Like acceleration, swiftness depends on the manipulator configuration. In particular, the greater the arithmetic average of the eigenvalues of the matrix $\mathbf{N}(\mathbf{x})$ corresponding to a manipulator configuration is, the lower is the acceleration achievable by the end-effector starting from that configuration. Therefore, the value of the index κ_s which quantify the robot swiftness, is defined as

$$\kappa_s = \frac{3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{5.20}$$

and it is called *index of manipulator swiftness*. The weakness of this index is that it is dimensional, thus it is difficult to manage when different manipulators are compared.

5.4 Example: 3-CPU

The proposed algorithm in Section 5.2 has been implemented using a simplified model of the 3-CPU SPM. The inertial and geometric parameters of the links assumed for the manipulator are reported in Tables 5.1 and 5.2. The inertia matrices of each link are easily written according to a reference system fixed to the corresponding link: the entries of this matrix are constant not being dependent on the configuration of the robot. Such inertia matrices in order to computer the total kinetic energy of the manipulator must be expressed in the fixed reference system $\mathscr{F}_0$. To this aim, the rotation matrices between each local reference system of the link and the fixed reference frame are defined according to the parameters of the manipulator pose.

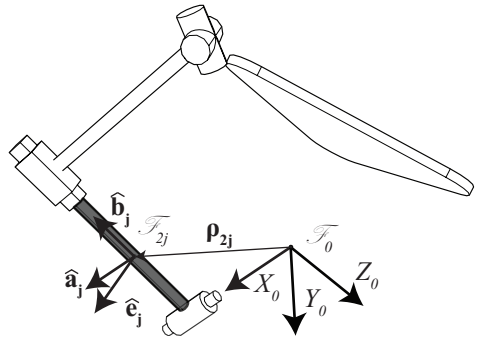


$$[\mathbf{I}_C]_{1j} = \text{diag}\,[1.830, 0.396, 1.830] \cdot 10^{-6}\,\text{kg m}^2$$

$$\mathbf{Q}_{1j} = \begin{bmatrix} \mathbf{b}_c^T\mathbf{a}_1 & \mathbf{a}_c^T\mathbf{a}_1 & \mathbf{e}_c^T\mathbf{a}_1 \\ \mathbf{b}_c^T\mathbf{a}_2 & \mathbf{a}_c^T\mathbf{a}_2 & \mathbf{e}_c^T\mathbf{a}_2 \\ \mathbf{b}_c^T\mathbf{a}_3 & \mathbf{a}_c^T\mathbf{a}_3 & \mathbf{e}_c^T\mathbf{a}_3 \end{bmatrix}$$

$$\boldsymbol{\rho}_{1j} = \mathbf{a}$$

$$m_{1j} = 0.024\,\text{kg}$$

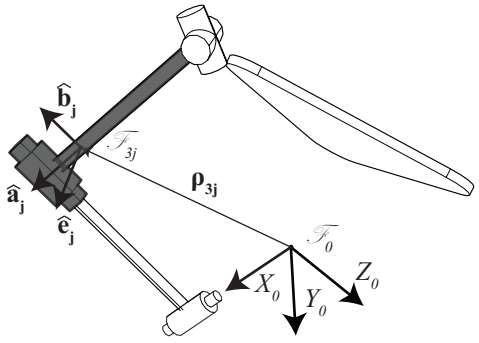


$$[\mathbf{I}_C]_{2j} = \text{diag}\,[0.01, 18.57, 18.57] \cdot 10^{-6}\,\text{kg m}^2$$

$$\mathbf{Q}_{2j} = \mathbf{Q}_{1j}$$

$$\boldsymbol{\rho}_{2j} = \mathbf{a} + c_2\hat{\mathbf{b}}$$

$$m_{2j} = 0.022\,\text{kg}$$



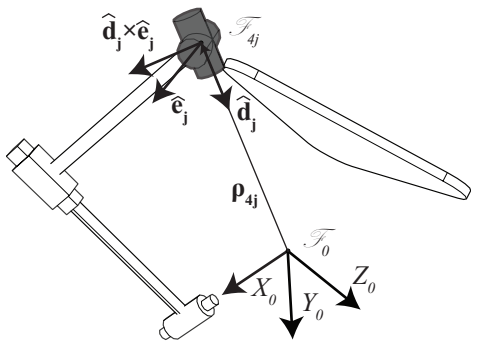
$$[\mathbf{I}_C]_{3j} = \text{diag}\,[52.57, 3.138, 54.89] \cdot 10^{-6}\,\text{kg m}^2$$

$$\mathbf{Q}_{3j} = \mathbf{Q}_{1i}$$

$$\boldsymbol{\rho}_{3j} = \mathbf{a} + \mathbf{b} + c_3\hat{\mathbf{c}}$$

$$m_{3j} = 0.055\,\text{kg}$$

Table 5.1: 3-CPU: Inertia matrix, rotation matrix, center of mass position vector and mass of the members 1,2 and 3 composing each leg j.

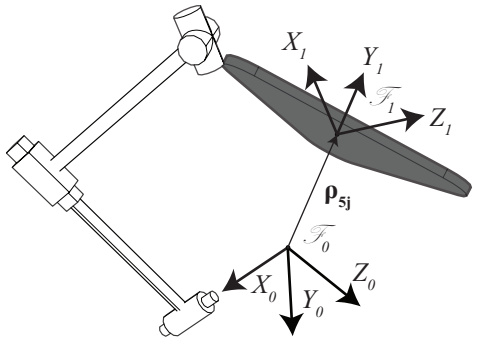


$$[\mathbf{I}_C]_{4j} = \text{diag}\,[1.448, 1.448, 2.258] \cdot 10^{-6}\,\text{kg}\,\text{m}^2$$

$$\mathbf{Q}_{4j} = \begin{bmatrix} \mathbf{d}_c^T \mathbf{a}_1 & \mathbf{e}_c^T \mathbf{a}_1 & (\mathbf{D}_c \mathbf{e}_c)^T \mathbf{a}_1 \\ \mathbf{d}_i^T \mathbf{a}_2 & \mathbf{e}_c^T \mathbf{a}_2 & (\mathbf{D}_c \mathbf{e}_c)^T \mathbf{a}_2 \\ \mathbf{d}_i^T \mathbf{a}_3 & \mathbf{e}_c^T \mathbf{a}_3 & (\mathbf{D}_c \mathbf{e}_c)^T \mathbf{a}_3 \end{bmatrix}$$

$$\boldsymbol{\rho}_{4j} = -c_4 \mathbf{d}$$

$$m_{1j} = 0.034\,\text{kg}$$



$$[\mathbf{I}_C]_{5j} = \begin{bmatrix} 254 & -63.4 & -63.4 \\ -63.4 & 254 & -63.4 \\ -63.4 & -63.4 & 254 \end{bmatrix} \cdot 10^{-6}\,\text{kg}\,\text{m}^2$$

$$\mathbf{Q}_{5j} = \mathbf{Q}$$

$$\boldsymbol{\rho}_{5j} = \mathbf{g}$$

$$m_{1j} = 0.258\,\text{kg}$$

Other data:
$d = 0.100$ m
$c = 0.100$ m
$c_2 = 0.5$ m
$c_3 = 0.5$ m
$c_4 = 0$
$\mathbf{g} = -1/30\,\text{m}[1\ 1\ 1]^T$

Table 5.2: 3-CPU: Inertia matrix, rotation matrix, center of mass position vector and mass of the members 4 and 5 composing each leg j.

Chapter 6

Singularity analysis with ERPs

Singular configurations are important in the analysis of manipulators because they should generally be avoided and because they limit the workspace of the machine. The topic is widely studied in the literature, then without going into too much detail on the description of singularities, this chapter emphasizes the representation of singularity surfaces by means of parameters used to describe the manipulator pose, i.e. the ERP. The 3-CPU robot is taken as example: the relationships between the parameters of the moving platform pose corresponding to the different singularity configurations are established and plotted. The aim of this chapter is to study the singularity surfaces with the parameters also used to write the objective function, so these equations could be used as additional constraint conditions to bond the research of the minimum.

6.1 Singular configurations

Jacobian matrices represent the mapping of both velocities and forces between the actuators and the moving platform and when one of them become singular we are not able to perform the inverse computations. Chapter 4.1 has shown how the determination of reciprocal screw $\mathbf{s}^{\perp}$ allows a particular form of the velocity problem to be obtained (Eq. (4.5)). Another way to introduce singular configuration is to examine the relations obtained from inverse kinematics [5]. For both serial and parallel robots, there will usually be a relation of the following type:

$$\mathbf{f}(\mathbf{q}, \mathbf{x}) = \mathbf{0} \tag{6.1}$$

where $\mathbf{q}$ represent the actuated joint variables and $\mathbf{x}$ the generalized coordinates. We differentiate both terms of this equation to obtain a more general relation,

$$\mathbf{J}_d \dot{\mathbf{x}} = \mathbf{J}_i \dot{\mathbf{q}} \tag{6.2}$$

This is a linear relation between the moving platform velocity vector $\mathbf{x}$ and the actuators velocity vector $\dot{\mathbf{q}}$. From Eq. (4.1) or (6.2) we can distinguish three singular cases: $\mathbf{J}_d$ is singular, $\mathbf{J}_i$ is singular, both $\mathbf{J}_d$ and $\mathbf{J}_i$ are singular. For serial manipulators, $\mathbf{J}_d$ is

the identity matrix, while for many parallel manipulators, including the 3-CPU robot, $\mathbf{J}_i$ is the identity matrix. Thus, the Jacobian matrix for a parallel manipulator for which $\mathbf{J}_i = \mathbf{1}$ is usually defined in an inverse sense with respect to the Jacobian for a serial manipulator. For manipulator designs for which neither $\mathbf{J}_i$, nor $\mathbf{J}_d$ become the identity matrix, a single Jacobian can be defined as either $\mathbf{J}_i^{-1}\mathbf{J}_d$, or $\mathbf{J}_d^{-1}\mathbf{J}_i$, provided $\mathbf{J}_d$, or $\mathbf{J}_i$ is invertible, the difference being the direction (from actuators to moving platform or vice versa) in which the Jacobian is defined.

- If $\mathbf{J}_i$ is singular, there will be a non-zero velocity vector $\dot{\mathbf{q}}$ for which the platform does not move. This singularity correspond to a pose located on the boundary of the workspace. This is a structural limit that is not related to constraint on the articular coordinates.

- If $\mathbf{J}_d$ is singular, there will then be a non-zero twist $\mathbf{t}$ for which the actuator velocities are zero. In the neighborhood of such a configuration, the robot will be able to have an infinitesimal motion without any change the link positions. As a consequence, certain degrees of freedom become uncontrollable.

- If $\mathbf{J}_i$ and $\mathbf{J}_d$ are singular, there is a singularity where the moving platform may be moved while the actuators are locked, and vice versa.

When reciprocal screws are identified, they represent the rows of the direct Jacobian matrix $\mathbf{J}_d$ as shown in Eq. (4.5). The linear dependence between these screws determines a singularity condition and analytically it corresponds to

$$\mathbf{n}_1 \times \mathbf{n}_2 \cdot \mathbf{n}_3 = 0 \tag{6.3}$$

because in general this relationship between the rows $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and the determinant of a 3×3 matrix $\mathbf{A}$ subsists

$$\mathbf{a} \times \mathbf{b} \cdot \mathbf{c} = \det(\mathbf{A}) \tag{6.4}$$

The inverse kinematics Jacobian matrix $\mathbf{J}_i$, being a diagonal matrix, becomes singular if one or more diagonal values vanish.

The singularity analysis through the Jacobian matrices, in which the contribution of passive joints is neglected multiplying reciprocal screws to them, is in general non-exhaustive. The avoidance of passive-constraint singularities [50] cannot be guaranteed using these matrices. This type of singularity is not addressed in this manuscript but they should be identified and appropriately avoided.

6.2 Rendering workspace with ERP

To describe the orientation of the MP three independent parameters are necessary. However, to avoid representation singularities, the four parameters representation with

ERPs has been chosen. The four scalar ERPs $\mathbf{r}$ and r_0 are dependent because their norm is bound to be unitary

$$\mathbf{r}^T\mathbf{r} + r_0^2 = 1 \tag{6.5}$$

This relationship represents a surface of a four-dimensional sphere. Therefore, a rotation, or an EE pose, can be represented on a four-dimensional space as a point in a spherical surface centered at the origin of the space. The appearance of such spherical surface in a three-dimensional space can be carried out by projecting a four-dimensional object in a three-dimensional subspace. A possible option is to represent a rotation matrix as a point in a three dimensional sphere whose position vector is equal to the vector $\mathbf{r}$ associated with that rotation.

$$\|\mathbf{r}\| \leq 1 \tag{6.6}$$

This point collects both information of rotation axis $\mathbf{e}$ and angle of rotation θ because $\mathbf{r} = \mathbf{e}\sin(\theta/2)$. For example, the point at the origin of the reference frame represents all the poses of the manipulator in the reference relative position between the mobile and the fixed frames, or initial pose, because it corresponds to a null rotation around any directions. In our case the initial configuration corresponds to coaxial axes of the two frames. The point $\mathbf{p} = \sqrt{3}/3[1\ 1\ 1]^T$ identifies instead a rotation of π around $\hat{\mathbf{p}}$ of the two frames. Singularity configurations are found and plotted for the 3-CPU manipulator using this 3D representation in the ERPs space. Nevertheless, the reader should note that the shape of the singularity surfaces depends on the invariants of the rotation matrix used to represent it, e.g. the use of linear invariants $\mathbf{q} = \mathbf{e}\sin(\theta)$ provides different shapes. However, parametrization with ERPs is known not to suffer of singularities of representation: this is the reason why the rendering of the workspace with these parameters is convenient.

6.3 Example: 3-CPU

The singularity condition of the Jacobian matrix $\mathbf{J}$ corresponds to a MP pose whose condition number attains an infinite value. In fact the condition number gives an indication of how much close a pose is from a singularity configuration. For this reason we begin our research of singulaties from the denominator of the 3-CPU condition number of the inverse Jacobian matrix written in Eq. (4.31) in terms of the ERPs:

$$\begin{aligned}\mathbf{n}_1 \times \mathbf{n}_2 \cdot \mathbf{n}_3 = \det(\mathbf{J}_d) = d^3(r_1 + r_2 + r_3 - r_0)(-r_1 - r_2 + r_3 - r_0) \\ \cdot(r_1 - r_2 - r_3 - r_0)(-r_1 + r_2 - r_3 - r_0)\end{aligned} \tag{6.7}$$

Clearly, the determinant vanishes when one of its factors vanishes, i.e. when an ERPs set respects one of these equations:

$$\det(\mathbf{J}_d) = 0 \quad \Leftrightarrow \quad \pi_1 = 0 \vee \pi_2 = 0 \vee \pi_3 = 0 \vee \pi_4 = 0 \tag{6.8}$$

having defined

$$\begin{aligned} \pi_1 &: r_1 + r_2 + r_3 - r_0 \\ \pi_2 &: -r_1 - r_2 + r_3 - r_0 \\ \pi_3 &: -r_1 + r_2 - r_3 - r_0 \\ \pi_4 &: r_1 - r_2 - r_3 - r_0 \end{aligned} \tag{6.9}$$

The geometrical parameter d is a constant value defined at the design stage and it is assumed different from zero. It is pointed out that $\pi_1 = 0$, $\pi_2 = 0$, $\pi_3 = 0$ and $\pi_4 = 0$ represent four affine hyperplanes in the four-dimensional space of ERP. Such equations are associated with four dimensional geometric entities that identify the borders of the robot workspace. We may illustrate them in a three-dimensional space using the representation with ERPs by removing from such equations the explicit dependence on r_0. To this aim we calculate the square of Eq. (6.9) to replace r_0^2 into Eq. (6.5). In this way, we obtain for the four equations

$$\begin{aligned} \pi_1 &: 2(r_1^2 + r_2^2 + r_3^2 + r_1 r_2 + r_2 r_3 + r_1 r_3) - 1 \\ \pi_2 &: 2(r_1^2 + r_2^2 + r_3^2 + r_1 r_2 - r_2 r_3 - r_1 r_3) - 1 \\ \pi_3 &: 2(r_1^2 + r_2^2 + r_3^2 - r_1 r_2 - r_2 r_3 + r_1 r_3) - 1 \\ \pi_4 &: 2(r_1^2 + r_2^2 + r_3^2 - r_1 r_2 + r_2 r_3 - r_1 r_3) - 1 \end{aligned} \tag{6.10}$$

which can be rewritten in the quadratic form

$$\mathbf{r}^T \mathbf{A}_i \mathbf{r} = 1 \qquad i = 1,2,3,4 \tag{6.11}$$

where

$$\begin{aligned} \mathbf{A}_1 &= 2\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \qquad \mathbf{A}_2 = 2\begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \\ \mathbf{A}_3 &= 2\begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{A}_4 = 2\begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{aligned} \tag{6.12}$$

It's interesting to note that these quadratics forms represent analytical expressions of ellipsoid surfaces centered at the origin. Each ellipsoid surface is shown in Fig. 6.1. Carbonari et al. [59] have already identified and studied these geometric entities for the 3-CPU in the space of the actuator displacement where they assume the shape of

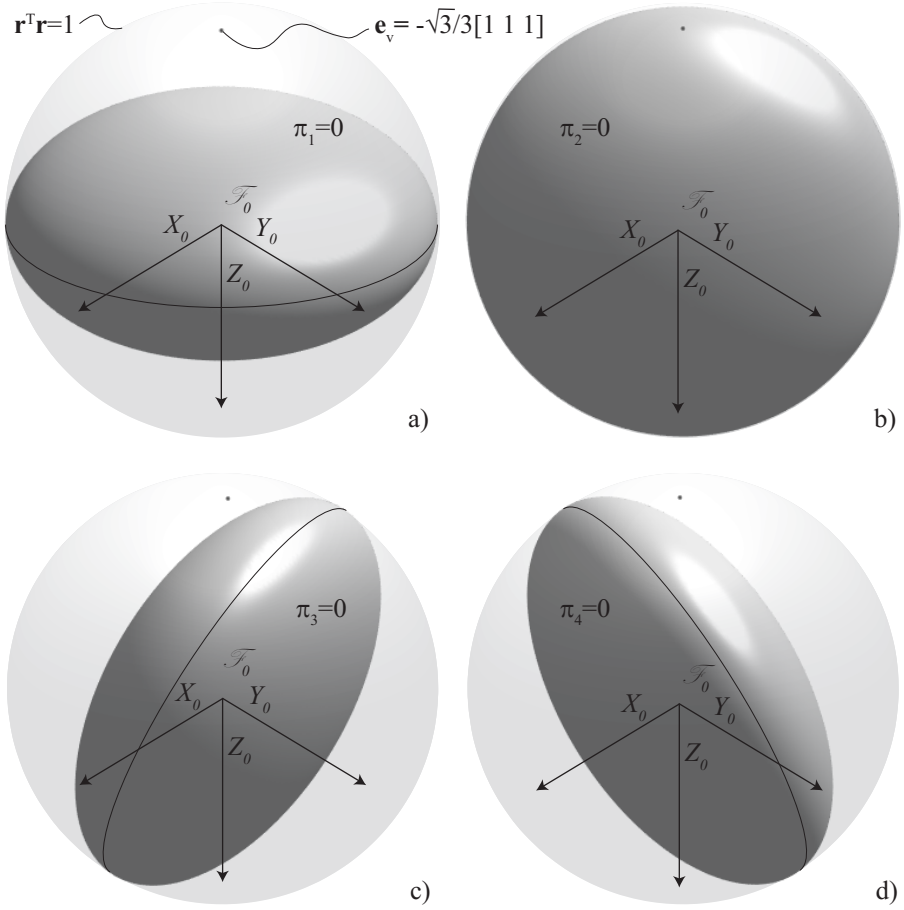


Figure 6.1: Renderings of the singularity ellipsoid surfaces with the ERPs 3D representation of the 3-CPU SPM: a) $\pi_1 = 0$, b)$\pi_2 = 0$, c) $\pi_3 = 0$ and d) $\pi_4 = 0$. Lines indicate the intersection with the unit sphere.

planes. They showed that, when the EE pose reaches these surfaces, the robot can change its own working mode: in particular, the transition of the robot configurations between them corresponds to transition between pure rotational and hybrid behavior (translational and rotational) of the SPM. Whatever the space of representation is, these geometric entities identify the limits of the robot workspace. The four ellipsoid surfaces are illustrated together in Fig. 6.2. The eigenvectors of $\mathbf{A}_i$ define the principal axes of the ellipsoid, while the eigenvalues of $\mathbf{A}_i$ are the reciprocals of the squares of lengths of the semi-axes. As we can see from Fig.s 6.1 and 6.2 two of the ellipsoid semi-axes are equal to one thus their intersections with unit sphere are the circumferences which are drawn in the Figures. The intersection of the ellipsoid volume gives us the singularity-free region of the workspace.

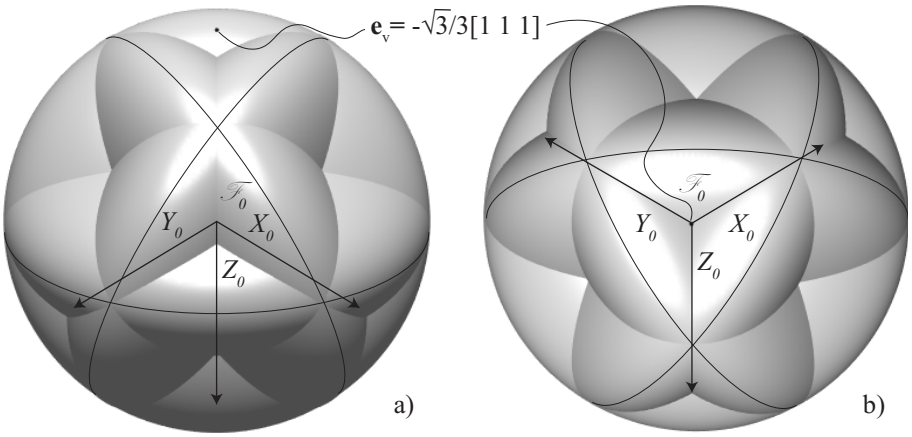


Figure 6.2: Renderings with ERPs representation of the singularity ellipsoid surfaces of the 3-CPU SPM: a) isometric view and b) upper view. Lines indicate the intersections with the unit sphere.

The visualization of the axis of rotation between the fixed and mobile frames and its magnitude depending on the angle of rotation does not give an easy information about the pointing direction that the mobile platform attains after this transformation. A more intuitive representation is given by substituting the axis of rotation with a pointing direction $\mathbf{e}_v$ of the mobile platform after the application of the rotation, in mathematical terms

$$\mathbf{p} = \mathbf{Q}(\mathbf{r}, r_0)\mathbf{e}_v \sin(\theta/2) \tag{6.13}$$

where $\mathbf{p}$ is a point inside the unit sphere having chosen $[\mathbf{e}_v]_1 = -\sqrt{3}/3[1\ 1\ 1]^T$ as a reference vector fixed with the mobile reference system. In the home configuration the two reference systems are coincident and the vector $[\mathbf{e}_v]_0 = -\sqrt{3}/3[1\ 1\ 1]^T$ indicates the vertical direction also in the fixed reference frame $\mathscr{F}_0$.

The singularity surfaces with this representation ε_i assume a more complex shape: Fig. 6.3 shows these entities from the same point of view, while they assume different orientations as in the case of ellipsoids. One more time, the intersection volume between these surfaces enclosures the pointing workspace of the 3-CPU. The assembly of the four entities ε (Fig. 6.4) reveals a symmetry with respect to planes shifted 120 degrees one from the other deriving from the tripod architecture of the SPM. These surfaces are linked to particular poses of the MP where the robot changes its working modes; they represent singular points of the robot workspace in which the legs can change their configuration independently from the actuation. Verifing exactly that a trajectory is singularity-free is not a trivial problem, however, the equations of these surfaces can be considered in the constraint equations of optimization problems to limit the manipulator workspace. Moreover, for a real robot also the limits imposed by the joints must be added to the constraint equations system. This problem has not

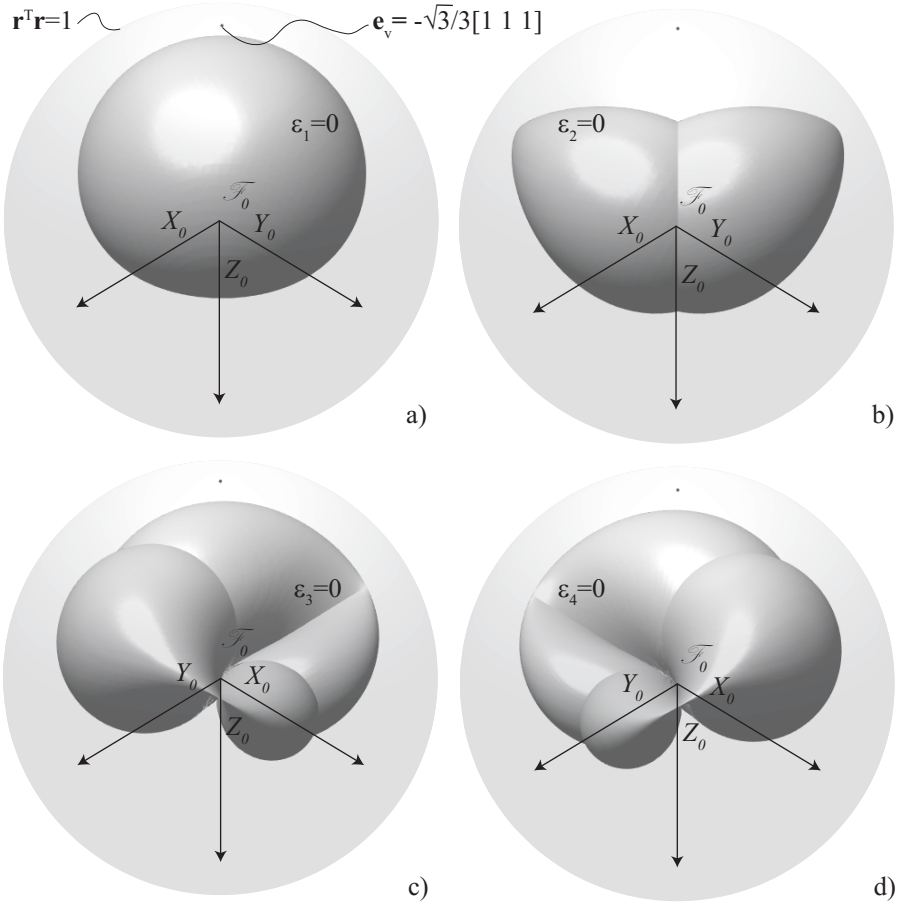


Figure 6.3: The four singularities surfaces for the 3-CPU SPM; Renderings with ERPs representation of the pointing direction of the moving platform.

been addressed in this work but it could be part of future developments.

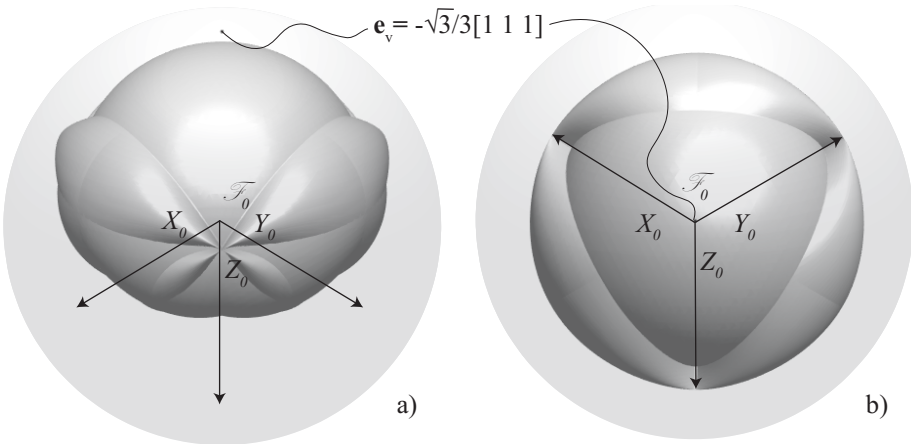


Figure 6.4: Renderings with a pointing ERPs representation of the moving platform singularity surfaces of the 3-CPU SPM: a) isometric view and upper view b).

Chapter 7

Posture Optimization

This chapter addresses the posture optimization of a redundant spherical robot with one dof of functional redundancy and no intrinsic redundancy. The functional redundancy that arises when a spherical PKM performs a pointing task is exploited to maximize the machine performance using indices that have been already proposed in the dissertation. As an example, the minimization of the condition number of the Jacobian matrix is addressed to reach higher dexterity performance of the PKM avoiding singularities. Different case studies of spherical manipulators performing a pointing task are reported, in order to show how posture-optimization can be used as a redundancy-solution for functionally redundant PKMs. The indices will be the objective function of the problem, while the orientation of the pointing task is used to build the constraint equations using ERPs. Sequential Quadratic Programming is conducted to numerically solve the nonlinear constrained optimization problem and to find the moving platform pose corresponding to the robot posture of maximum performance for a given pointing direction of the MP. Lastly, the constrained problem is rewritten as an unconstrained optimization problem with an objective function in one design variable. The optimization problems are then extended to every pointing direction.

7.1 Formulation of the Optimization Problem

The problem of the functional redundancy of a SPM performing a pointing two-dof task can be formulated as a constrained optimization. The constraints are imposed by the pointing specification, while the objective function can be defined as the square of the condition number $\kappa_F^2(\mathbf{x})$ as shown in Eq. (4.28) for the dexterity optimization of the 3-CPU manipulator. Thus, the problem takes the form:

$$f(\mathbf{x}) \equiv \kappa(\mathbf{x}) \rightarrow \min_{\mathbf{x}}, \text{ s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{7.1}$$

where $\mathbf{h}$ is a vector of constraints and $\mathbf{x}$ the array of unknowns. The constrained problem can be formulated by means of points lying on a sphere with unit radius. Choosing this approach, a rotation about the hole axis that passes through the center of the spherical manipulator is the functional redundancy. The constraint is then the

coincidence of the vertical unit vector $\mathbf{e}_v$, taken as reference vector and mapped by the rotation matrix, and the prescribed unit vector $\mathbf{e}_h$, both expressed in the same reference frame, i.e. the fixed frame $\mathscr{F}_0$:

$$\mathbf{h}(\mathbf{x}) = \mathbf{Q}\mathbf{e}_v - \mathbf{e}_h = \mathbf{0} \quad \Rightarrow \quad \mathbf{h}(\mathbf{x}) = [\mathbf{Q}(\mathbf{x})]_0[\mathbf{e}_v]_0 - [\mathbf{e}_h]_0 = \mathbf{0} \tag{7.2}$$

According to the two reference systems that have been chosen, $[\mathbf{e}_v]_0 \widehat{=} [\mathbf{e}_h]_1 = -\sqrt{3}/3 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$.

7.2 Pointing Constraint Equations with ERP

So far the constraint equation has been written in terms of the rotation matrix $\mathbf{Q}$ that needs a proper parametrization to describe the orientation of the mobile platform. We use the same representation already used in the previous chapter to describe rotations, i.e. the Euler-Rodrigues Parameters (ERPs) consisting in the four scalars $\mathbf{r}$ and r_0 [55]. This representation, shown in Eq. (4.27), is free of singularities. Such property is significant when the attitude of a body must be controlled in 3D space: in this case both the information of orientation and angular velocity are used in order to plan the motion, and smoother trajectories can therefore be obtained. The kinematics of rigid-body rotations is well documented in the literature [55], although most of the time analyses based on coordinates are found. A coordinate-free discussion is invoked here, with the purpose of stressing the invariant properties of rotations, i.e. those quantities $\mathbf{r}$ and r_0 that, if scalar, remain immutable under a change of the coordinate frame. In general, the scalar function $f([\mathbf{p}])$ is said to be *frame invariant*, or *invariant* for brevity, if

$$f([\mathbf{p}]_{\mathscr{B}}) = f([\mathbf{p}]_{\mathscr{A}}) \tag{7.3}$$

where $\mathscr{A}$ and $\mathscr{B}$ are two different reference frames. However, the *invariance* concept can be extended also to vectors and matrices. The vector quantity $\mathbf{f}$ is said to be invariant if

$$[\mathbf{f}]_{\mathscr{A}} = [\mathbf{Q}]_{\mathscr{A}}[\mathbf{f}]_{\mathscr{B}} \tag{7.4}$$

Let the proper orthogonal matrix $[\mathbf{Q}]_{\mathscr{A}}$ denote the rotation of coordinate frame $\mathscr{A}$ into $\mathscr{B}$. Finally, the matrix quantity $\mathbf{F}$ is said to be invariant if

$$[\mathbf{F}]_{\mathscr{A}} = [\mathbf{Q}]_{\mathscr{A}}[\mathbf{F}]_{\mathscr{B}}[\mathbf{Q}^T]_{\mathscr{A}} \tag{7.5}$$

The array of unknowns becomes $\mathbf{x} = \begin{bmatrix} \mathbf{r}^T & r_0 \end{bmatrix}^T$, which are dependent, holding the constraint

$$\|\mathbf{r}\|^2 + r_0^2 = 1 \tag{7.6}$$

A non-linear system of four algebraic equations in four unknowns is thus obtained:

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{cases} \mathbf{Q}\mathbf{e}_v - \mathbf{e}_h = \mathbf{0} \\ \mathbf{r}^T\mathbf{r} + r_0^2 = 1 \end{cases} \tag{7.7}$$

If the rotation matrix $\mathbf{Q}$ is written as function of the Euler-Rodrigues parameters $\mathbf{r}$ and r_0, previous equations becomes

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{cases} r_0^2\mathbf{e}_v - \mathbf{r}^T\mathbf{r}\mathbf{e}_v + 2\mathbf{r}^T\mathbf{e}_v\mathbf{r} - 2r_0\mathbf{E}_v\mathbf{r} - \mathbf{e}_h = \mathbf{0} \\ \mathbf{r}^T\mathbf{r} + r_0^2 = 1 \end{cases} \tag{7.8}$$

We obtain a system of four equations in four unknowns, that seems to leave no room for optimization. To find whether the system of equations hides a dependency between the variables we study the gradient of the constraint equations, that is

$$\nabla\mathbf{h}(\mathbf{x}) = \begin{bmatrix} -2\mathbf{e}_v\mathbf{r}^T + 2\mathbf{r}\mathbf{e}_v^T + 2\mathbf{r}^T\mathbf{e}_v\mathbf{1} - 2r_0\mathbf{E}_v & (2r_0\mathbf{1} + 2\mathbf{R})\mathbf{e}_v \\ 2\mathbf{r}^T & 2r_0 \end{bmatrix} \tag{7.9}$$

Actually, the system conceals a nonlinear dependency between the variables [53], since the 4×4 gradient of the system of equations is singular. The proof of this result is reported.

Proof: $\nabla\mathbf{h}(\mathbf{x})$ is singular

$\nabla\mathbf{h}(\mathbf{x})$ is written as a block matrix. It can be decomposed (block-LU-decomposition) when $\mathbf{A}$ can be inverted in order to find a simplified expression of its determinant.

$$\nabla\mathbf{h} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{c}^T & d \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{c}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{0}^T & d - \mathbf{c}^T\mathbf{A}^{-1}\mathbf{b} \end{bmatrix} \tag{7.10}$$

It follows that

$$\det(\nabla\mathbf{h}) = \det(\mathbf{A})(d - \mathbf{c}^T\mathbf{A}^{-1}\mathbf{b}) \tag{7.11}$$

where, in our case, it is $\mathbf{c}^T\mathbf{A}^{-1}\mathbf{b} = 2r_0 = d$. The determinant of $\mathbf{A}$ is difficult to handle, therefore an alternative decomposition of the matrix, when $d \neq 0$, leads to

$$\begin{aligned} \det(\nabla\mathbf{h}) &= d\det(\mathbf{A} - \frac{1}{d}\mathbf{b}\mathbf{c}^T) \\ &= \Big(d\det \quad -2\mathbf{e}_v\mathbf{r}^T + 2\mathbf{r}\mathbf{e}_v^T + 2\mathbf{r}^T\mathbf{e}_v\mathbf{1} - 2r_0\mathbf{E}_v - \frac{2}{r_0}(r_0\mathbf{1} + \mathbf{R})\mathbf{e}_v\mathbf{r}^T\Big) \end{aligned} \tag{7.12}$$

After simplification we obtain

$$\det(\nabla \mathbf{h}) = 16 r_0 \left(\det \quad \mathbf{r}\mathbf{e}_v^T - r_0 \mathbf{E}_v - \frac{1}{r_0}(2r_0\mathbf{1} + \mathbf{R})\mathbf{e}_v\mathbf{r}^T + \mathbf{r}^T\mathbf{e}_v\mathbf{1} \right) \tag{7.13}$$

If there exists a vector $\mathbf{v} \in \mathbb{R}$, different to the null vector, that lies in the null space of the matrix inside the parenthesis, then its determinant must be zero. We first try $\mathbf{e}_v$

$$\begin{aligned} &\left(\quad \mathbf{r}\mathbf{e}_v^T - r_0\mathbf{E}_v - \frac{1}{r_0}(2r_0\mathbf{1} + \mathbf{R})\mathbf{e}_v\mathbf{r}^T + \mathbf{r}^T\mathbf{e}_v\mathbf{1} \right) \mathbf{e}_v \\ &= \mathbf{r} - \frac{\mathbf{r}^T\mathbf{e}_v}{r_0}(2r_0\mathbf{1} + \mathbf{R})\mathbf{e}_v + \mathbf{r}^T\mathbf{e}_v\mathbf{e}_v \end{aligned} \tag{7.14}$$

If $\mathbf{r} = \mathbf{e}_v$, then

$$\mathbf{e}_v - 2\mathbf{e}_v + \mathbf{e}_v = \mathbf{0} \tag{7.15}$$

This proves that the determinant of the matrix is zero and , as a consequence, $\nabla \mathbf{h}$ is singular.

As a matter of fact, the 4^{th} equation of the system naturally comes out from the Euclidean norm of the two sides of Eq. (7.2) by imposing that $\mathbf{e}_v$ and $\mathbf{e}_h$ are unit vectors.

Proof

The vector constraint equation $\mathbf{Q}\mathbf{e}_v = \mathbf{e}_h$ is manipulated by means of its Euclidean norm

$$\mathbf{e}_v^T\mathbf{Q}^T\mathbf{Q}\mathbf{e}_v = \mathbf{e}_h^T\mathbf{e}_h \tag{7.16}$$

and introducing

$$\mathbf{Q} = (r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R} \tag{7.17}$$

it becomes

$$\mathbf{Q}^T\mathbf{Q} = ((r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R})^T((r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R}) \tag{7.18}$$

The first two terms of Eq. (7.16) are symmetrical matrices, while the third one is a skew-symmetric matrix; thus the transpose matrix becomes

$$\mathbf{Q}^T\mathbf{Q} = ((r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T - 2r_0\mathbf{R})((r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathbf{R}) \tag{7.19}$$

Many terms vanish expanding the equation:

$$\begin{aligned} \mathbf{Q}^T\mathbf{Q} =& (r_0^2 - \mathbf{r}^T\mathbf{r})^2\mathbf{1} + 2(r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{r}\mathbf{r}^T + 2r_0(r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{R} + 2(r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{r}\mathbf{r}^T \\ &+ 4\mathbf{r}^T\mathbf{r}\mathbf{r}\mathbf{r}^T - 2r_0(r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{R} - 4r_0^2\mathbf{R}^2 \end{aligned} \tag{7.20}$$

So we obtain

$$\mathbf{Q}^T\mathbf{Q} = (r_0^2 - \mathbf{r}^T\mathbf{r})^2\mathbf{1} + 4r_0^2\mathbf{r}\mathbf{r}^T - 4r_0^2\mathbf{R}^2 \tag{7.21}$$

Being $\mathbf{R} = \mathrm{CPM}(\mathbf{r})$, it can be proven that $\mathbf{R}^2 = -\mathbf{r}^T\mathbf{r}\mathbf{1} + \mathbf{r}\mathbf{r}^T$.

$$\mathbf{Q}^T\mathbf{Q} = (r_0^2 - \mathbf{r}^T\mathbf{r})^2\mathbf{1} + 4r_0^2\mathbf{r}\mathbf{r}^T + 4r_0^2\mathbf{r}^T\mathbf{r}\mathbf{1} - 4r_0^2\mathbf{r}\mathbf{r}^T \tag{7.22}$$

$$\mathbf{Q}^T\mathbf{Q} = (r_0^2 + \mathbf{r}^T\mathbf{r})^2\mathbf{1} \tag{7.23}$$

By substituting this expression into Eq. (7.16) we obtain

$$(r_0^2 + \mathbf{r}^T\mathbf{r})^2\mathbf{e}_v^T\mathbf{e}_v = \mathbf{e}_h^T\mathbf{e}_h \tag{7.24}$$

If we impose that $\mathbf{e}_h$ and $\mathbf{e}_h$ are unit vectors it follows

$$r_0^2 + \mathbf{r}^T\mathbf{r} = \pm 1 \tag{7.25}$$

This leads to the ambiguity of proper-improper orthogonality of the rotation matrix, because $\det(\mathbf{Q}) = (r_0^2 + \mathbf{r}^T\mathbf{r})^3)$. Isometry preservation of the Euclidean vector norm of the $\mathbf{Q}$ mapping for this form of the rotation matrix leads to have the single solution. What will eliminate the ambiguity is the invariant form of a reflection $\mathbf{R} = \mathbf{1} - \mathbf{n}\mathbf{n}^T$. A reflection in $\mathbb{R}^3$ is necessarily symmetric; given that $\mathbf{Q}$ has a skew-symmetric component, this disqualifies $\mathbf{Q}$ from being a reflection, thereby eliminating the ambiguity. In conclusion, since $r_0 \in \mathbb{R}$ and $\mathbf{r} \in \mathbb{R}^3$, we obtain

$$r_0^2 + \mathbf{r}^T\mathbf{r} = 1 \tag{7.26}$$

This demonstrates that the unit norm equation is redundant; furthermore, the system provided by the first three equations outputs an ERPs set of values with unit norm. This equation, as we will see later on, will be removed from the system of constraint equations in order to reach a better computational efficiency.

Projection of the constraint equations

In order to provide the system with a straightforward physical interpretation, a projection of Eqs. 7.2 is operated along known directions. In fact, the constraint vector can be rewritten expanding the first equation

$$r_0^2\mathbf{e}_v - \mathbf{r}^T\mathbf{r}\mathbf{e}_v + 2\mathbf{r}\mathbf{r}^T\mathbf{e}_v + 2r_0\mathbf{R}\mathbf{e}_v - \mathbf{e}_h = \mathbf{0} \tag{7.27}$$

and multiplying all terms by $\mathbf{r}^T$

$$(r_0^2 + \mathbf{r}^T\mathbf{r})\mathbf{r}^T\mathbf{e}_v + 2r_0\mathbf{r}^T\mathbf{R}\mathbf{e}_v - \mathbf{r}^T\mathbf{e}_h = 0 \tag{7.28}$$

Being $\mathbf{r}^T\mathbf{R}\mathbf{e}_v = \mathbf{e}_v^T\mathbf{R}\mathbf{r} = 0$ and $r_0^2 + \mathbf{r}^T\mathbf{r} = 1$, it results

$$\mathbf{r}^T\mathbf{e}_v = \mathbf{r}^T\mathbf{e}_h \tag{7.29}$$

Multiplying all terms of the Eq. (7.27) by $\mathbf{E}_v$ we obtain

$$r_0^2\mathbf{E}_v\mathbf{e}_v - \mathbf{r}^T\mathbf{r}\mathbf{E}_v\mathbf{e}_v + 2\mathbf{r}^T\mathbf{e}_v\mathbf{E}_v\mathbf{r} + 2r_0\mathbf{E}_v\mathbf{R}\mathbf{e}_v - \mathbf{E}_v\mathbf{e}_h = \mathbf{0} \tag{7.30}$$

The first two terms vanish while the triple vector product can be rewritten as $\mathbf{ABc} = \mathbf{a}^T\mathbf{cb} - \mathbf{a}^T\mathbf{bc}$, thus

$$2\mathbf{r}^T\mathbf{e}_v\mathbf{E}_v\mathbf{r} + 2r_0(\mathbf{r} - \mathbf{r}^T\mathbf{e}_v\mathbf{e}_v) - \mathbf{E}_v\mathbf{e}_h = \mathbf{0} \tag{7.31}$$

Multiplying by $\mathbf{r}^T$

$$2\mathbf{r}^T\mathbf{e}_v\mathbf{r}^T\mathbf{E}_v\mathbf{r} + 2r_0(\mathbf{r}^T\mathbf{r} - (\mathbf{r}^T\mathbf{e}_v)^2) - \mathbf{r}^T\mathbf{E}_v\mathbf{e}_h = 0 \tag{7.32}$$

we obtain another equation

$$\mathbf{r}^T\mathbf{E}_v\mathbf{e}_h = 2r_0(\mathbf{r}^T\mathbf{r} - (\mathbf{r}^T\mathbf{e}_v)^2) \tag{7.33}$$

Finally we multiply Eq. (7.27) by $\mathbf{e}_v^T$

$$\mathbf{e}_v^T\mathbf{e}_h = r_0^2 - \mathbf{r}^T\mathbf{r} + 2(\mathbf{r}^T\mathbf{e}_v)^2 \tag{7.34}$$

or

$$\mathbf{e}_v^T\mathbf{e}_h = 1 - 2\mathbf{r}^T\mathbf{r} + 2(\mathbf{r}^T\mathbf{e}_v)^2 \tag{7.35}$$

These three equality constraint equations are collected in the following system:

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{cases} \mathbf{r}^T\mathbf{e}_v = \mathbf{r}^T\mathbf{e}_h \\ \mathbf{r}^T\mathbf{E}_v\mathbf{e}_h = 2r_0(\mathbf{r}^T\mathbf{r} - (\mathbf{r}^T\mathbf{e}_v)^2) \\ \mathbf{e}_v^T\mathbf{e}_h = 2(r_0^2 + (\mathbf{r}^T\mathbf{e}_v)^2) - 1 \\ \|\mathbf{r}\|^2 + r_0^2 = 1 \end{cases} \tag{7.36}$$

From the first scalar equation it results that the angle between the vector $\mathbf{r}$ and the two unit vectors $\mathbf{e}_v$ and $\mathbf{e}_h$ of the vertical and hole axes must be equal: this vector is bound to lie on the bisecting plane defined by to the two unit vectors $\mathbf{e}_v$ and $\mathbf{e}_h$. Vector $\mathbf{r}$ can sweep this plane by rotating around the origin of the frames. Within this set of vectors $\mathbf{r}$ that brings $\mathbf{e}_v$ to overlay with $\mathbf{e}_h$, the solution of the problem is the one that optimally

orients the EE in the $\mathbf{e}_h$ pointing direction. The gradient of these equations is

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_h^T - \mathbf{e}_v^T & 0 \\ -\mathbf{e}_v^T \mathbf{E}_h - 4r_0(\mathbf{r}^T - \mathbf{r}^T \mathbf{e}_h \mathbf{e}_h^T) & -2(\mathbf{r}^T \mathbf{r} - (\mathbf{r}^T \mathbf{e}_h)^2) \\ 4\mathbf{r}^T - 4\mathbf{r}^T \mathbf{e}_h \mathbf{e}_h^T & 0 \\ 2\mathbf{r}^T & 2r_0 \end{bmatrix} \tag{7.37}$$

Constraint equation with Natural Invariants

For the sake of completeness, it should be also provided the relations of constraint equations using linear invariant as defined by Angeles [55]. The rotation matrix $\mathbf{Q}$ can be written as a function of its natural invariants:, i.e. the axis of rotation $\mathbf{e}$ and the angle of rotation ϕ.

$$\mathbf{Q} = \mathbf{e}\mathbf{e}^T + \cos\phi(\mathbf{1} - \mathbf{e}\mathbf{e}^T) + \sin\phi\mathbf{E} \tag{7.38}$$

A fourth equation can be introduced, that is the constraint

$$\|\mathbf{e}\|^2 = 1 \tag{7.39}$$

Thus the equality constraints are

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{cases} (\mathbf{e}\mathbf{e}^T + \cos\phi(\mathbf{1} - \mathbf{e}\mathbf{e}^T) + \sin\phi\mathbf{E})\mathbf{e}_v - \mathbf{e}_h = \mathbf{0} \\ \|\mathbf{e}\| = 1 \end{cases} \tag{7.40}$$

that can be rewritten as

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{cases} \mathbf{e}^T \mathbf{e}_v = \mathbf{e}^T \mathbf{e}_h \\ \mathbf{e}^T \mathbf{E}_v \mathbf{e}_h = \sin\phi(1 - (\mathbf{e}_v^T \mathbf{e})^2) \\ \mathbf{e}_v^T \mathbf{e}_h = (1 - \cos\phi)(\mathbf{e}^T \mathbf{e}_v)^2 + \cos\phi \\ \|\mathbf{e}\|^2 = 1 \end{cases} \tag{7.41}$$

The gradient of this system is

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} (\mathbf{e}_v - \mathbf{e}_h)^T & 0 \\ (\mathbf{E}_v \mathbf{e}_h + 2\sin\phi(\mathbf{e}^T \mathbf{e}_v \mathbf{e}_v))^T & \cos\phi(1 - (\mathbf{e}_v^T \mathbf{e})^2) \\ (2(1 - \cos\phi)(\mathbf{e}^T \mathbf{e}_v \mathbf{e}_v))^T & \sin\phi(1 - (\mathbf{e}_v^T \mathbf{e})^2) \\ 2\mathbf{e}^T & 0 \end{bmatrix} \tag{7.42}$$

7.3 Unconstrained Optimization Problem

The constrained optimization can be rewritten as an unconstrained problem by decomposing the $\mathbf{Q}$ matrix into two rotation matrices, namely, $\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2$. For the sake of simplicity, the two factors are described here in terms of their linear invariants

$\mathbf{Q}_i(\mathbf{e}_i, \vartheta_i)$, even though ERPs were used for practical implementation. The first rotation is determined by imposing the coincidence of the two given vectors: among the infinite number of rotation matrices, the geodesic one that minimizes the angle of rotation ϑ_1 is chosen [60]

$$\mathbf{Q}_1 \leftarrow \mathbf{Q}_1 \left(\frac{\mathbf{e}_v \times \mathbf{e}_h}{\|\mathbf{e}_v \times \mathbf{e}_h\|}, \text{atan2}\left(\|\mathbf{e}_v \times \mathbf{e}_h\|, \mathbf{e}_v^T \mathbf{e}_h\right) \right) \tag{7.43}$$

In this notation the parenthesis includes the arguments of $\mathbf{Q}_1$, namely, the unit vector in the direction of the axis of rotation and the angle of rotation. The matrix $\mathbf{Q}_2(\mathbf{e}_h, \vartheta_2)$ describes a rotation around the unit vector $\mathbf{e}_h$ of an angle ϑ_2. Then, the whole rotation $\mathbf{Q}$ can be used to compute the square of the condition number κ_F^2 as a function of ϑ_2, which represents the only unknown. Such manipulations allow the writing down of the problem as an unconstrained optimization of the objective function κ_F^2 in the single design variable ϑ_2.

Compound Rotation with ERPs

The unconstrained optimization problem can be written in terms of ERP using the relation introduced in this section. Given the ERPs of two rotations $\mathbf{Q}_1(\mathbf{r}_1, r_{01})$ and $\mathbf{Q}_2(\mathbf{r}_2, r_{02})$ the compound rotation matrix $\mathbf{Q}(\mathbf{r}_f, r_{0f}) = \mathbf{Q}_2\mathbf{Q}_1$ is described by the ERPs

$$\mathbf{r}_f = r_{01}\mathbf{r}_2 + r_{02}\mathbf{r}_1 + \mathbf{R}_2\mathbf{r}_1 \tag{7.44}$$

$$r_{0f} = r_{01}r_{02} - \mathbf{r}_1^T\mathbf{r}_2 \tag{7.45}$$

The two sets of ERPs can be written as a function of the unit vectors:

$$\mathbf{r}_1 = \mathbf{e}_1 \sin\frac{\phi_1}{2} = \frac{\mathbf{E}_v\mathbf{e}_h}{\|\mathbf{E}_v\mathbf{e}_h\|}\sqrt{\frac{1-\mathbf{e}_v^T\mathbf{e}_h}{2}} \tag{7.46}$$

$$r_{01} = \sqrt{\frac{1+\mathbf{e}_v^T\mathbf{e}_h}{2}}; \tag{7.47}$$

$$\mathbf{r}_2 = \mathbf{e}_2 \sin\frac{\phi_2}{2} = \mathbf{e}_h \sin\frac{\phi_2}{2} \tag{7.48}$$

$$r_{02} = \cos\frac{\phi_2}{2} \tag{7.49}$$

Being $\mathbf{r}_1 \perp \mathbf{r}_2$, thus $\mathbf{r}_1^T\mathbf{r}_2 = 0$. We obtain

$$r_{0f} = r_{01}r_{02} = \sqrt{\frac{1+\mathbf{e}_v^T\mathbf{e}_h}{2}}\cos\frac{\phi_2}{2} \tag{7.50}$$

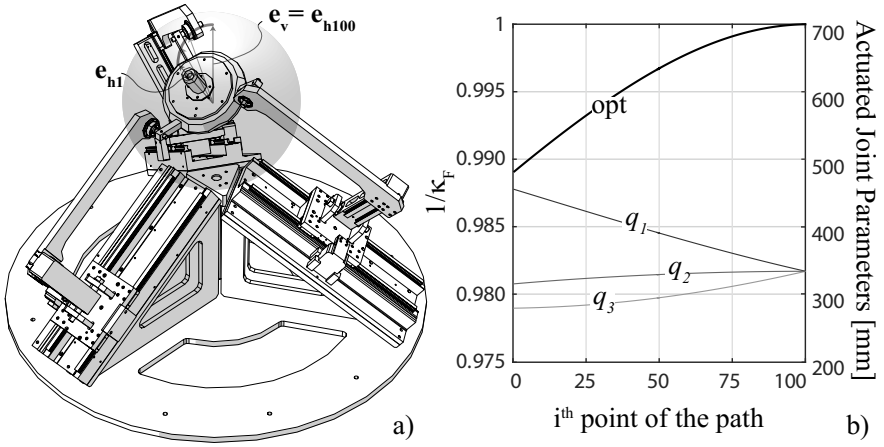


Figure 7.1: Sphe.I.Ro under trajectory-following: (a) at the initial posture; and (b) evolution of $1/\kappa_f$ (opt) along the trajectory, with the time-histories (q_i) of the joint-coordinates, for $i = 1,2,3$.

$$\begin{aligned}\mathbf{r}_f &= r_{01}\mathbf{r}_2 + r_{02}\mathbf{r}_1 + \mathbf{R}_2\mathbf{r}_1 \\ &= \sqrt{\frac{1+\mathbf{e}_v^T\mathbf{e}_h}{2}}\mathbf{e}_h \sin\frac{\phi_2}{2} + \cos\frac{\phi_2}{2}\frac{\mathbf{E}_v\mathbf{e}_h}{\|\mathbf{E}_v\mathbf{e}_h\|}\sqrt{\frac{1-\mathbf{e}_v^T\mathbf{e}_h}{2}} + \mathbf{E}_h \sin\frac{\phi_2}{2}\frac{\mathbf{E}_v\mathbf{e}_h}{\|\mathbf{E}_v\mathbf{e}_h\|}\sqrt{\frac{1-\mathbf{e}_v^T\mathbf{e}_h}{2}}\end{aligned} \tag{7.51}$$

The computation of these expressions fails when the two vectors are parallel, so there must be a special case to be considered in the optimization algorithm that is it not easy to implement. To overcome this problem the constrained optimization problem will be implemented in the following pages.

7.4 Numerical solution

The constrained optimization problem written so far with ERPs has been implemented with computer algebra and then numerically solved to find the optimum pose of the manipulator for a specific pointing direction of the mobile platform. Moreover, the SPMs that have been considered so far have no intrinsically redundant dof: to each pose it corresponds to a single configuration of the active and passive joints, i.e. a manipulator posture. Two examples show the results of the optimization of the 3-CPU performing pointing trajectories.

7.4.1 A practical case: 3-CPU Sphe.I.Ro

The 3-CPU prototype called Sphe.I.Ro has to perform a pointing task while keeping the Frobenius condition number of its Jacobian matrix at a minimum. In order to prove the effectiveness of the procedure, an arbitrary array of ERPs $\begin{bmatrix}\mathbf{r}^T & r_0\end{bmatrix}^T = \begin{bmatrix}0.140 & 0.210 & 0.280 & 0.926\end{bmatrix}^T$ is chosen to reproduce a geometric path for the EE. In practical cases instead, such path is given by the task. The geodesic rotation between the chosen orientation and the vertical vector $\mathbf{e}_v$ is taken as the pointing path of the MP, represented by $\mathbf{e}_{hi}$: a total of 100 path points on the unit sphere are used to describe the axis path, keeping fixed the axis of rotation while decreasing the angle of rotation φ by $\Delta\varphi$, as shown in Fig. 7.1a). For each prescribed direction of the hole axis an optimization problem is solved in order to find the EE orientation that minimizes the Jacobian condition number. To solve the optimization problem of a nonlinear function with nonlinear constraint equations we use the Sequential Quadratic Programming (SQP) algorithm. After having found the first point of the optimum joint trajectory, the remainder of the path follows in a similar way, using the previous trajectory point as an initial guess. The next posture is, consequently, the closest minimizer at the current posture. By doing this the result is a trajectory that keeps continuity in the series of postures. To verify the results of the SQP method, the condition number is evaluated at each iteration upon varying the angle of rotation ϑ_2 about the axis $\mathbf{e}_h$ of the unconstrained problem: the plots in Figs. 7.2a) and b) show the evaluation for a full rotation of 2π, in the middle and final point of the path. In Fig. 7.2b) the reciprocal of the condition number indicates the existence of an isotropic and a singular configuration; moreover, the plot rightfully appears as $2\pi-$ periodic. From the first results of the optimization problem we can conclude that the implementation of Eq. (7.2) together with $\|\mathbf{r}\|^2 + r_0^2 = 1$, leads to an excessive number of iterations, often without reaching a minimum. The exclusion of the unit norm equation from the system makes the dof of the functional redundancy explicit, thereby leading to a faster convergence. The comparison of the history of the condition number along the path for the optimization problem, Fig. 7.1b), vs. the unconstrained function evaluation, shows the same results; hence, the solver is capable of finding the local minimum. In Figs. 7.1 and 7.2 the histories of the actuated joints $q_i(t)$ are also plotted, for the respective condition number obtained using the inverse kinematics relationships. The pointing-path was verified looking for potential problems such as joint-limit violation and singular postures. No such problems were found for this particular path. For every practical purpose, this result shows that the optimization scheme led to a constant condition number, at a remarkably low level. It is noteworthy that the optimization problem avoids singularities only if at least one non-singular posture is available for a particular pointing direction; since this is not ensured for every pointing direction, it can happen that the solution of the problem is a pose either singular or close to a singular configuration. In any case, the proposed procedure is not aimed at seeking

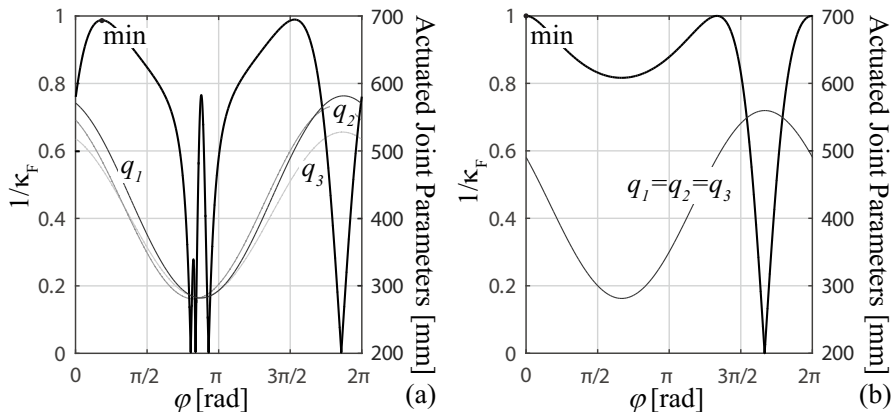


Figure 7.2: Reciprocal of the condition number and joint values (a) time-histories for the middle (50th); and (b) final (100th) trajectory point.

a singularity-free trajectory, but at optimizing a path that should be generated a priori and verified for singularity avoidance. In particular, for each path it should be guaranteed that the condition number is lower than a safe threshold.

7.4.2 2-dof Vs. 3-dof Pointing PKM

Let us consider another case. In a laboratory it is required to perform a tool pointing task and a spherical parallel manipulator is available, e.g. the Sphe.I.Ro PKM. The rotation of the tool around the symmetry axis is irrelevant for the task to be accomplished, thus the operator could evaluate, for energy-saving purposes, whether deactivate one of the three actuator or solve the optimization problem to exploit the functional redundancy. A spiral pointing trajectory is given as shown in Fig. 7.3 and the results of the numerical solution of the two problems are plotted in Fig. 7.4.

From the figure Fig. 7.4a) the reciprocal of the condition number of the functionally optimal 3-CPU has always higher values than the two-dof case. As regards the time histories of the actuated parameters Fig. 7.4b), the stroke amplitudes of the two actuators are higher than the three-dof case. The performance of the robot is clearly better due to the solution of the optimization problem since several poses for a single pointing direction are available and the algorithm is aimed at choosing the best one. Nevertheless, the improvement of performance is low: the average value of the reciprocal of the condition number along the trajectory is 0.996 vs. 0.986. The main reason is that the starting point of the trajectory is an isotropy posture of the robot and the spiral trajectory remains in a small neighborhood of such point. The improvement of performance is not too evident for this particular trajectory of the robot, thus the user can choice to lock an actuator if strict robot performance is not required.

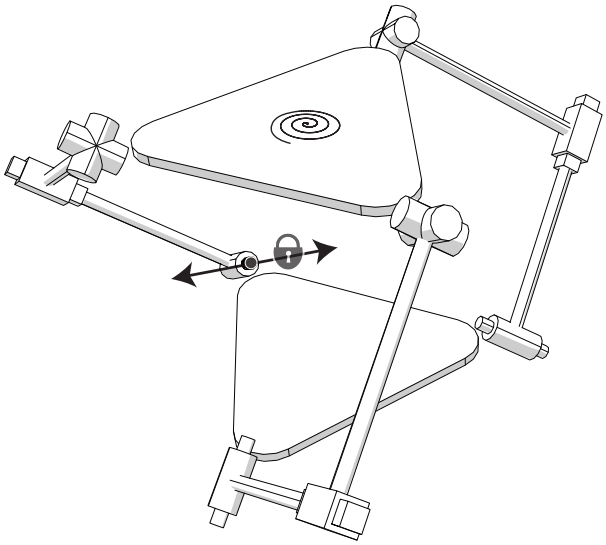


Figure 7.3: 3-CPU with a locked actuator to avoid functional redundancy under spiral trajectory-following. The video shows first the task of the 2-dof 3-CPU and then the trajectory of the functionally optimal 3-dof machine.

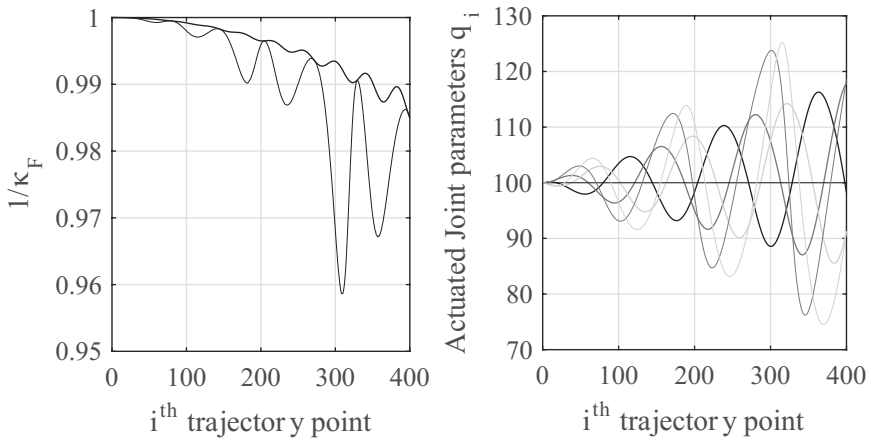


Figure 7.4: Reciprocal value of the condition number a) and joint values time-histories b) under spiral trajectory-following. Bold lines indicate the optimal 3-dof PKM while fine lines are the result for the 2-dof 3-CPU.

7.5 Extended Optimization

The optimization algorithm which exploits the functional redundancy allows the posture of the robot to be obtained with respect to the direction that maximizes the parameter taken into consideration. In this section, the algorithm is extended to each

pointing direction and the results are plotted on a spherical surface. The maps are then used for trajectory planning purposes in the next chapter.

3-CPU dexterity index

Within the infinite set of ERPs that bring $\mathbf{e}_v$ to overlay with $\mathbf{e}_h$, the solution of the problem is the one that optimally orients the EE in the $\mathbf{e}_h$ pointing direction. The optimization problem that involves the dynamic indices is then solved at each pointing direction and the results are plotted on sphere surfaces. The results obtained for the robot 3-CPU, searching the maximum value of the condition number based on the Frobenius norm of its direct Jacobian matrix, are shown in figure 7.5 on a spherical surface. Each point is associated to the robot posture that maximizes the dynamic

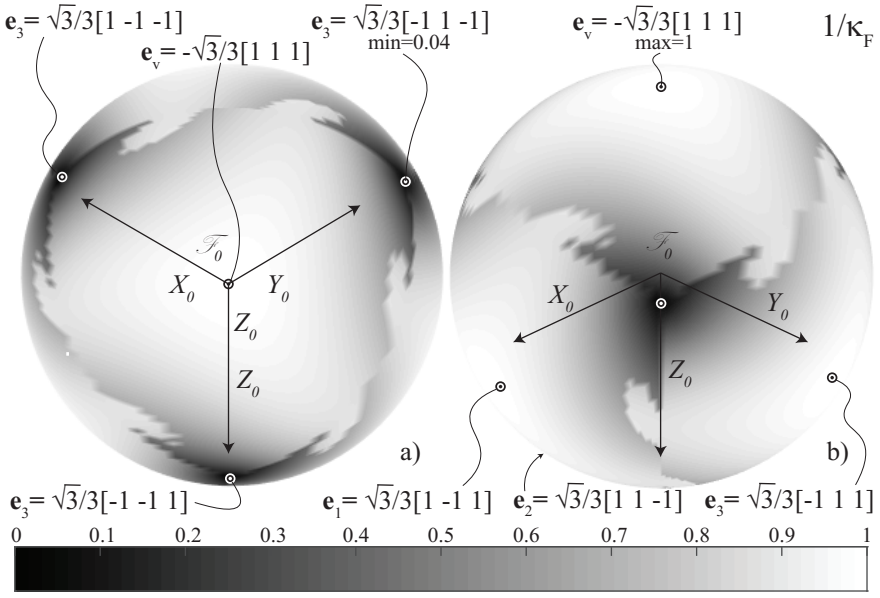


Figure 7.5: 3-CPU optimization map of the condition number values: upper view a) and isometric view b); directions of minimum and maximum values are indicated.

index in that pointing direction of the vertical axis $\mathbf{e}_v$ and the grey-scale map indicates this maximum value. First of all, four points of kinetostatic isotropy appear: the condition number attains the unitary value. These points lie on the vertices of a tetrahedron inscribed to the sphere whose coordinates are shown according to the fixed reference system. Around these points there are regions of the sphere where the condition number has a symmetrical decreasing trend, accordingly to symmetry planes coplanar with the coordinated axis. This trend decreases continuously until a limit: the reader should note that the triangular area (the one that can be seen from an

upper point of view) is very similar to the one shown in the singularity analysis. This is probably due to the limit on range of possible poses in a given pointing direction by the singularity surfaces. The discontinuity between these areas is due to the initial guess of the optimization problem: in fact, the initial guess is set to be the closest isotropic points from the actual pointing direction. This criterion was adopted because the trajectory will extend within the upper region around the initial configuration of the robot; outputs from this area are not expected because these poses are characterized by very low condition numbers. In opposition to the isotropic point, we can distinguish other four points of minimum condition number value which are very close to the singularity condition. During the planning of the 3-CPU pointing trajectory these dark areas must be avoided since the manipulator attains singular postures.

3-UPU: $\mathbf{J}_d$ Vs. $\mathbf{J}_i$ condition number

The same work has been done for the 3-UPU manipulator, that represents a more general case because it is characterized by two Jacobian matrices which relate the velocities of the actuators with the angular velocity of the mobile platform. The optimal values of condition number related to the direct Jacobian matrix $\mathbf{J}_d$ are shown in Fig. 7.6a). The results of the optimization of the inverse kinematics Jacobian matrix condition number are always very close to the isotropic condition of this matrix for all the pointing trajectory (the plot is not shown): the upper and the lower limits of the $1/\kappa_F$ range are respectively 1 and 0.975. We can conclude that the inverse kinematics Jacobian matrix has no influence for optimization purposes: as illustrated in Fig. 7.6b) the optimization of the condition number of the single Jacobian matrix $\mathbf{J}_i^{-1}\mathbf{J}_d$ gives an almost identical result with respect to the $\mathbf{J}_d$ condition number optimization. For this reason this second optimization map related to the single Jacobian $\mathbf{J}_i^{-1}\mathbf{J}_d$ will be used for the 3-UPU trajectory planning. Readers are warned that this could be a special case in the chosen class of isostatic manipulators whose inverse matrices differ very little from the isotropy condition. In the author opinion, the particular PKM at hand should reserve a special study of its Jacobian matrices when the optimization is the goal and the comparison of the two optimizations provides a good help. If the influence of the two matrices is comparable one can think also to write an optimization problem with two objective functions.

3-UPU Vs. 3-CPU

The comparison of the two optimization maps of the 3-UPU and 3-CPU (Fig. 7.7) reveals the same arrangement of areas. Isotropic points occupy the position of the vertices of the same tetrahedron and the same happens to the darker regions with a rotated tetrahedron. The main difference is that the darker regions for the 3-CPU are related to poses very close to singularity condition while for the 3-UPU the values are considerable less burdensome (0.66 Vs. 0.04). The geometric arrangements of the

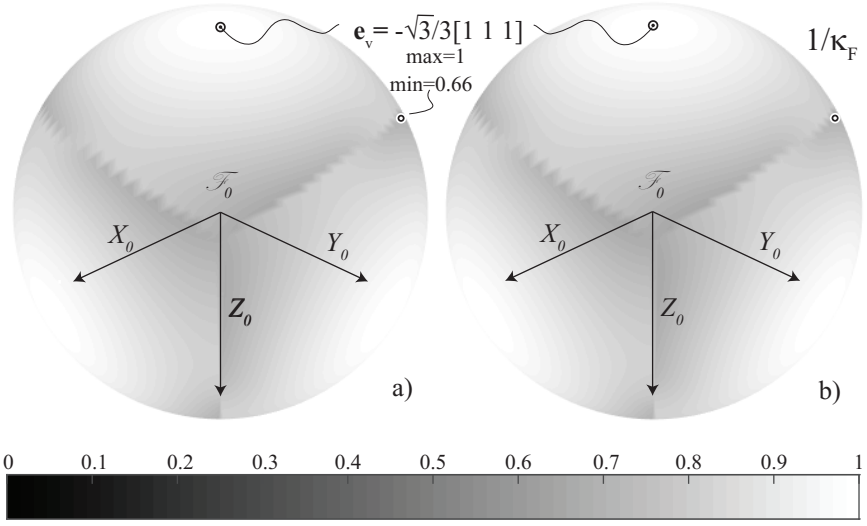


Figure 7.6: 3-UPU optimization map of the condition number values: direct Jacobian matrix $\mathbf{J}_d$ optimization a) Vs. single Jacobian matrix $\mathbf{J}_i^{-1}\mathbf{J}_d$ optimal values b).

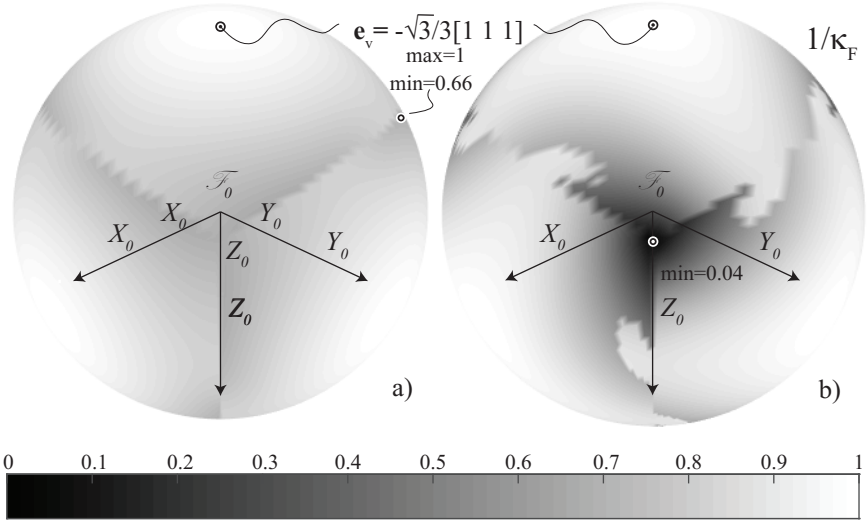


Figure 7.7: 3-UPU a) Vs. 3-CPU b) optimization maps of the Jacobian condition number.

links of the SPM class considered determines an important isotropy area for a vertical pointing direction of the MP albeit they have a different topology. This behavior is also expected for the others manipulators of the non-constrained SPM class. The

	3-CPU	3-UPU	
GCI	0.54	0.61	+13%
GCI_{opt}	0.80	0.85	+6%
	+48%	+39%	

Table 7.1: Comparison between the 3-CPU and 3-UPU GCI, both before and after functional redundancy optimization.

particular setting of the legs planes in space not only grants that its reciprocal screws are linearly independent, but also greatly improves robot performance because, even if this arrangement changes during operation of the machine, this configuration is the most far from the singular configuration, therefore granting a better kinematic manipulability of the wrist.
The calculation of the global condition number (GCI) using the ERP space allows a comparison of the performance improvements of the two architectures, both before and after the optimization. In the second case a global condition index (GCI_{opt}) is used: it takes into account only the optimal manipulator poses for any pointing direction (surface integral). The results of Table 7.1 quantify the better performance of the 3-UPU compared to the 3-CPU in both cases; however the dexterity improvement thanks to optimization for the 3-CPU is higher than 3-UPU.

3-CPU: Dynamics Performance indices

In this paragraph, the optimization involves the dynamic performance indices defined in Chapter 5. Once the inertia matrix is obtained we can also calculate its condition number based on the Frobenius norm of the matrix whose proprieties are already discussed. The optimization of this value over the sphere is shown in Fig. 7.8. The figure shows a similar trend of the optimal values with respect to the dexterity performance index. These results should not surprising the reader because both Jacobians and inertia matrix are function of the robot posture. For example in the home configuration (with a vertical $\mathbf{e}_v$) the robot assumes a symmetric posture and both condition numbers are equal to unit. A rotation from this configuration ensures the same performance, independently from the direction and the performance index considered. The condition number based on the Frobenius norm is correlated with the dynamic manipulability index d_1 according to [56]: Fig. 7.9 shows their comparison and reveals a strong analogy. Thanks to the properties of the Frobenius norm, the condition number based on this index is preferred and will be used in the trajectory planning. Instead, a comparison between d_1 and d_2 indices, that describe the robot swiftness, is illustrated in Fig. 7.10. The d_2 dynamic index has the main drawback to be dimensional, thus a different scale is required for its plot. The overall map keeps the same trend for the three indices. In conclusion, the plotted results reveal a similar trend of the dynamic indices

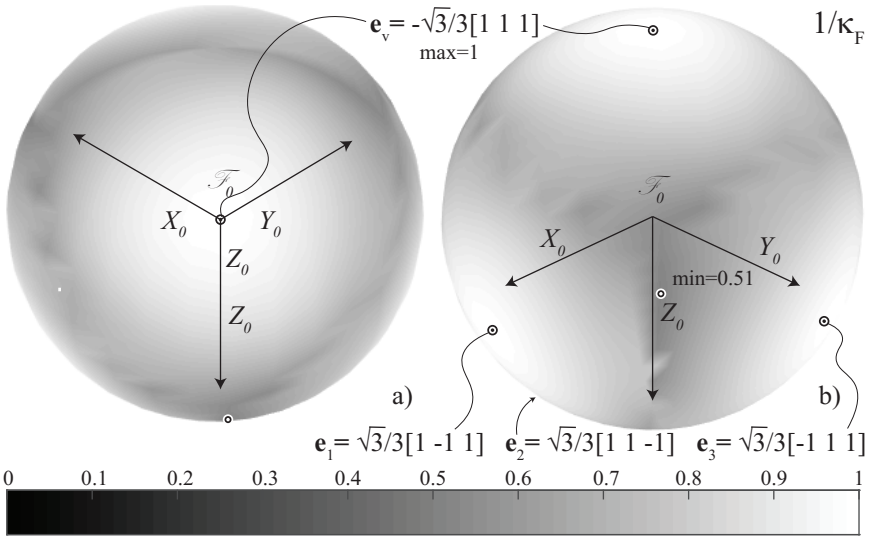


Figure 7.8: Reciprocal value of the Condition Number $1/\kappa_F$ based on the Frobenius norm for some views of the sphere: a) is an upper view while b) an isometric view.

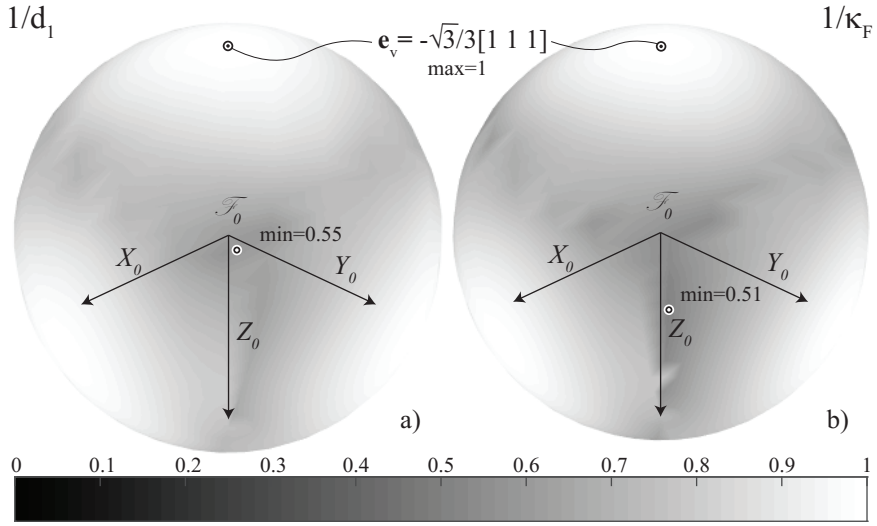


Figure 7.9: Reciprocal value of the dynamics indices of the inertia matrix; dynamic isotropy index a) and condition number based on the Frobenius norm b).

with respect to the kinematic indices and an independence from the specific topology of the robot limbs. The condition number based on the Frobenius norm of the generalized inertia matrix allows the computational time to be reduced for the search for

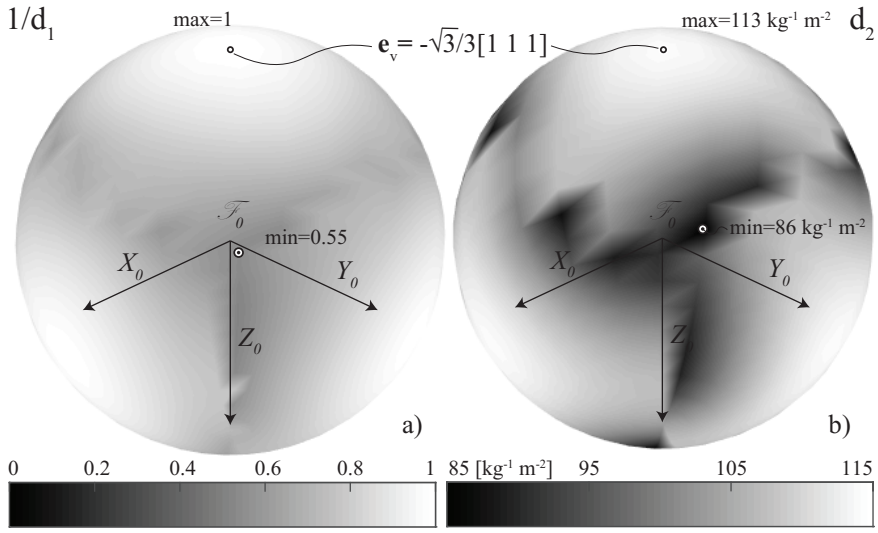


Figure 7.10: Reciprocal value of the dynamics indices of the inertia matrix. Dynamic isotropy index d_1 a) and swiftness b).

the minimum, thanks to properties of the matrix norm.

Chapter 8

Optimal Pointing Trajectory Planning

One of the robotics research target in Europe is to devise robust strategies for motion planning of multiple degrees of freedom mechanical systems. Robots need to plan the path in order to move the EE from one place to another. Complex mechanical systems may have multiple paths to reach a destination but each with a different cost: optimization strategies are called to find the best solution for complex motion planning. In this chapter, the maps obtained by solving posture optimization problems for all pointing directions are used to plan the pointing trajectories of spherical manipulators. For each direction of the trajectory, the terminal will assumes the optimum posture that is the solution of the previous problem, while the definition of the trajectory will be formalized in a second optimization problem at a higher level; it aims to minimize an index associated to the particular trajectory, such as the average value of the condition number. This type of problem may be of industrial interest when the initial and final orientation are known and a trajectory that ensures a high degree of accuracy is requested. Recent studies of Wu et al. [61] have turned to build pointing paths with Bézier curves constructed on spheres and based on the real projective plane geometry. The geometric properties of this space allow geodesic curves to be created using the exponential mapping to define the orientation of the platform. These curves of minimum distance between two points on the spherical surface are then used for the construction of Bézier curves using the De Casteljau algorithm. Recently, new applications of this tool for the optimal control problems have arisen, because the exponential coordinates are often convenient for their greater computational speed compared to other representation systems.

8.1 Bézier Curves lying on spheres

The Bézier curve is a parametric curve frequently used in computer graphics and it is also used in the robotic field for trajectory planning. A Bézier curve can be drawn using a recursive method called *De Casteljau's algorithm*. Although the algorithm is slower for most architectures when compared to the direct approach, it is more numer-

ically stable. The geometric interpretation of De Casteljau's algorithm is straightforward and can be described as follows:

1. Consider n control points of the curve $P_1^1, P_2^1, ..., P_n^1$.
2. Connect the consecutive control points to create the control polygon of the curve.
3. Multiply each line segment of this polygon by $t \in [0,1]$ and connect the points: we obtain a new polygon that has one fewer segment.
4. Repeat the process until you arrive at the single point: this is the point of the curve corresponding to the parameter t.

Fig. 8.2 shows this process for a cubic Bézier curve. The algorithm can also be ex-

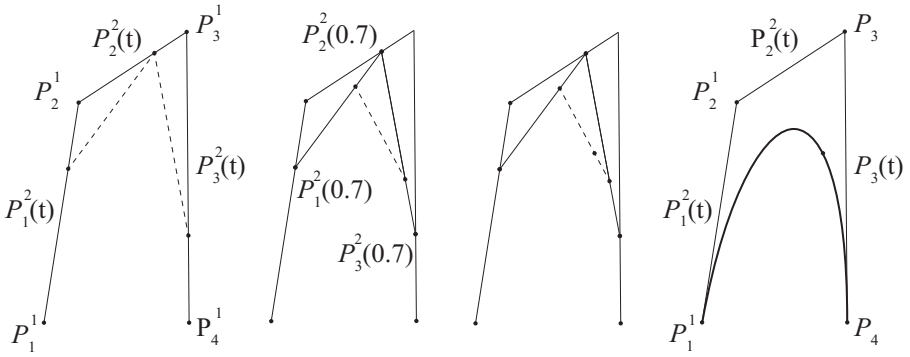


Figure 8.1: Steps of the *De Casteljau's algorithm* to determine the point of the Bézier curve corresponding to $t = 0,7$.

tended to the planning of Bézier curves which lie on a sphere. With this aim, the polygon, rather than be made by segments, will consist of geodesics, that are a generalization of the notion of straight lines to curved spaces. A geodesic between two points is the shortest arc between them on the sphere surface. The construction of the Bézier curves on a sphere is a recent topic of research and it is used in this thesis for the definition of optimal pointing trajectories.

8.1.1 Exponential Mapping

Wu et al. [61] proposed a novel construction of Bézier curves for pointing tasks, using the geometry of real projective plane. They showed that this space has interesting features, as the one to generate any geodesics using the representation of rotations with exponential mapping. The exponential representation allows to express the rotation matrix in terms of its *natural invariants*, i.e. the unit vector of the axis of rotation $\mathbf{e}$ and the angle of rotation θ. These four scalar parameters are termed *invariants*

because they are indipendent to the coordinate axes chosen to represent the rotation under study. The relatioships to pass from natural invariants to ERPs and vice versa are reported in [55]. The proper orthogonal matrix representing a rotation around an unit axis $\mathbf{e}$ with angle θ is

$$\mathbf{Q} = e^{\mathbf{E}\theta}, \quad \mathbf{E} = \mathrm{CPM}(\mathbf{e}) \tag{8.1}$$

For the inverse problem it is

$$\mathbf{e} = \frac{\mathrm{vect}\,(\mathbf{Q})}{\|\mathrm{vect}\,(\mathbf{Q})\,\|} \tag{8.2}$$

and

$$\theta = \|\mathrm{vect}(\log\mathbf{Q})\|, \quad \mathrm{vect}(\mathbf{Q}) \equiv \mathbf{q} \equiv \begin{bmatrix} q_{32} - q_{23} \\ q_{13} - q_{31} \\ q_{21} - q_{12} \end{bmatrix} \tag{8.3}$$

having defined the vector operator on a matrix. The geodesic between two orientations $\mathbf{Q}_i = e^{\mathbf{E}_i}$ and $\mathbf{Q}_f = e^{\mathbf{E}_f}$ is explicitly given by [62]

$$\mathbf{Q}_g(\mathbf{Q}_i, \mathbf{Q}_f, s) = \exp\left(s \log(\mathbf{Q}_f \mathbf{Q}_i^T)\right) \mathbf{Q}_i \qquad s \in [0,1] \tag{8.4}$$

Defining a vector of values s, the geodesic curve will be discretized and the points of the geodesic are revealed by Eq. 8.2. Eq. 8.4 allows us to apply the De Casteljau algorithm: the following treatment closely follows that in [63]. Given $n+1$ control points $\mathbf{Q}_i^1 = e^{\mathbf{E}_\mathbf{i}} \in \mathbb{R}\mathrm{P}^2$, $1 \leq i \leq n+1$ and a parameter value $t \in [0,1]$, the De Casteljau algorithm computes a point $\mathbf{Q}(s) \in \mathbb{R}\mathrm{P}^2$ on the n^{th} order Bézier curve in a recursive manner:

$$\begin{aligned} &\mathbf{Q}_c^k(s) = \mathbf{Q}_g(\mathbf{Q}_c^{k-1}(s), \mathbf{Q}_{j+1}^{k-1}(s), s) \\ &2 \leq k \leq n+1 \quad 1 \leq j \leq n+2-k \end{aligned} \tag{8.5}$$

The points of the Bézier curve can be found using Eq. 8.2 with

$$\mathbf{Q}(s) = \mathbf{Q}_1^{n+1}(s) \tag{8.6}$$

8.1.2 Natural Invariants

In this section, an alternative construction of Bézier curves of two-dimensional orientations using ERPs is formulated. There is an infinite number of rotation matrices such that $\mathbf{e}_h = \mathbf{Q}\mathbf{e}_v$, i.e. rotation matrices characterized by an axis of rotation $\mathbf{e}$ which is located in the bisector plane between the two unit vectors $\mathbf{e}_h$ and $\mathbf{e}_v$. However, the

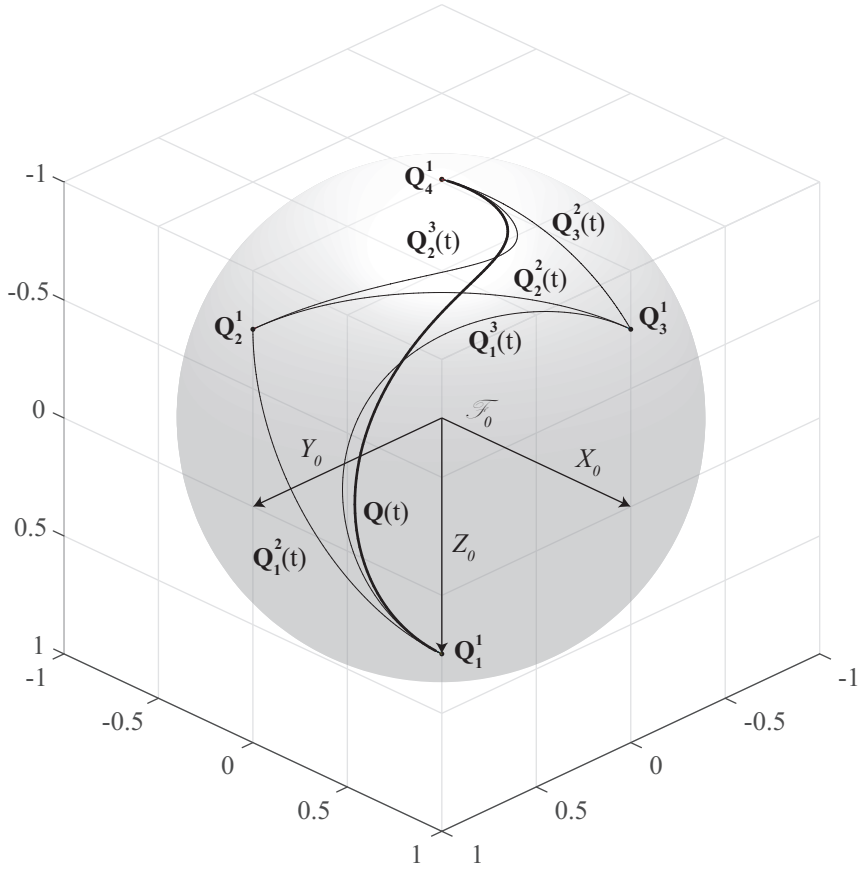


Figure 8.2: Cubic Bézier curve on $\mathbb{R}P^2$. $\mathbf{e}_1 = [0,0,1]^T$, $\mathbf{e}_2 = [-1,0,0]^T$, $\mathbf{e}_3 = [0,-1,0]^T$, $\mathbf{e}_4 = -1/\sqrt{3}[1,1,1]^T$.

only rotation matrix $\mathbf{Q}_g(\mathbf{e}_\perp, \theta_g)$ that gives a rotation with a minimum angle is

$$\mathbf{e}_\perp = \frac{\mathbf{e}_i \times \mathbf{e}_f}{\|\mathbf{e}_i \times \mathbf{e}_f\|} \tag{8.7}$$

and the angle of rotation θ associated to this rotation matrix is

$$\theta_g = \text{atan2}(\|\mathbf{e}_i \times \mathbf{e}_f\|, \mathbf{e}_i^T \mathbf{e}_f)) \tag{8.8}$$

Parameterizing the angle of rotation θ with the parameter t we can obtain all the points of the geodesic lying in a unit sphere

$$\mathbf{Q}_g(t) = \mathbf{e}_\perp \mathbf{e}_\perp^T + \cos(t\theta)(\mathbf{1} - \mathbf{e}_\perp \mathbf{e}_\perp^T) + \sin(t\theta)\mathbf{E}_\perp \tag{8.9}$$

The unit vectors that indicate the geodesic points will be

$$\mathbf{e}_g(t) = \mathbf{Q}_g(t)\mathbf{e}_i \tag{8.10}$$

This algorithm can be collected in a function called geo($\bullet$)

$$\mathbf{e}_g(t) = \text{geo}(\mathbf{e}_i, \mathbf{e}_f, t) \tag{8.11}$$

As in the previous section, given $n+1$ control points $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{n+1}$ and a parameter value $t \in [0,1]$, the De Casteljau algorithm computes a point $\mathbf{e}_B(t)$ on the n^{th} order Bézier curve:

$$\mathbf{e}_c^k(t) = \text{geo}(\mathbf{e}_c^{k-1}(t), \mathbf{e}_{j+1}^{k-1}(t), t); \quad 2 \leq k \leq n+1,\ 1 \leq j \leq n+2-k \tag{8.12}$$

The points of the Bézier curve can be found using Eq. (8.2) with

$$\mathbf{e}_B(t) = \mathbf{e}_1^{n+1}(t) \tag{8.13}$$

8.2 Optimization Problem

The construction of Bézier curves with the De Casteljau algorithm can be used for the planning of optimal trajectories. The algorithm allows to find a curve of a degree n with $n+1$ control points and, if one of them is free to move on the unit sphere surface, an optimization problem can detect its position minimizing an objective function. As an example, a typical manufacturing problem is to find the best pointing path having fixed the initial and the final point. Being a pointing trajectory, the best posture of the SPM for each direction could be the solution the posture optimization problem to exploit the functional redundancy. The optimal trajectory can be found employing the foregoing optimal maps to define an objective function. To this aim, an average index related to the trajectory is computed using the local dynamic indices already defined. Equivalently to the global condition index (GCI) defined by Angeles and Gosselin in [64], an index related to the trajectory can be defined as follows

$$\text{TCI} = \frac{\int_s \frac{1}{\kappa_F} ds}{\int_s ds} \tag{8.14}$$

where s indicates the curvilinear coordinates of the trajectory. The problem can be formulated as a constrained optimization where the constraint equation bounds the free control point $\mathbf{e}_f$ to lie on a sphere with unit radius. Then the problem takes the form

$$f(\mathbf{e}_f) \equiv \frac{1}{\text{TCI}} \rightarrow \min_{\mathbf{x}},\ \text{s.t. } \mathbf{e}_f^T \mathbf{e}_f = 1 \tag{8.15}$$

where the objective function to maximize is the TCI.

8.3 Problem solution

A numerical solution algorithm is applied to the optimization problem to find the direction of the free control point. Three types of problems have been implemented and solved to determine the optimal Bézier curve: in all of them the initial and the final pointing directions are fixed.

- In the first problem, the intermediate control point that optimizes the TCI of a quadratic Bézier curve is found.
- In the second problem, the angular velocity of the final control point is given and a cubic Bézier curve is obtained.
- The last problem considers a 4^{th}-degree Bézier curve where both angular velocity and angular acceleration of the first point are given. The algorithm detects the fourth control point.

The figures are accompanied by videos that prove the feasibility of the trajectory.

8.3.1 Quadratic Bézier curve: two-points boundary value problem

As a first example of trajectory planning, a quadratic Bézier curve is chosen. It is uniquely defined by three control points $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$. The initial and the final pointing directions are fixed, while the intermediate point $\mathbf{e}_2$ represents the degree of freedom for the optimization problem. Moreover, this free point is bound to lie in the unit sphere surface $\mathbf{e}_2^T\mathbf{e}_2 = 1$ and the objective function is the TCI based on the foregoing indices. The optimization problem subject to this constraint equation is solved to find the optimal direction $\mathbf{e}_{2opt}$: Fig. 8.3 compares the results for the 3-CPU and the 3-UPU using the dexterity index based on the Frobenius norm. From this figure one can see that the optimal TCI attains a higher value for the 3-UPU PKM than the 3-CPU because the starting point is close to a singularity condition for the 3-CPU manipulator. Moreover the trajectory of the 3-UPU is more flattened to the origin since the 3-CPU has a more pronounced triangular symmetry of the gray-scale map. Moreover, in both cases the trajectory is flattened to the origin, but without reaching the final direction with a minimum distance trajectory; this because the highest average value of the dexterity index is obtained with motions around the point of isotropy than the geodesic path. This is due to the definition of the objective function that does not depend on the length of the trajectory.

With the aim of verifying the optimal pointing trajectory with the dynamic index, an inverse dynamic simulation is performed with a multi-body software. The simulation

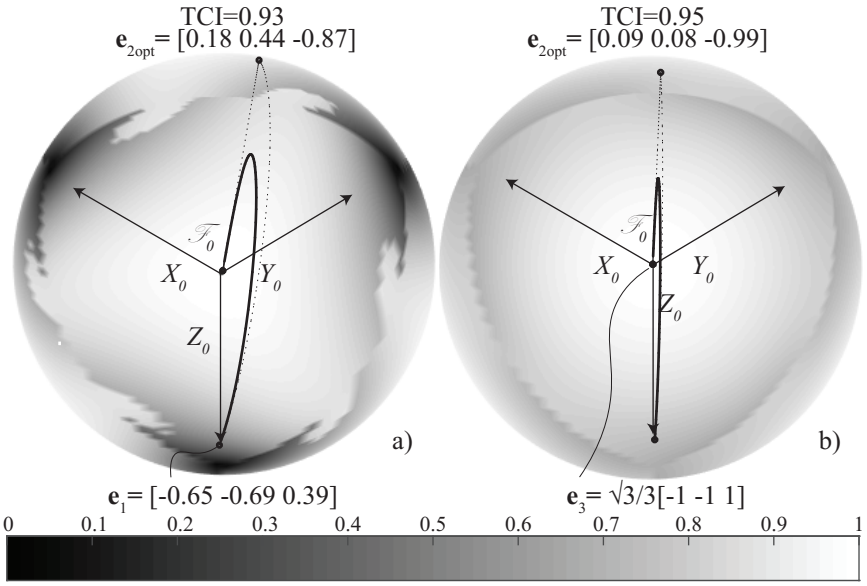


Figure 8.3: Optimal quadratic Bézier curve: 3-CPU a) Vs. 3-UPU b) dexterity index.

allows us to obtain the actuators forces time-histories, once motion law is defined; in fact, up to now the path does not contain time information. A symmetrical-triangular velocity law is imposed to the optimal trajectory and to the geodetic path to reach the final trajectory point in 2 seconds, as shown in Fig. 8.4a). The root mean square values of the force time-history for the geodetic is $F_g^{rms} = 5.63,\mathrm{N}$ against $F_{opt}^{rms} = 5.33\,\mathrm{N}$ for the optimal trajectory. Obviously, a higher acceleration is imposed to the optimum path, however, the mean force decreases (-0.53%). In general, this is not always true: the trajectory determines the directions of the motion, while the index follows way-points that overall give better acceleration when equal forces are given, regardless of the direction of motion. For this reason a trajectory index which takes into account this fact would be more appropriate, despite the optimization problem will assumes a greater complexity.

Fig. 8.5 illustrates the differences for the 3-CPU between the use of kinematic and dynamic performance indices, both based on the Frobenius norm. The trajectory associated to the 3-CPU dynamic index guarantees better performance than the dexterity index.

8.3.2 Cubic Bézier curve: two-points and one-velocity boundary value problem

This second problem uses a cubic Bézier curve which is defined with four control points. In addition to the definition of the initial $\mathbf{e}_1$ and final directions $\mathbf{e}_4$ of the

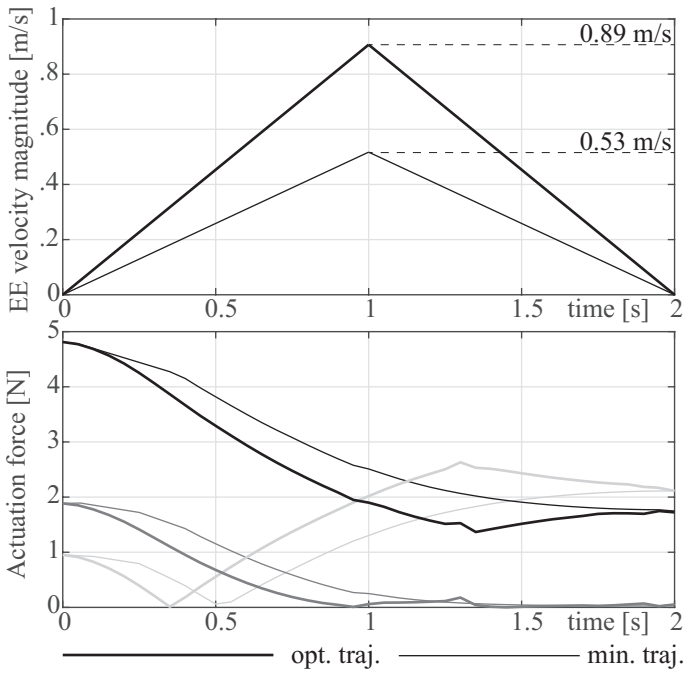


Figure 8.4: Inverse dynamics simulations of the geodesic and the optimal trajectories. Symmetric-triangular velocity profiles on the top and the actuator force time histories on the bottom.

trajectory, the final geometrical angular velocity vector $\boldsymbol{\omega}_{34}$ is provided. It defines the initial tangent to the trajectory. This geometrical angular velocity defines the position of the third control point $\mathbf{e}_3$ that can be computed using Eq. 8.2 once $\mathbf{Q}_3^1$ is calculated from

$$\mathbf{Q}_3^1 = \exp(\mathrm{CPM}(\exp(-\boldsymbol{\Omega}_{34}\mathbf{e}_4)), \quad \boldsymbol{\Omega}_{34} = \mathrm{CPM}(\boldsymbol{\omega}_{34})/3 \tag{8.16}$$

Otherwise, if the initial angular velocity vector is defined instead of the final one, the second control point could be derived from the same Eq.s 8.16 but omitting the minus sign. The data of the problem are shown in Fig. 8.6 together with the optimal control point, i.e. the pointing direction $\mathbf{e}_{2opt}$ that maximizes the TCI of the cubic Bézier curve. In this case, the geodesic between the initial and the final trajectory points is far away from the isotropic configuration. Moreover, the final velocity obliges the curve to be tangent to a direction that does not indicate that point. However, the solutions shown in Fig. 8.6 are trajectories that pass close to the kinematic isotropic configuration, both for the 3-CPU and the 3-UPU. Fig. 8.7 shows a comparison between the use of kinetostatic and dynamic optimum maps of the 3-CPU PKM.

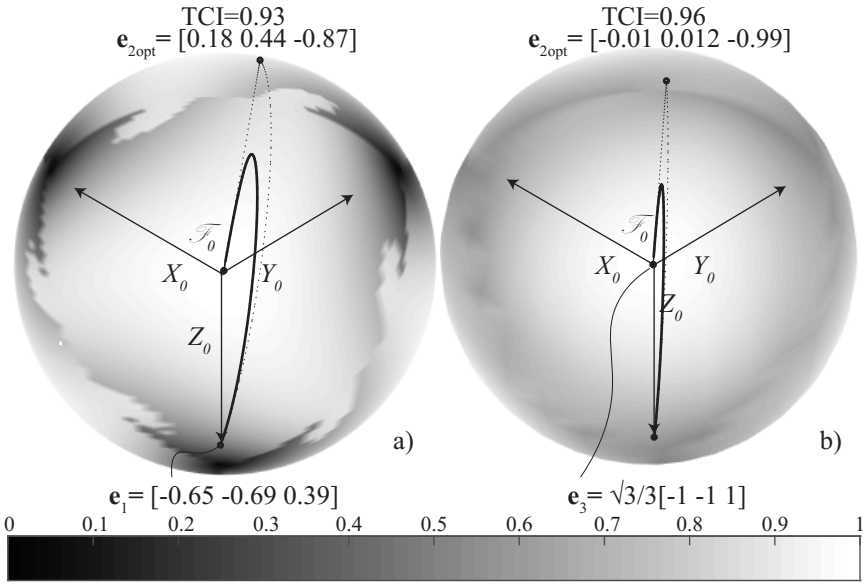


Figure 8.5: Optimal quadratic Bézier curve: 3-CPU dexterity index a) and dynamic condition number b).

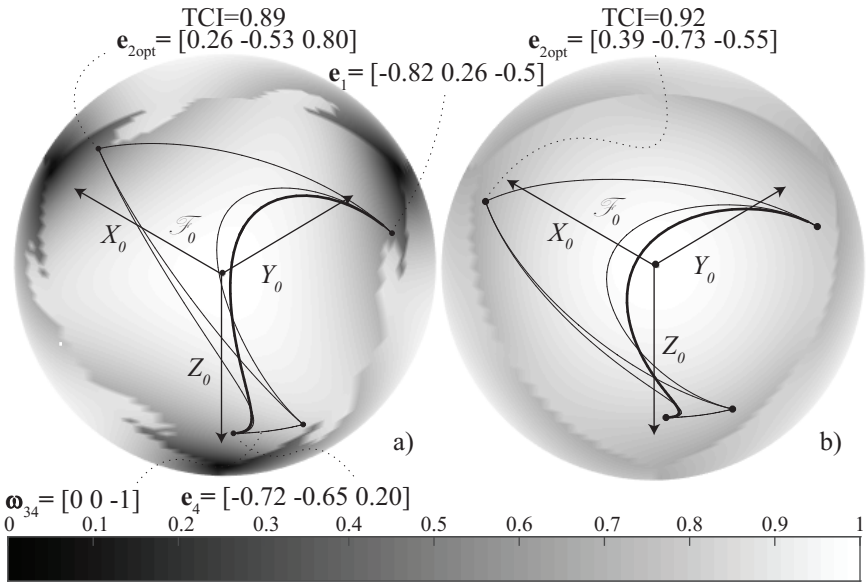


Figure 8.6: Optimal cubic Bézier curve: 3-CPU a) Vs. 3-UPU b) dexterity index.

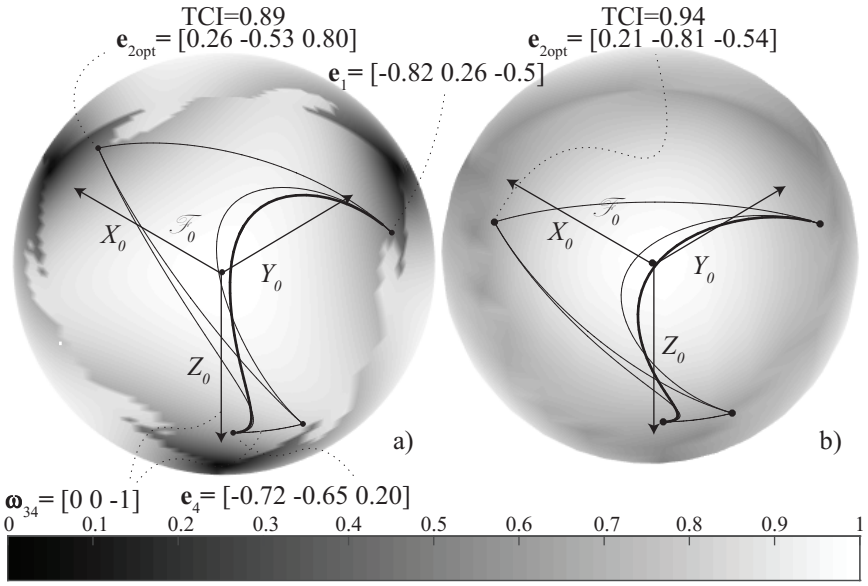


Figure 8.7: Optimal cubic Bézier curve: 3-CPU dexterity index a) and dynamic condition number b).

8.3.3 4^{th}-degree Bézier curve: two-points, one-velocity and one-acceleration boundary value problem

The fourth control point of a 4^{th}-degree Bézier curve is determined given the initial and final pointing directions of the path ($\mathbf{e}_1$ and $\mathbf{e}_5$), the initial angular velocity vector $\boldsymbol{\omega}_{12}$, which defines the second control points $\mathbf{e}_2$ and the initial geometrical angular acceleration $\boldsymbol{\alpha}_0$, that gives us information on the initial curvature of the path and defines the third control point $\mathbf{e}_4$. According to Eq. 8.16 the unit vector $\mathbf{e}_2$ is computed, while $\mathbf{e}_3$ comes from

$$\boldsymbol{\omega}_{22} = \boldsymbol{\omega}_{12} + \frac{1}{6}\int_0^1 \exp(u\boldsymbol{\omega}_{12})\mathrm{CPM}(\boldsymbol{\alpha}_{12})du \tag{8.17}$$

$$\mathbf{Q}_3^1 = \exp(\mathrm{CPM}(\boldsymbol{\omega}_{22}))\mathbf{Q}_2^1 \tag{8.18}$$

Fig.s 8.7 and 8.9 show a comparison between the case studies. Whatever is the map or the employed manipulator, the final trajectories are very similar each other, because less freedom is left to the trajectory than the previous problems.

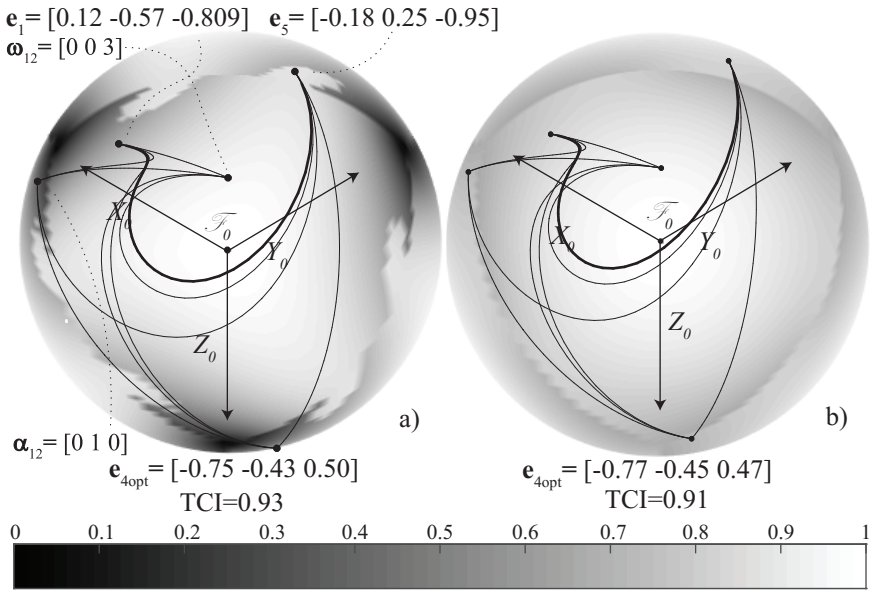


Figure 8.8: Optimal 4^{th}-degree Bézier curve: 3-CPU a) Vs. 3-UPU b) dexterity index.

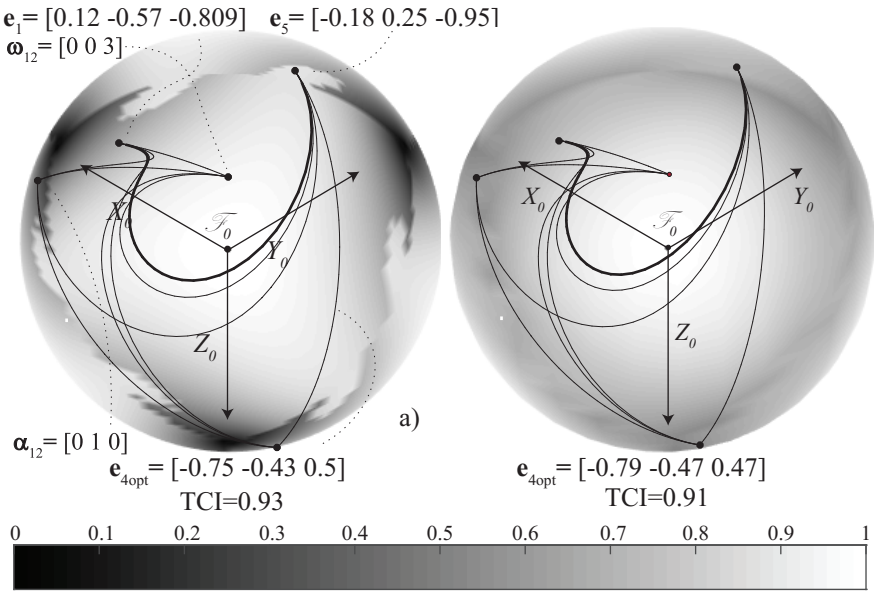


Figure 8.9: Optimal 4^{th}-degree Bézier curve: 3-CPU dexterity index a) and dynamic condition number b).

Chapter 9

Concluding remarks and future works

The research project aims to take advantage of the functional redundancy between a parallel manipulator and the task at hand. This situation arises in the industry when a generic manipulator with six-dof is chosen to ensure flexibility and the task requires the use of an axially symmetric tool. A particular case study is taken into account where a six-dof parallel robot is decomposed into two lower mobility machines. Since the functional redundancy belongs to the spherical parallel machine, a class of these robots is considered and studied in order to show the advantages of a special arrangement of the limb planes that maximizes the kinetostatic and dynamic performance. However, the thesis presents analysis methods which can be applied to any parallel machine:

1. A comprehensive and general redundancy analysis is presented to quantify the degrees of kinematic redundancy: the method is valid for both serial and parallel kinematics machines.

2. A general procedure is developed to identify kinetostatic and dynamic performance indices by means of the *screw algebra*.

3. Workspace analysis of the 3-CPU is addressed to represent the singularity surfaces in the *Euler Rodrigues Parameters* space.

4. Formulation and numerical solution of the posture optimization problem for functionally redundant PKMs. The problem is extended to any pointing direction of the MP to create maps of optimum postures.

5. Pointing trajectory planning with Bézier curves using optimum maps: exponential mapping is use to ensure computational efficiency.

The optimization problem could be extended to a general case of a six-dof parallel manipulator using the orientation kinematics that has been developed and introducing that of positioning, while a more complex expression will result for the chosen indices. The generalization of the method might involve the Stewart-Gough platform

already introduced that represents a widespread example of parallel machine for manufacturing.

The optimization problem does not take into account the physical limitations of the joints and this is a limit of the method. The planned trajectory is maintained within the area near the vertical isotropic point that assures best manipulator's performance. A possible choice of the variation range of active and passive joints in the design stage may be made so as to ensure this workspace area. For the practical case of Sphe.I.Ro the verification of the joint limit space are carried downstream of the optimization problem, however these limits should be embedded in the constraint equations at the expense of a greater computation time. The condition for singularities avoidance is instead inherent in the objective function of the kinematic dexterity index, so it can be avoided.

Another future development of this work could involve the validation of results. Although the PKM prototypes I.Ca.Ro and Sphe.I.Ro are available, their accidental failure did not allow the implementation of the resulting trajectories. This gap was partly offset by kinematic simulations that show a continuity in the postures assumed by the manipulator for different pointing directions of the MP. For this reason, the author believes that the implementation of the trajectory, having solved the off-line problem is feasible. The reader might think to implement the optimal posture maps in the control algorithm so that the robot knows in advance which postures it had to assume for a particular orientation of the terminal.

In trajectory planning the solution is usually distant from the one that minimizes the distance of the trajectory since the objective function aims to the only optimization of the condition number average value. An idea would be to combine more objective functions in order that the optimization problem could find a good compromise solution.

Another fact to note is that this dissertation does not elaborate on the topic of the solution algorithm but uses the SQP algorithm as a closed box. A future work could cover this study. Furthermore, the author has attempted a solution in closed form of the problem but without success due to the non-linearity equations of the optimization problem. However, qualitatively the author observed that the optimization problem guarantees excellent computational performance, indicating that the numerical solution is a great approach.

This work, in addition to the trajectory planning, could be employed for the functional design of manipulators. The kineto-static indices of Sphe.I.Ro do not depend on the geometric parameters but this represents a special case. In general, these parameters are used as unknowns in a optimization problems to lead the optimal design of machines. Furthermore, the average value of the condition number of the functional redundant optimization could guide the choice of a machine rather than another as a catalog parameter.

Bibliography

[1] L. Huo and L. Baron, "Kinematic inversion of functionally-redundant serial manipulators: application to arc-welding," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 29, no. 4, pp. 679–690, 2005.

[2] J. Léger and J. Angeles, "Off-line programming of six-axis robots for optimum five-dimensional tasks," *Mechanism and Machine Theory*, vol. 100, pp. 155–169, 2016.

[3] R. S. Ball, "The theory of screws: A study in the dynamics of a rigid body," *Mathematische Annalen*, vol. 9, no. 4, pp. 541–553, 1876.

[4] K. H. Hunt, *Kinematic geometry of mechanisms*. Oxford University Press, USA, 1978, vol. 7.

[5] J.-P. Merlet, *Parallel robots*. Springer, 2012.

[6] I. Fassi and G. J. Wiens, "Multiaxis machining: Pkms and traditional machining centers," *Journal of Manufacturing Processes*, vol. 2, no. 1, pp. 1–14, 2000.

[7] M. S. Narayanan, S. Chakravarty, H. Shah, and V. N. Krovi, "Kinematic-, static- and workspace analysis of a 6-pus parallel manipulator," in *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME, 2010, pp. 1456–1456.

[8] I. A. Bonev, *Geometric analysis of parallel mechanisms*. Ph.D. Thesis, 2002.

[9] I. Bonev, "The true origins of parallel robots," *ParalleMIC*, 2003.

[10] J. Gallardo-Alvarado, *Kinematic analysis of parallel manipulators by algebraic screw theory*. Springer, 2016.

[11] J. Tindale, "Discussion on the stewart paper," *Proc. Instn. Mech. Engrs*, vol. 180, no. 15 Pt 1, pp. 383–384, 1965.

[12] G. Pritschow, "Research and development in the field of parallel kinematic systems in europe," in *Parallel Kinematic Machines*. Springer, 1999, pp. 3–15.

[13] L. Tunc and J. Shaw, "Investigation of the effects of stewart platform-type industrial robot on stability of robotic milling," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 1-4, pp. 189–199, 2016.

[14] L. T. Tunc and J. Shaw, "Experimental study on investigation of dynamics of hexapod robot for mobile machining," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 5-8, pp. 817–830, 2016.

[15] P. Threadgill, A. Leonard, H. Shercliff, and P. Withers, "Friction stir welding of aluminium alloys," *International Materials Reviews*, 2013.

[16] R. S. Mishra and Z. Ma, "Friction stir welding and processing," *Materials Science and Engineering: R: Reports*, vol. 50, no. 1, pp. 1–78, 2005.

[17] W. Thomas and E. Nicholas, "Friction stir welding for the transportation industries," *Materials & Design*, vol. 18, no. 4, pp. 269–273, 1997.

[18] C. B. Smith, "Robotic friction stir welding using a standard industrial robot," in *2nd International Friction Stir Welding Symposium*, 2000.

[19] M. Callegari, A. Forcellese, M. Palpacelli, and M. Simoncini, "Robotic friction stir welding of aa5754 aluminum alloy sheets at different initial temper states," in *Key Engineering Materials*, vol. 622. Trans Tech Publ, 2014, pp. 540–547.

[20] J. Shi, Y. Wang, G. Zhang, and H. Ding, "Optimal design of 3-dof pkm module for friction stir welding," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9-12, pp. 1879–1889, 2013.

[21] A. W. Momber and R. Kovacevic, *Principles of abrasive water jet machining*. Springer Science & Business Media, 2012.

[22] M. Carricato and D. Zlatanov, "Persistent screw systems," *Mechanism and Machine Theory*, vol. 73, pp. 296–313, 2014.

[23] X. Kong and C. Gosselin, "Type synthesis of three degree of freedom spherical parallel manipulators," *The International Journal of Robotics Research*, vol. 23, no. 3, pp. 237–245, 2004.

[24] L. Carbonari, M. Callegari, G. Palmieri, and M.-C. Palpacelli, "A new class of reconfigurable parallel kinematic machines," *Mechanism and Machine Theory*, vol. 79, pp. 173–183, 2014.

[25] L.-W. Tsai and S. Joshi, "Kinematic analysis of 3-dof position mechanisms for use in hybrid kinematic machines," *Journal of Mechanical Design*, vol. 124, no. 2, pp. 245–253, 2002.

[26] M. Callegari and M.-C. Palpacelli, "Prototype design of a translating parallel robot," *Meccanica*, vol. 43, no. 2, pp. 133–151, 2008.

[27] L. Carbonari, M. Battistelli, M. Callegari, and M. Palpacelli, "Dynamic modelling of a 3-cpu parallel robot via screw theory," *Mech. Sci*, vol. 4, no. 1, pp. 185–197, 2013.

[28] M. Callegari, L. Carbonari, G. Palmieri, and M.-C. Palpacelli, "Parallel wrists for enhancing grasping performance," in *Grasping in Robotics*. Springer, 2013, pp. 189–219.

[29] L. Carbonari and M. Callegari, "The kinematotropic 3-cpu parallel robot: analysis of mobility and reconfigurability aspects," in *Latest Advances in Robot Kinematics*. Springer, 2012, pp. 373–380.

[30] M. Karouia and J. M. Hervé, "A family of novel orientational 3-dof parallel robots," in *Symposium on Robot Design, Dynamics, and Control*. Springer, 2002, pp. 359–368.

[31] Q. Li and Z. Huang, "A family of symmetrical lower-mobility parallel mechanisms with spherical and parallel subchains," *Journal of Robotic Systems*, vol. 20, no. 6, pp. 297–305, 2003.

[32] D. J. Cox and D. Tesar, "The dynamic model of a three degree of freedom parallel robotic shoulder module," in *Advanced Robotics*. Springer, 1989, pp. 475–487.

[33] W. M. Craver, *Structural analysis and design of a three-degree-of-freedom robotic shoulder module*, Ph.D. Thesis, 1989.

[34] C. Gosselin and J. Angeles, "The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator," *Journal of mechanisms, transmissions, and automation in design*, vol. 111, no. 2, pp. 202–207, 1989.

[35] P. Vischer and R. Clavel, "Argos: a novel 3-dof parallel wrist mechanism," *The International Journal of Robotics Research*, vol. 19, no. 1, pp. 5–11, 2000.

[36] R. Di Gregorio, "Kinematics of a new spherical parallel manipulator with three equal legs: The 3-urc wrist," *Journal of Robotic Systems*, vol. 18, no. 5, pp. 213–219, 2001.

[37] J. M. Hervé and M. Karouia, "The novel 3-ruu wrist with no idle pair," in *Proceedings of the workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, 2002, pp. 3–4.

[38] J. M. Hervé, "The planar-spherical kinematic bond: implementation in parallel mechanisms," *online at http://www.parallemic.org/Reviews/review013.html*, 2003.

[39] R. Di Gregorio, "A new parallel wrist using only revolute pairs: the 3-ruu wrist," *Robotica*, vol. 19, no. 03, pp. 305–309, 2001.

[40] K. Al-Widyan, B. Monsarrat, and J. Angeles, "The robust analysis and design of parallel spherical manipulators," in *Proceedings of Dresden Symposium Geometry*, 2003, pp. 23–32.

[41] M. Callegari, *Design and prototyping of a spherical parallel machine based on 3-CPU kinematics*. International Conference on Innovative Technologies, 2008.

[42] M. Karouia and J. Hervé, "A three-dof tripod for generating spherical rotation," in *Advances in robot kinematics*. Springer, 2000, pp. 395–402.

[43] J. M. Selig, *Geometric fundamentals of robotics*. Springer, 2004.

[44] D. R. Walter, M. L. Husty, and M. Pfurner, "A complete kinematic analysis of the snu 3-upu parallel robot," *Contemporary Mathematics*, vol. 496, p. 331, 2009.

[45] D. A. Cox, J. Little, and D. O'Shea, *Using algebraic geometry*. Springer, 2006, vol. 185.

[46] X. Kong and M. Pfurner, "Type synthesis and reconfiguration analysis of a class of variable-dof single-loop mechanisms," *Mechanism and Machine Theory*, vol. 85, pp. 116–128, 2015.

[47] M. Mohamed and J. Duffy, "A direct determination of the instantaneous kinematics of fully parallel robot manipulators," *Journal of mechanisms, transmissions, and automation in design*, vol. 107, no. 2, pp. 226–229, 1985.

[48] L.-W. Tsai, "The jacobian analysis of a parallel manipulator using reciprocal screws," in *Advances in Robot Kinematics: Analysis and Control*. Springer, 1998, pp. 327–336.

[49] D. Zlatanov, R. Fenton, and B. Benhabib, "A unifying framework for classification and interpretation of mechanism singularities," *ASME Journal of Mechanical Design*, vol. 117, pp. 566–572, 1995.

[50] M. Conconi and M. Carricato, "A new assessment of singularities of parallel kinematic chains," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 757–770, 2009.

[51] J.-P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006.

[52] W. A. Khan and J. Angeles, "The kinetostatic optimization of robotic manipulators: the inverse and the direct problems," *Journal of mechanical design*, vol. 128, no. 1, pp. 168–178, 2006.

[53] L. Brand, *Advanced calculus: an introduction to classical analysis*. Courier Corporation, 2013.

[54] C. Gosselin, "Kinematic analysis, optimization and programming of parallel robotic manipulators," Ph.D. Thesis, 1988.

[55] J. Angeles, *Fundamentals of robotic mechanical systems*. Springer, 2002.

[56] R. Di Gregorio and V. Parenti-Castelli, "Dynamic performance indices for 3-dof parallel manipulators," in *Advances in Robot Kinematics*. Springer, 2002, pp. 11–20.

[57] T. Yoshikawa, "Dynamic manipulability of robot manipulators," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 1033–1038.

[58] O. Ma and J. Angeles, "Optimum design of manipulators under dynamic isotropy conditions," in *Innovation, Entrepreneurship, and Real-world Solutions (ICRA)*, 1993, pp. 470–475.

[59] L. Carbonari, M. Callegari, G. Palmieri, and M. Palpacelli, "Analysis of kinematics and reconfigurability of a spherical parallel manipulator," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1541–1547, 2014.

[60] G. Taubin, "Graphics math & code editor: 3d rotations," 2011.

[61] Y. Wu, M. Andreas, and M. Carricato, "The 2d orientation interpolation problem: A symmetric space approach," in *Advances in Robot Kinematics (ARK)*, 2016.

[62] F. Park and B. Ravani, "Bezier curves on riemannian manifolds and lie groups with kinematics applications," *Journal of Mechanical Design*, vol. 117, no. 1, pp. 36–40, 1995.

[63] P. Crouch, G. Kun, and F. S. Leite, "The de casteljau algorithm on lie groups and spheres," *Journal of Dynamical and Control Systems*, vol. 5, no. 3, pp. 397–429, 1999.

[64] J. Angeles and C. Gosselin, "A global performance index for the kinematic optimization of robotic manipulators," *Journal of Mechanical Design*, vol. 113, no. 3, pp. 220–226, 1991.