Novel Energy-Optimal Solutions to the Inverse Kinematics Problem of Redundant Manipulators

By

Alessandro Tringali

A thesis submitted to the University of Strathclyde

for the degree of

Doctor of Philosophy

Department of Design, Manufacturing and Engineering Management

University of Strathclyde

Glasgow, UK

August 2020

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.'

Signed: Date: 23/02/2021

Abstract

Robotic manipulators are today used in many industrial and field applications, especially in the kinematically redundant ones, which offer the possibility to approach a specific end-effector pose in infinitive different ways. This freedom allows to optimise secondary task alongside the main goal to drive the end-effector through a specific trajectory, and such optimisation problem has motivated the work of many researchers in the last decades. Historically, the energy of the manipulator has been considered a particularly important optimisation cost function: it is relevant in industrial settings, where manipulators operations are a standing cost, and even more in field environments, such as space, where the available power is limited.

This thesis presents a study about inverse kinematics algorithms for redundant manipulators, aimed at optimising the energy required to perform manipulation tasks. First, a literature review surveying inverse kinematics and optimisation for both fixed-base and free-floating manipulators is presented. This presents the state of the art in the field and illustrates the motivation for this thesis. It also outlines the main challenges encountered in the development of optimisation algorithms for redundant manipulators.

After this, two algorithms are presented and discussed within the thesis, a global and a local one. Both are based on nonlinear optimisation techniques. The global problem is talked first, and a method is proposed that can optimise different cost functions related to either kinetic energy or torques, with linear and nonlinear constraints, such as torque, power, and periodic motion. Furthermore, the algorithm is able to individuate multiple optima when they are present, thus increasing the chances to find the best (global) optimum.

A local algorithm based on prediction of kinetic energy integral has also been developed. In order to illustrate related challenges, a workspace analysis is first presented that illustrates difficulties in providing reliable prediction of kinetic energy values along a specific end effector trajectory. Kinematic indexes are discussed through a qualitative and quantitative analysis aimed at assessing their correlation with kinetic energy, and results of a canonical correlation analysis is presented. Furthermore, it is illustrated that the a spaceborne robotic manipulator can be controlled concurrently with the Attitude and Orbit Control Systems of spacecraft, adding extra degrees of freedom.

Following this, a local algorithm based on a predictive estimation of kinetic energy integral along a specified trajectory is presented and discussed. This algorithm is based on a simplified optimisation problem that allows to assess the direction of motion that will cause the smallest increase in the kinetic energy integral. This produces solutions that are closer to the global optimum respect to traditional algorithms.

Simulations with a 3-DoF planar robot are used to validate the results. The global method is validated against a global algorithm existing in literature and shown to be able to solve a wider class of problems. The local algorithm is statistically compared against existing inverse kinematics methods, showing an reduction in kinetic energy up to 30%. The thesis is completed by a discussion about limits and further improvements of the work hereby presented.

Acknowledgements

Many people have shared some of my journey towards the completion of this thesis. First, I'd like to express my gratitude to my first supervisor Prof. Xiu-Tian Yan for allowing me to become a very autonomous and independent researcher, and my second supervisor Dr. Erfu Yang for the friendly support and for helping me especially in the final steps. Special thanks should go to Dr. Silvio Cocuzza, who has been both a mentor and a friend in many occasions, at times believing in me more than I did myself. Other academics to whom I am grateful for their advice are Prof. Bongiorno in Paris, Edmondo, Annalisa, Karen in Glasgow, and my old friend Luca in London.

My PhD colleague and flatmate Scott has shared with me most of the journey and has become one of my the closest friends in the process. This would have been much harder without him, and I am very grateful for his company.

I would also like to thank all of my Scottish, Italian, Serbian friends in Glasgow, who made my time in Scotland extremely enjoyable. Most of my PhD stress has been released through healthy doses of sailing, I'd like to thank the University of Strathclyde Sailing Club and everyone who sailed with me, especially Murray and Colin. Thanks for being on those boats with me. Special thanks should also go to my friends from home, especially Giuseppe, Giuseppe and Michela, and Marco (x2).

Finally, I would like to thank my family for supporting me through all this journey, and last but not least my partner Daniela. Her support has been invaluable every single second of the last few years, and this thesis is dedicated to her.

Alessandro

Oxford, August 8th 2020

Table of contents

List of abbreviations

ACS	Attitude Control System
Cond. Number	Condition Number
CSA	Canadian Space Agency
DART	Demonstration for Autonomous Rendezvous Technology
DARPA	Defense Advanced Research Projects Agency
DEOS	Deutsche Orbital Servicing
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Space Agency)
DM	Disturbance Map
DOF	Degrees-of-freedom
Dyn. Man.	Dynamic Manipulability
EDM	Enhanced Disturbance Map
EE	End-Effector
EF	Exposed Facility
ERA	European Robotic Arm
ESA	European Space Agency
EVA	Extra-Vehicular Activity
FREND	Front-end Robotics Enabling Near-term Demonstration
GA	Genetic Algorithm
GEO	Geostationary Orbit
GKP	Global Kinematic Planner
IBA	Inspection Boom Assembly
IBGKP	Interpolation-Based Global Kinematic Planner
IK	Inverse Kinematics
ISS	International Space Agency
JAXA	Japanese Space Agency
JEM	Japanese Experiment Module
JEMRMS	Japanese Experiment Module Remote Manipulator System

Kin.	Kinetic
LEO	Low Earth Orbit
LMKE	Local Minimisation of Kinetic Energy
LQR	Linear Quadratic Regulator
LSV	Least-Squares minimisation of Velocity
LSE	Least-Squares with Equality constraints
Man.	Manipulability
MPC	Model Predictive Control
MS	Multi-Start
NASA	National Aeronautics and Space Administration
NLP	Nonlinear Programming
NMPC	Nonlinear Model Predictive Control
ORU	Orbital Replacement Unit
PDW	Path Dependent Workspace
PIW	Path Independent Workspace
PMCI	Predictive Minimisation of Correlated Indexes
PMKE	Predictive Minimisation of Kinetic Energy
PSO	Particle Swarm Optimisation
ROV	Remotely Operated Vehicles
SA	Simulated Annealing
SFA	Small Fine Arm
SPDM	Special Purpose Dexterous Manipulator
SRMS	Shuttle Remote Manipulator System
SSD	Solid State Disk
SSRMS	Space Station Remote Manipulator System
SUMO	Spacecraft for the Universal Modification of Orbits
TECSAS	Technology Satellites Demonstration and Verification of Space Systems
Tor.	Torques
TPBVP	Two-Point Boundary Value Problem

ZRM	Zero Reaction Maneuver

List of figures

List of tables

List of related published work

Conference papers

The material contained in this thesis is to different degrees linked to three conference papers.

[C1] Tringali A., Cocuzza S., *Predictive control of a space manipulator through error expectation and Kinetic energy expectation,* In: 69th INTERNATIONAL ASTRONAUTICAL CONGRESS, Bremen (Germany), Oct. 2018

The work reported in this paper corresponds to a first iteration of the algorithm presented in the Local Algorithm chapter. Dr. Cocuzza provided code for the simulator used, and general review of the conference paper content.

[C2] Cocuzza S., Tringali A., Yan X. T., *Energy-efficient motion of a space manipulator,* In: 67th INTERNATIONAL ASTRONAUTICAL CONGRESS, Guadalajara (Mexico), Sept. 2016

The work reports the first studies of energy-efficient motion born from the interaction between the author of this thesis and Dr. Cocuzza. Dr. Cocuzza provided the simulator and most of the writing of the paper, while the author of this thesis provided kinetic energy formulation and validation, and computational results.

[C3] Cocuzza S., Tringali A., *Extended reactionless workspace of a space manipulator through reaction wheels,* In: 69th INTERNATIONAL ASTRONAUTICAL CONGRESS, Bremen (Germany), Oct. 2018

This work is an extension of previous one. Dr. Cocuzza provided the simulator and most of the writing of the paper, while the author of this thesis provided improved mathematical formulation for attitude control and kinetic energy, and computational results as shown in Workspace Analysis chapter.

Journal papers

The material contained in this thesis has been partially presented in one journal paper.

[J1] Tringali A., Cocuzza S., *Globally Optimal Inverse Kinematics Method for a Redundant Robot Manipulator with Linear and Nonlinear Constraints,* In: *Robotics* 2020, *9*, 61.

The work reported in this paper corresponds to the algorithm presented in the Global Algorithm chapter. Dr. Cocuzza provided general advice, code and support for the writing of the paper content.

Signature: Date: 23/02/2021

Chapter 1. Introduction

1.1 Motivation

Mechatronic systems and space exploration are two of the great technological breakthroughs of the XX century, and they are closely related to each other. Robotics, in fact, have been part of the effort for space exploration and exploitation since, in a wide sense, every unmanned space mission is a "robot", comprising the very first satellite Sputnik 1, launched on the 4th of October 1957. However, more specifically, a robot can be intended as a system of rigid bodies connected by joints [1]. According to this definition, the most obvious tasks that a roboticist will face are those related to controlling pose and velocity of a robot. Both need to be defined in an unambiguous way, characterised from the mathematical point of view, and set according to the operational needs of the robot user. These problems are usually expressed in terms of translating the position and velocity of each one of the joints of a robotic manipulator into the position and velocity of the so called end-effector, which is the part of the robot executing its main task (usually, a tool or a gripper fixed at the free end of the manipulator, although this may vary, and one example will be presented in this document, where the attitude of a spacecraft will be considered part of the end-effector).

The science that tackles these problems is kinematics, and in fact one of the most classical problem in robotics is the computation of **direct or inverse kinematics of robotic manipulators**. The first one is the computation of the end-effector position given specific joint positions, while the latter one is the opposite problem: estimating which joint positions are required to obtain a specific end-effector position. Both problems present specific issues, and especially the inverse kinematics one has been widely studied for serial manipulators, which constitute the vast majority of robotic manipulators in industrial applications. The problem of computing the inverse kinematics of a robot lies in the nonlinearity of the problem, which forces to resort to solution methods based on the inversion of a Jacobian matrix of the manipulator. Such technique, called differential inverse kinematics, is prone to numerical problems and instabilities, and this makes the inverse kinematics problem particularly tricky.

The complexity is further increased for specific kinds of robotic manipulators, as some structures generate more complex problems than others. Two classes of manipulators for which extra complications arise are redundant and free-floating ones. Both the former and the latter ones feature special characteristics respect to their degrees of freedom: **redundant manipulators** feature more joints (degrees of freedom) than what is strictly necessary to execute its end effector task. Such manipulators are peculiar due to the fact that infinite solutions exist for the inverse kinematics problem, which allows for greater flexibility as it allows to approach each task in a variety of ways. However, in order to exploit the added possibilities, it is necessary to resort to extra constraints, or to optimisation techniques, in order to individuate a specific solution. This further complicates the already difficult inverse kinematics problem.

Free-floating manipulators, instead, are manipulators that are not mounted on a fixed base, such as Remotely Operated Vehicles (ROVs) for underwater operations, or the Space Shuttle arm. The specific complication for this kind of manipulator is that, as soon as the manipulator moves, the base may also move, depending on the reactions transmitted from the robotic arm to the base. The motion is governed by the momentum conservation law, which constitutes a **nonholonomic (non-integrable) constraint** to the motion. This constraint is particularly difficult to tackle, since it implies that a specific position of the joint doesn’t unambiguously correspond to a specific position of the end-effector, due to the fact that the base position depends not only on the current moment, but on the history of the manipulator motion as well. It should be noticed that a manipulator can be both redundant and free-floating at the same time. Mathematical techniques exist to adapt inverse kinematics techniques initially developed for fixed-base manipulators to free-floating manipulators, so that algorithms developed for ground use can be adopted for free-floating systems as well.

From the nature of redundant manipulators, an optimisation problem naturally arises: their added freedom of motion allows in fact to solve the inverse kinematics problem as an optimisation problem, attempting to minimize a cost function such as kinetic energy, torque, or distance from obstacles. Redundant manipulators and novel optimisation techniques to solve their inverse kinematics problem are indeed the main topic of this thesis. Particularly, the main concern of this thesis is the optimisation problem that arises from the objective to minimise the kinetic energy and the torque integral of a redundant robotic manipulator. These functions are commonly associated with the energetic cost of manipulation tasks in literature, and many works have been published regarding their minimisation. **This thesis particularly presents developments that extend the state-of-the-art of global and local inverse kinematics solution methods for redundant manipulators, with a particular focus on energy optimal and near-optimal solutions**.

The topic is relevant for a variety of applications. Particularly, a case can be made for robotic systems used on field applications such as space and remote environments on Earth. For such systems, a big challenge is clearly posed by autonomy and battery life, making it obvious that a case exists for energy efficient inverse kinematics. This can have positive effects on the ability of an autonomous robotic system to perform prolonged operations.

Furthermore, powerful on-board computers are nowadays available for field robotics, making it possible to deploy algorithms featuring higher complexity than before.

Industrial manipulators, although they feature less limited power resources, can also benefit from improved energetic performance, although in a different way. They usually perform a restricted set of motions throughout their whole operational life. As these motions can be optimised offline rather than in real-time, it is particularly relevant for them to be able to access algorithms that can find the best possible solution, taking into account all the real constraints they face during operations.

1.1.1 Space robotics and redundancy

As mentioned earlier, space and robotics are very closed related topics, as most of the human artifacts that are sent to space are in fact automated. However, when it comes to the specific topic of interest for this thesis, free-floating manipulators, their history is much shorter. This kind of mission has become particularly interesting in recent years due to the growing interest in both space debris removal and in-orbit servicing, however very few space robotic arms have flown so far, and none of them has been explicitly designed for debris removal. The first robotic manipulator to see service in space was Shuttle Remote Manipulator System (SRMS), deployed from the Space Shuttle Columbia in 1981. SRMS, also knowns as Canadarm, is a 15.2 m long robotic manipulator with 6 Degrees of Freedom. Since when it has first been developed by Canadian Space Agency, and it has deployed in space several times, performing on-orbit service missions. A further version of the whole design is the so called Canadarm2, which official name is Space Station Remote Manipulator System (SSRMS): it is 16 metres long, and its number of DOF has been increased to 7. This has been possible with the addition of the Special Purpose Dexterous Manipulator (SPDM), it is able to perform activities previously carried out by astronauts through Extra Vehicular Activities

(EVA) [2]. Canadarm2 and its SPDM add-on are currently deployed on a movable platform, Mobile Remote Servicer Base System, which constitutes with them the so-called ISS Mobile Servicing System. However, it is not the only robotic system used on the ISS: Japanese Space Agency (JAXA) developed the Japanese Experiment Module Remote Manipulator System (JEMRMS), meant to support experiments performed in the Exposed Facility (EF) of the Japanese Experiment Module (JEM). It includes a 10 m long main arm with 6 Degrees of Freedom, and a 6 DOF 2 m long Small Fine Arm (SFA) [3]. Another robotic arm, European Robotic Arm (ERA), 11 m long with 7 DOF, has been deployed by European Space Agency (ESA), on a re-allocable base attached to the Russian segment of the ISS [4].

It shall be noticed that most of these manipulators are either redundant or were made redundant at a later stage with additions providing extra degrees of freedom. Indeed, redundant manipulators allow for greater flexibility than nonredundant ones, in that they allow to perform tasks while respecting additional constraints, which can vary from energy minimisation to obstacle avoidance, and in space environment are most likely to be constraints on the motion of the base which, like already said, depends on the motion of the arm through the equation of conservation of momentum.

1.1.2 Current limitations and challenges

Given this additional flexibility, redundant manipulators are a very interesting field of research, yet the additional mathematical difficulties pose a number of challenges, due to the fact that inverse kinematics is a particularly difficult problem to optimise since, as already mentioned, it is nonlinear and prone to numerical problems. Generally, this problem is solved in different ways depending on the application the manipulator is designed for.

Such solution methods can be considered a subfield of robotic motion planning, which refers more generally to the topic of controlling the motion of a robot. For manipulators, this does not only include the inverse kinematics problem, but also choosing an end-effector trajectory that is considered suitable to execute the prescribed task. **In this thesis, the focus is on those problems where a specific end-effector trajectory must be tracked, and it is necessary to compute a joints motion to perform it.** Several practical situations of this kind exist in robotics, for example those where the end-effector is required to perform a specific manufacturing task, or inspection tasks where a camera or a similar instrument is mounted on the end-effector.

The inverse kinematics problem of redundant manipulators can be solved either locally or globally. The difference between the two classes of methods is the scope of the solution: the local methods are based on dividing the desired trajectory in steps and provide an optimal solution for each step sequentially. That is, each step is optimised locally and independently from the others, using the result of the previous step as initial conditions. This means that the algorithm will act in a greedy way, looking for early gains while neglecting what further steps may bring. On the other hand, global methods have the advantage of considering the whole trajectory to be performed at once. That is, the algorithm will avoid an improvement of the solution at a given step if this produces a bigger loss at some other step. This kind of algorithm is usually based on an integral cost (integral along the trajectory under consideration) and it can provide higher quality (lower cost) solutions, at the price of algorithm complexity. Usually, roboticists prefer to use local methods when they have the necessity for a fast or even online solution, while they exploit global methods when they have less time constraints and can afford a solution with increased computational cost.

For local optimisation, and possibly for inverse kinematics problem in general, the most used method is Moore-Penrose pseudoinverse, which is based on a simple local least squares minimisation of joint velocities [5]. What makes it effective is that it is computationally simple, and it allows for countless variations based on a framework called task prioritisation [6]. This consists in modifying the solution obtained with Moore-Penrose pseudoinverse by adding an extra term to joint velocities, that doesn't modify end-effector trajectory (i.e. the result is still a solution of the inverse kinematics problem). This extra term can be the result of a further optimisation, and allows to include secondary objectives such as obstacle avoidance.

This flexibility allows for a huge number of specific adaptations of the Moore-Penrose pseudoinverse to different requirements and frameworks. The specific choice of the implementation to be used for a certain inverse kinematics problem can be found by asking what the most effective way is to solve inverse kinematics given the specific operational needs of the manipulation task under examination. For many manipulators tasks, the most obvious answer to this question is the avoidance of singularities. These are specific configurations where a specific end-effector velocity can only be achieved through infinite joint velocity, and they are the main issue related with Moore Penrose pseudoinverse, or any other pseudoinverse-based inverse kinematics algorithm. This problem is made even more complex in the case of orbiting manipulators, since they not only feature kinematic singularities, but dynamic singularities as well, which depend on their inertia tensor [7]. Their avoidance can be implemented with similar methods as classical manipulators, although with increased mathematical complexity, as it will be explained later in this thesis. The main reason for this issue is that any algorithm based on the Moore-Penrose pseudoinverse simply incorporates local information. It is particularly hard to develop an effective method of solution that allow for better-than-local solutions because increases in computational cost mean that the algorithm will be harder to use online. The possibility to incorporate more than local information in an online method is something that researchers have been looking at in the last decades, and it is indeed one of the challenges that will be investigated in this thesis.

The global problem is different than the one that is solved by pseudoinverse-based methods, in that in this case the motion is not locally optimised for each time step, but the inverse kinematics *global* problem is solved instead: that is, the whole trajectory is considered at once, and an integral cost is optimised, as opposed to the value of a specific function at a specific point. This kind of problem is of course much more mathematically complex, and features completely different solution methods, most of which are concentrated on point-to-point motion rather than precise tracking of a trajectory. The solutions methods that have been developed to the inverse kinematics problem are generally based on optimal control frameworks, such as Pontryagin maximum principle [8], or calculus of variations, and try to solve the problem by setting up a Two-Point Boundary Value Problem (TPBVP).

Global methods feature a number of limitations that is quite consistent. An important limit lies in the necessity to solve a TPBVP, which is a mathematically complex task requiring a good initial guess, and presenting complications related to the imposition of boundary conditions. Furthermore, as outlined by some researchers [9], it may well happen that methods based on the calculus of the variations get caught in a local optimum, depending on the nature of kinematic constraints and of the initial conditions. One more limit is that real manipulators feature limits on joints excursions, velocity, as well as power and torques, which are usually hard to introduce in existing optimisation methods, which are in the best cases able to incorporate linear constraints, but fail in considering nonlinear constraints, such as torque limits. Ideally, such nonlinear constraints should be extendable to obstacles and environmental constraints to the motion of the robotic arm, and be able to output periodic trajectory, where the robotic arm reaches the initial joint configuration again at the end of the motion. This kind of trajectory is particularly relevant for repetitive industrial applications. Challenges that invite for further investigation in the field of global inverse kinematics optimisation are the complexity of use of existing solutions, and their limitations regarding the ability to impose constraints which apply to real manipulators.

1.2 Research aim and hypothesis

1.2.1 Research Aim

This thesis investigates the problem of optimal inverse kinematics solutions for redundant manipulators. More specifically, it aims at investigating if the problem of developing novel mathematical optimisation techniques to the inverse kinematics problem to obtain solutions that feature a minimisation of the integral energy cost of a predefined trajectory, while avoiding specific limitations of existing algorithms. It should be noticed that the particular target of this work is to obtain an overall performance increase over trajectories performed by a manipulator, as opposed to a local result. The problem will be discussed in a way that can be applied to both fixed-base and free-floating manipulators, making a case for the use of the results in space.

The inverse kinematics problem of redundant manipulators is easily posed as an optimisation problem, however the methods used so far to individuate a solution feature several limitations. Such limitations are particularly striking considering that the field of robotics has generally made use of a huge variety of optimisation techniques to tackle its problems, such as inverse kinematics itself, or the point-to-point motion problem.

Conventional solutions to the IK optimisation problem rely on least squares optimisation for the local problem, and to calculus of variations or Pontryagin maximum principle for the global problem. The local schemes, exploiting the Jacobian matrix of the manipulator to calculate joint accelerations or velocities [10], feature, as already mentioned, kinematic singularities, where the manipulator Jacobian (i.e. the matrix characterising the problem) cannot be inverted. Singularities are an important reason why manipulators trajectories may become energetically expensive, since moving in their proximity increases joints velocities. In order to be mitigated, they require special implementation and numerical expedients, one of them being numerical damping, which however reduces tracking precision [11]. In a milestone paper for robotics, Yoshikawa [12] proposed to use manipulability as an index to implement singularity avoidance, but its use as a singularity avoidance tool has been doubted in literature (Staffetti et al. [13]). Much like Yoshikawa, this thesis aims at reducing the local optimality of the solution to produce an overall improvement along the complete trajectory being performed. In particular, this thesis aims to produce a substantially improved energetic cost for manipulation rather than concentrating on improving manipulability or other simplified measures. This is expected to bring as a side effect that singularities are way less frequent, since they naturally mitigated by the fact that the lowest-energy joint trajectory is generally far from them.

For what concerns global algorithms, current implementations are limited in their ability to include constraints or varied complexity, and are limited in their flexibility and ease of use by the fact that they require to solve a Two-Point Boundary Value Problem (TPBVP), a problem featuring consistent mathematical complications and requiring educated guesses from the user to be solved properly. Coherently with the aim of this thesis, a global solution method is sought, that allows to include the maximum possible variety of constraints (linear and nonlinear) and cost functions, while being easier to use than existing algorithms. Furthermore, it should be able to provide solutions of multi-objective optimisation problems and individuate multiple optima when these are present.

The contribution to knowledge of this thesis is novel inverse kinematics optimisation methods that reduce operational costs of redundant manipulators by minimising their kinetic energy or joint torques norm integral along the complete end-effector trajectory. Although the impact of such an optimisation vary on the application, it is safe to say that many applications of redundant manipulators can benefit from reduced energetic needs, especially those fielded in environments such as space, where the amount of available power is limited and depending on factors not always under control. However, a reduction in power requirements can have an important impact even on more mundane industrial manipulators, since running costs are consequently reduced.

1.2.2 Research hypothesis and scope

In this thesis, a number of developments are presented to advance the state-of-the art of energy-optimal inverse kinematics algorithms for redundant manipulators. These developments include a global algorithm, a study of the workspace of fixed-base and free-floating manipulators in relationship to energy-saving patterns, and a local algorithm. Each algorithm has been developed in order to set the following research hypothesis:

“Mathematical optimisation tools can be developed for the inverse kinematics problem of redundant manipulators to obtain optimal or near-optimal solutions that feature better results and less limitations than solutions provided by optimisation algorithms currently in use.”

The **scope of the research** is thus the development of mathematical tools to solve the inverse kinematics problem of redundant manipulators. Such mathematical tools should be able to compute solutions which cannot be computed with currently existing algorithms, and provide practical advantages such as being closer to optimality, or being able to take into account practical constraints and limitations in a better way than existing mathematical tools.

1.2.3 Research questions

To develop the algorithms presented in this work, a number of research questions have been asked, following the early findings from a first literature review. These questions are concerned with current limitations of the solutions of the inverse kinematics problem for redundant manipulators, and they have been proven difficult to answer when compared to existing solution of the problem found in literature. By addressing them, the aim of the thesis would be satisfied and the state-of-the-art in inverse kinematics solutions would be extended.

1. What considerations are necessary to solve the inverse kinematics problem for redundant manipulators end-effector trajectory tracking through mathematical optimisation of energy-related integral cost functions?
2. How can the limitations of conventional solutions of the problem mentioned in question 1 be tackled by applying optimisation techniques that have not been used this way before? And how can such techniques be applied to free-floating manipulators?

3. Are there different limitations that come into play by applying different methods to the problem? How can they be tackled?

1.2.4 Research objectives

In order to achieve the desired research aim, it is necessary to divide it into separate objectives which constitute the milestones to reach in order to answer to the research questions.

1. Conduct a detailed literature review on the state-of-the-art of the inverse kinematics problem, and on the use of optimisation algorithms for robotic motion planning, both considering fixed-base and free-floating manipulators. Limits to existing works should be identified and used as a starting point for further research.
2. Identify representative cases for the development of energy-optimal inverse kinematics trajectories.
3. Identify the specific challenges of energy optimisation in relationship with the manipulator pose and tasks within its workspace.
4. Develop optimisation techniques that advance the state-of-the-art in redundant manipulators inverse kinematics by providing solutions that are closer to the global optimum compared to existing algorithms, and that allow to provide solutions not achievable with existing algorithms.
5. Validate the techniques as per objective 4 by testing the introduced research hypothesis and verifying it on inverse kinematics problems.

1.3 Research methodology

The methodology adopted in this thesis follows the subdivision of the inverse kinematics problem into a local and a global version. The two problems introduce several unique considerations that are peculiar to their specific mathematical formulation, although they also have some trait in common. They differ for number of parameters under consideration, initial and boundary conditions, and for the kind of cost function they use, as the global problem features an integral cost. This is reflected in the subdivision of the thesis, which tackles them separately. In order to analyse the challenges of predicting the global effects of a local optimisation, a workspace study has also been considered necessary, and performed observing how specific inverse kinematics algorithms change its shape and characteristics. The results of such study are presented before other results related to the local problem, since they allow for a better understanding of the challenges met during the development of this thesis.

1.3.1 General methodological considerations

Since this thesis is focussed on mathematical tools for the inverse kinematics problem, the main methodological assumption is that simulations can be used to validate the results, as opposed to physical experiments with manipulators. This assumption is considered reasonable because the main goal of the thesis is to produce mathematical solutions with peculiar characteristics (i.e. optimality, constraints, simplicity of use), as opposed to specific implementations of such solutions. Tools to develop such solutions purely in simulation are readily available, as formulations to compute kinetic energy, torque and other figures used to quantify manipulators energy consumption are can be easily found in literature and robotic textbooks [14]. On the other hand, performing physical experiments does not capture any specific aspect of a newly developed solution method, since the challenges of physical experiments are comparable for all inverse kinematics methods: inverse kinematics algorithms usually output the expected manipulator joints trajectory in terms of positions, velocities or accelerations. Such trajectories are routinely implemented on real manipulators in research and in the industry, and the implementation process itself features similar challenges irrespectively from the algorithm exploited to compute the trajectory in the first place. This is supported by the amount of algorithms in literature validated solely by simulation (see for example [15]–[17]) or featuring similar results for simulations and experiments [18]. It is particularly to be noted that literature provides an optimal kinetic energy solution [15], which features local properties validated by simulation and global properties solely validated by mathematical proof.

From this, it follows that it must be assumed that the inertial properties of the robotic manipulators under examination are well known. This assumption has already been made in inverse kinematics literature (see for example [19]) and allows to focus solely on the quality of the IK solution. It is noted that lack of knowledge of inertial properties of robotic manipulators is also a common problem for any motion planning algorithm involving kinetic energy or torques, and an important body of literature has been dedicated to manipulators system identification, with first published work being, to the author knowledge, by Atkeson et al. [20]. The accuracy and optimality by which trajectories obtained through inverse kinematics algorithm can be implemented on real manipulators depend thus on the knowledge of the manipulator dynamics, which in turn depend on the specific manipulator and system identification method. It is therefore considered to be beyond the scope of this work, as it introduces elements which do not depend on the IK solution.

Having decided to validate the new algorithms by simulation, a further choice must be done about the manipulator to use in the simulation. Given the focus on mathematical tools, the validation has been performed on a simple 3-DoF planar manipulator, described in detail in Chapter 3. This manipulator topology has been kept the same throughout the whole thesis. This uniformity allows to readily compare the solutions presented in the thesis to one another and can give a visual intuition of the difference between traditional algorithms and the newly developed ones. It in turn means that the implications of the use of the algorithms hereby presented on more complex manipulators are only touched briefly in Chapter 7, dedicated to the Conclusions, and no numerical examples are provided. As the scope of the work is to provide inverse kinematics solutions with enhanced mathematical properties than currently known ones, this is considered an acceptable assumption. A final methodological note regarding the choice of the manipulator is that most of the simulations are performed on fixed-base manipulators since, as illustrated in the chapter dedicated to the mathematical background of the thesis, it is possible to extend algorithms from a fixed-base to a floating-base manipulator by means of specific mathematical tools [19].

As already specified as part of the research aim, this thesis will generally look at the performance along a complete trajectory rather than local results, which means that, while the local problem is optimised according to a local cost function, the quality of the solution will be evaluated observing the overall performance on a complete trajectory. This is considered methodologically important because it is assumed that further emphasis on strictly local algorithms, given the wealthy body of such algorithms available in literature, is not necessary. Many examples of such existing methods will be presented in next chapter.

1.3.2 Literature review

The literature review aims at providing a critical analysis of the state-of-the-art in robotic arms end-effector motion planning. The field is very vast and a complete literature review is beyond the scope of this thesis, it has been thus narrowed down to inverse kinematics for fixed-base and redundant manipulators, and to optimal control and optimisation algorithms applied to robotic arms. The works chosen have been mentioned based on either historical relevance for the field, direct possibility of application to energy minimization, of relevance for the field of optimization applied to inverse kinematics.

The goal of this chapter is to provide the reader with an understanding of current limitations in existing state-of-the-art in the field, and to identify the key gaps in literature, which are tackled by the research presented in this document.

1.3.3 Global algorithm for inverse kinematics

The first problem studied in this thesis is the global inverse kinematics problem, where a redundant robotic manipulator is supposed to perform a predefine end-effector task (tracking of a trajectory) while keeping an integral cost alongside the trajectory as low as possible. The problem is first introduced and considerations that must be taken into account in the development of a global algorithm are analysed, answering to research question 1. Looking for an answer to question 2, it is then observed that several optimisation methods are able to overcome specific limitations of the traditional approach: they are easier to use since they do not require to solve a TPBVP, they allow to take into account a variety of cost functions and constraints, and they can find multiple optima, allowing for a more complete search of solution space. Based on the problem under examination, it has been chosen to explore the direction of multi-start methods, exploiting sequential quadratic programming as underlying local method, to solve constrained kinetic energy and torque integral minimisation problems. The choice of multi-start is motivated by the nature of the optimization problem under examination: the required solution is continuous, differentiable, and features a very high number of parameters and nonlinear constraints. Other methods, such as Genetic Algorithms (GAs) are certainly more frequently adopted in robotics literature [21]–[23], however they are usually based on parametrizing continuous solutions in terms of coefficients of continuous, or piece-wise continuous, polynomials, which in this case does not reduce the dimensionality of the problem, given the high number of constraints. Furthermore, the number of non-linear constraints consistently restricts the set of acceptable solutions and suggests the use of algorithms which feature an embedded capacity to incorporate such constraints at every optimization step. A simple approach based on gradient-based methods appears thus more promising. The quantity and nonlinear nature of constraints supports the case for sequential quadratic programming, which simplifies both the cost function and complex nonlinear constraints into a single quadratic problem (the algorithm is described in detail in Chapter 4). The resulting algorithm constitutes the research contribution of this thesis for what concerns global algorithms. In order to explore its potentialities, question 3 is first addressed, by finding that a potential problem related with the so called *curse of dimensionality* may appear. This is a known problem of many algorithms, and it refers to the fact that they may require an amount of time and memory to converge that grows very quickly with the number of variables, up to the point of becoming exponential to the number of variables. This may cause their use to be very lengthy and impractical, to the point that they would require too many resources to converge and would become intractable.

Particular care has been taken to solve this issue, further refining the new algorithm into an iterative version, which only takes a subset of the parameters of the complete inverse kinematics problem at a given step. The closer the iteration gets to the global solution, the higher the number of parameters considered, up to the point when the full problem is solved. In order to show its capabilities, this final version of the algorithm is evaluated by simulation. Particularly, the validation of the algorithm is performed against a solution method based on calculus of variations, taken from literature [15]. This comparison shows that the new global algorithm presented in this thesis matches the performance of variational methods when it is used to solve problems for which traditional variational methods are able to find a solution. Finally, a complete investigation of the new algorithm is performed, exploring its capabilities to solve a set composed by various global optimisation problems, with kinetic energy and with torque integral as cost functions, and with a variety of constraints on joint displacement, velocity, torque and power and on cyclicity of the motion. The set of problems has been chosen considering the physical limitations and operational requirements of real robotic manipulators, and with the aim to demonstrate that a wide variety of global inverse kinematics problems can be tackled by the new algorithm with minimal tuning, and among them a few which have never been addressed before in literature (particularly torque and power constrained ones).

1.3.4 Workspace analysis

The biggest challenge when designing a local inverse kinematics algorithms is related to the non-linearity of the problem under consideration, and on the difficulties in understanding how reaching the local optimum at a certain point in the trajectory will influence the reminder of the motion [24]. In order to show the reader how this affects the resolution of the local inverse kinematics problem, workspaces of redundant manipulators are analysed from the energetic point of view. The analysis aims to investigate how kinematic indexes that literature links with well-behaved trajectories (i.e. non-singularity of the Jacobian) matrix are related with energetic performance. To this purpose, the end-effector of the manipulator under analysis is moved all along a planar workspace with constant velocity, and the resulting kinematic indexes values are analysed. The selection of kinematic indexes to be included in the analysis has been based either on their use in the field of inverse kinematics or on their close relationship with variables that influence the energy and power figures of robotic manipulators. Further details are contained in the relevant chapter

Since the relationship between kinetic energy (or power) and manipulator motion does not have a clear analytical form, the possibility to include a further research contribution in the thesis has been identified in performing an analysis on a statistical basis to extract data features regarding correlations between kinematics indexes of redundant manipulators and kinetic energy and other energy-related variables. This is considered relevant both for the goal of motion planning, and for kinematic design considerations: in fact, minimization or maximization of kinematic indexes has been proposed and used as a design method in literature [25]. For the analysis hereby performed, the instrument of canonical correlation is used to assess which kinematic indexes influence the ability of a manipulator with a reduced energetic cost. This is a statistical technique that allows to quantify the correlation with linear combinations of variables rather than single variables, allowing to visualise which properties of a manipulator, and manipulator Jacobian, are more relevant for energetic performance. In order to allow for an understanding of the difficulties of predicting integral (global) energy costs when local algorithms are involved, the linear combination of kinematic indexes that features the highest correlation with kinetic energy is highlighted. Based on the analysis results, it is considered this function is a good approximator of the energetic behaviour of the manipulator, at least when considering only local variables to compute the approximation. Its optimization will thus will be used as a comparison with actual kinetic energy optimization in the reminder of the thesis, showing how much results can differ between a local approximator and the exact formulation involving joint velocities..

Afterwards, the analysis is extended to space manipulators and their workspace, particularly to observing how energy-saving inverse kinematics change the size and nature of the workspace compared to the traditional way space manipulators are controlled.

A peculiarity of space manipulators, in fact, is the implementation of constraints on the base. This is particularly important in the space environment, as the base must be kept in a specific orientation in order to maintain communication systems, solar panels and payload pointed to the right direction. The effort to do so can be partially demanded to the Attitude Control System (ACS), but this is usually considered impractical because of the mathematical complexity introduced by interactions between its actuators and the robotic arm joints. In this thesis, the workspace of the manipulator under analysis is extended by activating the ACS concurrently with the robotic arm, and effects on the nature of the workspace (i.e. size and manipulability of the overall ACS-manipulator system) are discussed.

Some limitations in both fixed-space and free-base analyses occur due to the fact that the manipulators used are relatively simple: it is well possible that more complex robots would feature a different workspace structure. Simple manipulators have however already been used in literature to assess general properties of manipulators' workspaces, even for free-floating ones[26], and the analysis should be considered under this light. Particularly, the properties that the thesis focuses on are the relationship between workspace and kinetic energy, and the amount of extension of the workspace of a free-floating manipulator that is made possible through the use of the ACS. It is noted in particular that space manipulators feature complex, dynamic coupling between their axes and that this is not captured with a planar manipulator. Practical concurrent control of a free-floating 6-DoF robotic arm and a 3-axis base attitude control has however been achieved in literature with a formulation based on the conservation of momentum [27], similarly to the workspace extension method presented in this thesis, which supports the case of the generalizability of the results hereby presented.

1.3.5 Local algorithm for inverse kinematics

The methodological approach to the local algorithms in this thesis has been chosen in the light of workspace analysis results. Taking the point that no discernible patterns have been individuated on a strictly local basis, it has been elected to include a prediction of the best direction of motion when the local optimisation is performed, which means that the optimization does not only rely on local information, but on insights about future direction of motion as well. This introduces a number of challenges in that the prediction has to be performed in a time that has to be as short as possible, yet it has to be accurate enough to effectively reduce the kinetic energy cost of the trajectory. A further challenge is the one introduced by the need to minimise a global (integral) cost by using a local algorithm. This generally means that results cannot be unambiguously mathematically proven to be optimal or better than any other local algorithm, since (with some notable exceptions presented in the literature review) local algorithms usually bear no relationship with global costs. Both these points dictate the shape of the proposed solution and the methodology adopted to validate it.

In light of these considerations, a local algorithm is presented in this work that features the ability to introduce a prediction of the future energetic cost in the local optimisation. It is based on two additional hypotheses:

1. At least a part of the future end-effector trajectory is known in advance and can be used for a prediction.

2. The solution does not necessarily have to be the globally optimal one, but it must clearly outperform traditional local inverse kinematics algorithms on a consistent number of cases.

The first additional hypothesis is acceptable for applications where the end-effector trajectory is not being computed in real-time itself. For application where it is also computed in real time, the method would include a sub-second delay (0.1s for most of the examples in this thesis, as explained in Chapter 6), which can also be acceptable depending on the application. It is, in any case, a feature that is shared with Model Predictive Control (MPC) and other control algorithm exploiting predictions of future costs. The second additional hypothesis is instead directly derived from the nature of local algorithms: a local algorithms lacks the information to provide a provably optimal global solution (some formulations can be used for both local and global problems [15], but the method of solution changes radically). Thus, it is reasonable to look for a local solution which globally performs better than other local solutions over a set of analysed cases.

Under the hypotheses presented for the whole thesis, and the additional two introduced here, it is possible to reduce the integral kinetic energy cost of a robotic manipulator inverse kinematics problem. The solution that allows to do so is based on an approximate prediction of the value of the integral of kinetic energy in future steps of a discretized trajectory. This prediction is based on a very small subset of future points and it does not need to be updated at every time step. Its purpose, rather than providing a precise result for future kinetic energy expense, is individuating the direction for which the energetic cost will be lower.

The strategy pursued to reduce its computational cost is to update the prediction with a very large time step. This allows to compute it only a few times over the whole trajectory, reducing computational burden. In order to validate the method, three inverse kinematics method available in literature have been included as terms of comparison, featuring the minimization of joints velocities norm, kinetic energy, and reactions transmitted to the base of the manipulator. The first cost function is widely used in inverse kinematics of redundant manipulators, and has been chosen as the benchmark of the comparison due to its huge diffusion, while the second one is directly relevant to the topic of energy minimization, and the third one is the main cost function used for free-floating manipulators [28]. All the methods chosen are based on the Jacobian matrix, which is a widespread and well-understood inverse kinematics tool, used in practical robotics and in research. Jacobian-based algorithms have been used for minimization of countless different cost functions, with velocities norms and kinetic energy being mentioned already in the first work on the topic [5]. Many other methods are available today in literature, their goal is however usually to add additional features to the optimization (e.g. joint limits) rather than improving the solution itself, as detailed in the literature review in Chapter 3, while Jacobian-based methods already provide the locally optimal solution for many cost functions. The use of the Jacobian is so frequent also due to the fact that it allows to add secondary objectives to the optimization task. In the light of this, the local algorithm presented in this thesis has been designed as a secondary task for Jacobian-based algorithms, allowing it to be incorporated in existing frameworks.

A further term of comparison has been added, developing a version of the new predictive algorithm based on the function computed in workspace analysis with the highest correlation with kinetic energy, as a mean to show the difficulties in mapping characteristics of the manipulator to kinetic energy cost.

Thus, a total of five local algorithms has been simulated and compared. The number of trajectories simulated is 60 for every algorithm, of which 45 rectilinear and 15 circular, resulting in 300 simulations. This has allowed for a comparison of average results of the algorithms under examination, showing the ability of the new predictive method to outperform all the others, and its validity as a novel knowledge contribution.

1.4 Thesis organisation

The thesis is structured in seven chapters, considering this introduction.

Chapter 2 is the literature review chapter, where the literature about inverse kinematics and motion planning of redundant manipulators is discussed. The chapter discusses ground and orbiting manipulators, and analyses existing literature for local and the global inverse kinematics problem, while also providing some background regarding the general application of optimisation in robotics planning, and some illustration of workspace analysis methods available in literature. The findings of the literature are then discussed and commented individuating the gap that this thesis aims to fill.

Chapter 3 illustrates mathematical background of the thesis, and the specific implementation used in the thesis as well. It is aimed at providing a general overview of the mathematical knowledge required throughout the course of this document, and the way it has been used in the validation of the thesis. It introduces manipulators’ kinematics and dynamics model, provides a complete description of the Moore-Penrose pseudoinverse, its numerical issues, task prioritisation, and the general flexibility provided by pseudoinverse-based algorithms. After this, the generalised Jacobian is presented. This is a mathematical tool that allows to compute free-floating manipulators kinematics with the same methods that are normally used to solve the inverse kinematics problem when the base of the manipulator is fixed to the ground. The chapter also explains the important difference between local and global optimisation, what their specific peculiarities are, and what most used global optimisation meta heuristics are. Furthermore, the chapter presents the robotic manipulator used in the simulations and explains in detail which simulations are performed to present and validate the algorithms. The chapter is closed by a description of the simulating software used to perform all the analyses presented in this thesis.

Chapter 4 is devoted to the presentation of the Interpolation-Based Global Kinematic Planner, the global optimisation method developed as part of this thesis. Such algorithm is able to optimise different cost functions with different sets of linear and nonlinear constraints, comprising periodicity of the trajectory, which is often required for industrial applications. It is based on an implementation of the multi-start algorithm and allows to optimise different integral cost functions (kinetic energy, joint torques squared norm) along an end-effector trajectory. It only requires a first guess of the manipulator initial configuration, from which it is able to generate a population of candidate trajectories and select those that yield a higher probability to be close to an optimum. These trajectories are then optimised with a method that does not involve a Two-Point Boundary Value Problem in order to find a solution. Furthermore, several optima are found at each run of the algorithm, overcoming a known limitation of variational methods, and allowing for more chances to find the effective global optimum. The chapter starts with a general overview of the algorithm, where a first version of the algorithm is presented. Since this version converges slowly for more complex problems, an improved, iterative version based on interpolation is presented and validated against an existing algorithm available in literature [15]. Several optimisation problems with constraints on joints displacement, velocity, torque or power are then solved and their solutions are commented.

Chapter 5 features an analysis of the test manipulators workspace. The purpose of this analysis are several: on one hand, workspaces are characterised in terms of size and patterns in preferential direction of motion depending on the algorithm and on the type of manipulator (fixed-base vs free floating, and free floating with attitude control system). On the other hand, a statistical analysis is performed to find correlations between the kinematic indexes of robotic manipulators (manipulability, Jacobian condition number etc…) and their kinetic energy along the motion on a certain trajectory, with the aim to characterise the relationship between manipulator configuration and expectation of kinetic energy cost. For this purpose, canonical correlation is used, which represents the correlation of linear combinations of variables. From this statistical analysis, it is observed that no kinematic index features high enough correlation with kinetic energy to be used as an alternative cost function in optimisation, and that a linear combination of them is selected as a possible approximator of kinetic energy. For free-floating manipulators, it is observed that reducing the base motion generally also reduces the workspace, and that attitude control system can be actively used to retain the manipulability of the manipulator (and thus extend the workspace).

Chapter 6 is dedicated to the presentation of the local algorithm developed for this thesis, based on predictive minimisation of a cost functions. The algorithm is meant to integrate with traditional Jacobian based pseudoinverse methods rather than substitute them, due to the simplicity, versatility and diffusion of pseudoinverse schemes. It is based on computing the value of a rough prediction of the kinetic energy integral along the trajectory, and exploit task minimisation to drive the manipulator in the direction where such prediction has a minimum. The prediction does not need to be updated at every time step, and does not need an extensive set of parameters, thus allowing to be used quite flexibly and reducing computational cost for real time execution. The predictive algorithm has been tested in two versions, PMKE (Predictive Minimisation of Kinetic Energy), based on the minimisation of a prediction of the kinetic energy, and PMCI (Predictive Minimisation of Correlated Indexes), which is based on the minimisation of a linear combination of kinematic indexes correlated with kinetic energy, as obtained from the workspace analysis. The first one has shown to provide much better results, being superior to traditional algorithms on a large set of trajectories, thus justifying its computational overhead, while the latter one illustrates the difficulty in predicting kinetic energy without precise knowledge of the motion.

Finally, **Chapter 7** features conclusions, limitations and further development of this thesis, with some points being made about computational time reduction for both the algorithms, and about their application to more complex manipulators. Furthermore, the possible introduction of environmental constraints and obstacle avoidance in the global algorithm is discussed.

Chapter 2. Kinematic motion planning and optimisation in robotics

2.1 Introduction

In this chapter, a survey of literature is conducted, about the state-of-the-art of redundant manipulators motion planning. This literature review mostly focuses on inverse kinematics and does not pretend to be exhaustive on the topic. Its main goal is to present previous research that the author considers relevant to have a complete perspective of the field, and to provide readers with the necessary knowledge to understand the contributions presented in this thesis in relation with their field of application. The chapter will be organised as follows: first, a general survey about methods to solve inverse kinematics and kinematic planning optimization method for redundant robotic arms is presented. This part is fairly general and can be applied to a number of different manipulators configurations, although some of the mentioned works are related to peculiar cases (e.g. hydraulic actuators). After this part, a more specific section is devoted to research that has been especially dedicated to motion planning of free-floating robotic systems. A further section reviews the workspace analysis papers that have been inspirational for the approach taken in this thesis. Finally, a discussion about literature review findings to completely individuate the gap filled by this thesis will be presented.

2.2 Kinematic planning algorithms for redundant manipulators

2.2.1 Pseudoinverse based local methods

Redundant robotic systems motion planning has different factors of complexity. Most notably, the main issues come from the use of redundancy: while it allows far more movements and tasks to be accomplished. It also increases control complexity. So far, the most used methods for motion planning optimisation of redundant robotic arms, as already mentioned in the introduction, have been pseudoinverse-based algorithm. They have been used in plenty of variations and have several interesting properties: it has fast execution times, can be used to prioritise specific tasks over others and it can be used to optimise a variety of different cost functions while driving the robot end-effector through a predefined path. This comes however at a price: it only provides local optima, which can be quite far from the global one. Local methods are also vulnerable to singularities, which cause large errors and mechanical stress on the manipulator. This makes IK frameworks not ideal; yet, offline optimisation methods are too cumbersome for real-time applications, due to their heavy computational requirements.

To the best of author knowledge, the first systematic review of about pseudoinverse-based inverse kinematics is in Klein and Ching-Hsiang [29], which is indeed a milestone in the research on the use of pseudoinverse for robotic control. The paper outlines how the pseudoinverse generally keeps the instantaneous power required low, since it minimises joint velocities, which is a rough approximation of kinetic energy minimisation. It also states that the pseudoinverse, minimising velocities, keeps the manipulator away from singularities, which was however proven incorrect by Bailleul et al. [30]. Many of the subsequent works evolved their algorithms from the concept of weighted pseudoinverse, as previously presented by Whitney [5]. This concept is based on optimising each joint with a different weight, rather than trying to reduce the velocities norm, in order to prioritise specific aspects of the motion. A possible variation of this exploits the inertia matrix of the manipulator as a weight matrix, resulting in a local minimisation of the kinetic energy at every time step. This kind of solution will be used as comparison for the local algorithms implemented in this thesis.

After that, IK have been proposed as a redundancy resolution method for several applications, most notably energy or torque minimisation, but obstacle avoidance, joint-limit avoidance, or simply singularities avoidance, have been explored as well. Nakamura et al. work about obstacle avoidance [6] is extremely important for robotic manipulators control, as it introduces the concept of task prioritisation to solve redundancy problems: this is a method to superimpose different motion tasks, so that a first one follows the end effector predefined path, while the others optimise secondary criteria. Its first application was, as mentioned, obstacle avoidance, but it has been used in most of IK optimisation schemes afterwards. In [31], Chevallereau and Khalil propose more applications of the task prioritisation framework, which they call null-space method, particularly they maximise the distance from singularities through manipulability, and the distance from joints limits. Several other applications have been developed along the years, with manipulability being a favourite cost function for being maximised as secondary task [12]. Manipulability, which has already been mentioned in the introduction, is a quantity depending on the determinant of the Jacobian matrix of a robotic manipulator, and it approaches 0 close to singularities. This makes its maximisation quite an effective method to avoid singularities. More complex applications have been developed along the years, with the concept going as far as being used to maintain the position of the centre of gravity of a legged robot, as proposed by Mistry et al. [32]. Such scheme uses the task prioritization framework to switch between different configurations of the robot, depending on which leg is on the ground. An alternative approach to take into account extra constraints in the motion of a redundant manipulator is to include them as extra rows in the Jacobian (augmented Jacobian), which was defined as state space augmentation method by Sciavicco and Siciliano [33]. This algorithm, however, features extra singularities, which do not normally appear in the original manipulator Jacobian.

Several researchers have followed the direction of pointwise minimising the joint torques as a mean to minimise the actuator consumption, associating such an algorithm to positive results regarding the minimisation of the kinetic energy integral along the whole trajectory. Some analysis on this approach is presented by Hollerbach and Suh [24], who however found out that a dynamics-based inverse kinematics resolution method can lead to instabilities on longer trajectories. Particularly remarkable for the scope of this thesis is one quote from the paper, “It seems that local tampering with the energetics of movements as led to global disaster”, which summarizes the complexity of achieving a satisfying global result based on local methods. Two works exploit the calculus of variation to provide local solutions with global properties: Kazerounian and Wang [34] presented a local solution that is able to minimise the integral of the norm of joint velocities, while a strictly energy-focused approach was followed by Nedungadi et al. [5], who used calculus of variation to find a formulation of torque minimisation that can lead to the global minimum of kinetic energy. Both works can be either local or global depending on the way they are used, however the latter case requires the solution of a two point boundary value problem, while the solution also presents stability issues that make the search particularly hard in both the local and the global case. The algorithm developed by Nedungadi is however provably optimal, which has led to it being selected as a mean to validate the global algorithms presented in this thesis.

The first research providing actuator models for local energy consumption minimisation was presented by Vukobratovic et al. [35], who introduced dynamic models of hydraulic and electrical actuators and showed indeed that a major advantage of energy minimisation compared to joint velocities minimisation is that it tends to move joints with low inertia more than those with high one, thus reducing consumption. This is a key point about energy minimisation algorithms and will often be observed in the results hereby presented as well. One further merit of the authors is that their algorithm is able to implement limits on both joint positions and rates. As pointed out by Nenchev [10] in his review, these early studies made it every apparent that a trade-off between local optimisation and global stability is generally unavoidable: pointwise minimisation must always be relaxed not to get stuck in singularities. For this reason, some authors, such as Wampler and Nakamura [11], [36] proposed damped least square formulations. These are solutions that balance between velocities and end-effector tracking error by adding an extra term to the least squares' formulation, so that the velocities norm is balanced against the tracking error. This concept was further refined by Schinstock et al. [37], who introduced a damped least squares scheme based on only damping in the neighbourhood of singularities, which allows to keep a low tracking error anywhere else in the workspace. A further refinement to the method has been developed by Maciejewski and Klein [38], which presented numerical filtering. This is a method based on individuating the direction with the lowest singular value and selectively damping only that one (i.e. only the actual direction where the singularity lies). This solution is today widespread as it is quite effective and simple to implement and, although it can lead to the manipulator losing the track, nevertheless can find uses where perfect tracking is not essential, or as a safety measure to avoid the manipulator to get damaged during the motion.

Despite approaches based on pseudoinverse matrix have been investigated for a long time, improvements on existing results are still being published in recent years. An idea based on switching between different tasks has been presented by Kelemen et al. [39], who change the priority of the tasks in order to choose the order that requires less computational time. This is particularly useful for hyper-redundant manipulators with an extremely high number of DOF. Another recent pseudoinverse-based approach has been presented by Woolfrey et al. [40], who present a framework able to keep the correct tracking when the manipulator is subject to large external forces.

2.2.2 Other local methods

Since stable local solutions are so elusive, especially when it comes to the torque minimisation problem, in the last 20 years a few authors attempted to solve the local problem with different methods other than pseudoinverse. These sounded more promising than traditional, and unstable, algorithms, and some results are particularly worth mentioning, especially regarding torque local optimisation. Most of these efforts are focused on solutions based on quadratic programming, an optimisation technique aimed at minimising a quadratic cost function subject to equality and inequality constraints. Particularly, the generalisation of a Quadratic Programming based inverse kinematics algorithm to inequality constraint is a breakthrough result obtained by Kanoun et al. [41]. The method allows to include constraints such as joints mechanical limits and actuators maximum velocities, but its main challenge lies in the ability to provide a result in real-time. Many algorithms have been proposed, featuring different principles from classic optimisation techniques to neural networks. Some practical solutions have been presented in the 90s, for example by Cheng et al. [42], who proposes workspace decomposition, a technique to divide the problem into smaller subproblems. This technique allows them to optimise distance from singularities with several kinds of constraints on joints angles, velocities and accelerations. The problem of torques minimisation has been tackled by a few researchers, especially using neural networks as a method to solve quadratic programming problems. An example is provided by Tang et al. [43], who propose two different kinds of neural networks, Lagrangian Networks and Primal Dual Networks. Both methods are successfully used for local minimisation of torques, while the second one also manages to enforce a maximum torque limit in the solution. A similar approach has been sought by Zhang et al. [44], who proposed a simple neural network to locally optimise torques while considering joint limits. A more recent work has been presented by Zhang et al. [45], proposing a unified approach for velocity and acceleration level quadratic programming solution methods, and introducing a new neural networks based solver featuring piecewise linear dynamics, which they show to be able to tackle the problem of local optimisation both on velocity and acceleration level. Recently, Faroni et al. [17] proposed a predictive-control based method to ensure respect of kinematic constraints, and aiming at avoiding situations where a manipulator moves in such a way that kinematic constraints cannot be respected in future steps.

Some other prediction techniques for inverse kinematics, unrelated with quadratic programming, are available in literature, and they usually involve nonlinear model predictive control (NMPC). It is worth mentioning Rybus et al. [46], who presented an NMPC for non-redundant free floating robots, aimed at minimising control action. The implementation they present doesn’t seem to be feasible in real-time. In fact, to the author knowledge, NMPC has not been applied to the general inverse kinematics problems in real-time yet. Another work, by Schuetz et al. [47], features an MPC solution based on Pontryagin Minimum Principle to reduce velocities and accelerations norm. However, it does feature a large time-step (10ms), and differentiability problems on velocities and accelerations. Furthermore, it features a small horizon which can only be improved at the price of renouncing to real-time capabilities. In fact, an issue for the application of predictive control is that robotic manipulators inverse kinematics requires a very high time resolution (in the order of milliseconds, or at least hundredths of second), while at the same time a good performance respect to optimality can only be reached when a large part of the trajectory is considered at once (tenths of second). Thus, an NMPC-based control would need to overcome these complications.

2.2.3 Optimal Control based global methods

Alongside local (and online) inverse kinematics method, offline methods based on optimal control have been pursued. These techniques are slower, but allow to find minima of a global cost function, or at least a function that takes into account a wider time horizon than local algorithms, although this happens at the price of much longer execution times. They usually minimise a function constituting the sum of error and "control cost" (usually joints kinetic energy or actuators torque) on a predetermined set of points, but might be used for other metrics such as execution times or such. The set may correspond to the whole trajectory or to a finite number of points after last completed integration step. Some authors have already been mentioned to use a variational approach provide local results that can be exploited for global solutions by solving a two-point boundary value problem. This is certainly one of the most used techniques, as is the Pontryagin maximum principle. Calculus of variations is based on variations, which are small changes in functions and functionals. A solution for this kind of problem can be derived considering the Euler-Lagrange equation of the calculus of variations. While the method allows to find analytical expressions for complicate solutions of optimisation problems involving integral costs, it is also prone to a number of limitation related to the need to solve a TPBVP in order to exploit such solutions. Furthermore, they make use of the Jacobian in their formulation, which leads to numerical instabilities during the solution process. Pontryagin maximum principle, on the other hand, is a method that establishes certain conditions under which a continuous optimal control problem can become a more manageable TPBVP on a limited set of points, through the use of a function called Hamiltonian [48], which makes the problem easier to solve than its original form, although still a difficult one due to its TPBVP nature.

The first authors to use an optimal control approach, to our knowledge, are Nakamura and Hanafusa [8], who exploit Pontryagin maximum principle to derive a minimum joints velocities norm solution, and provide the mathematical formulation for one based on integral of torques. Another early paper worth mentioning is by Suh and Hollerbach [16] which provide a global solution that minimises the joint torque integral along a trajectory through the use of calculus of variations. The proposed algorithm, however, has a complex mathematical formulation involving the second derivative of torques, and requires to solve a Two-Point Boundary Value Problem involving acceleration and jerk of the manipulator joints. Even then, as the author mentions, even with improvements it can at most consider joint limits, while no constraints can be introduced on velocities, accelerations, or other variables. Furthermore, methods based on calculus of variations have been showed to not necessarily lead to the best optimum, as observed by Martin et al. [9], who showed that the solutions of a specific global problem are not only several, but also do generally not lie in the same homotopy class. This is, they can’t be continuously deformed into each other, meaning an algorithm based on derivatives cannot reliably find them.

Despite their limitations, Pontryagin maximum principle and calculus of variations have been used to minimise different cost functions in literature. For example, some attempts have been done to minimise task execution time (i.e. time to move the manipulator end effector along a specific geometric path). First one, to author's knowledge is authored by Galicki [49], who showed that the optimal solution saturates some of the actuators while following the trajectory by adjusting the others. Similar results are also observed by Wa and Watanabe [50], and will indeed by observed in this thesis as well. Other attempts were made to optimise distance from joint limits, by Zhen-Lei Zhou et al. [51], which is based on adding terms to the cost function. This work also attempts to implement a periodic constraint on the motion, which bears some importance in industrial environments, since robotic arms are meant to perform cyclic tasks. Indeed, the global algorithm presented in this thesis also features the ability to produce periodic optimal trajectories. A work based on Pontryagin Maximum Principle to minimise energy consumption is presented in [52] by Halevi et al. for electromechanical linear actuators. This work features a mathematical model of the actuators to be included in the cost function evaluation, but the adaptability of the approach for revolute joints is doubtful.

Several other works address the issue of offline energy minimising motion planning. Schiller and Dubowsky [53] proposed a two phases algorithm, based on discretising the workspace as a grid first, and performing a local optimisation once the best path has been found on the grid. A variational, Jacobian-free approach was instead sought by Hirakawa et al. [18], who proposed an algorithm to find the minimum electrical energy consumption along a trajectory. This work is especially interesting for its cost functions, which introduces modelling of electrical motors, and because it approximates the solution through spline interpolation, a concept that will be exploited in this thesis as well to formulate an iterative solution of the global problem. In [54] Saramago et al. investigate offline path planning for non-redundant manipulators using a multi-objective cost function involving both trajectory time, mechanical energy, and the presence of obstacles. Also worth mentioning is that Ferrentino et al. [23] designed a Genetic Algorithm (GA) to perform time-optimal control of non-redundant robotic manipulators along specified paths, subject to torque constraints. Their method is suited for avoiding torque jitter close to singular points, which traditional solvers struggle to tackle.

Recent works aiming at time optimality were presented by Reiter et al. [55], who proposed a solution for the time-optimal path tracking problem of kinematically redundant manipulators that takes into account the technological limits of the robot, which is formulated as a nonlinear programming (NLP) problem solved with a multiple shooting method. In [56], the same authors presented a contribution to the solution of the time-optimal trajectory planning problem for kinematically redundant manipulators. In the proposed approach, the problem is divided into the trajectory optimization and an underlying inverse kinematics problem. The former is solved using a numerical computation scheme, augmented to fully exploit redundancy in an optimal way such that the latter problem yields optimal results.

2.2.4 Dynamic programming based global methods

Some of the most recent approaches to inverse kinematics are based on dynamic programming, which allows to solve the global inverse kinematics problem, although with some limitation. This is a method based on portioning a big problem into subproblems that are easier to solve, and whose result is related to the result of the bigger problem through a relationship known as the Bellman Equation [57]. Particularly, it requires a discretisation of both time and input parameters, which leads to a very high dimensionality. Despite this drawback, it has been proposed because it allows to incorporate different constraints and most importantly it avoids to solve TPBVP, which are very similar problems than the ones this thesis aims to solve. Among most notable works, Ferrentino et al. [58] discusses the flexibility of dynamic programming for multi-objective optimisation and for finding solutions in different homotopy classes and with different type of constraints. The numerical example he uses aims at the optimisation of the integral of joint velocities, thus no solution is provided for kinetic energy and torque. Dynamic programming approaches were previously sought by Guigue et al. [59], who proposed an algorithm to solve multi-objective optimisation problems for a 7-Degrees-of-Freedom (DOF) manipulator. Their work is particularly relevant because it is one of the few attempts, in the author’s knowledge, to solve a bi-objective optimisation problem with a different method than weighted squares in robotics.

Another approach to multi-objective optimisation in robotics has been proposed by Pashkevich et al., who proposed multi-objective optimization algorithms [60], [61] via graph representation of the search space and dynamic programming procedures, which allow generating smooth manipulator trajectories within acceptable time, considering simultaneously kinematic, collision, and singularities constraints. Also, Gao et al. [62] proposed a methodology based on dynamic programming to optimize the robot and positioner motions in redundant robotic system for the fiber placement process, which allows user to find time-optimal smooth profiles for the joint variables while taking into account maximum joint velocities/accelerations and collision constraints. Nevertheless, none of these works considers quadratic cost functions, which are necessary for the minimization of kinetic energy, joint torques, or reaction forces/torques.

Dynamic programming was also sought by Field et al. [63], who present a solution of the energy minimisation problem based on it. Their intuition does not serve the purpose to find the global optimum, since each one of the iterations only explores a subspace of the search space; however, it manages to avoid mediocre local minima, and keeps execution time at reasonable length by avoiding the curse of dimensionality usually involved with such kind of algorithms. Finally, Nurmi et al. [64] recently presented a dynamic programming solution for hydraulically powered redundant manipulators. Their approach works sensibly better for hydraulic actuators than general energy minimising algorithms, proving that actuators models might be beneficial for specific problems. However, the burden of the computational complexity means this method cannot be used online.

2.2.5 Point-to-point motion

The works mentioned so far are focused on the manipulator having to follow a specific geometric path in the Cartesian space. However, a consistent body of literature also exists for point-to-point problems, which are by nature a global problem. This is particularly interesting for this thesis despite it does not focus on the inverse kinematics problem, because it is one of the problems for which the robotics community made consistent and heterogeneous use of nonlinear optimisation algorithms. Most of the attention in this direction has been focused on obstacle avoidance through evolutionary algorithms, especially genetic ones (GA). This has been due to several advantages they have over other methods: they allow to find a global solution, do not imply the use of derivatives or gradients, and are suitable for a wide range of constraints. First author to extensively describe the possibility to use this methods in robotics, to the best of our knowledge, was Davidor [65]. Since then, many different works have been published on the topic. Shintaku [22] proposed a GA based solution to find the minimal energy solution for an underwater manipulator, based on solving the optimal control problem as a two-boundary value problem. McAvoy et al. [66] proposed to combine B-spline and GAs to obtain an optimal trajectory for a redundant manipulator tasked to perform a pick-and-place operation. Tian and Collins [67] present a floating point GA-based solution able to overcome obstacles and minimise joint displacements in the process. Collisions free trajectories were also proposed by Ata and Myo [68], who used a Generalised Pattern Search based on merging Genetic Algorithms with direct search. Saravanan and Ramabalan [69] proposed a complex cost function integrating transfer times, singularity avoidance, accelerations and other parameters, and obtained very promising results through Differential Evolution.

Another kind of evolutionary algorithm is the Particle Swarm Optimisation (PSO), which has also been exploited to solve robotics problems. Worth mentioning are Stevo et al. [70], who used it to calculate a point-to-point trajectory optimised respect to several different objectives (minimum time, energy consumption, joint displacement), and Hansen et al. [71], who used it to minimise electrical energy consumption with a realistic model of actuators and losses. More recently, PSO was used to plan a point-to-point motion of a 7'DOF arm by Jin et al. [72], who used it to plan a motion that could avoid singularities and obstacles.

2.2.6 Offline mapping based methods

Finally, some authors sought a hybrid approach based on using offline calculated global solutions to find an optimum in real time. To the author's knowledge, first attempt in this direction is presented by D'Souza et al. [73], who tackle the problem of learning a non-convex non-linear problem such as inverse kinematics. For such problems, averaging among learned solutions does not usually provide another valid solution; this problem was solved by the authors by constraining the learning set through a learning algorithm called Locally

Weighted Projection Regression, which allows learning of inverse kinematics only on localised region where convexity holds. The problem was further studied by Berenson et al. [74], who proposed a practical implementation of the learning process based on a planner and a retriever of past trajectories working in parallel. The planner would be used for new trajectories, while a retriever would be used to extract similar ones stored in memory and adapt them. The advantage of the method is that it does not need to calculate trajectories in advance, while at the same time it can improve its results through machine learning when the trajectories library is big enough. More recently, Hauser [75] proposed to create an offline problem optimum map, to use as a tool to dramatically reduce the search space during online calculations. His paper establishes theoretical foundations to assess the characteristics of the training data required to achieve a certain approximation quality. Finally, Raja et al. [76] presented a learning framework for the inverse kinematics of a mobile manipulator on uneven terrain. Their method deals with joint constraints, and with the necessity to keep the wheels on the ground, and maximize the robotic arm manipulability.

2.3 Kinematic planning for free-floating manipulators

2.3.1 Uses of free-floating manipulators and notable technology demonstration missions

Manipulators are generally meant to substitute humans in highly repetitive and time-consuming tasks. In orbit, this becomes even more important, since astronauts time is precious and limited, and furthermore humans require complex infrastructure and minimal exposure to risks. A list of tasks that are accomplished, or could be accomplished, by manipulators, include: EVA support, spacecraft deployment, assembly and maintenance, inspection, refuelling. More in detail, they can be described as follows:

- EVA support: astronauts need robotic support during their extra vehicular activities, in order to hold themselves or their tools in place, to grasp floating objects, and other tasks. The most important example of this use of space robotics is the fixing of the Hubble Space Telescope, which was first held in 1994, and again in 1997, 1999, 2002, 2009. During this operations, robotic manipulators were mostly used by astronauts to anchor themselves.

- Spacecraft deployment: for decades now, robotic manipulators have been used in space to deploy spacecraft or retrieve them. This also includes berthing (and de-berthing) from a space station.

- Assembly and maintenance: this use is typical of the ISS and consists in moving equipment and instruments around the station. As already mentioned, most of the manipulators on-board the ISS can be relocated, allowing to support human activities across the whole station.

- Inspection: instruments mounted on robotic arms can be used to inspect parts of the spacecraft that are normally unreachable. The most notable example is the Inspection Boom Assembly (IBA) used on the Space Shuttle as an extension to the SRMS. Being nearly 15 metres long, it permits visual inspection of the thermal shielding of the Space Shuttle, increasing the safety of the re-entry manoeuvre [77].

- Refuelling: extending satellites lifespan has always been a priority in space industry, and one of the bottlenecks is the quantity of fuel available on board for attitude and orbit control. NASA is one of the frontrunners to demonstrate such technologies in space, having operated a technology demonstration mission with Canadarm2 and Dextre, during which a satellite was refuelled after manipulating protective coatings, valves and pipe caps [78].

In addition to fully functioning robotic systems mentioned in the introduction, several other technology demonstration missions have been developed, or are under development, in order to make space manipulators able to accomplish as many tasks as possible. The first one was the Experimental Test Satellite VII (ETS-VII) developed by JAXA, which featured a 6-DOF robotic arm, and was launched in 1997 to verify some technologies and aspects of autonomous rendezvous and docking, and robotic servicing [79]. This included capture and berthing of a target satellite, which was however tied to avoid an accidental loss [80]. Another interesting demonstration mission was Orbital Express mission, launched in 2007, and developed by Boeing and Defense Advanced Research Projects Agency (DARPA). It featured technologies related to providing several on-orbit services, such as autonomous rendezvous and docking, in-orbit refuelling, and replacement of special components (an autonomous robotic arm transferred backup battery and computer to a target spacecraft, especially designed to be accessible for servicing [81]). DARPA projects also include Spacecraft for the Universal Modification of Orbits (SUMO), which later evolved in Front-end Robotics Enabling Near-term Demonstration (FREND). The former was initiated in 2002 to combine a detailed stereo photogrammetric image with robotic manipulators in order to grasp an existing spacecraft for servicing [82]. The latter was meant for a way broader objective: enabling autonomous rendezvous and docking with a spacecraft not explicitly designed to allow on-orbit servicing [83]. It featured a 7-DOF robotic arm and associated control system, and it accomplished a full-scale laboratory demonstration of autonomous grappling of a variety of possible interfaces [84], [85]. Afterwards, the program has further evolved to PHOENIX, which was aimed at recycling some components of old satellites in geosynchronous orbit [86], and today it is known as Robotic Servicing of Geosynchronous Satellites (RSGS), focused on providing a modular robotic manipulator toolkit to be added to a commercial spacecraft, to make it able to provide services to satellites in GEO orbit, mainly payload substitution or update. In 2005, NASA launched Demonstration for Autonomous Rendezvous Technology (DART) [87], with the objective to validate innovative components and algorithms for autonomous rendezvous and proximity operations. A mission planned, but so far never turn into practice, was Technology Satellites Demonstration and Verification of Space Systems (TECSAS), jointly developed by DLR, CSA and Russian Space Agency [88]. It consisted in a servicing satellite equipped with a robotic arm, and a target microsatellite, and it was meant to perform several phases of the on-orbit servicing, from rendezvous, to flying inspection, capture, stabilisation, manipulation of the target satellite, and control, either active with telepresence or passive during autonomous operations. Although the project was discontinued due to the participating agencies having different priorities, DLR still chose to continue with the development, renaming it Deutsche Orbital Servicing (DEOS). The objectives were reset to evaluate procedures and operations for capture and deorbiting of a non-cooperative spacecraft [89].

2.3.2 Inverse kinematics of free-floating manipulators

When a satellite equipped with a robotic manipulator has approached a non-cooperating spacecraft, the motion of the manipulator itself must be planned in order to ensure the target can be properly grabbed. This is mostly an inverse kinematics problem. Its analysis can start from the point that both linear and angular momentum are preserved in absence of external forces; however, the system is non-holonomic, as shown by Masutani et al. [90] in a milestone paper for space robotics. They consequently showed the conventional control method for industrial robots, based on local feedback at each joint, is not suitable for grasping a floating object. This problem has already been mentioned in the introduction, and its implication is that the motion of the base spacecraft and of the robotic arm are couple by a constraint on angular momentum, which is constant for the whole system. The main issue regarding this constraint is that non-holonomicity implies non-integrability, which is in fact what does not allow a simple joint position control.

Consequently, a first path planning attempt was made by Vafa and Dubowsky [91], [92], who presented a cyclic motion trajectory of a manipulator's joint, defined "Self-Correcting motions", to change the base orientation. Their idea was that a nominal trajectory was first selected and then, if any sensible deviation of the base orientation occurred, it would be corrected by adding small cyclic motions to the joints. The same authors later developed a tool called Disturbance Map (DM) in [93], which was meant to help selecting paths by identifying the direction of joint movements that would cause the maximum or minimum disturbance. This method was however pointwise, meaning that disturbances would only be calculated at specific points. Their work was further improved into an Enhanced DM (EDM), which was developed by Dubowsky and Torres [94], who also managed to use it to effectively plan the manipulator motion while reducing the reactions on the base spacecraft.

A fundamental result for the solution of the inverse kinematics problem for free-floating manipulators has been presented by Umetani and Yoshida [19], who introduced the use of a generalised Jacobian to control the manipulator. This Jacobian incorporates the dynamics of the base of the manipulator, allowing to include in the computation the end-effector motion caused by the non-holonomic constraint on the total momentum of the system. This result is of fundamental importance for the control of free-floating manipulators and can be used attitude control of satellites in general. It allows to use algorithms developed with traditional Jacobians for control of space manipulators, and indeed it will be used in this thesis to illustrate the possibility to use algorithms both for ground-based and free-floating manipulators. It is interesting to notice that, as the base inertia increases, its behaviour becomes more similar to the Jacobian of a ground-based manipulator with the same configuration, perfectly corresponding to it for an inertia tensor going to an infinitive value. This result is of fundamental importance for the field and its mathematical formulation is presented in detail in next chapter.

Although the generalised Jacobian allows effective kinematic control of free-floating manipulators, the fact that it incorporates dynamic characteristics of the system leads to the existence of a different kind of singularities, named dynamic singularities, which were investigated by Papadopoulous [7]. These are typical of free-floating manipulators and depend on the dynamic characteristics of the spacecraft. In order to avoid them, he developed a path planning technique in the Cartesian space [147], based on calculating the Path Dependent Workspace (PDW), which is the part of the workspace that can induce singularities, and subtracting it to the reachable workspace, in order to obtain a Path Independent Workspace (PIW), which is guaranteed not to cause dynamic singularities. Some years later, Siciliano and Sciavicco [95] proposed a task-space augmentation method to include the base motion as a constraint in the generalised Jacobian, adding a row for each constrained base motion. This formulation cold be shown to be equivalent to the fixed-attitude-restricted Jacobian Matrix proposed by Nenchev et al. [96]. Some researchers, such as Cocuzza et al. sought to minimize the reaction torque transferred to the base spacecraft in redundant manipulators [97]. This was done through different methods (Jacobian pseudoinverse and constrained least squares), and tested with a 2D manipulator, both in laboratory conditions and during a parabolic flight. Other researchers, such as Sabatini et al. [98], developed a coordinated control for the arm and the platform, aimed at reaching the Reaction Null Workspace by moving the platform first, and grab a manipulation target without further reactions on the base spacecraft afterwards. A different approach, certainly worth mentioning, was sought by Yoshida [27], who proposed a control able to activate the attitude control system based on a feedforward term computed from the conservation of momentum. A similar but more sophisticated, torque-based approach, was instead suggested by De Stefano et al. [99].

2.3.3 Path planning for free-floating manipulators

Alongside techniques to compute the inverse kinematics of a robotic manipulator, methods have been proposed to plan the motion of the manipulator in relationship of a target end effector position and velocity at a given time, without constraining the full trajectory, except for the need to keep the base spacecraft motion as limited as possible. Noticeably, Pandev and Agrawal [100] proposed a method called Mode Summation, to plan a Cartesian path of a free-floating system with prismatic joints. It is of particular interest since it avoids inversion of the Jacobian matrix and also provides a singularity-free path for the end effector, although it is to be considered that no requirement was formulated for the spacecraft final attitude.

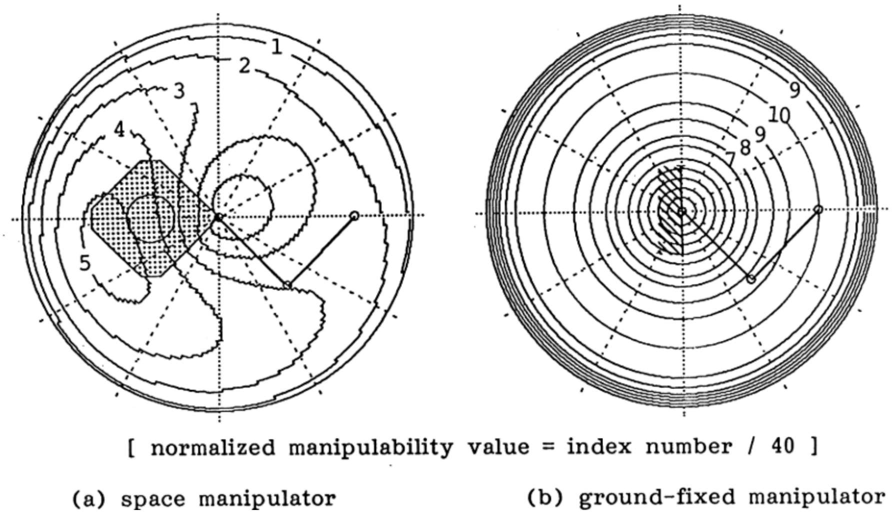
A further advancement was produced by Nenchev et al. [101], [102], who introduced the Reaction Null Space (RNS), representing the manipulator motion that causes no disturbance to spacecraft attitude. A reactionless trajectory generation strategy based on this was proposed by Piersigilli et al. [103] in order to find a path which would not affect base attitude. An approach based on Reaction Null Space was also proposed by Pisculli et al. [104].

In order to deal with the dynamic singularity problem of the Cartesian space path planning, Xu et al. [105] chose to use the direct kinematic equation instead. Their method exploits a parametrization of the joint trajectories by polynomial or sinusoidal functions, and a numerical (Newtonian) method to find the parameters through several iterations. This didn't address the possibility to go through different paths to reach the desired pose (due to the nonholonomic nature of the problem), and the long calculation times required for numerical methods, which didn't allow for real-time use of their findings. The problem of planning the path of a free-floating robot with angular momentum was first addressed by Yamada et al. [106], [107]. By proposing a variational optimisation approach in the joint space, they managed to obtain the required change in satellite attitude only by joint control. Suzuki and Nakamura [108] showed that a free-floating space robot with 6-DOF cannot follow an arbitrarily desired trajectory in the 9-D coordinate space (three for the base spacecraft and six for the manipulator) without making use of some additional control other than the joints' one. Then, they proposed to solve the problem of following a 9-D path in such a way by introducing a perturbation around the path itself, resulting in a Spiral Motion. Yoshida et al. [109] proposed the concept of Zero Reaction Maneuver (ZRM), which is obtained by equating to 0 the base velocity in the angular momentum equation. Its existence is however limited to 6-DOF manipulators. Another original strategy was formulated by Nakamura and Mukherjee [110], featuring a path planning scheme that deals with the total nonlinearity of the whole satellite/manipulator system. The idea behind it is to control both the base and the manipulator by only actuating the manipulator joints, and it was called Bi-Directional Approach, which was however affected by singularities, and only provided non-smooth trajectories. The latter problem was addressed by Papadopulos et al. [111], who proposed a path planning technique based on smooth, continuous polynomials, which can control both the endpoint location and the base attitude. Further work showed that it could be improved drastically by using high order polynomials. The same concept was later extended by Tortopidis and Papadopoulos [112] to n-DOF manipulators. Some researchers, starting with Franch et al. [113], have tried to solve the path planning problem for space manipulators by employing flatness theory. This is an extension of the concept of controllability from linear to nonlinear dynamical systems, and it can be used to design differentially flat systems, which are in turn easily linearizable and controllable. On the same path, Agrawal et al. [114] extended the method to a three-link space robot, obtaining also significant control improvements, as they managed to avoid the use of Nonlinear Programming (NLP) to solve the rate equations. Xu et al. [115] pursued a much different approach, trying to only employ direct kinematics for the path planning of a 6-dof manipulator, and solve them through a genetic algorithms based optimisation. This kind of solution provides a smooth solution, and constraints to both the motion path and the reactions on the base. Furthermore, since it does not exploit inverse kinematics, it does not incur in singularities. Same authors also proposed a similar algorithm, employing particle swarm optimisation, in [116]. However, while the first presents very long convergence time, the second may fail to converge depending on the constraints.

2.4 Workspace analysis

Many works have been published about workspace analysis of robotic manipulators, and some of those related with free-floating manipulators have already been mentioned, as they play an important role in understanding the inverse kinematics or the path planning of such systems. The oldest work that presents results on fixed-base manipulators' workspace, to the best author's knowledge, is by Kumar and Waldron [117]. Most of the techniques developed afterwards are devoted to individuating the boundaries of the workspace rather than proper characterisation. A survey of some of these methods has been presented by Haug et al. [118] in last century. A lot of effort has been devoted to Monte Carlo methods, with authors such as Cao et al. [119] and Guan et al. [120], while some authors used different search algorithms, such as branch-and-prune (e.g. Bohigas et al. [121], [122]). However, papers aiming to characterise the workspace in different ways other than individuating the boundaries are generally lacking, although some example can be found (e.g. Zacharias at al. [123]). This work presents an analysis of the manipulability of a humanoid robot and its approach is to create a manipulability map by dividing the workspace in so called reachability spheres, which quantify how many end-effector positions within the sphere are reachable. This work is however focused on producing a mapping, but does not elaborate on it relationship with energy.

In order to develop a method to study workspaces from the energetic point of view, there are two very interesting works that the author deems worth mentioning. Both of them refer to nonredundant space manipulators, but their approach can be useful for redundant and fixed-based ones as well. The first one, by Vafa and Dubowski [93], has already been mentioned in previous sections as an approach to free-flying manipulators motion planning. It proposes an algorithm for kinematic planning that is alternative to the generalised Jacobian used in this thesis. It is not as practical for kinematic planning, but it allows for an easier assessment of the workspaces size. The authors' key finding in this regard is that a free-floating manipulator workspace is influenced by the use of an attitude control system: it is in any case smaller than a fixed-base manipulator, but the loss of size can be limited by an active attitude control system, while it is maximum when the attitude is only determined by the forces ad torques transmitted to the base by the manipulator. The second paper is by Umetani and Yoshida [26], and it presents some observations about the difference in manipulability between fixed base and free-flying manipulators. The authors further expand the concepts in the previous work, and provide the interesting concept of straight-path workspace: that is, the workspace reachable by moving the end-effector on a straight path in every direction. Furthermore, they observe that, while for fixed-base manipulators the manipulability only depends on the distance from the base joint, free-flying manipulators are also influenced by the orientation of the base and the direction respect to it. **Errore. L'origine riferimento non è stata trovata.**, from the paper, illustrates the difference.



[normalized manipulability value = index number / 40]

(a) space manipulator

(b) ground-fixed manipulator

Figure 2-1 Manipulability distribution for space and ground-fixed manipulator [26]

2.5 Discussion and key findings

The field of robotics presents a huge amount of open research questions and possible approaches, and this holds true even restricting the research scope solely to inverse kinematics. The nature of its mathematical and technical problems is characterised by non-linearity and modelling hardships, and it has been tackled with a very diverse range of methods, with different objectives and scopes.

Naturally, computational techniques designed for industrial robotic manipulators evolved into maturity earlier than those invented for spaceborne systems, which feature increased complexity and a more restricted and less urgent field of application. Furthermore, they have a much higher number of applications, each one with its own specific requirements, while space robotics has a more specific scope, mostly limited to perform the manipulation with as less motion as possible of the base spacecraft.

For this reason, ground based systems feature a wide spectrum of challenges and constraints. The difference between online (local) algorithms and offline (global) ones has already been mentioned. This reflects into different applications, which allow for one or the other approach: for example, industrial manipulators can rely on offline planning and extended computational resources to solve their inverse kinematics problem, whereas robotic systems operating in on-field applications (e.g. ROVs) need to plan their motion online, or within a limited time frame at best, and their computational resources are much more limited. For ground systems, research has been produced for much longer than for orbiting systems, and the set of questions that have been asked is much wider and more established. Thus, while spaceborne manipulators have mostly seen inverse kinematics focused on base orientation control [95], [97], the number of optimisation criteria applied to ground manipulators is much wider, spacing from obstacle avoidance [53], [54] to minimum time [53], with a great deal of research on kinetic energy [15] or torque [44] minimisation.

These last two are particularly important problems in the context of redundant robotic manipulators because energy represents a key variable in manipulators operations. The reasons to minimise kinetic energy or torque are however different in nature depending on the application: for industrial facilities, the focus lies on reducing operational costs by reducing power consumption, while applications on the field, such as space, the case for power use reduction mostly lies in the necessity to not consume precious resources such as battery life. The direction for further improvement, for both global and local algorithm, lies in the limitations of the optimisation methods, which have partially been already mentioned, and can be observed by investigating the literature hereby reviewed in detail.

2.5.1 Global inverse kinematics key findings

The global methods feature limitations related to the nature of the solution methods, based on optimal control, such as Pontryagin maximum principle, or calculus of variations. In the works cited in the literature review, it has been observed that much work has been done on the use of Pontryagin maximum principle for the minimisation of various cost functions (time [53], velocities norm [124]), or applying it to specific kind of manipulators (e.g. with linear actuators [52]), while calculus of variations has been used to find explicit formulations to minimise kinetic energy and torques norm along a desired trajectory. These results all come however at the expensive cost of needing to solve a Two-Point Boundary Value Problem (TPBVP). Solving such problems can be extremely difficult and involves the computation of a first estimate of the joint space trajectory to follow, to the point that some authors [8] propose specific techniques to facilitate the problem in the case of robotics. It is also complicated by the fact that they exploit manipulators Jacobian matrices during their search for the optimum. Jacobians are prone to singularities just the same as they are for local methods, and thus they can cause instability and numerical issues in the solution process. Furthermore, imposing realistic boundary conditions can be difficult (for example, in Nedungadi [15] imposing zero velocity at initial and final time means no control over initial joints configuration, and thus end-effector position). What's more, variational methods cannot include constraints, ad Pontryagin maximum principle is only able to incorporate joint limits. Some works have been presented based on modifications of the methods to include specific kinds of constraints, such as acceleration or periodicity of the motion, but no method is currently used that can include them all. Martin et al. [9] also outlined a strong limitation in the fact that global approaches based on calculus of variations can get caught in a local optimum, depending on the nature of kinematic constraints. Thus, limitations in current global methods mostly lie in four categories:

- Necessity to solve a TPBVP, with all the complications it bears.
- Numerical problems caused by dependency on evaluations of the Jacobian matrix of the manipulator.
- Difficulties in implementing heterogeneous constraints, such as nonlinear constraints (actuators power, actuators torque) and cyclicity of the motion.
- The possibility to find the actual global optimum largely depends on the initial guess.

One more point could be made looking at the fact that all the algorithms present in literature present a formulation specifically developed for solving the problem under examination, and no general-purpose optimisation algorithm for the global inverse kinematics problem exists. Ideally, it would be beneficial to be able to optimise different cost functions without necessarily modifying the underlying algorithm, even more so if such underlying algorithm were able to efficiently tackle multi-objective optimisation problems. All these issues are properly addressed by the new global inverse kinematics algorithm developed for this thesis, the Interpolation-Based Global Kinematic Planner. This is a multipurpose inverse kinematics solution method that can address different cost functions and constraints, and can automatically produce a high number of initial guesses and evaluate the most promising ones.

2.5.2 Local inverse kinematics key findings

Online methods, contrary to global methods, are limited by the fact that their optimisation task is limited to a local scope. This leads to a number of limitations: optimality on the complete trajectory is not guaranteed, and furthermore the cost function local improvement can lead to instabilities on the long run. This has been observed very clearly in literature for Jacobian-based energy optimisation methods, which are all based on torque or acceleration optimisation [10]. These schemes are usually based on linearization of the Jacobian matrix around the current manipulator configuration, which is then exploited to compute either accelerations or velocities. This however does not take into account the problem conditioning: as explained in depth in next chapter, Jacobian matrix is not invertible for certain manipulator's configurations, called singularities. Local methods are normally not able to avoid such singularities, as they do only take into account the locally linearized relationship between joints and end-effector velocities. Mathematical tools have been developed to limit this problem (numerical damping has been already mentioned), but they come at the price of reducing tracking precision. In a milestone paper for robotics, Yoshikawa [12] proposed to use manipulability as an index to implement singularity avoidance. This is an index that depends on the Jacobian determinant, and which assumes a value of zero when the manipulator is in a singular configuration. It is still however locally optimised, which means it does only allow to locally avoid singularities but does not provide any guarantee to compute a trajectory with a reduced energy cost (they are only loosely correlated, as will be shown later in this document). Furthermore, even its main use as a singularity avoidance tool has been doubted in literature (Staffetti et al. [13]).

The other broad category of local methods is Quadratic Programming based methods, which do not necessarily involve a convergence time that is short enough to be online, and thus require ad-hoc methods to be solved fast enough. Quadratic programming constitute an improvement on classic Jacobian-based methods in that they allow to include kinematic constraints in the task more naturally, due to the nature of QP. They however come at the cost of requiring neural networks implementations, or complex solvers, although some results have been obtained in this regard, especially in the direction of respecting kinematic limits, although this still happens locally, which means that respecting the limits at a certain step of the trajectory does not imply the manipulator will be able to respect them in future. This problem has been partially tackled by Faroni et al. [17] with a predictive control, however the main issue with the quadratic programming approach remains the computational time, and the ability to predict how their local optimisation will influence future motion of the redundant manipulator.

It is evident that local inverse kinematics methods also feature limits in their ability to optimise the motion of redundant manipulators, however such limits are of a completely different nature than global methods: in this case, the question that arises is whether it is possible to develop inverse kinematics computation methods that lie in between global algorithms, able to optimise a complete trajectory at once, and local ones, which only consider one point at once. Some solutions that feature both local and global characteristics have been presented [15], but they are solved in completely different ways depending if they are used globally or locally (TPBVP or local least-squares), and in the latter case they do not feature any global characteristics. A method sometimes used in control systems to incorporate a longer time horizon than just the local one is predictive control, which however is relatively slow for solving IK, not having allowed so far an online implementation (for example [47]). Indeed, it would also be helpful to have solutions that are either less dependent on the Jacobian matrix condition number, or help keeping such condition number as low as possible. It is worth noticing that Jacobian-free approaches mentioned so far in this work either belong to the category of offline methods or are based on learning the inverse kinematics offline and approximate them online rather than calculate them.

The reasons why Jacobian matrix is so widespread in online methods despite its shortcomings are however many: it is a relatively simple and elegant solution to a complex problem, it is straightforward to implement, and guarantees a good degree of precision when far from singularities. As of today, it is widely known and comprehensively understood in the robotics community, and it is featured in an overwhelming amount of theoretical research and practical implementations. Furthermore, as already mentioned, a Jacobian-based method exists that allows any algorithm suitable for fixed base manipulators to be used for free-floating manipulators [19], and generalisations exist to include extra constraint and further optimisation tasks [6], [32], [125]. For such reasons, it sounds sensible that the use of improved optimisation methods compared to those available in literature should aim to improve the quality of traditional inverse kinematics by adding their ability to compute the solution beyond the local problem in an online fashion, rather than aiming to substitute such an established tool. Frameworks exist (most notably task prioritisation [6]) that allow to superimpose different kinematic solutions and, as already mentioned, they have been widely used to improve specific aspects of inverse kinematics solutions, such as manipulability [12] or distance from joint limits [51]. The process of developing an energy-saving online planning method for redundant manipulators in this thesis should thus be oriented in the direction of integration with existing algorithms. This set of considerations led to the development of the online algorithm presented in this thesis, which aims to complement inverse kinematics frameworks with the ability to estimate the direction of joint motion for which minimum kinetic energy will be achieved and incorporate it in the solution. This is achieved through a prediction based on a limited number of points along the trajectory. Such prediction doesn't need to be updated at every time step of the algorithm, which allows to save computational power and run the algorithm at sufficient speed to be of interest for problems which require an online solution.

2.5.3 Free-floating manipulators key findings

Free-floating manipulators are not as technologically mature as ground based systems, thus most of the research in the field of free floating manipulators aims to solve some of the fundamental issues related to the nature of such systems – which is understandable, since very few actual missions have flown, especially compared to the massive number of industrial robots, and to the massive amount of theoretical research in debris removal respect to the practical results (no single debris has ever been removed according to the author's knowledge). Thus, main points addressed so far are related to the control of systems with nonholonomic constraints [19], [126], to the dimensions and nature of the workspace [92], and to the ability to keep the base spacecraft attitude stable [95], [97].

The inverse kinematics problem for nonholonomic systems has been solved with many techniques, with the generalised Jacobian [19] being in the author's opinion the most versatile solution, not least because Jacobians are a familiar tool for the whole robotics community. The importance of this tool lies in the fact that algorithms normally adopted for ground systems can be easily ported to free-floating manipulators, although it should be noticed that extra singularities, called dynamic singularities, do exist [7], worsening the biggest downside of the Jacobian-based algorithms. This is very relevant because it allows to exploit any Jacobian-based algorithm for a free-floating system as well. However, care must be taken because it is usually advisable to keep the base spacecraft orientation constant, which results in an extra constraint. Thus, a degree of redundancy is lost in the transition from a fixed-base to a floating-base manipulator. Considering these peculiarities, it is observed that the problem of energy minimisation in a space environment has not been explored in depth, despite energy and power being scarce resources in orbit. An energy saving algorithm, in fact, can reduce the need for solar power and allow a robotic space manipulator to operate with a higher power margin. Such algorithm can be developed on fixed-base manipulators and, based on [19], be readily applied to space manipulation. A possible obstacle for this is the reduction in redundancy, caused by the necessity to keep the base stable, however room for research in this direction exists as well, since only few researchers explored the possibility to correct the manipulator disturbance to the base by using the ACS (most notably Yoshida [27]). The ACS may be used to obtain extra degrees of freedom and have more freedom in the choice of an inverse kinematics algorithm, provided that a suitable control method is developed. Based on literature findings, especially the work by Vafa et al. [91], it can be reasonably expected that the workspace size would be increased by the inclusion of the ACS in the inverse kinematics computation. This is discussed in deeper detail in next section, dedicated to the workspace analysis key findings.

2.5.4 Workspace analysis key findings

The goal of workspace analysis in this thesis is to investigate the relationship between a manipulator pose and position, and its energetic figures. In order to develop a method to study workspaces from the energetic point of view, two specific works have been identified as particularly worth attention, both on them focussed on nonredundant space manipulators [26], [93].

Both papers analyse manipulators workspace from different points of view, and provide directions to understand the workspaces of redundant manipulators, both free-flying and fixed-base. The straight-path workspace concept exploited by Umetani and Yoshida [26] is particularly interesting for the purpose of this document, in that moving the end-effector through straight paths till the limit of the workspace is reached allows to analyse it in terms of how reachable it is from a specific initial end-effector position, as opposed to analysing the reachability only in terms of existence of valid joints configurations for a certain end-effector position. Furthermore, the energetic cost of such straight paths can be quantified, allowing to observe not only how *reachable* a certain end-effector position is from the starting configuration, but also how *expensive* it is to reach it.

One more important point made by the authors of the paper [26] is that workspaces can be analysed depending on performance indexes such as manipulability. This is a sensible approach for this thesis as well: the amount of energy or power required for a manipulator to follow a specified trajectory depends on velocities or accelerations, and it is especially hard to assess in advance. On the other hand, several performance indexes have been proposed for design and performance assessment of robotic manipulators (e.g. [25]). Most of them only depend on the joint configuration (through the Jacobian or the Inertia matrix), which means they do not depend on velocities. Studying the relationships between such indexes and energy or power related performance indicators, it is possible to observe what kind of connection exists between configurations and energy requirements of a trajectory, allowing to assess the difficulties in adopting configuration-based planning for a trajectory-dependent cost function.

The work by Vafa bears instead a specific interest for free-floating manipulators, as it points at a specific direction to explore to improve free-flying manipulators workspaces:

using a control method that incorporates both the attitude and the manipulator control, the workspace of the manipulator can be extended. This suggests a possible use for energy saving algorithms for free-floating systems, since it may be argued that keeping the base fixed has a cost in terms of workspace size. Indeed, an increased size of the workspace can have positive effects on the amount of fuel required to approach a target, reducing the positioning requirement of the robotic manipulator. Control methods able to control both attitude and manipulator's joints at the same time are known in literature (e.g. Yoshida [27]), however they've never been used to perform an assessment of the workspace characteristics. This leads to the workspace analysis to be divided in two parts: an assessment of the workspace characteristics of fixed-base redundant manipulators, based on the correspondence between kinematic indexes and energetic costs, and an assessment of workspace characteristics of free-floating manipulators based on the algorithms used to control them, and on the inclusion of the ACS. The original work by Vafa can provide validation to the latter one, since it observes changes in workspace size, and the analysis presented in this thesis should of course observe them as well.

Starting from these considerations on literature, a workspace analysis is performed in this thesis using a 3-DoF manipulator. For the fixed-base case, a large amount of data has been generated by moving a manipulator end effector through rectilinear trajectories over the whole workspace, starting from different initial manipulator configurations. This wealth of data has been analysed through statistical techniques in order to find correlations with kinematic indexes of robotic manipulators, while qualitative analysis has also been performed observing the graphical results of the analysis. The problem of extending the results to free-floating manipulators through the use of the attitude control system, represented by a reaction wheel, has also been discussed and validated by simulation of free-floating manipulators workspace. This allows to consider energy minimisation algorithms as a viable alternative to reaction minimisation algorithms, and is thus discussed in detail, with some considerations about the different extension of workspaces with and without attitude control.

2.6 Conclusions

Based on the key findings observed in the different subtopics tackled in the literature review, a knowledge gap in current research can be identified in energy saving inverse kinematics strategies for redundant manipulators. The gap exists in both local and global motion planners, and it is relevant for fixed-base and free-floating manipulators alike. It is observed in the lack of algorithms implementing mathematical optimisation methods able to overcome the typical issues related with the limited scope of the local solution, and with the limitations of optimal control methods for the global problem. This thesis presents three research contributions aimed at improving the state-of-the-art in the field of energy saving inverse kinematics for redundant manipulators:

- A global algorithm that is not limited by the choice of initial condition, does not require the solution of a TPBVP, is not limited by Jacobian-related numerical issues, and can incorporate a variety of different constraints corresponding to actual operational constraints of robotic manipulators.
- A workspace study that is aimed at relating kinetic energy to kinematic indexes of a robotic manipulator, and at analysing the effects of ACS on the workspace size and manipulability distribution of a free-floating manipulator.
- A local algorithm that incorporates a prediction on future energy cost, and allows for a local solution that features more stability (less singularity issues)

and produced trajectories with a reduced kinetic energy cost compared to traditional algorithms.

The research questions presented in the introduction of this work has thus shown to be relevant and in need for an answer. Relevant material about how it is tackled will be further presented in next chapter, featuring the mathematical and implementational background of the thesis.

Chapter 3. Theoretical background and implementation

3.1 Introduction

This chapter is dedicated to present the mathematical background of the thesis, and how it has been implemented in order to develop the results presented in the reminder of this document. It is mostly a summary of all the relevant mathematics available in the literature on the field of inverse kinematics and mathematical optimisation, but it has been included as a separate chapter from literature review, because of the complexity of the relevant concepts, which deserves a separate in-depth description, and because of their fundamental role in the challenges and the solutions adopted in the reminder of the thesis. First, an overview of direct and inverse kinematics of manipulators is briefly presented. Afterwards, a discussion about singularities and workspaces is introduced, mostly re-elaborated from content by Siciliano and Sciavicco [14], unless otherwise stated. This part is then followed by an explanation of the subtleties of kinematics of free-flying manipulators. The mathematical background is concluded by a general discussion about optimisation and its most important techniques, with references to its uses in robotics. A final section of the chapter introduces the manipulator model and the mathematical expressions of the end-effector trajectories used for this thesis, alongside the structure of the simulators that have produced all the numerical results presented in the reminder of the document.

3.2 Mathematical background

3.2.1 Kinematics and dynamics of manipulators

According to Siciliano and Sciavicco [14] “a manipulator can be schematically represented from a mechanical viewpoint as a kinematic chain of rigid bodies (links) connected by means of revolute or prismatic joints.” The manipulator has two ends: one is constrained to a base, while the other is the so called *end-effector*. The latter one is the object of interest of study for the manipulator motion planning. More specifically, its motion along a specific trajectory is the main concern of motion planning. The difficulties arise from the fact that each link contributes, adding its elementary motion to the previous one. Furthermore, while obtaining the end effector position from the joint positions (*Direct Kinematics*) is a straightforward task for open chain manipulators, the opposite problem, called *Inverse Kinematics*, is not as easy. In fact, it is nonlinear and rarely features a closed-form solution, and it may have multiple or infinitive solutions, or even no solution, in reason of some peculiar manipulator structure. For the purpose of this thesis, only those manipulators with more degrees of mobility than degrees of freedom are taken into examination. These are called *redundant manipulators*, and they play an important role in robotics since, while their inverse kinematics problem almost always presents an infinite number of solutions, they also allow to take further constraints or minimization of cost functions into account, which is indeed the aim of this work.

Direct kinematics of robotic manipulators is a mathematical operation that transforms *joint space coordinates* into *Cartesian coordinates*. That is, its input is the joint vector of size $n \times 1$, with *n* number of joints, and it results in the corresponding $m \times 1$ end effector position vector in Cartesian space, with *m* being the number of end effector degrees of freedom. Direct Kinematics is usually calculated by performing a sequence of coordinates transformations from joint *0* (base joint) to *N*, in a sequential fashion:

$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \dots {}^{N-1}_{N}T \tag{3-1}$$

In this formula, each one of the *T* represents a transformation matrix from one link to the following one. Its general formulation is:

$$ {}^{i-1}_{\ i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ sin\theta_i cos\alpha_{i-1} & cos\theta_i c\alpha_{i-1} & sin\alpha_{i-1} & -sin\alpha_{i-1}d_i \\ sin\theta_i sin\alpha_{i-1} & cos\theta_i c\alpha_{i-1} & cos\alpha_{i-1} & cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-2} $$

In this formulation, the parameters represent geometrical features of the link, as per Denavit Hartenberg convention [127]. According to Craig definitions [128], and referring to figure 1, the meaning of the parameters is:

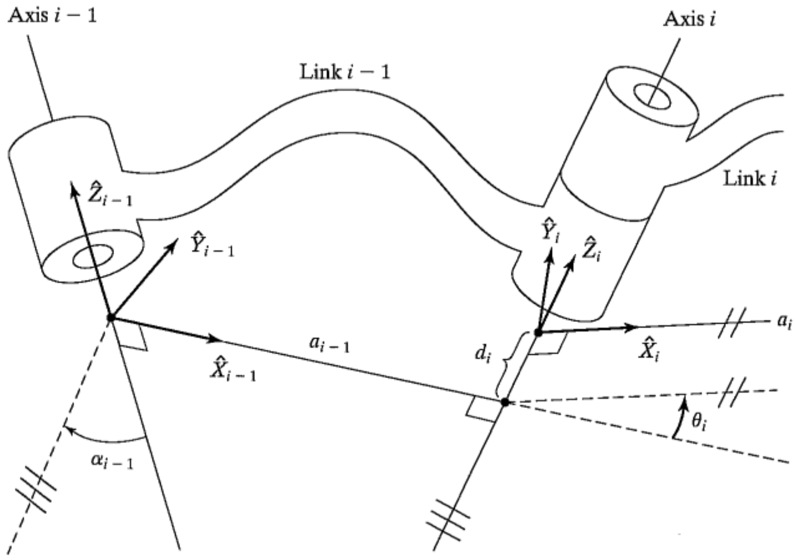


Figure 3-1 Denavit Hartenberg axis naming convention [128]

$$a_i = distance\ from\ joint\ \hat{Z}_i to\ \hat{Z}_{i+1}\ axis\ along\ \hat{X}_i$$

$$\alpha_i = angle\ between\ \hat{Z}_i and\ \hat{Z}_{i+1}\ measured\ about\ \hat{X}_i$$

$$d_i = distance\ from\ joint\ \hat{X}_{i-1} to\ \hat{X}_i\ axis\ along\ \hat{Z}_i$$

$$\theta_i = angle\ between\ \hat{X}_{i-1}\ and\ \hat{X}_i\ measured\ about\ \hat{Z}_i$$

As already mentioned, the opposite problem, called *Inverse Kinematics*, is much more complex: generally, it is possible to obtain the joint vector from the Cartesian vector only for very simple manipulators. To overcome this issue, *differential kinematics* are used.

This is because joint velocities and Cartesian velocities can be related witch each other through a linear mapping, although dependent on the configuration, called *Jacobian Matrix*, or *Jacobian* for short. The Jacobian of a manipulator is surely the most important mathematical instrument used to deal with the inverse kinematics problems. Its size is $r \times n$, where *r* is the number of degrees of freedom of the end effector, while *n* is the number of degrees of mobility of the manipulator (i.e. the number of joints). The Jacobian only depends on joint positions and, once it is known, end effector velocities can be calculated with:

$$\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3-3}$$

In this formula, $\boldsymbol{v}$ is the end effector velocities vector, $\boldsymbol{q}$ is the joint positions vector and $\dot{\boldsymbol{q}}$ the joint velocities vector, while $\boldsymbol{J}(\boldsymbol{q})$ is the Jacobian of the manipulator. This equation constitutes the aforementioned differential kinematics of a robotic manipulator. The inverse problem is expressed as:

$$\dot{\boldsymbol{q}} = \left(\boldsymbol{J}(\boldsymbol{q})\right)^{-1}\boldsymbol{v} \tag{3-4}$$

As already pointed out, this is a way more complex problem to solve, since the Jacobian might be non-invertible or ill conditioned. There are configurations at which mobility is reduced and it is not possible to impose an arbitrary motion to the end effector. These configurations are called *singularities*, and their characteristics are widely studied in robotics. When the manipulator hits a singularity, infinite solutions of the inverse kinematics problem may exist. In the neighbourhood of the singularity, small velocities in the Cartesian space determine high velocities in the joint space.

Singularities can be divided in two categories: *boundary* singularities, and *internal* singularities. The former ones are not a true issue, since they lie on the limit of the reachable manipulator workspace. The internal ones, though, can occur inside the workspace because of the alignment of two or more joints, because of peculiar end-effector configurations, or because of special relationships between dynamic characteristics of the base and the manipulator in the case of free-base manipulators. They represent a serious problem to take into account for motion planning.

A further point to take into account is that, in the case of redundant manipulators, the Jacobian is not squared (the number of rows is lower than the number of columns). This means that a suitable substitute for the Jacobian must be used in place of it. The most used method, presented for the first time by presented in [5], is to formulate the problem as a constrained linear optimisation problem, featuring the satisfaction of the equation (3-3) and the *minimisation* of a quadratic cost functional of the joint velocities:

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{W} \dot{\boldsymbol{q}} \tag{3-5}$$

Where $\boldsymbol{W}$ is an $n \ \times n$ symmetric positive definite weighting matrix. This can be solved, for example, by the *method of Lagrangian multipliers*, which uses the modified cost functional:

$$g(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{W} \dot{\boldsymbol{q}} + \boldsymbol{\lambda}^T(\boldsymbol{v} - \boldsymbol{J}\dot{\boldsymbol{q}}) \tag{3-6}$$

Where $\boldsymbol{\lambda}$ is an $(r \ \times 1)$ unknown vector that permits to incorporate the constraint (3-3) in the functional to minimise. The minimum of the functional lies where its derivatives equal zero:

$$\left(\frac{\partial g}{\partial \dot{\boldsymbol{q}}}\right)^T = 0 \quad \left(\frac{\partial g}{\partial \boldsymbol{\lambda}}\right)^T = 0 \tag{3-7}$$

Which gives the optimal solution:

$$\dot{\boldsymbol{q}} = \boldsymbol{W}^{-1}\boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{T})^{-1}\boldsymbol{v} \tag{3-8}$$

A special case occurs when the weighting matrix is the identity matrix, yielding:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v} \tag{3-9}$$

Where:

$$\boldsymbol{J}^{+} = \boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{J}^{T})^{-1} \tag{3-10}$$

Is the *right pseudoinverse* or *Moore-Penrose pseudoinverse* of the Jacobian. Its solution locally minimises the norm of joint velocities, and it is the most widespread control strategy for redundant manipulators. Another possible choice for the weight matrix $\boldsymbol{W}$ is the inertia matrix of the manipulator $\boldsymbol{B}(\boldsymbol{q})$, which is used to locally minimise the kinetic energy of the manipulator. It should be noticed that, if $\dot{\boldsymbol{q}}^{*}$ is a solution of (3-3), $\dot{\boldsymbol{q}}^{*} + \boldsymbol{P}\dot{\boldsymbol{q}}_{\boldsymbol{0}}$ is also a solution, as long as ***P*** is the null space projector for matrix ***J***. This has important consequences, since it allows the optimisation of another objective, often referred as *secondary task*, as first presented in [6]. The expression of ***P*** can be easily obtained using the *Lagrangian multipliers* again, and it yields:

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J} \tag{3-11}$$

Which can be used to generate a solution such as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{0}} \tag{3-12}$$

The second term of this expression, called *homogeneous solution*, attempts to satisfy an additional constraint specified through $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$. In other terms, the operator ***P*** produces an internal motion that does not alter the end effector position, but locally optimises another constraint compatibly with it. There are several measures that have been used for $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$, and the solutions developed in this thesis in indeed also rely on the possibility to include special tasks within an inverse kinematics framework. The most interesting extra constraint for the purpose of this document is indeed the *manipulability*, expressed as:

$$w(\boldsymbol{q}) = \sqrt{\det\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\right)} \tag{3-13}$$

This figure is one of the indexes used to assess a manipulator's mobility and its physical meaning is related to the manipulator capability to turn joint velocities into end effector velocities. In order to understand what it means, the reader shall observe the expression:

$$\boldsymbol{v}^T\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\right)^{-1}\boldsymbol{v} = 1 \tag{3-14}$$

This equation represents the points on the surface of an ellipsoid in the end effector velocity space. For the given posture, end-effector can reach high velocities when moving along the direction of the major axis, and lower velocities when moving along the directions of the minor axes. The shape and orientation of the ellipsoid depend on the quadratic form $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})$, which in turn only depends on the manipulator configuration. The volume of the ellipsoid is proportional to the quantity $w(\boldsymbol{q})$, which can be used to provide a measure of the overall freedom of movement of the manipulator, since $w(\boldsymbol{q}) = 0$ when a singularity is reached. This, however does not necessarily mean that, when $w(\boldsymbol{q})$ assumes a high value, a redundant manipulator is far from singularities, as the ellipsoid might be very extended in some directions and very narrow in others.

In order to address the issue of minimising energy and power consumption of robotic manipulators, it is necessary to also know the expression of torques. Its derivation can be again found in many robotics manuals ([1], [14], [128]), here only the formulation will be presented:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \ \boldsymbol{g}(\boldsymbol{q}) \tag{3-15}$$

Where $\boldsymbol{M}(\boldsymbol{q})$ is the joint-dependent inertia matrix, $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the term that comprises Coriolis and centrifugal forces, and $\boldsymbol{g}(\boldsymbol{q})$ is the gravity term. It is worth noticing that an index similar to kinematic manipulability can be obtained through the manipulator dynamics, through the equation:

$$\boldsymbol{\tau}^T\boldsymbol{\tau} = 1 \tag{3-16}$$

This expression represents the *dynamic manipulability ellipsoid* and its complete derivation can be found, again, in [14]. For redundant manipulators, the expression would be:

$$\left(\dot{\boldsymbol{v}} + \boldsymbol{J}\boldsymbol{B}^{-\mathbf{1}}\boldsymbol{g}\right)^T \boldsymbol{J}^{+T}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{J}^{+}\left(\dot{\boldsymbol{v}} + \boldsymbol{J}\boldsymbol{B}^{-\mathbf{1}}\boldsymbol{g}\right) = 1 \tag{3-17}$$

Where $\boldsymbol{g}$ is the gravity acceleration and all the other symbols have already been introduced. This equation individuates an ellipsoid for which, in each direction, the distance between the surface of the ellipsoid and the end effector is proportional to the accelerations that can be imposed to the end effector in that direction while respecting equation (3-16). Considering the core of the quadratic form, the *dynamic manipulability will be defined as:*

$$w_d(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}^{+T}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{J}^{+})} \tag{3-18}$$

Which, oppositely to the kinematic manipulability, is higher when the robot is closer to singularities. Both kinematic and dynamic manipulability allow to quantify the freedom of movement of robotic manipulators, and their evolution along trajectories will be used throughout this thesis for analysis purposes.

One further thing to be noticed about expressions (3-11) and (3-12) is that they can be used to add further tasks, by using the null operators to project them on the previous one. This is particularly important for manipulators with a high number of degrees of freedom, and allows for generalisation of results obtained with only one secondary task.

Even for simple robots, equation (3-3) rarely has an analytical solution. For this reason, all the work hereby presented has exploited discrete numerical integration, based on the simplest possible technique, the so called *Euler integral*:

$$\boldsymbol{q_k} = \boldsymbol{q_{k-1}} + \dot{\boldsymbol{q}}_{\boldsymbol{k}} \Delta t \tag{3-19}$$

Where *k* is a time step index. This technique works no matter how solvable the system is, but it still needs the Jacobian to be *square* and *full rank* which, apart from the already mentioned specific issues of redundant manipulators, also requires special handling in proximity of singularities [1].

Along with Euler integration, another important numerical technique is the *Finite difference* method. Finite difference generally consists in approximating the derivative of a function through the difference between two of its values at different points. It can generally be defined like:

$$\dot{\boldsymbol{q}}_{\boldsymbol{k}} = \frac{\boldsymbol{q_k} - \boldsymbol{q_{k-1}}}{\Delta t} \tag{3-20}$$

In a way that is specular to (3-19). Both formulations will be used along this document depending on the context.

3.2.2 Kinematics of Free - Flying Manipulators

When dealing with space manipulators, kinematics are different than fixed base manipulators due to the fact that the base is not fixed on the ground and, for this reason, the momentum generated by the arm will influence the basis orientation. This translates into an extra constraint on the manipulator kinematics, that the momentum remains constant along the motion. Since this constraint acts on velocities, and is in fact non-integrable, it is called a *non-holonomic constraint*. Such a constraint, for the translational momentum of system composed by *n* rigid bodies, can be expressed as:

$$\sum_{i=0}^{n} m_i \dot{\boldsymbol{r}}_{\boldsymbol{i}} = const. \tag{3-21}$$

Where m_i is the mass of the *i* element, and $\boldsymbol{r}_{\boldsymbol{i}}$ is the position of the centre of mass of the *i* element. For rotational momentum, the expression is:

$$\sum_{i=0}^{n} (\boldsymbol{I}_{\boldsymbol{i}} \boldsymbol{\omega}_{\boldsymbol{i}} + m_i \boldsymbol{r}_{\boldsymbol{i}} \times \dot{\boldsymbol{r}}_{\boldsymbol{i}}) = const. \tag{3-22}$$

Where $\boldsymbol{I}_{\boldsymbol{i}}$ is the inertia matrix of the *i* element with respect to the centre of mass, and $\boldsymbol{\omega}_{\boldsymbol{i}}$ is the angular velocities vector of the *i* element. The main issue when trying to solve the inverse kinematics of a problem involving (3-21) and (3-22), is that not even forward kinematics feature a closed-form solution, since the end-effector position depends on the inertia matrix, which in turn changes according to the joints configuration. This means that it is necessary to know the history of the postural change in order to derive a solution. Historically, two approaches have been sought: one by Vafa and Dubowsky [93] features the use of imaginary mechanical links to simulate the basis kinematic behaviour, while the other one, provided by Umetani and Yoshida [19], uses the inertia properties of the manipulator to compute a *Generalised Jacobian*, which allows to solve the problem in the same form as equation (3-4) for fixed-base manipulators. The results obtained by the first one are interesting especially concerning the analysis of the workspace of a free-flying manipulator. The latter one, however, provides an elegant solution that does not require any change from the classic Inverse Kinematics solution methods, apart from using a different Jacobian, making it extremely valuable. For this reason, algorithms that are presented in this thesis are mostly simulated with fixed-base manipulators, since their floating-base implementation has been demonstrated to be straightforward in [19].

In order to solve the inverse kinematics problem for free-flying manipulators, Umetani and Yoshida observe that equation (3-4), in this specific case, can be rewritten as:

$$\boldsymbol{v} = \boldsymbol{J_s}\dot{\boldsymbol{q}}_{\boldsymbol{s}} + \boldsymbol{J_m}\dot{\boldsymbol{q}}_{\boldsymbol{m}} \tag{3-23}$$

Where $\boldsymbol{J_s}$ is the Jacobian of the spacecraft attitude angular velocities, $\boldsymbol{J_m}$ is the Jacobian of the manipulator, $\dot{\boldsymbol{q}}_{\boldsymbol{s}}$ is the base spacecraft attitude rates of change, and $\dot{\boldsymbol{q}}_{\boldsymbol{m}}$ are the manipulator joints velocities. Considering *m* to be the End Effector degrees of freedom, and *n* the degrees of redundancy, $\boldsymbol{J_s}$ is *m x 3*, while $\boldsymbol{J_m}$ is *m x n*. This leaves the inverse problem with *m* equations and *n x 3* unknown variables. In order to fill the gap, momentum conservation equations are used. They are rewritten with the expression:

$$\boldsymbol{I_s}\dot{\boldsymbol{q}}_{\boldsymbol{s}} + \boldsymbol{I_m}\dot{\boldsymbol{q}}_{\boldsymbol{m}} = cost. \tag{3-24}$$

Where $\boldsymbol{I_s}$ and $\boldsymbol{I_m}$ are respectively the spacecraft part and the manipulator part of the inertia matrix of the system, and their sizes are *3 x 3* and *3 x n*. This, considering the momentum equal to zero, leads to the relationship:

$$\dot{\boldsymbol{q}}_{\boldsymbol{s}} = -\boldsymbol{I_s^{-1}}\boldsymbol{I_m}\dot{\boldsymbol{q}}_{\boldsymbol{m}} \tag{3-25}$$

Which can be substituted into (22) to obtain:

$$\boldsymbol{v} = (\boldsymbol{J_m} - \boldsymbol{J_s}\boldsymbol{I_s^{-1}}\boldsymbol{I_m})\dot{\boldsymbol{q}}_{\boldsymbol{m}} \tag{3-26}$$

The term $\boldsymbol{J_m} - \boldsymbol{J_s}\boldsymbol{I_s^{-1}}\boldsymbol{I_m}$, summarised as $\boldsymbol{J}^*$in literature, is the *Generalised Jacobian Matrix* of the manipulator and can be used in the inverse kinematics general case, where the manipulator base is not fixed. Its expression is generally much more complex than the fixed base case, and can lead to significant complications in computation, nevertheless it is an efficient way to solve the free-flying manipulator problem.

This result is particularly important because it shows that any Jacobian-based algorithm that is proven successful for inverse kinematics of fixed base manipulators can be used for free-flying manipulators. Furthermore, it is worth noticing that the generalised Jacobian can be used to compute the base position and orientation once the manipulator velocities are known even in cases for which the manipulator velocities have not been calculated through a traditional, Jacobian-based method. This will have important consequences on the methodology of this thesis, as next chapter will show.

3.2.3 Local versus global optimisation in motion planning

In previous chapter, the concepts of local and global optimisation have been mentioned several times, in reference to scientific findings about motion planning of robotic systems. However, the meaning of these concepts has not been systematically presented yet. When it comes to mathematical optimisation, the first thing to do is to properly define a *standard, continuous optimisation problem* [129]:

$$\begin{array}{ll} \underset{x}{\text{minimise}} & f(x) \\ subject\ to & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_j(x) = 0, \quad j = 1, \dots, p \end{array} \tag{3-27}$$

Where:

$f(x) : \mathbb{R}^n \Rightarrow \mathbb{R}$ is the objective function to be minimised over the *n-variable* vector *x*,

$g_i(x) \leq 0$ are the *inequality constraints*,

$h_j(x) = 0$ are the *equality constraints*,

$m \geq 0$ and $p \geq 0$.

When *m* and *p* equal zero, the problem is said to be *unconstrained*. The standard form defines a *minimisation* problem, however the definition for *maximisation* problems can easily be obtained by negating the cost function.

Alongside with the *standard form*, another definition is important to understand how optimisation is used in robotics, and why it is so difficult to apply it to certain problems. This is the definition of *convex function*. A function $f(x) : \mathbb{R} \Rightarrow \mathbb{R}$ is convex if:

$$\forall\, x_1, x_2 \in \mathbb{R}, \forall t \in [0,1]: \quad f(tx_1 + (1-t)x_2) \leq f(tx_1) + (1-t)f(x_2) \tag{3-28}$$

This definition is important for optimisation because it implies that, if a *local minimum* of the function is found, it is also a *global minimum*, which means the function assumes the lowest possible value in that point. *Convex optimisation problems* can be solved with a number of techniques illustrated, for example, in [129]. One possible, very simple method for optimisation of such problems is the *least squares method*. The aforementioned Moore-Penrose pseudoinverse is a possible application of *least squares*, and in fact it is used to find the minimum of joint velocities norm.

Non-convex optimisation problems are, on the other hand, way more complicated in that they feature a number of *local minima*, only a subset of them (at least one) being *global minima*. This means that, for such problems, finding a *minimum* does not guarantee the function cannot assume lower values elsewhere. This complicates the optimum search in that several classical methods, efficient for convex functions, will only find a minimum without any further guarantee about its nature. In order to overcome this limitation, global optimum methods often use *heuristics* to increase the chance to find a local optimum that is also the global minimum [130].

Heuristic methods are techniques that are developed to solve a problem by trading completeness, accuracy, and ultimately mathematical rigorousness for increased speed and chance of convergence to the desired result. They're often used when traditional methods are too slow or impractical, or simply when they do not bear enough chances of convergence. In order to analyse methods implying heuristics, a further distinction must be done, between actual *heuristics* and *metaheuristics* [131].

Heuristics are non-rigorous methods based on the nature of the problem. An expert with extensive knowledge of the field can make an *educated guess* or implement an empirical search method that will allow an optimisation algorithm to reduce its *search space* only to the part of the *solution space* where a global optimum is likely to be found. On the other hand, *metaheuristics* are general, practical methods who still lack mathematical guarantees of *non-heuristic* methods, but have been proved successful on a wide family of possible problems. They can be used as a sort of black-box when classical methods do not provide a good solution, still the available knowledge about the problem is not sufficient to provide a tailored heuristic method.

Many metaheuristic methods have been developed by imitation of natural phenomena, such as *Genetic Algorithms*, probably the most successful family of metaheuristic methods for non-convex optimisation, which found many uses in robotics, as mentioned in previous chapter. Another nature inspired method that found some inspiration in robotic is the *simulated annealing* method, while less attention from the robotics community was received by the *multi-start algorithm*, which will be used to provide a solution for the global problem presented in this thesis, indeed by the use of a problem-specific heuristic.

While all these methods share the purpose of avoiding the pitfalls of only searching for solutions in the *attraction basin* of a certain local optimum, they exploit different tactics to reach such a goal. Genetic Algorithms are inspired by genetics, and exploit a set of candidate solutions called *chromosomes*, from which they derivate new solutions through a procedure that involves an evaluation of *fitness*, *crossover* of most fit solutions, and eventual *mutation* [132]. The first step consists in giving each chromosome a score through a fitness function, which usually is the *cost function* of the optimisation problem. Following this, some of the chromosomes are selected for crossover, which implies mixing the two solutions through a specific operator to obtain *offspring*: that is, new solutions that contain traits of their parent solutions, and are hopefully closer to the global optimum. Once this process is completed, some solutions undergo occasional *mutation*, which is a stochastic change in its parameters, in order to avoid pitfalls of local minima and introduce further variety, in case some parts of the solution space were not covered by the initial population.

Simulated annealing is also inspired by nature, but in a completely different way: it mimics thermodynamic phenomena related to the freezing and crystallization of liquids, or with the cooling and annealing of metals [133]. This method has been developed by observing that solidification of crystalline materials naturally reaches a low energy state when materials are cooled slowly enough. In fact, cooling them quickly doesn’t give the atoms enough time and energy to reposition themselves and reach an ordered state. By analogy, *simulated annealing* is used for problems that feature a cost function with a high number of minima. For such problems, a slow convergence is required to find the best optimum by allowing the solution parameters to line up in a similar fashion to atoms exploiting their thermal mobility to solidify into a crystal. The subtleties of how it is done in practice are beyond the scope of this thesis, but it essentially based on the Boltzmann probability distribution:

$$Prob(E) \sim e^{-\frac{E}{kT}} \tag{3-29}$$

This equation states that a system at temperature *T* has its energy probabilistically distributed among states *E*, while *k* is the *Boltzmann constant*. This means that, even at low temperature, there's a non-zero possibility of the energy being in a higher state and, conversely, that by reducing temperature, the energy could go uphill rather than downhill. By setting a proper temperature evolution and an *annealing schedule* this can be exploited to escape local minima of the energy function (which is in fact the cost function to optimise) and find the global optimum.

Finally, Multi-start algorithms are based on the generation of several sets of initial conditions and the use of traditional deterministic algorithms, such as the gradient method, to generate a possible solution from each one of them (e.g. see [134]). This allows them to search for several local minima at once in a part of the search space, which can be the whole of it, or a portion chosen by a heuristic method. The have been somewhat less exploited in robotics, due to the difficulties in the individuation of suitable initial conditions: most choices are simply too far away from the best trajectory to stand any chance of convergence. Still, once a set of initial conditions is generated close enough to a minimum, the gradient-based nature of the optimisation implies the algorithm will converge on it in a very straightforward way and without further complications. A manipulator specific heuristic method to complement Multi-start algorithm will be proposed in this thesis.

The presented methods are only a small subset of the non-convex optimisation methods available in literature (see for example [135]), yet they have been exploited in countless variations and applications. For the purpose of robotics systems, their main use, as seen in previous chapter has mostly been *path planning* in the Cartesian space rather than *motion planning* in the joint space. Focussing on actual motion planning, some distinctions need to be made in order to understand what kind of problems can be defined from the optimisation point of view. Motion planning problems are usually solved by discretising the path in several points, called *path points* [14]. This may lead to different definitions of the problem: the motion planning might move the end effector to the next path point without considering the rest of the path, or take into account a wider scope. In the first case the solution is always local, in the latter one it might still be local or global, depending on the number of path points and the nature of the algorithm exploited.

The weighted pseudoinverse method described in previous section allows to find the minimum of weighted velocities through the least square method for a single path point. For this reason, it is usually implemented as sequential optimisation problem: at every step, the solution for the next path point is computed using the previous path point as initial condition. This method only produces a sequence of local minima and it is unreliable to optimise trajectories as a whole since the solution might deteriorate the joint mobility for the following ones. Indeed, the stability issues of such algorithms have been highlighted by several researchers (e.g. [10]).

The problem of optimising a trajectory as a whole requires instead the definition of a more complex cost function, as discussed for example by Nakamura and Hanafusa [124].

The two authors evidence that two different quantities must be minimised, one being the trajectory error, and the other one being a cost to keep low during the trajectory, which may vary depending on the actual objective of the optimisation. It can be time, but it is usually a control cost, related to the power and energy required to move the end effector along the trajectory. Most frequent choices are squared joint velocities (sometimes referred as pseudo-kinetic energy), squared torques, or kinetic energy. It must be noticed that, due to the nature of the problem, the cost function is most of the times expressed as an integral along the trajectory. In the case of Nakamura and Hanafusa, it is formulated as an integral cost subject to a trajectory constraint, such as:

$$\begin{aligned} &\underset{x}{\text{minimise}} \quad \int_{t_0}^{t_{fin}} C(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, t)\, dt \\ &\textit{subject to} \quad \boldsymbol{x_{ref}}(t) = f(\boldsymbol{q}(t)) \end{aligned} \tag{3-30}$$

Where t_0 and t_{fin} are the initial and final times, $C(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, t)$ is the cost whose integral is to minimise, and $\boldsymbol{x_{ref}}$ is an end-effector trajectory dependent on joints motion. Another suitable form is the performance index of a non-linear quadratic regulator [136]:

$$C(\boldsymbol{q}, \boldsymbol{u}, t) = \int_{t_0}^{t_{fin}} \left[\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right)^T \boldsymbol{Q}(\boldsymbol{q}, \boldsymbol{u}, t)\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right) + \boldsymbol{u}^T \boldsymbol{R}(\boldsymbol{q}, \boldsymbol{u}, t)\boldsymbol{u} \right] dt \tag{3-31}$$

Where t_0 and t_{fin} are the initial and final times, $\boldsymbol{x_{ref}}$ is the reference end-effector trajectory, $\boldsymbol{x}$ the actual end-effector trajectory, $\boldsymbol{Q}$ is the trajectory error weight matrix, $\boldsymbol{u}$ (e.g. joint velocities for kinetic energy) is the control vector and $\boldsymbol{R}$ is the control cost weight matrix (e.g. the inertia matrix for kinetic energy). The formulation (3-30) is the one that will be used mostly in this thesis. In the discrete case, it becomes:

$$\begin{aligned} &\underset{q}{\text{minimize}} \quad \sum_{i=t_0}^{t_{fin}} G(\boldsymbol{q_i}, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i, i)\Delta t \\ &\textit{subject to} \quad \boldsymbol{x}(\boldsymbol{q_i}) = \boldsymbol{x}_{i,ref} \qquad for\ i = t_0 \,..\, t_{fin} \end{aligned} \tag{3-32}$$

Where $\boldsymbol{x_{i,ref}}$ is the reference end-effector position at time i, $\boldsymbol{x}(\boldsymbol{q_i})$ the actual end-effector position at time i, $\boldsymbol{q_i}$, $\dot{\boldsymbol{q}}_i$, $\ddot{\boldsymbol{q}}_i$ are the joint positions, velocities and accelerations at time i, $G(\boldsymbol{q_i}, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i, i)$ is the cost function at time i, and Δt is the discrete time step.

In literature, solutions of sequential optimisations problems on the path points are often referred as *local solutions*, while solutions of problems such as the one expressed by (3-31) and (3-32) are regarded as *global solutions*. The terms might give room for some confusion: it is necessary to point out that a global solution to a motion planning problem is not necessarily the globally optimal solution. It has already been mentioned that some researchers, such as Martin et al. [9] noticed indeed that, under certain conditions, optimal control methods may fail to find the global optimum, as two solutions might be in different *homotopy classes*: in the language of topology, this means that the most expensive one cannot be deformed continuously into the less expensive one. In such cases, optimisation algorithms that operate in continuous space (e.g. exploiting derivatives) will fail to find the global optimum.

3.3 Implementation setup

It is frequent, in the domain of engineering, to rely on simulations for a variety of tasks. This is frequently done since experiments with actual hardware are operationally expensive, and may be risky as well, due to the possibility of mechanical failures. For this reason, an approach where simulation is done first and tests on the physical system are only performed when strictly required is considered essential for most successful engineering projects (e.g. [137]). Robotics research is no exception to this, and simulations of robotic manipulators reduce costs, as they reduce the need for testing on actual systems.

For this reason, this thesis validation of results is based on simulations. This is for several reasons: first, a high number of tests is required in order to be confident about the performance of a new algorithm, and performing them with an actual manipulator would be very expensive. Secondly, several simulation parameters need to be tuned, sometimes with a trial and error approach, which may be hard when a physical system is involved. Finally, the environmental situation might be difficult to reproduce, as in the case of free-flying manipulators, which require experimental setups able to allow the base spacecraft to move freely, or difficult to measure, such as angular accelerations of robotic manipulators.

The choice of simulation as a validation tool is thus quite straightforward. Some requirements are outlined regarding the simulation setup:

1. The system simulated must be representative of a real robot.
2. The system simulated must be representative of the kind of problem under examination.
3. The system simulated must be as simple as possible and validity of the solutions must be easy to assess.

This section of the thesis illustrates the simulation setup used to fulfil such requirements.

3.3.1 Simulators

Two different simulators have been used to perform the simulations required for this thesis, one for ground manipulators, and one for free-floating manipulators. Both are based on the finite difference formulations (3-19) and (3-20).

The two simulators use slightly different manipulators, although they bear many similarities. They represent an existing prototype robot used by University of Padova, and both have been validated and used in published research [97], fulfilling requirements 1 and 2 regarding representativity. They are planar and feature 3 degrees-of-freedom, with all degrees of freedom provided by revolute joints, which is the one of simplest possible redundant configurations, allowing for ready visualisation of results, thus fulfilling requirement 3. Geometrical and inertial characteristics of the fixed-base manipulator are as per Table 3-1.

Table 3-1 Simulated fixed-base manipulator geometrical and inertial characteristics

Link #	Mass [kg]	Length [m]	Moment of Inertia [Kg*m^2]	Centre of Gravity [m]
1	*0.615*	*0.176*	*0.001811*	*0.0950*
2	*0.615*	*0.176*	*0.003173*	*0.0717*
3	*0.307*	*0.1375*	*0.002103*	*0.0526*

For the floating-base case, characteristics are presented in Table 3-2. It can be noticed that characteristics of the base and of a reaction wheel have been added. The position of the centre of Gravity of the base is calculated respect to the base joint rotational axis, while the reaction wheel mass is considered to be part of the base spacecraft mass for what concerns the centre of gravity.

Two end-effector coordinates are controlled for both the fixed and the floating base simulator, *x* and *y* while, depending on the algorithm, base orientation φ might be controlled as an end-effector coordinate as well. In the latter case, the extra-degree of freedom would be used to keep the base fixed, thus turning the manipulator into a non-redundant one.

Hence the reaction wheel has been added, to provide one further degree of freedom with an actuator that is already available on board: this permits to retain the possibility to optimise additional tasks while the base is being kept fixed, and to observe cooperation between an Attitude Control System and a robotic manipulator being controlled concurrently. This can be done, again, by including the momentum conservation law into the Jacobian in a similar fashion to the generalised Jacobian as explained in the reminder of this thesis.

Table 3-2 Simulated free-floating manipulator geometrical and inertial characteristics

Link #	Mass [kg]	Length [m]	Moment of Inertia [Kg*m^2]	Centre of Gravity [m]
Base	2.136	-	0.1455	0, -0.0870
Reaction Wheel	-	-	0.002	-
1	0.815	0.176	0.003173	0.0717
2	0.815	0.176	0.003173	0.0717
3	0.507	0.1375	0.002103	0.0526

Summarising, the controlled end-effector coordinates are, respectively for the fixed-base and floating-base cases:

$$\boldsymbol{x} = [\, x \; y \,] \qquad\qquad \boldsymbol{x} = [\, x \; y \; \varphi] \tag{3-33}$$

As for the simulators architecture, both are structured as per *Figure 3-2*Figure 3-2.

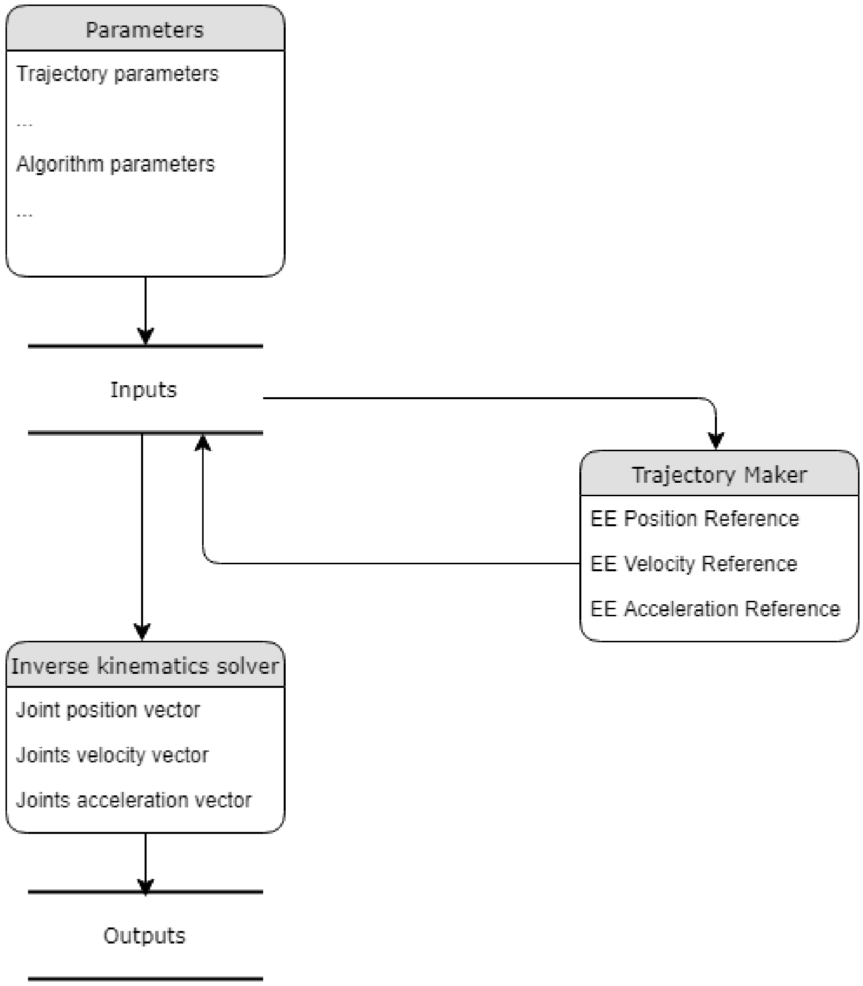


Figure 3-2 Simulator structure

A Parameters script is used to provide the inputs for the simulation. These are divided in two categories, trajectory parameters, which will be further fed to a trajectory maker script, and algorithm parameters, which will be directly used without further elaboration. The trajectory maker script's role is to output the end-effector position, velocity and reference at each path point, its inputs are:

- Joints initial configuration.
- Type of trajectory: segment, or circle.
- Direction of movement (only needed for segment).

- Characteristic dimension (length for segment, diameter for circle).

- Total trajectory time.

- Integration step.

The inputs that are directly fed into the inverse kinematics solver are instead:

- Type of algorithm.

- Integration step.

These will be integrated with the reference outputted by the trajectory maker. The inverse kinematic solver is tasked to calculate joints position, velocity and acceleration steps for each path point. Around this piece of software, three different kinds of planners have been built: a local one, a global one, and a workspace analyser. Their details will be discussed in dedicated chapters.

3.3.2 End-Effector Trajectories

End-effector trajectories are generally better suited for use when they feature smoothness and regularity, to avoid mechanical shocks to the transmission. For this reason, it is preferred that they feature continuous derivatives till the highest possible order. The end-effector velocity profiles for this thesis have been chosen according to this criterium. For this reason, it has been decided to make sure they have a smooth accelerations' profile (continuous and with continuous derivative), as shown in Figure 3-3 for a trajectory being tracked on an interval of 1 second. The equation of motion of the end-effector is:

$$\frac{d\boldsymbol{x}}{dt} = \begin{cases} 2 * \left(\frac{\frac{\mathrm{L}}{2}}{\left(\frac{\mathrm{T}}{2}\right)^2}\right) * \left(\frac{\mathrm{t}^2}{2} + (2 * \beta)^{-2} * \cos\left((2 * \beta) * \mathrm{t}\right) - (2 * \beta)^{-2}\right), & t \leq \frac{T}{2} \\ \mathrm{L} - \left(2 * \left(\frac{\frac{\mathrm{L}}{2}}{\left(\frac{\mathrm{T}}{2}\right)^2}\right) * \left(\frac{(\mathrm{T} - \mathrm{t})^2}{2} + (2 * \beta)^{-2} * \cos\left((2 * \beta) * \left(\frac{\mathrm{T}}{2} - \mathrm{t}\right)\right) - (2 * \beta)^{-2}\right)\right), & t > \frac{T}{2} \end{cases} \tag{3-34}$$

Where T is the total motion time and $\beta = \frac{2\pi}{T}$. This expression has already been used in literature [138] due to its differentiability up to the second derivative.

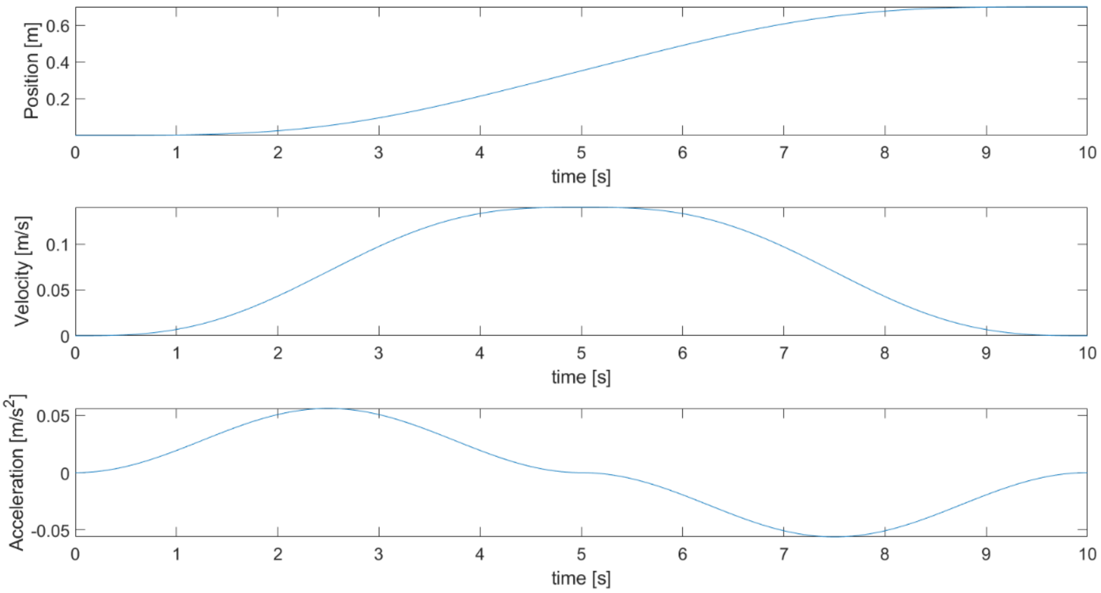


Figure 3-3 Sample End-effector trajectory

For the workspace analysis, trajectories with constant end-effector velocity has been used instead.

$$\frac{d\boldsymbol{x}}{dt} = K \tag{3-35}$$

This has the issue of a strong discontinuity in terms of acceleration at the beginning of the motion, but it allows to estimate the energy needed to reach a specific point without end-effector velocity changes unequally affecting different parts of the trajectory, and thus allows for a less trajectory-dependent analysis.

3.4 Conclusions

In this chapter, the mathematical background for this thesis has been outlined, both from the theoretical point of view, and the specific implementation point of view. First, the inverse kinematics problem has been introduced and main resolution methods have been discussed, both for fixed-base and free-floating manipulators, have been presented, highlighting how the latter ones differ with respect to traditional robotic arms. Then, the field of optimisation for robotic manipulators has been outlined, and main optimisation algorithms used for robotic manipulators have been briefly explained. After this, a rationale for the implementation choices of simulators and trajectories is presented, and the specific simulation setup used for this thesis is introduced to the reader.

Chapter 4. The Interpolation-Based Global Kinematic Planner

4.1 Introduction

In this chapter, a new algorithm is presented for the resolution of the global tracking problem of a redundant manipulator, the Interpolation-Based Global Kinematic Planner (IBGKP). It is based on nonlinear optimisation techniques rather than classic optimal control methods. This allows the new algorithm to outperform traditional methods in that, differently from existing techniques, it can find the optimum of different cost functions with minimal adaption, and it can not only handle joint motion and velocity limits, but nonlinear constraints, on torques and power, as well. Furthermore, it is able to solve cyclic trajectories (those with same joints configuration and velocities at the starting and final point of the trajectory), and it can compute multiple optima and solve bi-objective optimisation problems, while featuring much lower dependency on the choice of initial conditions than traditional methods, and not having Jacobian-related numerical problems. In order to achieve these results, IBGKP uses a specifically developed pre-optimisation step to pick most promising initial conditions for the optimisation from a randomly generated initial population, and it exploits multi-start methods to find several optima at once. It is thus able to find constrained global optima to the inverse kinematics global problem.

The chapter starts with an overview of the optimisation method hereby used, then it describes the problem to be solved in detail, and presents a first working version of the algorithm, which is however slow in converging. Thus, a more sophisticated version is introduced, which features reduced computational time and increased chances to find the global optimum. This latter, faster version is proven to effectively find optima of kinetic energy through comparison with the unconstrained results obtained with existing methods. After this, the results obtained with the new method are shown to be effective to solve the constrained problem for different trajectories and boundary conditions, especially the cyclic ones, which are relevant for industrial tasks. The chapter is concluded with a brief discussion about bi-objective optimisation problems.

4.2 Problem under consideration

4.2.1 Overview

An inverse kinematics global algorithm is an algorithm that optimises a cost function taking into account the whole trajectory. The cost function is usually an integral cost, such as the one presented in equation (3-31).

There are several factors that may increase the complexity of optimising such a cost function. A linear version of this problem, called the *Linear Quadratic Regulator* (LQR) [139], is very well understood and its solution is widely known. Most of the literature about optimal control is concerned with solving nonlinear problems that are linearizable or bear some similarity with the LQR, such as Nonlinear optimal tracking for *control affine systems*, which are expressed in the form (e.g. [140], [141]):

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x}(t))\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{x}(t))\boldsymbol{u}(t) \\ \qquad \boldsymbol{y} = \boldsymbol{C}(\boldsymbol{x}(t))\boldsymbol{x}(t) \end{cases} \tag{4-1}$$

Where the cost function assumes the form:

$$F(\boldsymbol{u},t) = \int_{t_0}^{t_{fin}} \left[\left(\boldsymbol{y}(t) - \boldsymbol{y}_{\boldsymbol{ref}}(t)\right)^T \boldsymbol{Q}(t) \left(\boldsymbol{y}(t) - \boldsymbol{y}_{\boldsymbol{ref}}(t)\right) + \ \boldsymbol{u}^T \boldsymbol{R}(t)\boldsymbol{u} \right] dt \tag{4-2}$$

Solution methods with some proof of convergence have been provided for this kind of problem, and can be used for nonlinear problems provided that form (4-1) is a good approximation for them. However, considering $\boldsymbol{x}(t)$ to be the joints trajectories and $\boldsymbol{y}(t)$ to be the end-effector trajectory, it is not possible to rewrite direct kinematics of a robotic manipulator as an control affine system.

Global optima for nonlinear problems usually require stochastic techniques to be found. This is because the structure of the problem doesn't allow to readily recognise patterns to rely on for deterministic search, thus making statistic approaches necessary. In the field of robotic manipulators motion planning, genetic algorithms have been extensively used to minimise time, torques, energy, and enforce obstacle avoidance for point-to-point problems, but, while such problems have been extensively discussed, the same level of attention has not been given to tracking problems, where the end effector desired position and velocity are defined along the full length of the trajectory.

As already mentioned in the literature review, some of the early work specifically presented for the tracking problem of redundant manipulators, tackled the global inverse kinematics problem through Pontryagin Maximum Principle [124]. However, it still has issues when applied to energy minimisation, most notably two: its computation requires to solve a two points boundary values problem (TPBVP), which is strongly dependent on the initial guess, and its mathematical formulation features high complexity and dependency on the Jacobian, which is a source of numerical problems.

Another widely used approach is the one based on calculus of variations. As mentioned in the literature review, this kind of approach allows to find an explicit formulation for the optimisation problem, which can then be solved by solving a two points boundary value problem. Nedungadi et al. [15] provided an already mentioned solution for kinetic energy, which is however prone to instability which makes the computation hard. Furthermore, it has been observed in literature [9] that trajectory tracking optimisation of redundant manipulators may feature multiple local minima, which hampers the ability of methods based on the calculus of variation to find the global one. These methods also have difficulties with respecting joint and velocity limits. Some global solution for the constrained problem has been presented in literature, for example by Guo and Zhang [142], but no general solution exists to the knowledge of the author, that allows to take both joint and velocity limits into account, not to mention torques or power limits.

A point about this can be made that it is hard to consider such limits with either Pontryagin Maximum principle or with calculus of variations. A possible solution for joint limits mentioned in the literature review by Zhou et al. [51], but it is limited in that it relies on adding terms to the cost function, and consequentially weight factors to prioritise them. This means that the optimisation becomes multi-objective, yielding sub-optimal results for each one of the terms.

In addition to these points, some researchers, such as Martin et al. [9] noticed that, under certain conditions, methods based un Euler-Lagrange equations may fail to find the global optimum, as solutions might be distributed among different homotopy classes: this means that the most expensive ones cannot be deformed continuously into the less costly one. In such cases, optimisation algorithms that are based on Euler-Lagrange equations fail if their initial conditions are not in the right homotopy class. Furthermore, not many solution methods exist for the constrained case, and they are usually limited in the cost function or constraints (usually posed on joints displacements and velocities), while their ability to find the actual global optimum, rather than a local solution to the global problem, has only rarely been discussed (an example is available in Nakamura [124]).

Despite the shortcomings of methods based on optimal control and calculus of variations, application of nonlinear optimisation techniques is not easy either, as the dimensionality of this problem can easily become very high, since it involves dividing the trajectory in *path points*, which need to be as many as possible in order to allow precise tracking. On the other hand, each steps adds *n* parameters to the optimisation, where *n* is the number of Degrees of Freedom of the manipulator. Furthermore, only small portions of the search space are relevant to the problem: most of the possible solutions feature excessive errors on trajectory tracking, to the point they are not only unworthy of consideration in order to find the solution, but considering them might considerably slow down the solution process. Even if it were possible to restrict the search space only to solutions with no tracking error, the redundant nature of the manipulator means there is still an infinitive amount of them.

Many optimisation techniques, such as Genetic algorithms, focus heavily on the ability to search a wide space of possible solutions. This case is rather different, as the global inverse kinematics problem requires to restrict the search to very specific patches of the search space, as opposed to the wide search scope of GAs and other methods. This is due to the fact that the tracking of the trajectory imposes a number of nonlinear constraints equal to the number of DoF of the end-effector at every time step. Using a method that makes use of such constraints to reduce the search space is obviously preferable. Furthermore, GAs methods developed for robotics motion planning usually exploit a parametrization of joint trajectories as polynomials, with the coefficients of the polynomial being the parameters of the optimization. The high number of constraints, which does not allow to reduce the number of parameters this way. Multi-start algorithms, on the other hand, are based on the generation of several sets of initial conditions and the use of traditional deterministic algorithms, such as the gradient method, on each one of them (e.g. see [134]). This approach allows to use deterministic algorithms that exploit augmented cost functions which include nonlinear constraints directly, such as Sequential Quadratic Programming (SQP). Multi-start optimisation algorithms exploit multiple sets of initial conditions to fully explore the search space of the cost function. They belong to the class of Random Search Methods, which have been proven to require mild assumptions to converge to the global optimum as the number of search attempts grows (for a proof, see, for example, [143]). The algorithm works as follows:

1. Construct initial condition *i*.
2. Apply a local search method to improve *i*. Let *x* be the solution obtained.
3. If *x* is the best solution, save it.
4. Repeat until a stop criterion is fulfilled.

In modern implementations, rather than repeating the steps in a sequential way, steps from 1 to 3 are performed in parallel on different processors, and the best solution is chosen among all their outcomes. It should be noticed that several local methods have been developed specifically to perform constrained optimisation without the need to weigh fulfilment of constraints against other objectives (i.e. adding further terms to the cost function), such as Sequential Quadratic Programming or methods based on barrier functions [144]. Thus, an algorithm based un multi-start methods can be easily used for constrained optimisation.

Multi-start ability to search for several local minima at once allows, depending on the chosen initial conditions, to investigate the whole solution space, or just a portion of it. This latter case is important because in the case of redundant manipulators because, if it were possible to setup the search in such a way that only subspaces of the search space with low tracking error are selected, the search for an optimum would be much easier. In fact, a possible reason why multi-start methods have been somewhat less exploited in robotics could be the difficulties in the individuation of suitable initial conditions: most choices are simply too far away from the best trajectory to stand any chance of convergence. Still, once a set of initial conditions is generated close enough to a minimum, the gradient-based nature of the optimisation implies the algorithm will converge on it in a very simple and straightforward way.

. A huge amount of methods to construct initial conditions has been proposed (for some references, the reader can look, for example, at [145]), but none has been specifically developed for robotic manipulators precision tracking. Most of the existing proposals are aimed to explore as much of the search space as possible, while here the main aim is to only explore the part of the search space containing the kinematically compliant solutions, which should be continuous and differentiable functions respecting all end-effector trajectory constraints. A good strategy is to use existing local inverse kinematics methods, such as weighted pseudoinverse, while using randomly generated weights. In such a way, arbitrarily many random solutions can be generated that respect kinematic constraints, while each one of them is potentially very different from each other.

The main intuition behind the new algorithm developed for this thesis is thus a new heuristic for multi-start algorithms, suited for generation of optimal kinematic solutions for the tracking problem of redundant manipulators. The approach hereby presented, called Global Kinematic Planner (GKP) is based on the pre-generation of sets of random pseudoinverse weights. Each one of these weight matrices can be used to obtain a different solution for the tracking problem under examination, although it is not necessarily an optimal one. The most promising sets with respect to a cost function are then picked as initial conditions for a multi-start global optimisation. This approach is an improvement respect to a general multi-start framework, in that all sets of random starting points, being solutions for the end-effector trajectory, respect kinematics constraints, although in a non-optimal way, and most of them are continuous and differentiable (those for which a Jacobian singularity occurred during their computation are not continuous and differentiable in the specific point where the singularity occurred). This allows an easier optimisation of extra costs, such as kinetic energy or torques norm integral, as opposed to optimise trajectory error and additional objectives at the same time. This has a positive effect on chances of convergence and convergence times. The method hereby presented improves on optimal control methods, in that it may look for solutions in different homotopy classes, and generally explore the whole solution space rather than the neighbourhood of an initial guess. It allows for cost functions usually hard to optimise, and it is specifically designed for manipulators kinematics. This method also improves on solutions based on the calculus of variations in that it is especially suited for constrained problems: in this chapter, not only constraints on joints positions and velocities, but also on torques and power, are shown to be effective. Not only they can be imposed, but they may even allow for faster optimisation, reducing the search space.

The main downside of such algorithm is can quickly fall into the curse of dimensionality, especially with problems with many degrees of freedom and/or path points. In order to extend the method to such problems, a further refinement, called Interpolation-Based Global Kinematics Planner (IBGKP), has been necessary. The IBGKP is based on a relaxed problem with a restricted number of parameters. The solution of this problem is then extended to a bigger set of parameters through interpolation, and further optimised, till the number of parameters of the complete problem is reached in an iterative fashion. This optimisation method is proven to be able to reach the same optimum in less time, and to be able to process a higher number of candidate solutions, providing a practical tool for optimisation of high dimensionality kinematic planning problems.

4.2.2 Mathematical formulation and constraints

As already mentioned in the chapter dedicated to mathematical background, the problem that is tackled by the IBGKP is the problem of minimising an integral cost function alongside a trajectory. This cost can be expressed in several different ways. One of them is to have a cost functions composed by two parts which weighted by gain matrices, the tracking error and a quadratic control cost. This is equation (3-32), repeated here:

$$C(\boldsymbol{q},\boldsymbol{u},t) = \int_{t_0}^{t_{fin}} \left[\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right)^T \boldsymbol{Q}(\boldsymbol{q},\boldsymbol{u},t)\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right) + \boldsymbol{u}^T \boldsymbol{R}(\boldsymbol{q},\boldsymbol{u},t)\boldsymbol{u} \right] dt \qquad \text{(3-31)}$$

Where all the symbols have the meaning already illustrated in previous chapter. Another one is to optimise a cost function subject to a trajectory constraint, such as (3-30):

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & \int_{t_0}^{t_{fin}} C(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},t)\, dt \\ \textit{subject to} \quad & \boldsymbol{x_{ref}}(t) = f(\boldsymbol{q}(t)) \end{aligned} \qquad \text{(3-30)}$$

This latter formulation is the one used for the global algorithm presented in this chapter.. The integral function in (3-30) is defined to be the control cost. Several possible choices for it can be considered, as energy consumption has been measured in different ways in literature. For the purpose of this thesis, three of them have been particularly focused. One of them is the kinetic energy integral along the end-effector trajectory:

$$E(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \int_{0}^{L} \frac{1}{2} \dot{\boldsymbol{q}}^{T} \boldsymbol{W} \dot{\boldsymbol{q}} \, dl \tag{4-3}$$

where L is the overall length of the trajectory. This function is kind of a natural choice, being a basic and established way to represent energy in mechanics, and having been already used many times in literature accordingly (for example, see [15]).

Another index that has been widely used in literature as a measure of energy consumption, and is also used here, is the integral of the squared norm of torques, expressed as:

$$F(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, t) = \int_{0}^{L} \boldsymbol{\tau}^{T} \boldsymbol{\tau} \ dl \tag{4-4}$$

This expression, widespread in literature (for example [44], [146]), is a useful operational parameter to quantify the dynamic effort made by a manipulator during a certain trajectory and will be used as well. The expression of torques can be as simplified as (3-15) or include further terms, such as friction, in which case it would take the form:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{k}_{\boldsymbol{vis}}\dot{\boldsymbol{q}} \ + \ \boldsymbol{g}(\boldsymbol{q}) \tag{4-5}$$

Where $\boldsymbol{k}_{\boldsymbol{vis}}$ are viscous friction coefficients depending on the materials interacting. The result is a damping term added to the torques. Both these expressions can be used as control costs of the IBGKP, and the one exploiting torques will be presented both with and without friction. This results in three cost functions: kinetic energy, joint torques squared norm integral and joint torques squared norm integral with viscous friction.

While control costs such as kinetic energy or torques norm integral are featured in the second term of the expression (3-13), the first term refers to the error from a reference trajectory. In fact, tracking problems such as the global inverse kinematics problem are characterised by a reference end-effector trajectory, expressed as a function of time in the cartesian space:

$$\boldsymbol{x_{ref}}(t) = \boldsymbol{f}(t) \tag{4-6}$$

However, initial conditions can be posed in different ways. For the purpose of this work, conditions on joint positions and velocities have been taken into account, since they are the most reasonable for the task a robotic manipulator is supposed to perform, especially the latter ones on velocities: realistic working conditions feature the manipulator to start and conclude its motion with motionless joints. For this reason, a condition such as what follows are used in this work:

$$\dot{\boldsymbol{q}}(t_0) = \boldsymbol{0} \tag{4-7}$$

Indicating that the initial velocities are set to zero. Initial conditions on joint positions are enforced by the fact that any $\boldsymbol{q_0}$ should respect the constraint:

$$\boldsymbol{x}(\boldsymbol{q_0}) = \boldsymbol{x_{ref}}(t_0) \tag{4-8}$$

In case no further condition is posed, the trajectory will be said to have free initial configuration in the reminder of this work, although it is not exactly free, as it still must respect the constraint on initial end-effector position. The “free” term is thus referred to the fact that joints can assume any configuration within this limit. In case a precise condition on the joints configuration is posed, the problem will be said to have constrained initial configuration. Two kinds of constrained initial configuration problems are worth being mentioned for the purpose of this thesis. One of them is said to have fixed initial configuration, expressed as:

$$\boldsymbol{q}(t_0) = \boldsymbol{q}_0 \tag{4-9}$$

While the other is only defined for closed end-effector trajectories, and it encloses the cases where it is desired that the arm reaches its starting configuration again at the end of the trajectory, with the same velocity it had at the beginning of the trajectory. In this case initial conditions are still free in the sense that they're not forced into a specific configuration. However, a cyclicity constraint is posed that can be expressed as:

$$\begin{gathered} \boldsymbol{q}(t_0) = \boldsymbol{q}(t_{fin}) \\ \dot{\boldsymbol{q}}(t_0) = \dot{\boldsymbol{q}}(t_{fin}) \end{gathered} \tag{4-10}$$

This is a case that is useful for industrial applications, since repetitive and cyclic motions are often the case for industrial manipulators. For the purpose of this thesis, only free initial conditions have been examined: it will become apparent in the rest of the chapter that the case with fixed initial configuration is easier to solve with the algorithm hereby presented, than the one with free initial configuration. Hence, simulations with fixed initial configuration wouldn't provide any extra insight.

Further consideration must be given to limits and constraints of the joints: real robotic manipulators feature joint limits, and constraints on joints velocities, torques, and power. The IBGKP is able to incorporate such constraints through the use of a local Sequential Quadratic Programming algorithm, which allows it to generate solutions that can be applied straightaway. From the mathematical point of view, the limits on joints can be expressed as

$$-q_{lim} < q_i < q_{lim} \; for \, i, \ldots, n \tag{4-11}$$

Where n is the number of degrees of freedom of the manipulator. With symbols retaining the same meaning, constraints on velocities can be expressed as:

$$-\dot{q}_{lim} < \dot{q}_i < \dot{q}_{lim} \; for\, i, \ldots, n \tag{4-12}$$

While limits on torques and power feature the following expressions:

$$-\tau_{lim} < \tau_i < \tau_{lim} \; for\, i, \ldots, n \tag{4-13}$$

$$-W_{lim} < \tau_i \dot{q}_i < W_{lim} \; for\, i, \ldots, n \tag{4-14}$$

4.3 Description of the new global algorithm

4.3.1 The Global Kinematic Planner

The new algorithm is first explained in a simple version featuring the main characteristics of the new method. This first version is called Global Kinematic Planner (GKP), and is able to provide good results, but usually has slow convergence times. A more sophisticated implementation, which has already been referred as Interpolation-based Global Kinematic Planner (IBGKP), is explained later in this chapter. This latter one features faster convergence times and allows for a more complete search over the solution space, and it is the one that has been used to compute the results provided in this thesis.

As already mentioned, the global algorithm presented in this thesis is based on the choice of suitable initial conditions for a multi-start optimisation. The underlying intuition is that a very high number of trajectories are randomly generated through the use of classic inverse kinematics algorithms. This populates a set of candidate initial conditions, where each randomly generated trajectory is a candidate initial condition for the algorithm. The set is then ranked according to a criterion that reflects the chances of that specific initial condition to converge to a good optimum. This criterion might be the cost function without any modification, or a cost function modified to take into account a penalty for the violation of constraints. A subset constituted by the initial conditions with the best ranking is then used as initial conditions for the multi-start algorithm.

More specifically, the algorithm that follows exploits weighted pseudoinverse as per formula (3-12) to generate the initial conditions for the multi-start search. As already mentioned in the Mathematical Background chapter, this is a widespread technique in robotics, when a task is meant to be locally optimised by giving different weights to each joint depending on the optimisation objective. Such objective usually is a local minimisation of the norm of velocities, but might be any local solution obtained exploiting least squares minimisation, which is indeed the case here. Reminding formula (3-12), the most usual cases, are where $\boldsymbol{W} = \boldsymbol{I}$, which has already been identified as the *Moore Penrose pseudoinverse*, and is used to calculate the minimum-norm joint velocities required to obtain a specific end effector velocity, and the case were $\boldsymbol{W} = \boldsymbol{B}(\boldsymbol{q})$, where the weight matrix is the inertia matrix. In this case, the algorithm would provide a local minimization of the kinetic energy. However, it is possible to exploit any set of weights that form a symmetric positive definite matrix, and indeed this is the case examined here, where different solutions are computed by randomly generating compliant weight matrices. The resulting algorithm is divided in three phases:

- Initialisation: random weight matrices are generated.

- Population: the population of candidate trajectories is generated by using inverse kinematics schemes.
- Optimisation: the best candidate solutions are used as initial conditions to perform a multi-start optimisation.

More in detail, considering the trajectories to be divided in n_{steps} path points, and the robot to have n_{joints} joints, the steps of the Global Kinematic Planner are hereby presented. In their description, bold notations such as q, q˙ etc… identify vectors as usual, while a notation with braces, such as $\{\boldsymbol{q}\}$, $\{\dot{\boldsymbol{q}}\}$ etc… identify a set of vectors, each one representing a different time step.

Initialization

1. Manipulator physical and inertial parameters, simulation parameters and end-effector trajectory are taken as input.
2. A set of joint configurations compliant with the desired end-effector initial position is taken as input.
3. A set of symmetric positive definite weight matrices is generated with all eigenvalues randomly assigned between 0 and 1 (excluding 0 and 1 themselves). The number of weight matrices to be generated depends on computational power, but it must generally be high enough that increasing it doesn't improve the best candidate solution anymore (see later in the chapter for further explanation). A feasible number is:

$$n_{candidates} = {n_{parameters}}^2 \qquad (4\text{-}15)$$

Where $n_{parameters}$ is the number of parameters of the problem taken into account, which corresponds to:

$$n_{parameters} = n_{path\,points} * n_{joints} \tag{4-16}$$

Where $n_{path\,points}$ is the number of points that define the desired end-effector trajectory, and n_{joints} is the number of DOF of the manipulator.

Population

4. set of $n_{candidates}$ solutions $\{\boldsymbol{q_{random}}\}$ is computed for each robot initial configuration defined at point 2, using the weight matrices generated at point 3 to weight the pseudoinverse. Each solution is obtained through formula (3-4), repeated here:

$$\dot{\boldsymbol{q}} = \boldsymbol{J_W v} \tag{3-4}$$

Where $\boldsymbol{v}$ is the desired end-effector velocity.

Optimisation

5. All candidate solutions generated during the population phase are ranked according to a criterion. In case of unconstrained optimization, the criterion is the value of the cost function, while, in case of constrained optimization, the criterion is the value of the cost function plus an extra term to penalize the violation of constraints on joint mechanical limits. This term has the same expression as that presented in [31] by Liegeois in a different context:

$$w_{JL}(\boldsymbol{q}) = k_{JL} \sum_{i=t_0}^{t_{fin}} \frac{1}{2n} \sum_{j=1}^{n} \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \tag{4-17}$$

The value k_{JL} is a weight to balance the extra term against the cost function, and the reminder of the expression is a quantity defined as the distance from the joint mechanical limits, computed for each time step and summed over the whole motion time. In Equation (14), q_{iM} (q_{im}) is the maximum (minimum) *i*-th joint limit, $\bar{q}_i$ the mean value between the two, and *n* are the robot DOF. This term is included in the cost function to penalize candidate solutions that exceed joint limits by large amounts, as opposed to excluding all candidate solutions that violate the limits, and for this reason k_{JL} is set to 0.01. In this way, the solver is still able to consider candidate solutions that exceed the limits by a small amount on a limited number of path points. Other constraints are not considered for the ranking of solutions, since in all the simulations carried out joint limits appeared to be by far the most influential constraint for the ranking. The term in Equation (4-17) is not required to be explicitly part of the cost function used for the optimization (step 6 below), since all constraints, including trajectory tracking, are taken into account as part of the SQP algorithm.

6. Multi-start algorithm is launched with the best n_{runs} candidate solutions according to the selected criterion, all the others are discarded. Joint limits and the other constraints are enforced through the *fmincon* function, which takes their expression/value as an input. The number n_{runs} depends on the computational power available and on the complexity of the problem. For the results presented in this work, n_{runs} has been set to 24.

7. Best solution is picked from the results of the multi-start.

It must be highlighted that the use of a random weight matrix might feature very different results in term of joint displacements over the whole length of the trajectory despite apparently small changes in the weights. This makes it necessary for $n_{candidates}$ to be sufficiently high, while on the other hand increasing it above a certain value will require additional computational time but will not sensibly increase the quality of the best candidate solution anymore. In order to investigate the relationship between $n_{candidates}$ and the chances to find the best optimum, the value of the kinetic energy cost function for a three-DOF planar robot has been computed for sets with $n_{candidates}$ from 1 to $10^{4.5}$ for a problem with 39 parameters, corresponding to an initial point and 12 following path points along a line, which for three DOF gives (12 + 1)*3 parameters. In Figure 1, the x-axis shows the number of candidate solutions in the set, and the y-axis shows the difference between the cost function value (C) for the best member of the $n_{candidates}$ set and the best computed value of the cost function obtained in the simulations (see Results, case 1a) for the system under consideration, according to Equation (15).

$$y = \log\left(C_{best\ candidate} - C_{best\ optimum}\right) \tag{4-18}$$

It can be observed that, for a number of $n_{candidates}$ higher than the number of parameters, such difference tends to a linear function in logarithmic scale, which corresponds to a power law in linear scale. The best fit function is:

$$y = e^{-4.0733} * n_{candidates}^{-0.2030} \tag{4-19}$$

This suggests that, for the problem under consideration, increasing the number of $n_{candidates}$ progressively reduces the distance between the initial guess and the globally optimal solution. It is noted that the value of $n_{candidates}$ obtained by Equation (12) is 1521, which falls well within the linear part of the graph.

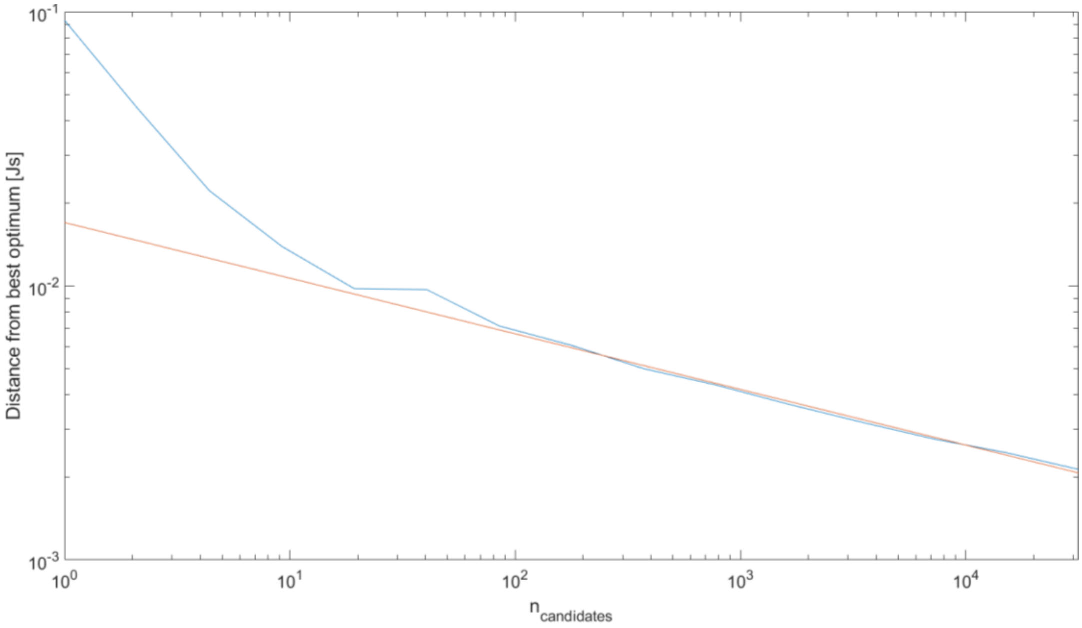


Figure 4-1 Difference between cost function value of best candidate solution and best computed value of cost function.

4.3.2 Generation of starting configurations

Step 2 of the Global Kinematic Planner involves the input of a set of initial joint configurations. However, it is most often the case where no known initial configuration is likely to be reasonably close to the one of the global optimum solution. In this case, the following method can compute more initial configurations starting from an existing one ($\boldsymbol{q}_{\boldsymbol{initial\,configuration}}$):

1. A set of $n_{path\,points}$ joint velocity vectors $\{\dot{\boldsymbol{q}}_{\boldsymbol{perturbation}}\}$ is randomly generated. In the examples provided in this paper, a normal distribution has been used, due to its effectiveness and the simplicity of implementation. Other methods are widely used for

finding initial conditions for multi-start algorithms, such as Latin hypercube [32]. This was not necessary here, but it could lead to better results when the chosen $\boldsymbol{q}_{\boldsymbol{initial\ configuration}}$ is thought to be far from the optimal one.

2. An inverse kinematics problem is solved with end-effector velocity set to zero, robot initial configuration $\boldsymbol{q} = \boldsymbol{q}_{\boldsymbol{initial\ configuration}}$, and secondary task $\{\dot{\boldsymbol{q}}_{\boldsymbol{perturbation}}\}$. This leads to a new initial configuration which does not affect the end-effector position:

$$\boldsymbol{q}_{\boldsymbol{new\ initial\ configuration}} = \boldsymbol{q}_{\boldsymbol{initial\ configuration}} + \int_{\boldsymbol{t_0}}^{\boldsymbol{t_{fin}}} (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{perturbation}} dt \tag{4-20}$$

Equation (4-20) is obtained by considering the general solution of the inverse kinematics problem with Moore–Penrose pseudoinverse, as per equation (3-12), repeated here:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{0}} \tag{3-12}$$

Where $\dot{\boldsymbol{x}}$ is the desired end-effector velocity and $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$ is a secondary task to be executed without modifying the end-effector velocity. This can be achieved by exploiting the null-space operator [33] $(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})$. If $\dot{\boldsymbol{x}}$ is set to zero in Equation (18), the joint velocities will not be affected by the end-effector velocity and will be equal to $(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{0}}$. Equation (17) is obtained by integrating Equation (4-20) from $\boldsymbol{t_0}$ to $\boldsymbol{t_{fin}}$.

The size of the set of initial configurations is an important parameter to ensure convergence on the best optimum, as the initial configuration of the manipulator influences the whole trajectory. In general terms, this set should be representative of the whole set of configurations that produce the desired end-effector initial position. Inverse kinematics as per Equation (3-12) is computationally inexpensive, which allows us to generate a high number of set members relatively inexpensively. The examples presented in this paper have been produced with a set of 66 initial configurations, which come from a trade-off between the overall computation time required and the probability to discard an initial condition that would give an optimum better than the previous ones found. Some of the 66 initial configurations may violate the manipulator joint limits. On the one hand, the ranking function will penalize initial configurations in which joint limits are violated by large amounts (which will not be processed further) and, on the other hand, the SQP optimization will always provide optimal solutions that are compliant with all the constraints, even if some of the initial configurations violate the joint limit constraints by a small amount.

4.3.3 The Interpolation-Based Global Kinematic Planner

While the Global Kinematic Planner can find global optima of the problem expressed by (3-31), its computational complexity grows with both the number of DOF of the manipulator and the number of path points. Due to this, its use might be computationally expensive for problems featuring multi-DOF manipulators or long trajectories with high precision requirements. In order to overcome this issue, a global optimal solution on a reduced set of parameters can be calculated and extended on a higher number of parameters through interpolation. Such new solution can then be optimised again with the new number of parameters. This two steps approach will reduce computational time, since the second round of optimisation starts from an initial guess that is already optimal for a similar problem. The two steps can be then repeated adding more parameters through interpolation, until the desired time step is reached (eventually, the same as the Global Kinematic Planner). This method also allows a more thorough search on the solution space: since the optimisation of a single candidate solution is much less time consuming with a low number of parameters, it is possible to first run the multi-start algorithm with a high number of candidate solution n_{runs}, and then focus the following optimisation steps on the most promising solutions obtained. This allows to save computational power and may increase the chances to find a global optimum. The method above outlined, called Interpolation-based Global Kinematic Planner, works as follows:

1. A subset of path points of the end-effector trajectory to track is selected. A sampling interval $\Delta t_{interp} = n * \Delta t$ with n integer and Δt discrete time step of the complete problem is used. Sampling can be thicker in parts of the trajectory where the cost function to be minimized is expected to be higher.
2. The Global Kinematic Planner is used to provide a solution $\{\boldsymbol{q}_{\boldsymbol{interp,subset}}\}$, as explained above.
3. A new Δt_{interp} is chosen, according to the formula $\Delta t_{interp,new} = \frac{n*\Delta t}{m}$, where m is an integer submultiple of n.
4. A new set of path points is selected with $\Delta t_{interp,new}$ as a time step.
5. Cubic splines are used to interpolate the values of $\boldsymbol{q}$ on the path points not included in the previous subset $\{\boldsymbol{q}_{\boldsymbol{interp,subset}}\}$, obtaining a complete $\{\boldsymbol{q}\}$ vector on the new set of path points.
6. A further gradient-based optimization based on SQP is run with initial guess corresponding to the solution obtained at the previous step.
7. Steps 3–6 are repeated, decreasing Δt_{interp} until the desired step size Δt is reached. Subject to available computational power, this can be as small as the one of the complete original problem.

The solution obtained at step 5 is likely to be close to the optimal solution of the full global problem since it is an interpolation of a global optimal solution of a simplified version of the problem. For this reason, step 6 can run a much easier optimization (with faster convergence) with respect to step 6 of the Global Kinematic Planner since the initial guess is already near-optimal. This allows for faster convergence than with the Global Kinematic Planner. For timing purposes, the IBGKP has been tested on a machine featuring an Intel i7 ninth generation exa-core processor with 32 GB RAM, SSD mass storage, and using Windows 10 and Matlab version 2019b. The problem under examination featured 303 parameters and the planner solved it with 4 optimisation steps, with Δt being progressively halved from an initial 0.08s to a final 0.01s, resulting in computational times around 300s and an exploration of 60 candidate solutions. For comparison, the global planner presented earlier in the chapter took times in the range of 600s for the same problem, while exploring only 6 candidate solutions. Similar computational times have been observed for all simulations in the reminder of this chapter, ranging from 160 to 400s.

Before analysing the results, it must be noticed that no explicit gradient formulation for the local search is available *a priori*, since the control cost function may vary. Due to this, numerical gradient methods have been used to obtain the results of this thesis. For the unconstrained case, used for the validation of the algorithm, a *Quasi-Newton* method has been used, which exploits the Broyden - Fletcher - Goldfarb - Shanno (BFGS) formula as proposed by [149] to approximate the Hessian of the function to optimise. For the constrained cases, the gradient optimisation has been performed through the use of Sequential Quadratic Programming algorithm [144]. This algorithm is based on the use of Lagrange multipliers to solve a problem that is equivalent to the nonlinear problem:

$$\begin{aligned} &\min_{z} f(x) \\ &s.t \qquad \begin{aligned} b(x) &\geq 0 \\ c(x) &= 0 \end{aligned} \end{aligned} \tag{4-21}$$

Given this problem, SQP solves it through a quadratic approximation of the Lagrangian function:

$$L(x,\lambda,\sigma) = f(x) + \sum_{i=1}^{m} \lambda_i b_i(x) + \sum_{j=1}^{n} \sigma_j c_j(x) \tag{4-22}$$

Where m is the number of inequality constraints, n is the number of equality constraints, and λ and σ are Lagrangian multipliers respectively for nonlinear and linear constraints. At each iteration of the algorithm, a search direction d is defined as a solution of the quadratic programming problem:

$$\begin{array}{ll} \min\limits_{d} & f(x) + \nabla \mathrm{f(x)}^{\mathrm{T}} d + \frac{1}{2} d^T \nabla_{xx}^2 L(x,\lambda,\sigma) d \\ s.t & b(x) + \nabla b(x)^T d \geq 0 \\ & c(x) + \nabla c(x)^T d = 0 \end{array} \tag{4-23}$$

The advantage of SQP compared to other algorithms, for this specific application, is that nonlinear (such as tracking) constraints in the quadratic expression (4-23) are much easier to handle than the nonlinear constraints in original expression (3-32). In all cases, the optimisation has been performed in Matlab trough *fmincon* or *fminunc* functions in MATLAB.

4.4 Results and discussion

4.4.1 Simulation setup for validation and analysis

In order to prove the capabilities of the IBGKP properly, it is necessary to perform a simulation campaign divided in two steps, one aimed at validating the new algorithm against existing methods, and one aimed at providing IK solutions that are not yet available in literature. The first step is in this case performed by optimising the kinetic energy integral alongside a specific end-effector trajectory. This problem has already been solved by Nedungadi et al. [15], and thus another solution is available for comparison. Results obtained by the new algorithm and by Nedungadi's one have been compared and their similarity used to validate the IBGKP. Once this has been done, a sequence of other solutions has been provided with an array of different constraints and cost functions to analyse the performance of the new algorithm on several possible IK constrained problems. Constraints on torque and power available by each joint have not been previously tackled in literature, hence no comparison is available.

Considering all the possible variations of cost function, limits and constraints reported in previous sections, the complete matrix of the possible conditions to simulate is illustrated in Table 4-1. They have been chosen to illustrate a wide number of specific unconstrained and constrained global inverse kinematics problems that the IBGKP can solve. The table lists in fact free and constrained motion cases, considering the three different cost functions proposed earlier in the chapter. The columns show different cost functions, which constitute the problems used as cases for illustrating the capabilities of the algorithm, while the rows shows the different sets of constraints applied to each one of them. Each cost function is defined by a number, and each constraints configuration by a letter, giving a total of 15 simulation conditions, each one identified by a number, for the algorithm used, and a letter, for the constraints used.

Simulation conditions are characterised by letters, and each one of them has been tested with a single trajectory. In the cases with letters from a to d, this is a rectilinear trajectory, while in the case e, it is a circular trajectory. Conditions a-d are solved with free initial configuration, while cases e are solved with cyclic boundary conditions, meaning that the initial and final manipulator configuration must be equal. The interest for these specific trajectories lies in the fact that the rectilinear one features multiple local optima for the integral of kinetic energy, while the circular one features tracking of the trajectory through cyclic motion, a notoriously important case for robotic manipulators in industrial environment, which is often mentioned in literature (e.g. [150]). All simulations are performed with the end-effector velocity profile discussed in previous chapter, according to the expression (3-34), in order to have continuous end-effector velocity and acceleration along the whole trajectory.

Table 4-1 IK problems used to illustrate the capabilities of the IBGKP

	Kinetic energy integral	**Torques squared norm integral**	**Torques squared norm integral with viscous friction**
No constraints	1a	2a	3a
Joints and velocities	1b	2b	3b
Joints and torques	1c	2c	3c
Joints and power	1d	2d	3d
Joints, velocity and Cyclicity	1e	2e	3e

Both trajectories start with the same end effector position, $\boldsymbol{x_{init}} = [0.4678; -0.000]$, and run for a total time T = 1 s. The rectilinear trajectory has a length of 0.40 m and reaches the end effector final position $\boldsymbol{x_{fin}} = [0.0983; 0.1526]$, while the final position is of course the same as the initial one for the circular one. The unconstrained kinetic energy integral solution for the rectilinear trajectory is also used for the validation of the algorithm against the solution method proposed by Nedungadi. Where constraints are present, their values are according to Table 4-2.

Table 4-2 Constraints used in the global algorithm simulations

Constraint	Limit without friction	Limit with viscous friction
Displacement [deg]	90 on 1st joint, 120 for 2nd and 3rd	90 on 1st joint, 120 for 2nd and 3rd

Velocity [rad/s]	3.8	3.8
Torque [Nm]	0.4	1
Power [W]	0.7	2.75

For all solutions presented in this chapter, $\Delta t = 0.01s$, which means each problem has 303 parameters, considering 3 Degrees of Freedom. All simulations have been performed by using the Interpolation-based Global Kinematic Planner, with a starting $\Delta t_{interp} = 0.08s$, leading to 39 parameters. The time step has been then progressively halved, doubling the number of parameters at each iteration. 1521 candidate solutions have been generated, and the best 48 ones have been chosen to be optimised via multi-start method. It is observed that the rectilinear trajectory has several local minima, allowing for considerations about the performance of the algorithm in such situations. All the characteristics of the simulated trajectories have been summarised in Table 4-3.

Table 4-3 Simulated trajectories characteristics

Number #	Shape	Characteristic dimension	Starting configuration
1	Rectilinear	0.40m (length)	Free
2	Circular	0.05m (radius)	Cyclic

4.4.2 Algorithm validation

In order to validate the algorithm, an unconstrained solution for the kinetic energy integral is computed and compared with a solution obtained with an algorithm available in literature, which is based on calculus of variations. Results are very similar, with a negligible advantage for the Interpolation-based Global Kinematic Planner presented here. The advantage is probably due to numerical differences caused by the implementation, on the other hand the similarity of the solutions shows that the new algorithm is for unconstrained problems as effective as a theoretically proven algorithm based on calculus of variations. The algorithm used for the validation has been proposed by Nedungadi et al. [15] and presents an acceleration-based inverse kinematics solution:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{B}}^{+}(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + (\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{B}}^{+}\boldsymbol{J})\boldsymbol{B}^{-\mathbf{1}}\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4-24}$$

Where $\boldsymbol{J}_{\boldsymbol{B}}^{+}$ is the pseudoinverse weighted with the inertia matrix, $\dot{\boldsymbol{J}}$ is the first derivative of the Jacobian, and $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, as introduced in the mathematical background chapter, is a term comprising Coriolis and centrifugal terms of the torque. A complete derivation of this equation can be found in the relevant paper, for the purpose of this thesis it is worth mentioning that this formula can be used both as a local optimisation inverse kinematic method when the initial condition problem is solved, and as a global optimisation one when the two point boundary value problem is solved. This latter case is the one used for validation here: the global optima for the kinetic energy integral of a specific end-effector trajectory have been computed with both the Interpolation-based Global Kinematic Planner and the algorithm proposed by Nedungadi. The problem features free starting configuration, for both cases a boundary condition on joints velocities at initial time t_0 has been posed as per (4-7):

$$\dot{\boldsymbol{q}}(t_0) = \mathbf{0} \tag{4-7}$$

For the Interpolation-based Global Kinematic Planner, no further condition is necessary: since the end effector velocity approaches to zero at the end of the trajectory, the algorithm naturally reduces kinetic energy (and thus joints velocities) to zero approaching the final time. On the other hand, a boundary condition at the final time t_{fin} is necessary for the Nedungadi algorithm to work:

$$\dot{\boldsymbol{q}}(t_{fin}) = \boldsymbol{0} \tag{4-25}$$

This condition allows to solve the tracking problem as a two-point boundary value problem.

The problem has been solved setting simulation parameters a explained in previous section, finding three local minima for the kinetic energy integral. The same problem has then been solved with Nedungadi variational algorithm with $\Delta t = 0.001s$, by solving the TPBVP with MATLAB routine *bvp4c*. It should be noticed that, in order to solve this problem an initial guess for each path point was required close to each optimum, while the same didn't apply for the Interpolation-based Global Kinematic Planner, which requires much less input from the user. Result for kinetic energy and mean difference in joint position between Nedungadi solution and the Interpolation-based global kinematic planner are shown in Table 4-4 (IBGKP in third column stands for Interpolation-based Global Kinematic Planner), while the three optimal trajectories are shown in Figure 4-2 starting from the best optimum found.

Table 4-4 Values of the optima for the validation trajectory

Optimum #	**Nedungadi solution**	**IBGKP solution**	**Mean difference in joints position**
1	0.0532	0.0528	$5.8*10^{-3}$
2	0.0567	0.0563	$2.2*10^{-3}$
3	0.0673	0.0671	$3.8*10^{-3}$

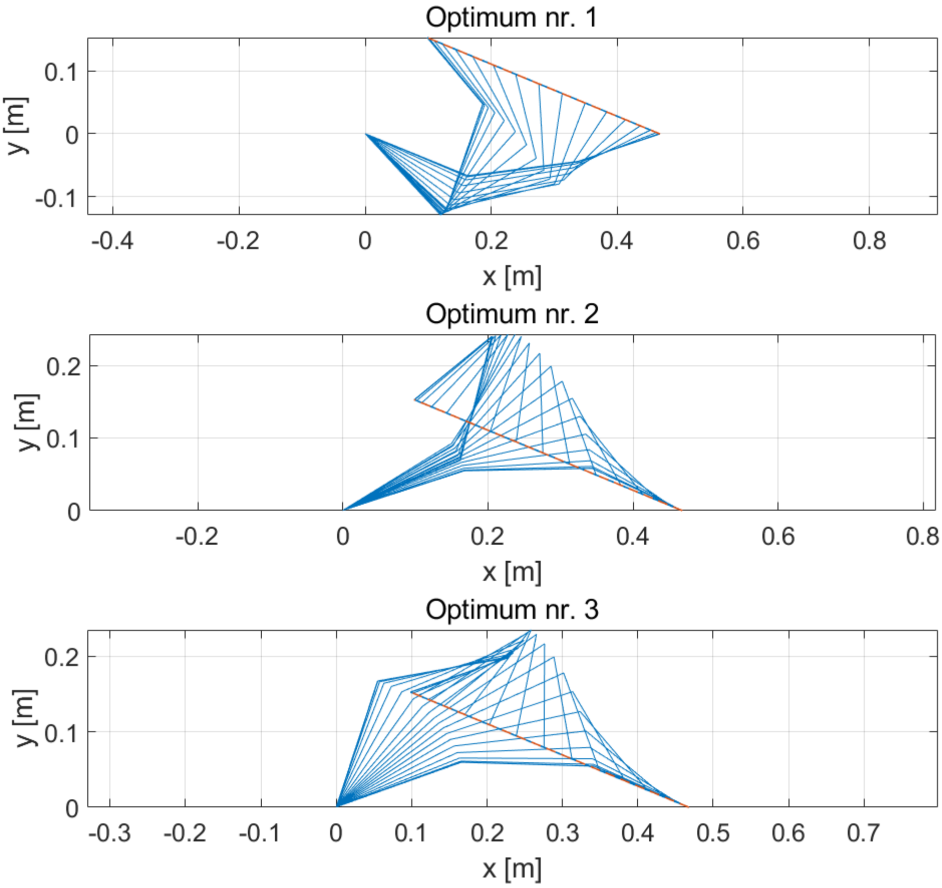


Figure 4-2 Kinetic energy optima for the validation trajectory

4.4.3 Simulations results

For all simulations, relevant variables are summarised in two figures. First one shows energy related variables, in this order: kinetic energy, power, torques norm integral and kinetic energy integral. Second one shows a stroboscopic plot of the motion, alongside with joints positions and velocities, torque and power for each joint. A blue line is used for first joint, red one for second, and yellow one for third. Conditions featuring rectilinear trajectories (conditions a-b-c-d) are discussed first, starting with those using kinetic energy as control cost. After this, cases featuring torques norm integral with no friction, and finally cases featuring torques norm integral with viscous friction. The friction coefficient that has been used is 0.3, a typical coefficient for aluminium-aluminium interface. Cases featuring cyclic motion (letter e) will be examined afterwards, since they feature different boundary conditions which are worth being examined against each other.

4.4.3.1 Rectilinear trajectories

CASE 1 (KINETIC ENERGY)

The values of kinetic energy and joint torques norm integral for each constraints' choice have been summarised in Table 4-5. It is important to notice that optimisation of a quadratic form of velocities does not guarantee continuity of the accelerations, which might be problematic for constraints involving accelerations.

Table 4-5 Results for case 1 rectilinear trajectories

Simulation	**Kinetic energy integral** **[Js]**	**Torques squared norm integral** **[((Nm)^2)s]**
1a (no limits)	0.0528	0.1827
1b (displacement and velocity)	0.0528	0.1820
1c (displacement and torque)	0.0550	0.1564
1d (displacement and power)	0.0529	0.1843

First simulation, 1a, with no constraints, has been already partially discussed as part of the validation of the algorithm, however the best solution that has been found is worth more comments. Comparison with Nedungadi kinetic energy minimising algorithm has shown slightly lower values of the kinetic energy integral: this difference is quite small, and it can probably be ascribed to small differences in implementation. Figure 4-3 and Figure 4-4 respectively show energy figures and kinematic and dynamic variables for each joint. It is possible to observe that the second and third joints feature enhanced motion, while the first one is less used: this is an effect of kinetic energy minimisation, which focuses on reducing the motion of the joints with higher inertial properties, trying to reduce the burden on them. It is however interesting to observe the torques figure: the first joint is now the one showing the highest value by far, reaching up to 0.7067 Nm, while for power, first and second joints are comparable, with maximum values being 0.7721 W and 0.7303 W. In both cases, the third joint is showing way lower values (maximum values are 0.0632 Nm for torque and 0.2545 W for power). This shows minimisation of kinetic energy doesn’t necessarily imply lower effort for actuators. In fact, even when minimising kinetic energy, it makes sense to limit actuators torques and powers as per the other simulations hereby presented.

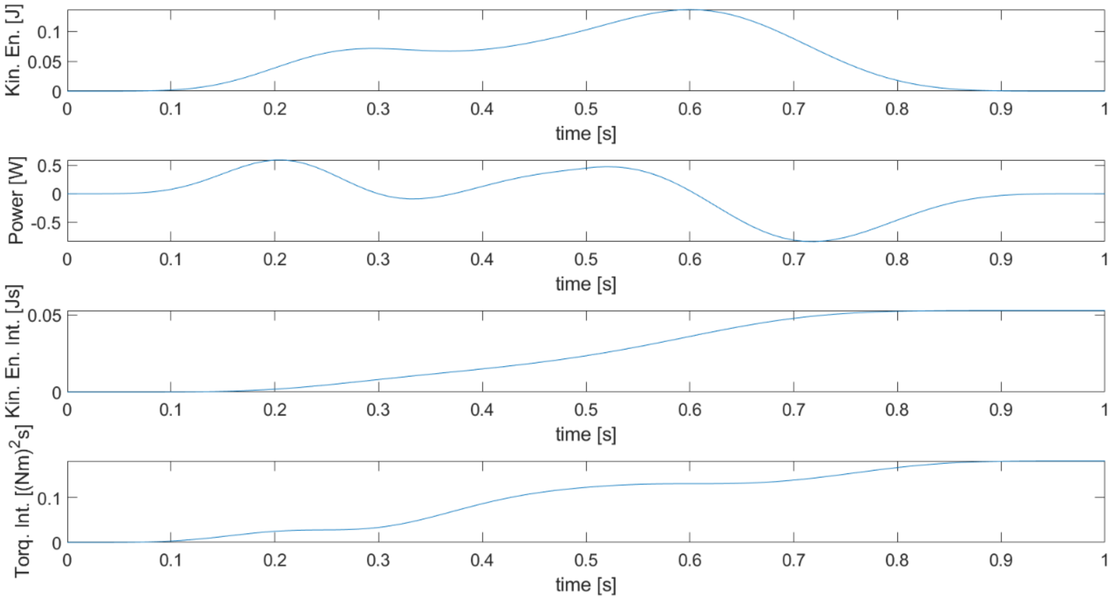


Figure 4-3 Energy figures for trajectory 1a

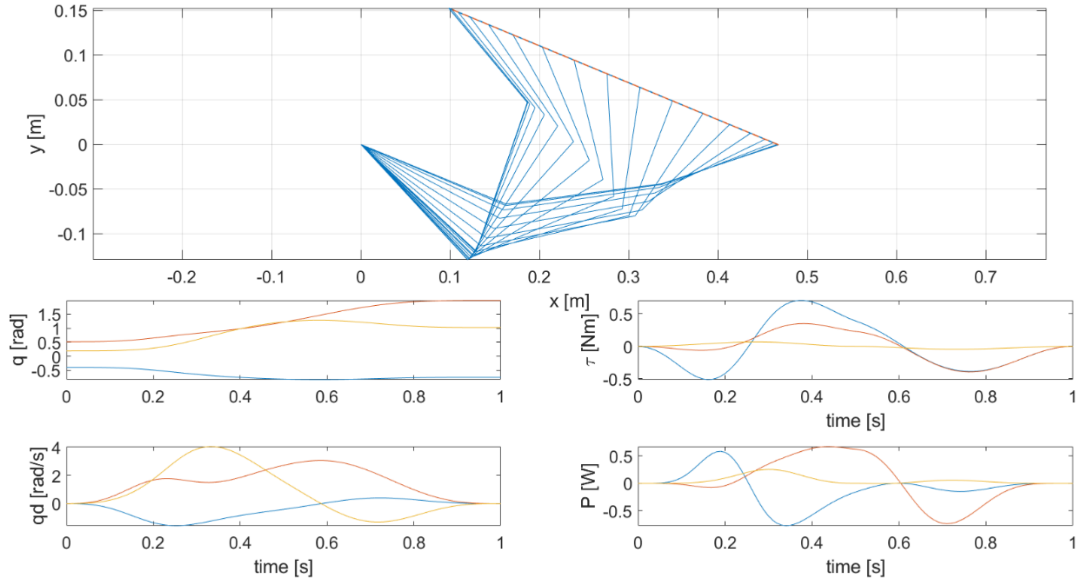


Figure 4-4 Joint figures for trajectory 1a

As already mentioned, constraints configuration 1b features limits in joints displacement and velocity, 1c in joints displacement and torque, and 1d in joint displacement and power. In all cases, the global optimum is close to the one obtained without constraints, and results are illustrated in the same order in Figure 4-5 and Figure *4-6* for 1b, Figure 4-7 and Figure *4-8* for 1c, and Figure 4-9 and Figure *4-10* for 1d. The stroboscopic plots are all similar, showing that the constrained optima are quite close to the unconstrained one. Constraints influence continuity and differentiability of the variables in different ways: in case 1b, with velocity constraints, it can be observed that, when third joint velocity reaches the limit, first and second joint velocities feature non-differentiable points as well to adapt the motion. Such non-differentiable points are due to the fact that optimisation of kinetic energy (or any cost based on a quadratic form of velocity) is only effective to provide continuity of velocity, and not necessarily differentiability, although it will guarantee it as much as possible. The case will be different for cases involving torques. It is interesting to observe, in Table 4-5, that the integral of joint torques norm is reduced as a result of this, possibly since second joint torque is suddenly reduced when third joint mobility is reduced by the limit. Simulation 1c features a nonlinear constraint on joint torques, which produces the highest kinetic energy and the lowest joint torques norm integral among all the simulations featuring kinetic energy as control cost. Abrupt changes in torques values are reflected on joint powers, which are also influenced by the jumps. The joints velocity profile is however perfectly acceptable, and fully differentiable along the whole trajectory. Finally, simulation 1d features a power limit. When the power limit is reached, by joint 1 at time 0.31s, the reduction in mobility caused by the constraint is offset by increasing velocity and torque on joint 2, as visible in Figure 4-10. As for joint limits, they are only reached by second joint at the end of the trajectory in both 1b and 1c, and no direct consequence of this is observable in the other variables.

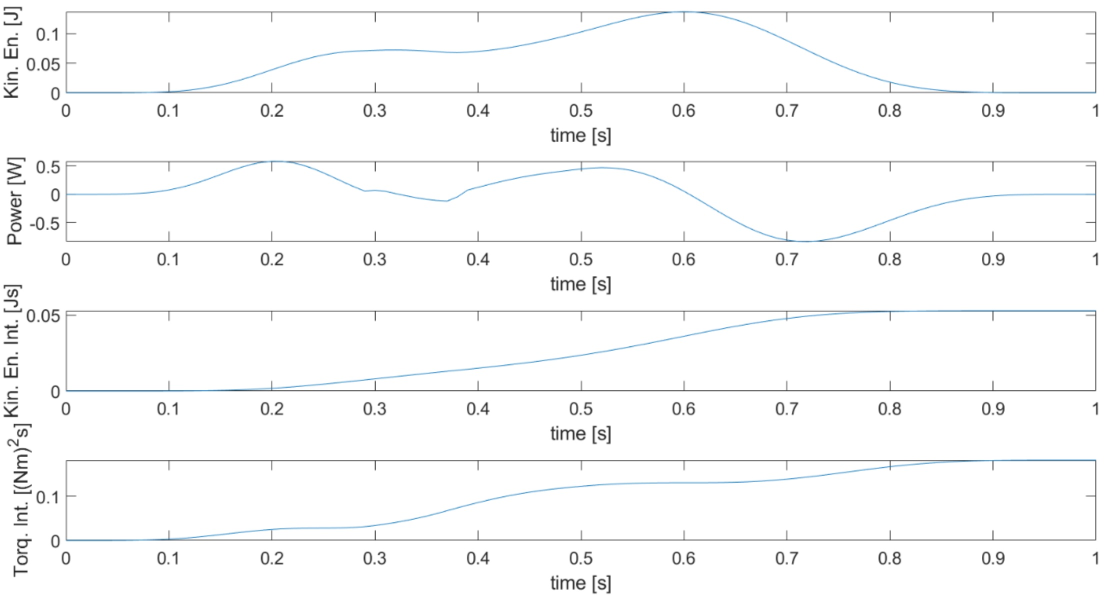


Figure 4-5 Energy figures for trajectory 1b

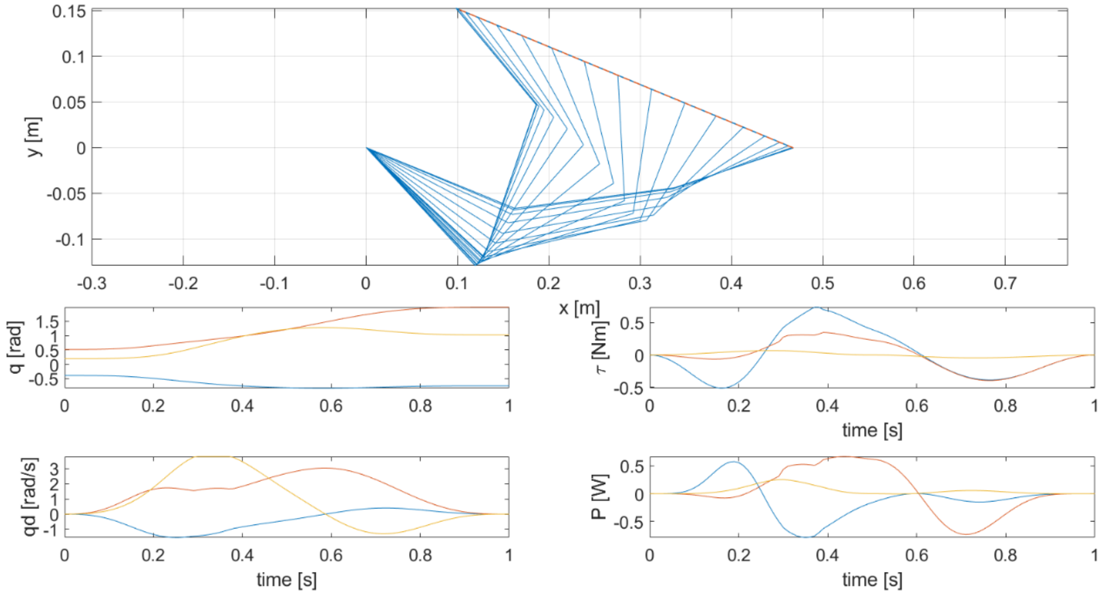


Figure 4-6 Joint figures for trajectory 1b

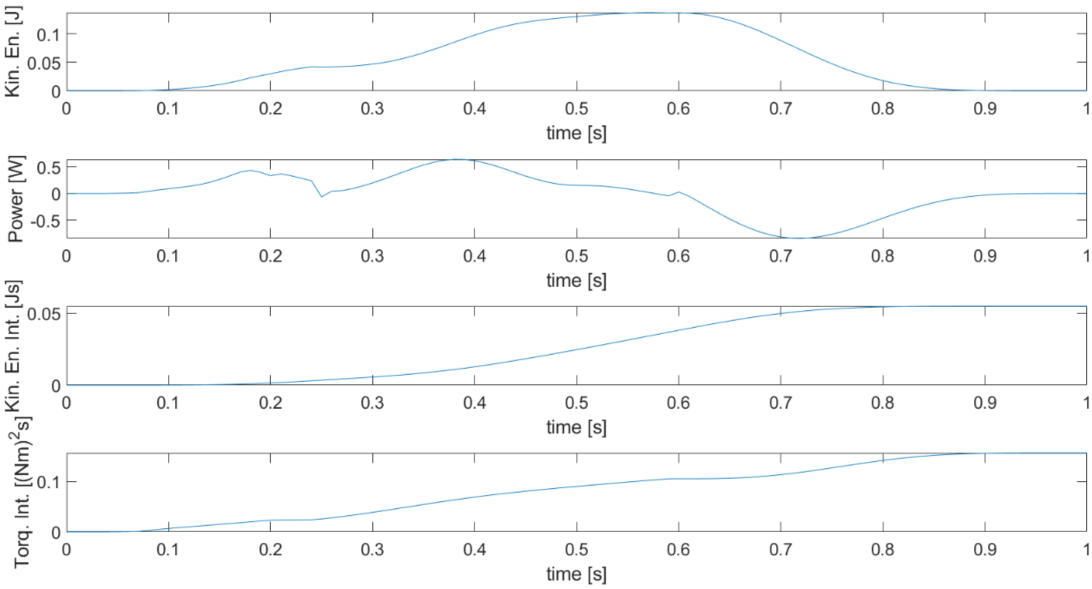


Figure 4-7 Energy figures for trajectory 1c

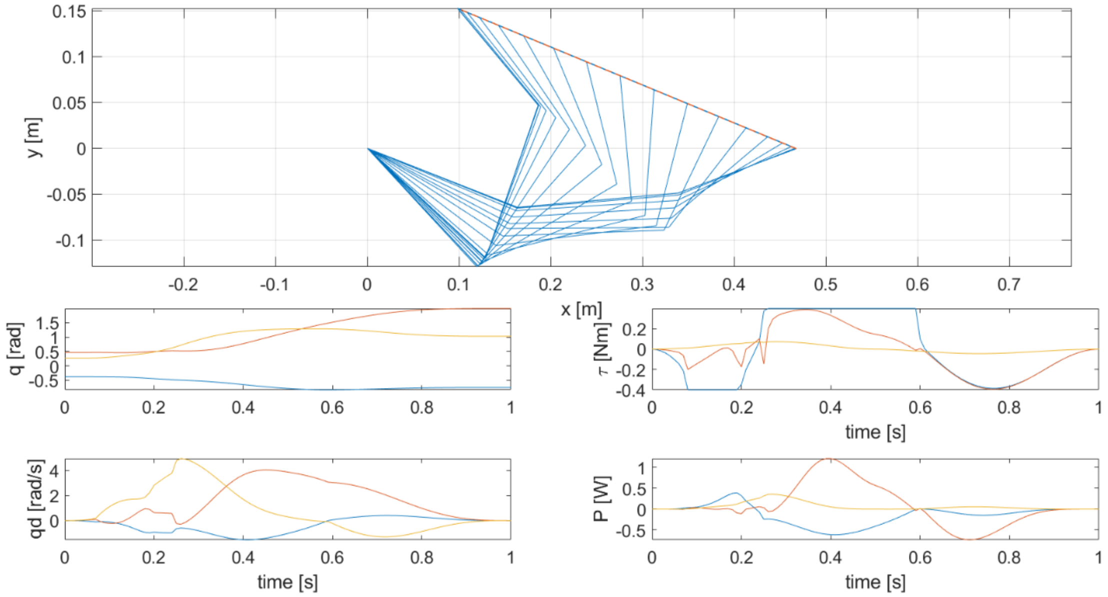


Figure 4-8 Joint figures for trajectory 1c

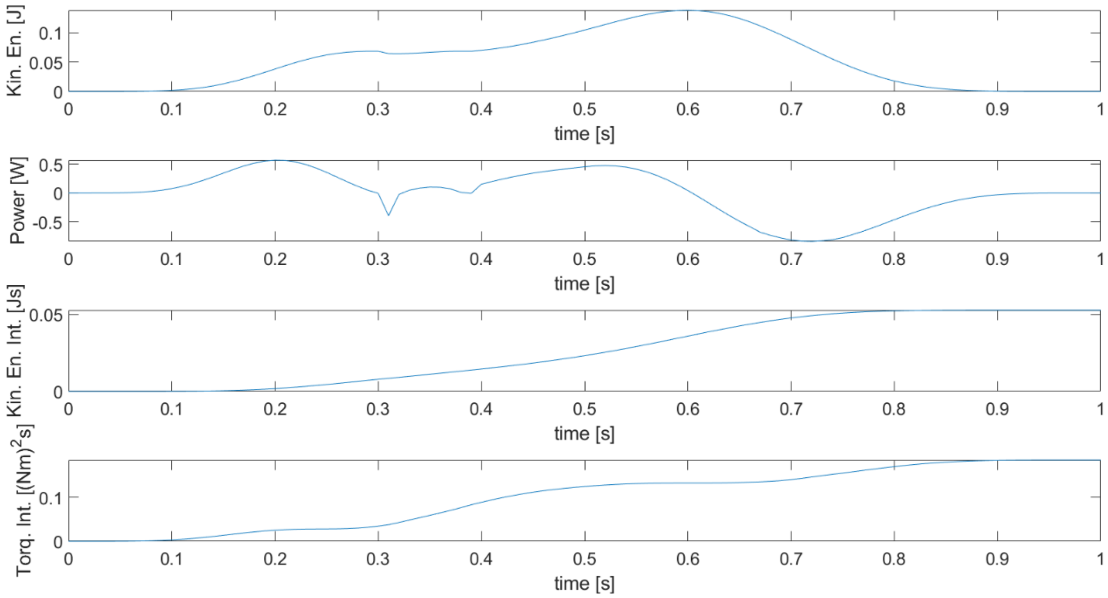


Figure 4-9 Energy figures for trajectory 1d

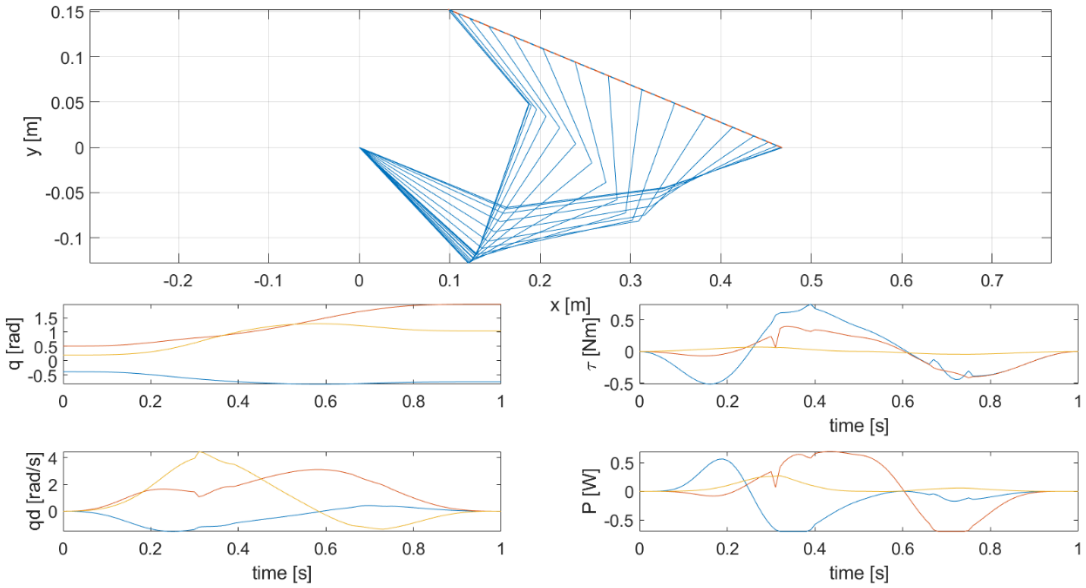


Figure 4-10 Joint trajectory for trajectory 1d

CASE 2 (JOINT TORQUES NORM WITHOUT VISCOUS FRICTION)

The results of the second cost function case for kinetic energy and torques norm integral are shown in Table 4-6.

Table 4-6 Results for case 2 rectilinear trajectories

Simulation	Kinetic energy integral [Js]	Torques squared norm integral [((Nm)^2)s]
2a (no limits)	0.0842	0.0398
2b (displacement and velocity)	0.0786	0.0932
2c (displacement and torque)	0.0854	0.0864
2d (displacement and power)	0.0849	0.0843

Results for simulation 2a are shown in Figure 4-11 and Figure *4-12*, while 2b, 2c and 2d are respectively shown in Figures from Figure *4-13* to Figure *4-18*. In simulation 2a, the result for torques is more than 450% less than simulation 1a, featuring kinetic energy as control cost, and the trajectory is completely different. This solution features low torques during the first part of the trajectory, and a maximum torque values in the interval 0.6-0.8s. No joint is completely motionless at the end of the trajectory: the absence of friction implies that no torque is necessary to maintain their velocity, and thus the optimal solution saves on braking torque, to the point of accelerating most towards the end of the trajectory rather than at the beginning. This effect will be lower for the third case, featuring viscous friction. Joint velocities are also more balanced compared to 1a, with second and third joint featuring similar velocities, although it is to be noticed this simulations features the highest maximum velocity of all rectilinear trajectories, being it 4.99 rad/s for second joint at t=0.37s. The second joint is now the one with the highest torque, but its maximum is now around 60% less of the highest joint torque in the kinetic energy case, being up to 0.315 Nm. Power values are also greatly decreased compared to the kinetic energy case, with the highest value being 0.4335 W at time 0.78s, which is reduced compared to kinetic energy optimisation highest value by around 50%. Looking at the energy figures of the whole manipulator, it is easy to observe that both kinetic energy is greatly increased from case 1a. The maximum in kinetic energy, 0.1398 J, corresponds to time 0.342s, when the values of all joint torques are close to zero, while the maximum of overall manipulator power (0.7325W at time 0.27s) corresponds to a local peak in power on all joints. Simulation 2b features a motion that is more like the one that has been seen in case 1, showing how the joint limits here affect the optimal solution. The cost function is also extremely increased, by almost 150%, while kinetic energy integral is the lowest one for case 2, due to the velocity limit. This shows the versatility of the Interpolation-based Global Kinematic Planner, which can adapt to the constraints' configuration looking for optima that are far apart. One more aspect of this simulation that is worth mentioning is the joints velocity profile: while in simulation 1b velocity featured non-differentiable points where velocity limit is reached, in simulation 2b the limit is approached more gently. This behaviour is due to the fact that the optimisation function now comprises a quadratic function of accelerations (the integral of the square of joint torques norm), which causes them to be continuous along the trajectory. Simulations 2c and 2d, with joint torque and power limits, are quite similar since the limit is only reached in small portions of the trajectory. It is however interesting to observe, in 2c, the change of slope in torques and power when the torque limit is reached. This is due to the fact that the manipulator only has one degree of redundancy left after considering EE trajectory constraints. Thus, when another constraint is active, has no degrees of redundancy left, and it is forced to follow the trajectory corresponding to the only feasible solution.

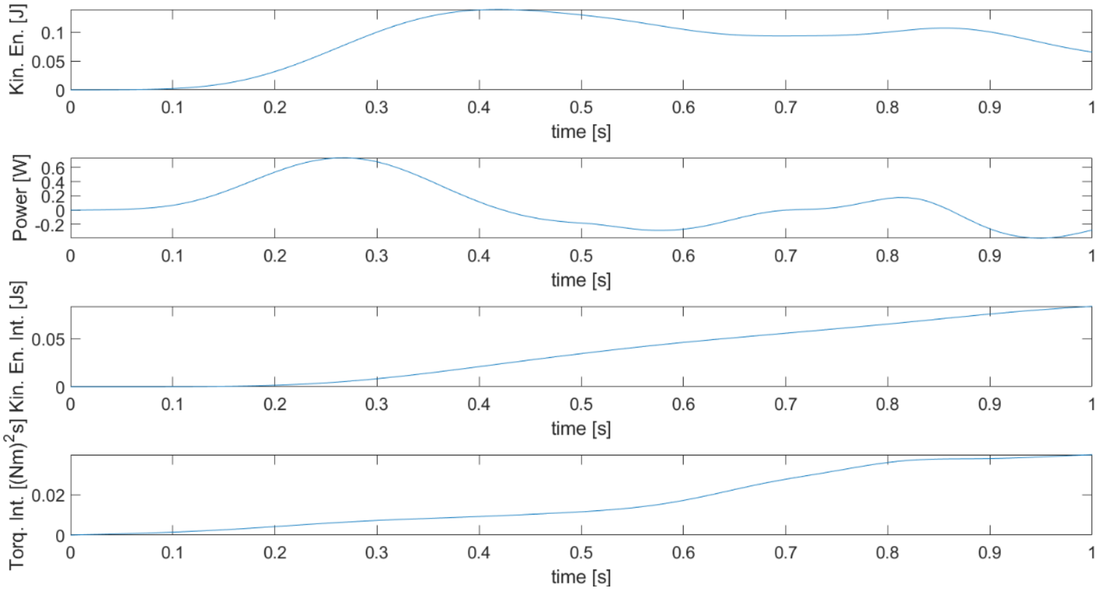


Figure 4-11 Energy figures for trajectory 2a

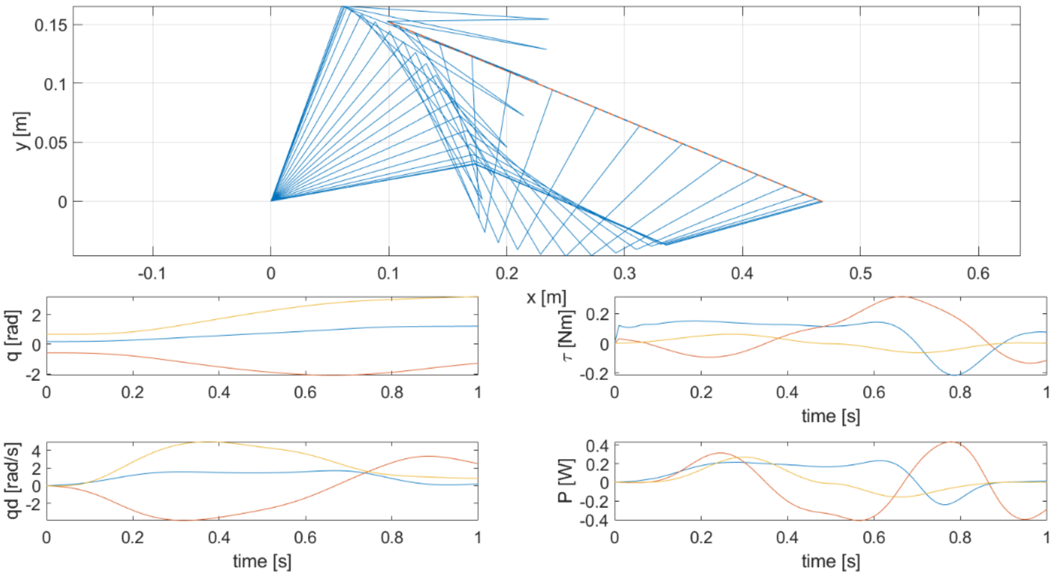


Figure 4-12 Joint figures for trajectory 2a

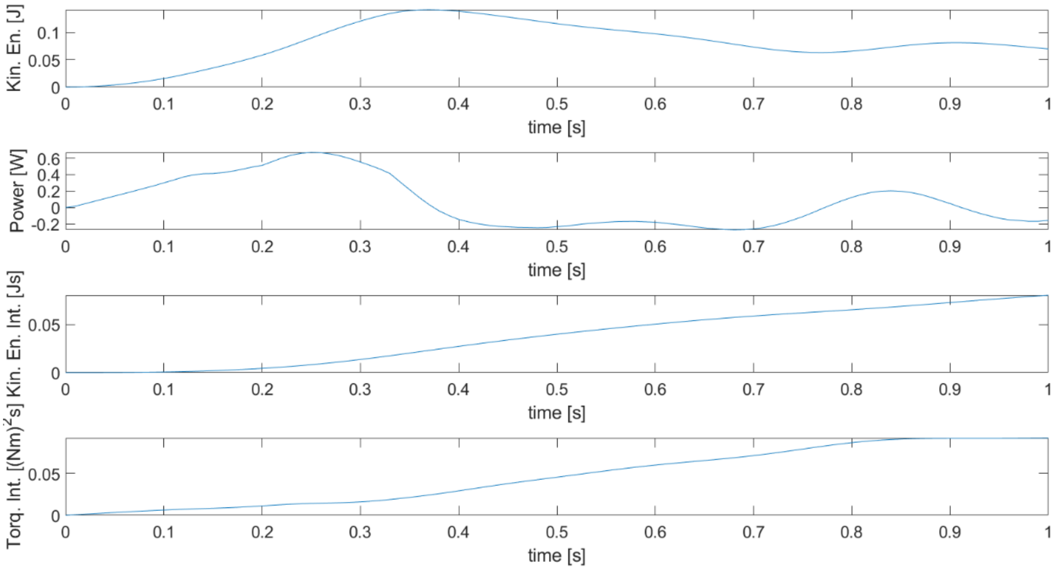


Figure 4-13 Energy figures for trajectory 2b

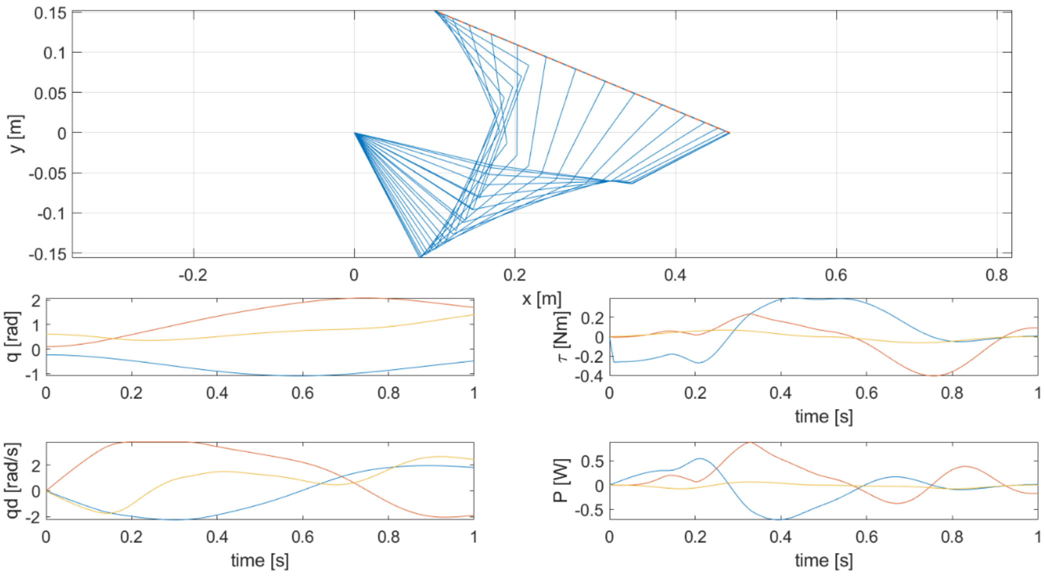


Figure 4-14 Joint figures for trajectory 2b

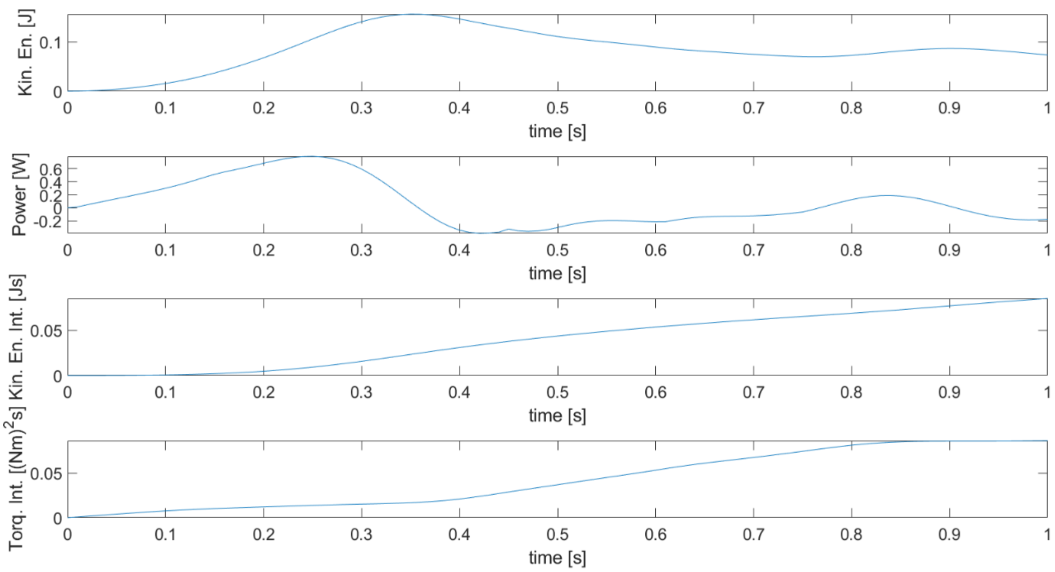


Figure 4-15 Energy figures for trajectory 2c

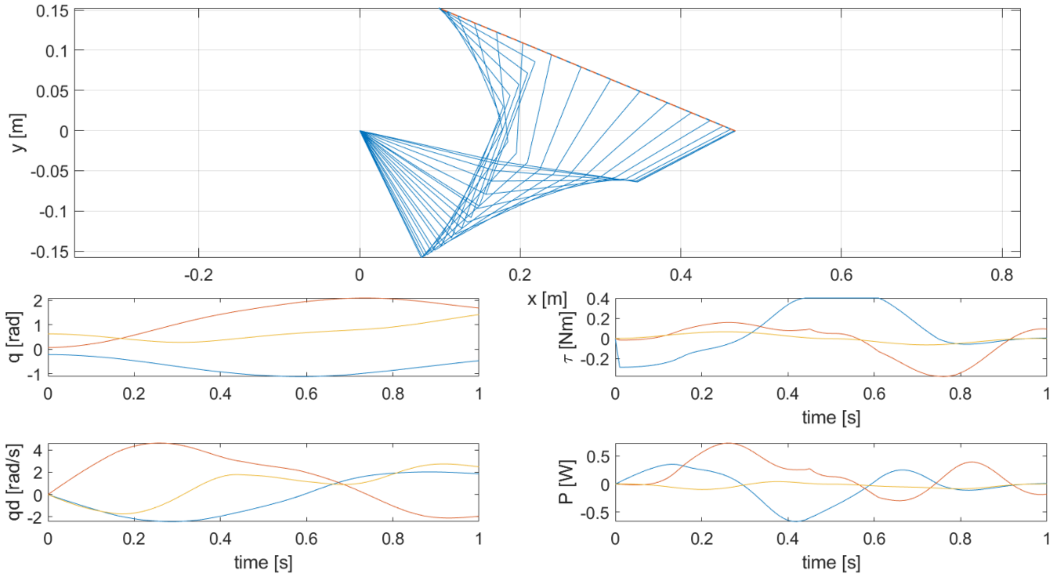


Figure 4-16 Joint figures for trajectory 2c

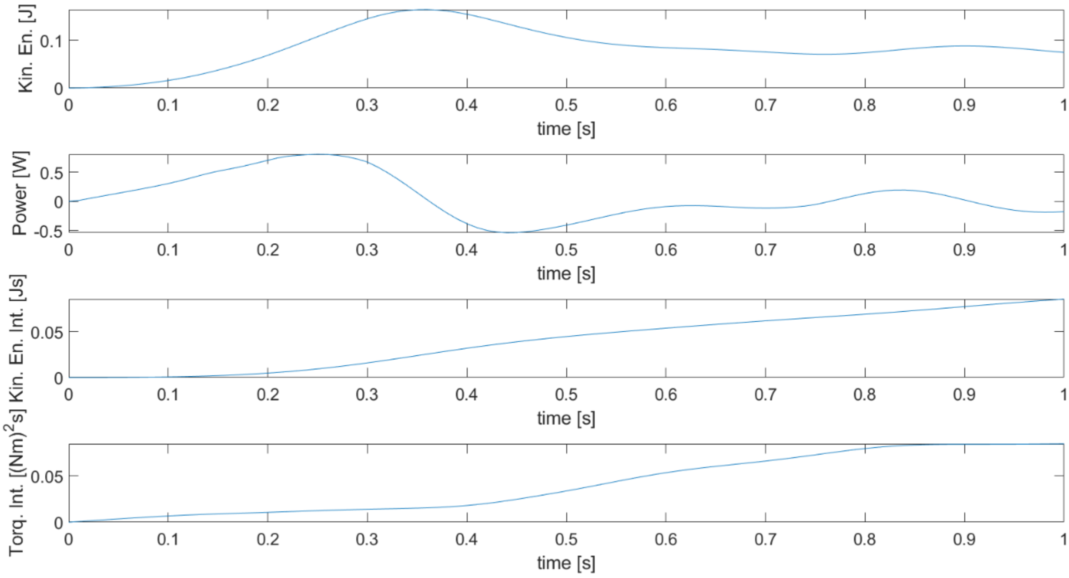


Figure 4-17 Energy figures for trajectory 2d

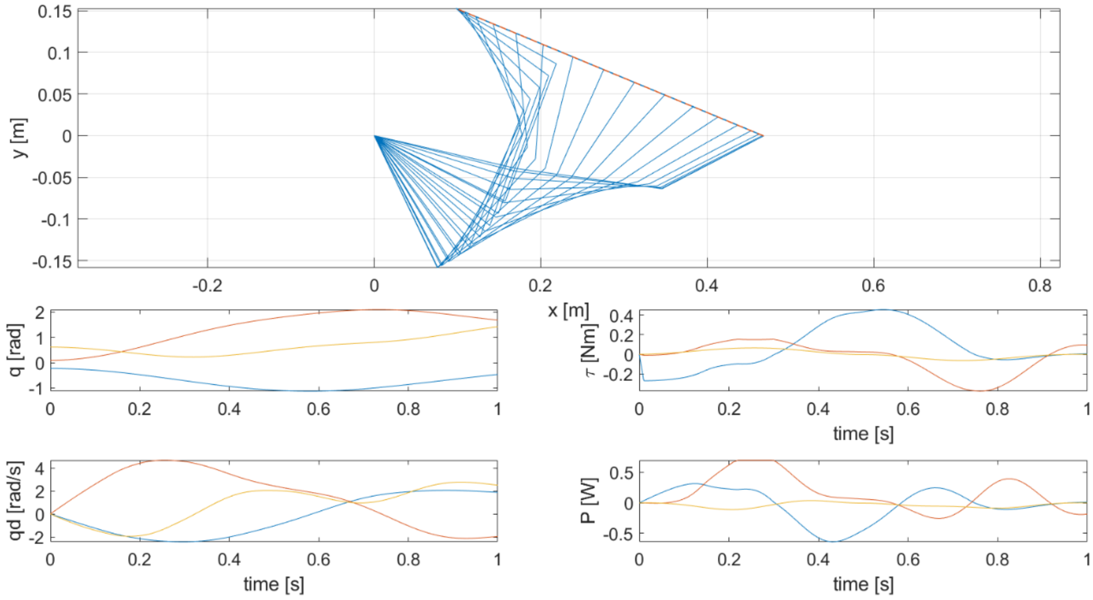


Figure 4-18 Joint figures for trajectory 2d

CASE 3 (JOINT TORQUES NORM WITH VISCOUS FRICTION)

The values for kinetic energy and torques' norm integral for case 3 are shown in Table 4-7. Joint torques squared norm is obviously much higher, since torque is now required to maintain velocity.

Table 4-7 Results for case 3 rectilinear trajectories

Simulation	**Kinetic energy integral** **[Js]**	**Torques squared norm integral** **[((Nm)^2)s]**
3a (no limits)	0.0569	0.6890
3b (displacement and velocity)	0.0569	0.6890
3c (displacement and torque)	0.0555	0.7019
3d (displacement and power)	0.0557	0.6976

A first observation regarding Table 4-7 is that the torque solutions featuring viscous friction, which is a function of velocity, feature solutions that are much closer to the kinetic energy global optimum. First simulation to be presented is 3a, illustrated in Figure 4-19 and

Figure *4-20*. This solution is closer to the kinetic energy-based solution 1a, rather than the torque-based solution 2a, and joint velocities at final time are also very close to 0. This can be explained by the fact that friction (and thus velocity) plays a major role in this case, forcing the algorithm to keep a low velocity profile. This is in sharp contrast with solution 2a, which features highest joint velocity of all rectilinear trajectories, and it also shows how much friction influences results. From the value in Table 4-7, it is evident that most of the torque is spent contrasting friction (it features an increase of more than 10 times compared to simulation 2a).

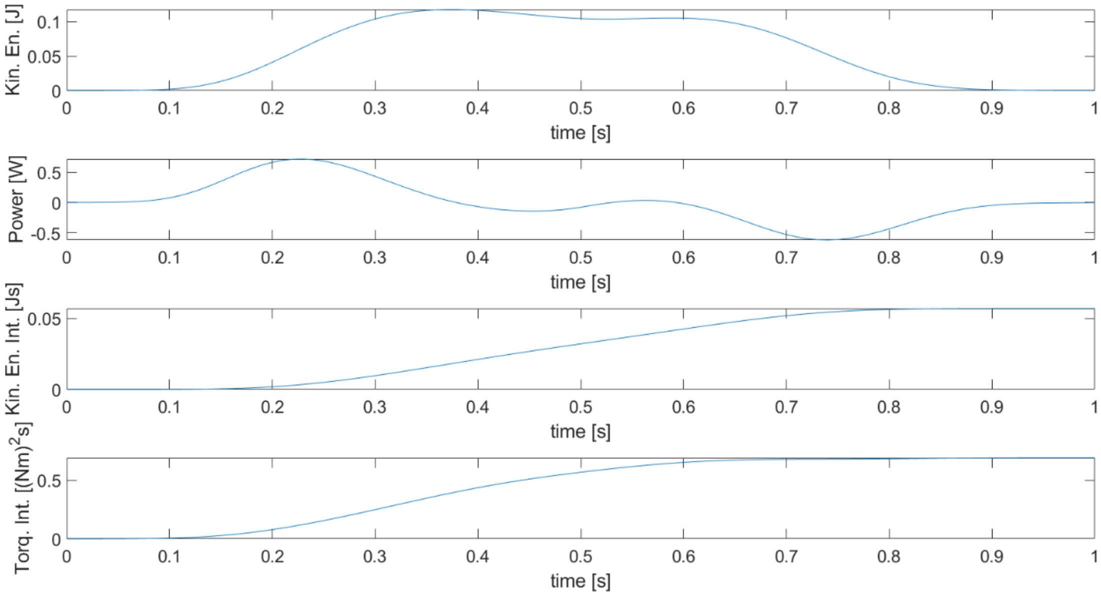


Figure 4-19 Energy figures for trajectory 3a

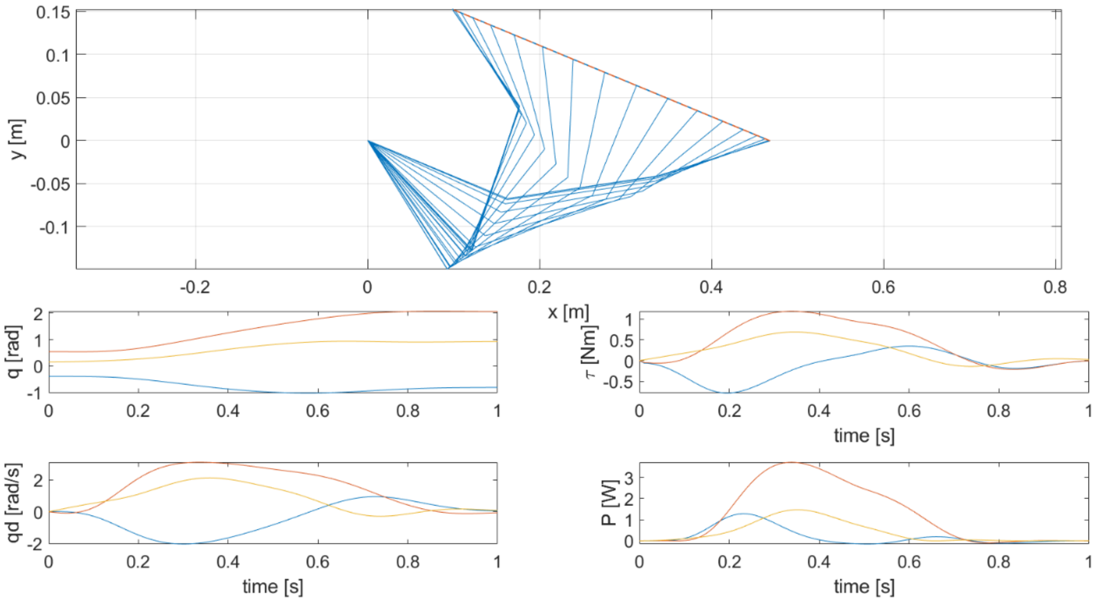


Figure 4-20 Joint figures for trajectory 3a

Figures for simulation 3b are omitted since the limits do not come into play and its result is the same as 3a. Results for simulation 3c and 3d, shown respectively in Figure 4-21 and Figure 4-22, and in Figure 4-23 and Figure 4-24, are more relevant since they show that, for the case with friction, limiting torques or power also reduces velocity: velocity only goes up to 2.952 rad/s in simulation 3c, and to 2.817 rad/s in simulation 3d. Simulation 3c features active torque constraints on joint 2 between times t=0.24s and t=0.49s. It can be clearly observed how the constraint on torque also limits velocity and power on joint 2 (maximum power is 2.876W), although it approaches its maximum value gently and without sudden changes. A similar phenomenon can be observed on simulation 3d as well, with power limiting velocity and torque as well (torque maximum value in this case is 1.052 Nm on joint 2). In this case, however, changes in torques when the power constraint is activated can be abrupt, with the torque featuring a non-differentiable point at t=0.26s. Case 3 can be considered the most realistic one and is the closest step to implementing the tracking algorithm presented in this chapter on a real manipulator.

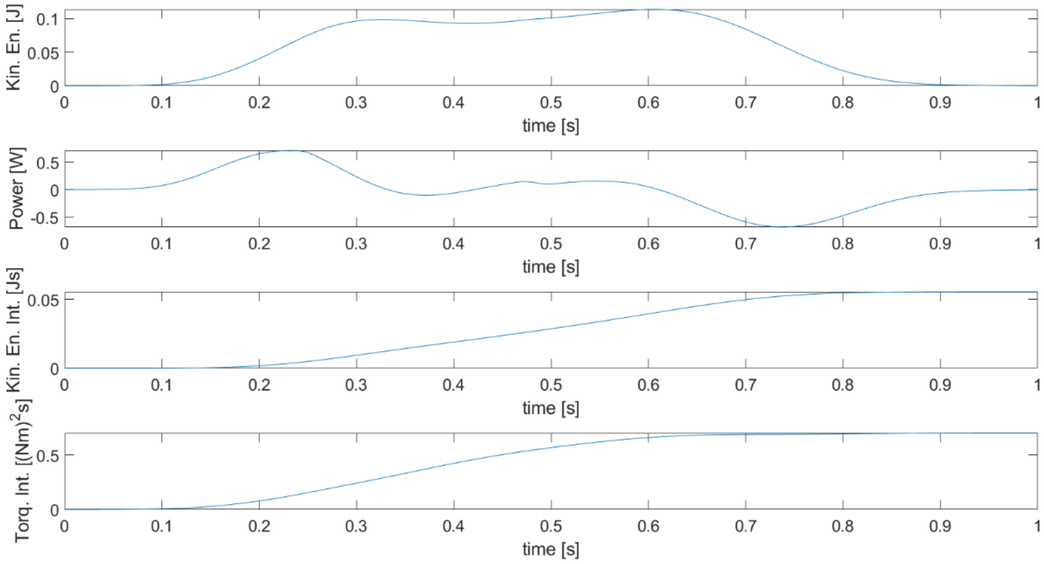


Figure 4-21 Energy figures for trajectory 3c

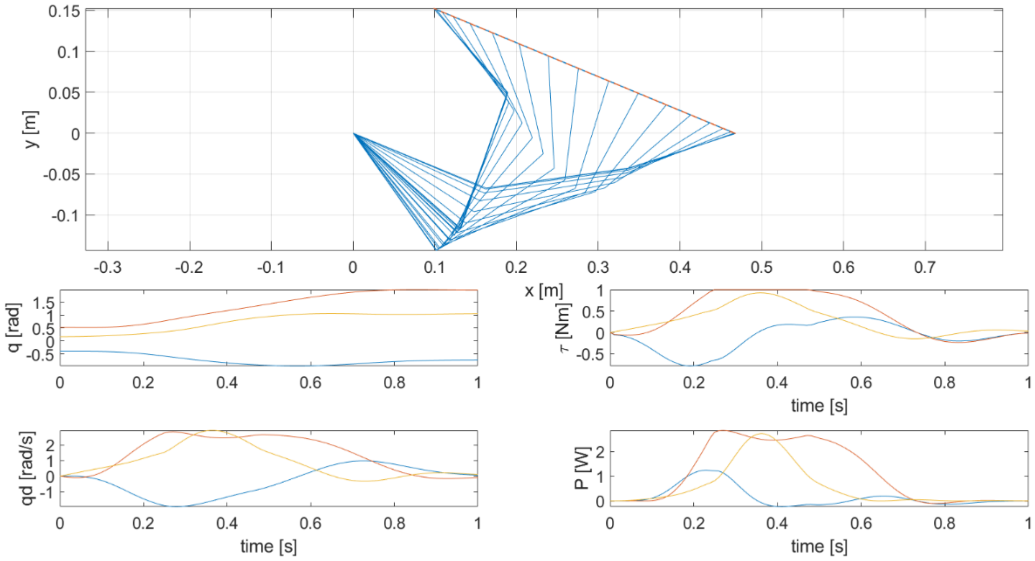


Figure 4-22 Joint figures for trajectory 3c

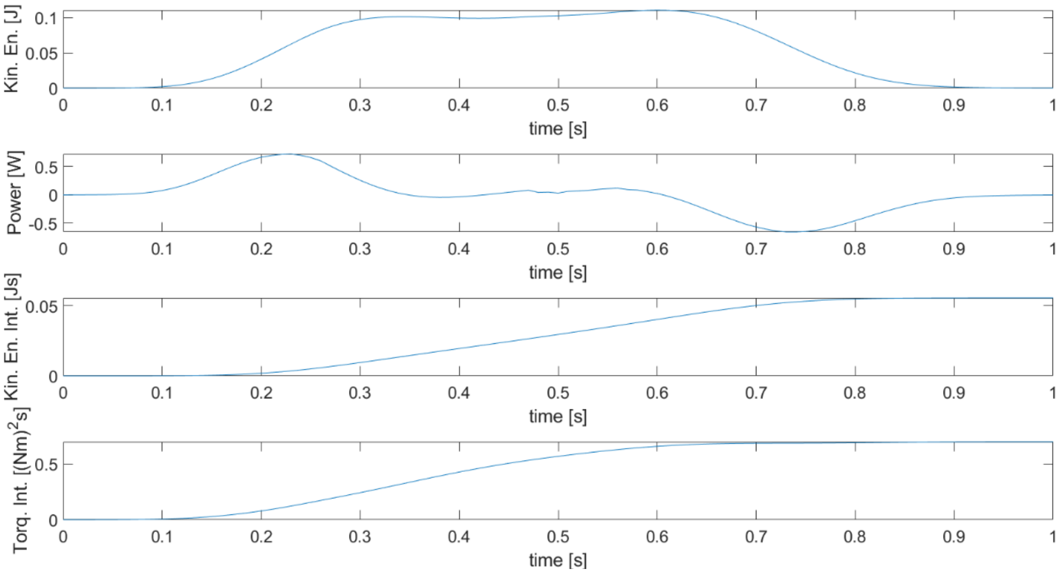


Figure 4-23 Energy figures for trajectory 3d

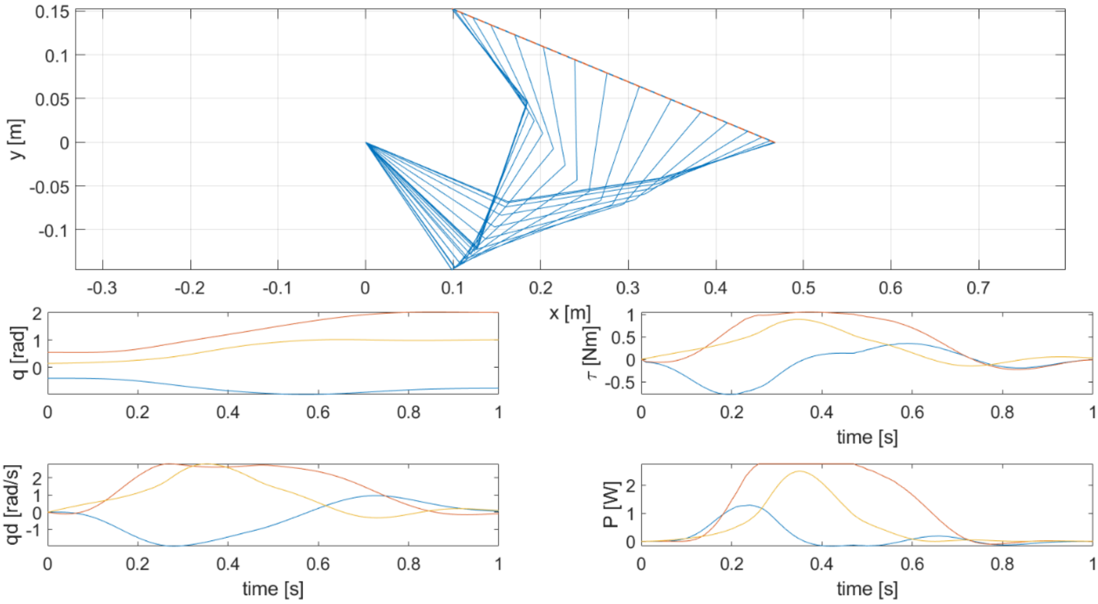


Figure 4-24 Joint figures for trajectory 3d

4.4.3.2 Cyclic trajectories

Cyclic trajectories present some different challenges than rectilinear ones. The most obvious one is that obviously displacements, velocities and accelerations must be continuous between the endpoint and the initial point. In order to obtain this, it is sufficient to simply include control cost between the endpoint and the initial point in the cost function. This enforces continuity and differentiability of velocity when kinetic energy is used as control cost, and continuity and differentiability of accelerations when joints squared torques norm is used as control cost. One more point to notice about this kind of trajectory is that, even for small radii, the motion of the joint appears much enhanced in circular trajectories than in rectilinear ones, as the simulations will show. Thus, for the simulations hereby presented velocity has been limited again to 3.8 rad/s. Results for kinetic energy and torques noem integral are visible in Table 4-8.

Table 4-8 Results for circular trajectories

Simulation	**Kinetic energy integral [Js]**	**Torques norm integral [((Nm)^2)s]**
1e (kinetic energy)	0.0554	1.898
2e (torques norm)	0.1031	1.246
3e (torques norm with friction)	0.0683	2.459

The value of the torque cost function is clearly much higher than for rectilinear solutions, surpassing them by one order of magnitude. The explanation for this probably lies in the big changes in joints velocity that are necessary to perform the trajectory under examination. The variables of interest for simulation with kinetic energy as control cost are illustrated in Figure 4-25 and Figure 4-26, while the ones for the simulation with torques norm without friction as control cost is shown in Figure 4-27 and Figure 4-28, and in Figure 4-29 and Figure 4-30 for the one with friction. Looking at velocity, torque and power values for each joint, the increase in control cost can be easily justified. For the cases featuring kinetic energy and torque without friction as control costs, velocity limit is reached much more often than in the rectilinear case. If more than three constraints were active at some timestep, the ability to track the trajectory would be completely lost at that path point (it must be remember that two constraints are imposed by the end-effector trajectory itself). However, the solutions computed by the algorithm for simulations 1e and 2e features at most three active constraints at the same time, retaining necessary freedom of movement, although in 2e the second joint reaches the velocity limit as the first joint also gets very close. Simulation 3e, on the other hand, gets close to the velocity limit only with second joint, due to the fact that high velocity, in this case, translates into a high torque value. Its value of kinetic energy integral also shows that in this case velocities play a much more important role in the control cost, putting the case with friction somewhere in between the other two as per joint motion profile. Looking at torques, their maximum values are respectively 4.144 Nm, 2.919 Nm, and 2.696 Nm, while for powers they are respectively 5.813 W, 7.216 W and 4.061 W. It is particularly interesting to observe the two cases featuring torque norm as control cost, as torque maximum value is higher in the one without friction rather than the one with friction. This peculiar result can be justified comparing the torque profile for each joint between the two simulations: in simulation 2e, the first joint features by far the highest torque (2.919 Nm against 1.177 Nm for the second joint and 0.1925Nm for the third), as it reaches similar velocity than the other two, but has by far the highest moment of inertia. In the case with friction, however, velocity has a cost in terms of torque regardless of the inertial properties. Considering that the algorithm optimised the squared norm of joint torques along the trajectory, the term for the first joint would become excessive if it were to have both high torque and high velocity. Thus, for the case of torque optimisation with friction, the third joint is the only one which can afford to move at a velocity close to the limit, and generally the friction term forces the algorithm to balance velocity and acceleration.

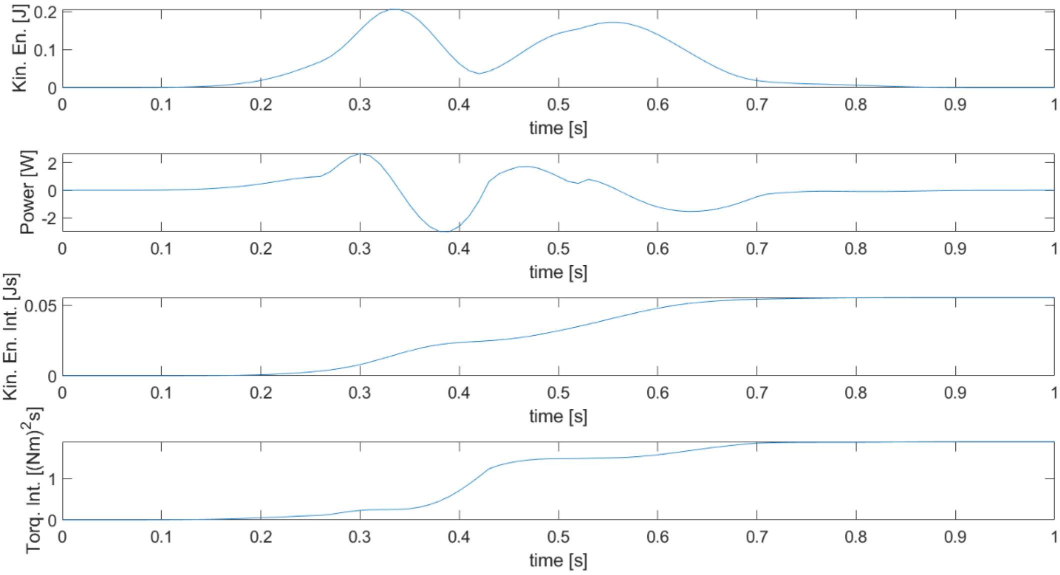


Figure 4-25 Energy figures for trajectory 1e

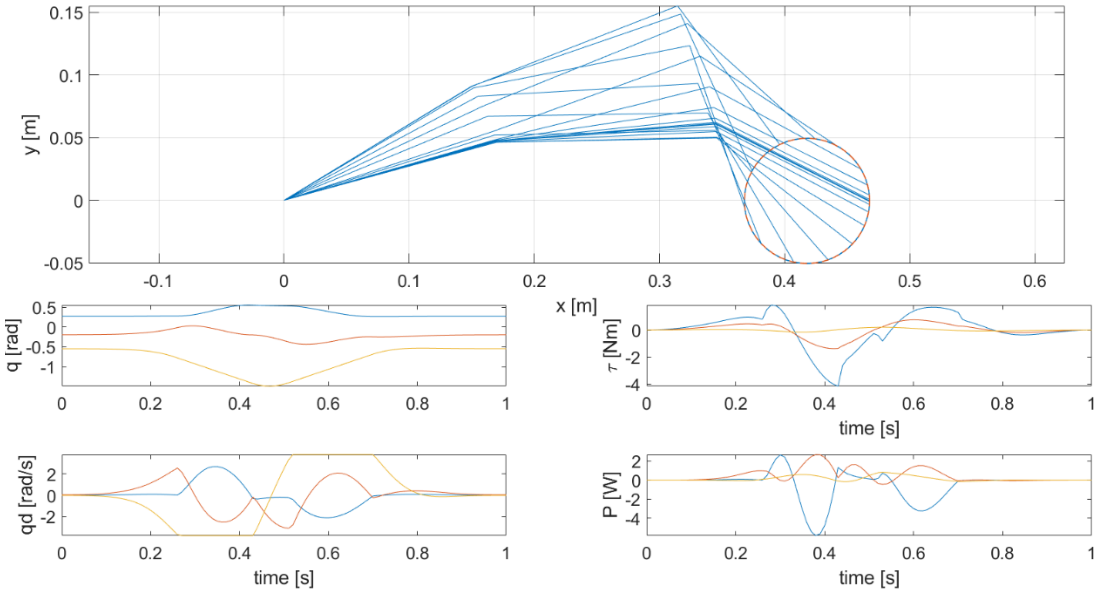


Figure 4-26 Joint figures for trajectory 1e

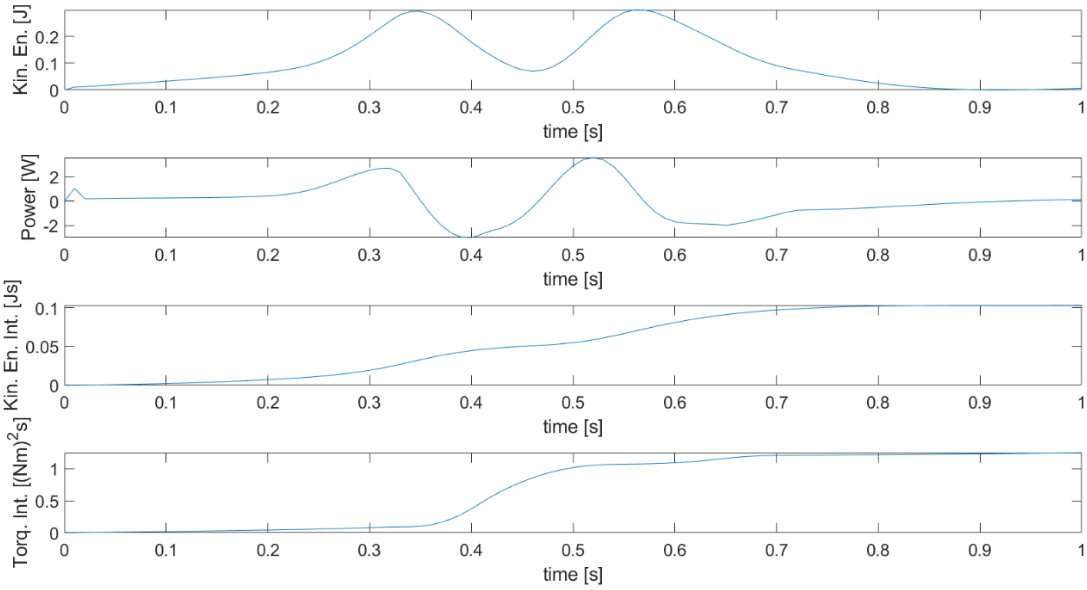


Figure 4-27 Energy figures for trajectory 2e

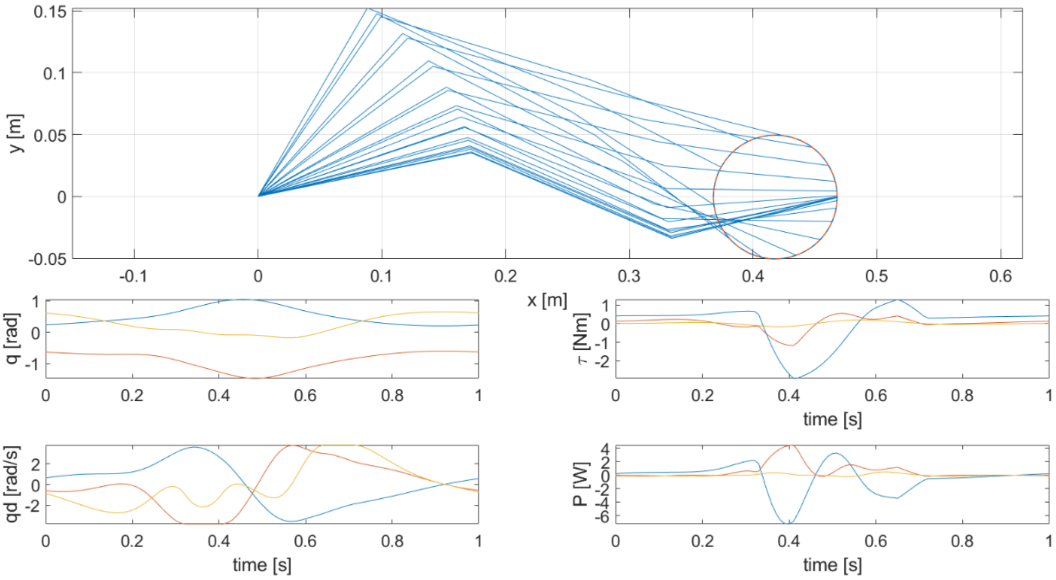


Figure 4-28 Joint figures for trajectory 2e

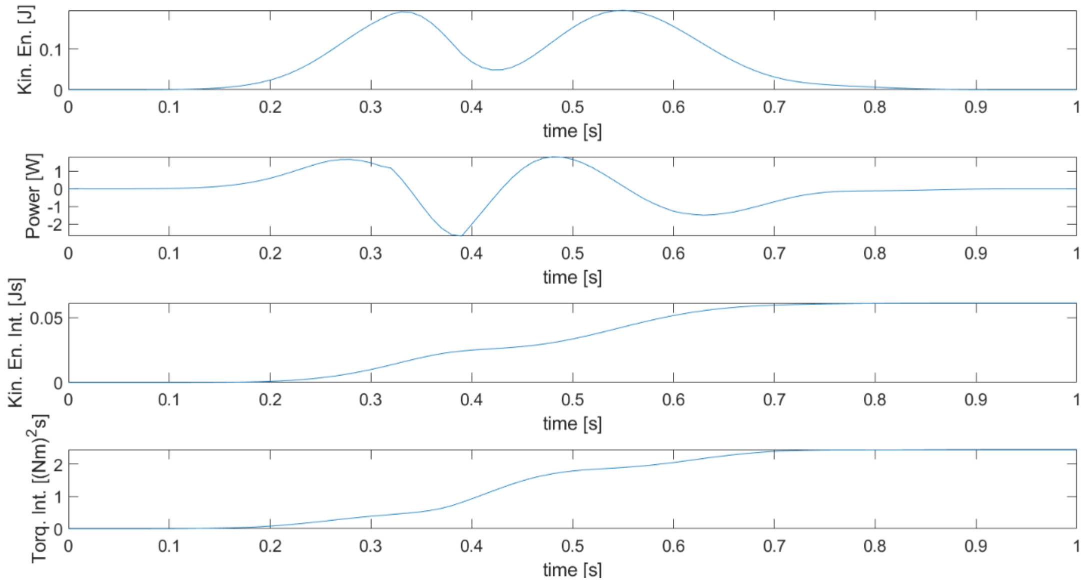


Figure 4-29 Energy figures for trajectory 3e

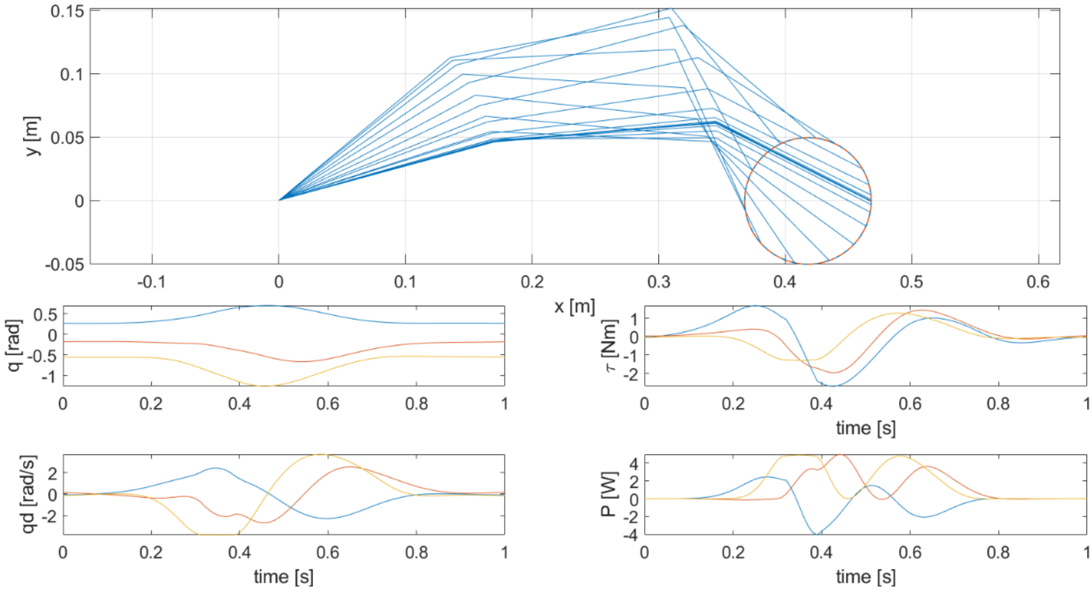


Figure 4-30 Joint figures for trajectory 3e

4.4.4 Multi-objective optimisation

For some robotic applications, it is required to balance between different cost functions [59]. This kind of problem is tackled by multi-objective optimization.

Considering a set of solutions X, and a set of cost functions $f = (f_1 \ldots f_p)$, a feasible solution $\hat{x} \in X$ is a Pareto optimal solution of a multi-objective optimization problem

$min(f(x): x \in X)$ if, and only if, no $x \in X$ exists such that $f(x) \leq f(\hat{x})$. The set of Pareto optimal solutions of a multi-objective optimization problem form the Pareto optimal set, or Pareto front.

When solving a multi-objective optimization problem, it is usually desired to find a solution that is as close as possible to the Pareto optimal set. In the ideal case, the full Pareto optimal set can be exactly computed, and it is possible to choose the preferred solution among its members. Exhaustive computation of the complete Pareto optimal set is, however, expensive and, in many cases, not possible, thus the problem is usually approached by computing some of the members of the set and using them as a representation of the full set. The most used method to do so in robotics is the weighting method [59], which, however, does not produce evenly distributed solutions, cannot individuate all members of the Pareto optimal set and, moreover, fails in the case of non-convex Pareto fronts [151]. A possible alternative to the weighting method is the ε-constraint method, which can capture the shape of the Pareto front in a more complete and representative way than the weighting method, even when the Pareto front is non-convex [152]. The use of the ε-constraint method has however not been possible so far in global inverse kinematics problems resolution because it requires the imposition of nonlinear constraints to the optimization problem under examination. This is possible through the use of the Interpolation-based Global Kinematic Planner presented in this work, which thus allows the use of the ε-constraint method. This has been demonstrated by analysing the bi-objective optimization problem resulting from optimizing both the kinetic energy integral and torques squared norm integral without friction while tracking the rectilinear trajectory used in rectilinear simulation cases. Particularly, its Pareto optimal set has been searched considering joint and velocity limits as per Table 4-2.

For the equally spaced implementation used here, the ε-constraint method steps are as follows:

1. The feasible solutions resulting in the minima of the two objective functions, $\{\boldsymbol{q}\}_{kin}$ (for the minimum of kinetic energy integral) and $\{\boldsymbol{q}\}_{tor}$ (for the minimum of torques squared norm integral), are computed separately.
2. The intervals $int_{kin} = G_{kin}(\{\boldsymbol{q}\}_{tor}) - G_{kin}(\{\boldsymbol{q}\}_{kin})$ and $int_{tor} = G_{tor}(\{\boldsymbol{q}\}_{kin}) - G_{tor}(\{\boldsymbol{q}\}_{tor})$ are computed.
3. Each interval is divided in *k* equally spaced steps $\varDelta G_{kin}$ and $\varDelta G_{tor}$, so that $G_{kin}(\{\boldsymbol{q}\}_{tor}) = G_{kin}(\{\boldsymbol{q}\}_{kin}) + k\ \varDelta G_{kin}$ and $G_{tor}(\{\boldsymbol{q}\}_{kin}) = G_{tor}(\{\boldsymbol{q}\}_{tor}) + k\ \varDelta G_{tor}$.

For each $h = 1\,..\,k$ a single-objective kinetic energy integral optimization problem is solved with the formulation:

$$\begin{array}{ll} \underset{q}{\text{minimize}} & \displaystyle\int_{t_0}^{t_{fin}} G_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)\, dt \\ subject\ to & G_{tor} = G_{tor}(\{\boldsymbol{q}\}_{tor}) + h\ \varDelta G_{tor} \end{array} \tag{4-26}$$

Likewise, a single-objective torques squared norm integral optimization problem is solved with the formulation:

$$\begin{array}{ll} \underset{q}{\text{minimize}} & \displaystyle\int_{t_0}^{t_{fin}} G_{tor}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, t)\, dt \\ subject\ to & G_{kin} = G_{kin}(\{\boldsymbol{q}\}_{kin}) + h\ \varDelta G_{kin} \end{array} \tag{4-27}$$

Following these steps, a set of Pareto optimal solutions has been computed, see Figure 4-31.

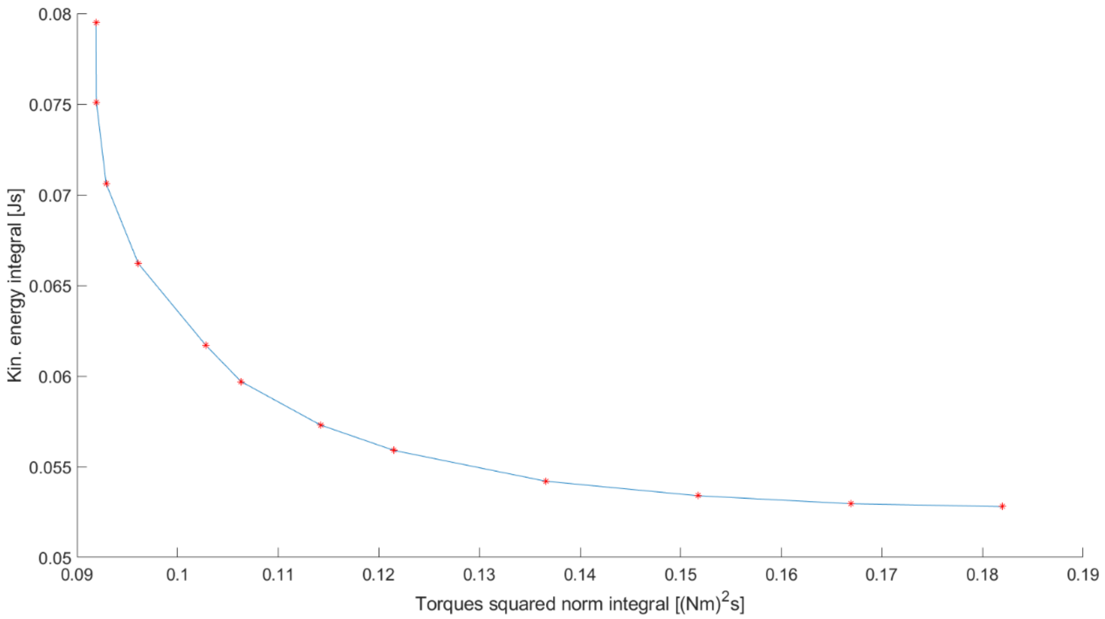


Figure 4-31 Pareto front of bi-objective optimization problem under consideration.

This result shows the suitability of the Interpolation-based Kinematic Planner for individuating Pareto optimal sets with a more reliable method than the weighting method.

4.5 Conclusions

This chapter has been focused on the introduction and validation of a novel global inverse kinematics algorithm for redundant manipulators, the Interpolation-based Global Kinematics Planner. Such algorithm is an improvement over current state-of-the-art in that it is able to find multiple optima of different cost functions, with a wide set of constraints, without having to solve TPBVP and without including Jacobian matrices in the control cost. The cost functions that have been used is composed by a tracking error term and a control cost. Control costs that have been analysed are kinetic energy, joint torques squared norm, and joint torques squared norm with viscous friction. Different constraints, both linear and nonlinear, have been introduced as well, computing inverse kinematics solutions with constraints on joint displacements, velocities, torques, and powers. The Interpolation-based Global Kinematic Planner has been validated against an existing solution of the unconstrained kinetic energy global problem, and then results for two different trajectories and different control costs and constraints have been presented and commented.

The novel algorithm has correctly performed providing feasible solutions for the global kinematics problem of a simple planar manipulator in all simulations. Usual issues with global algorithms, featuring long computational times and high number of parameters, have been mitigated through a sequential solution procedure based on interpolation and introduction of new parameters at each sequence step. The algorithm has also been shown to be able to provide results concerning the Pareto front of bi-objective optimisation problems, individuating several points of the Pareto front of the problem of optimising kinetic energy and torques without friction, through the ε-constraint method.

Chapter 5. Workspace analysis of fixed-base and free-floating redundant manipulators

5.1 Introduction

It has already been observed in the literature review chapter that workspace analysis has been mostly performed with specific attention to reachability rather than energy or power costs to move the end-effector across the workspace itself. This chapter presents a different method of analysis, as the scope is in this case to map how energetically expensive it can be to reach specific points of the workspace from a fixed starting position. To this goal, the workspace analysis performed here starts from a specific end-effector position and performs straight line motions with constant end-effector velocity across the workspace, till the point where a singularity or the boundary of the workspace are reached. At each time step, the value of several energy-related variables, and of the manipulator kinematic indexes, is logged. A qualitative and a quantitative analysis are conducted on these data, aiming in both cases at assessing patterns in the energy consumption distribution along the workspace.

The goals of the analysis, in detail, are:

1. To show what patterns exist that relate kinetic energy with manipulator configuration. To this goal, a qualitative and a quantitative analysis are performed on fixed-base manipulators. The first one aims at observing visually recognisable patterns in the manipulator workspace, while the quantitative one aims at correlating kinetic energy with kinematics indexes dependent on the manipulator configuration. This latter part of the analysis will exploit canonical correlation to individuate the linear combinations of kinematic indexes that better correlate with kinetic energy.

2. To show how IK algorithms influence the size of the workspace and the position of singularities. This part of the analysis is especially developed for free-floating manipulators, as it is shown that the use of the ACS in concurrence with the manipulator’s operation can sensibly increase the workspace size and retain the manipulator manipulability.

3. To observe if redundant manipulators follow specific patterns in the distribution of kinematic indexes in their workspaces.

The first part of the chapter is a description of how the analysis is performed: it explains which algorithms are included in the analysis, which kinematic indexes are computed, and which energy-related figures are taken into account. Second part is about the simulations and results obtained with fixed base manipulators, while the third part is about free-floating manipulators. Conclusions are drawn from these results, about the difficult predictability of redundant manipulators energetic cost, and about the relationship between free-floating redundant manipulators inverse kinematics algorithm and workspace size and characterisation.

5.2 Choice of the algorithms under examination

In order to produce meaningful comparisons, three algorithms have been selected. These three methods have been considered particularly relevant for the problem for different reason: the first one is the most important pseudoinverse-based method, while the second one is a local minimization of kinetic energy, and the third one is a typical local optimisation method for free-floating manipulators. The first one of the three algorithms is expressed by (3-9), repeated here:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v} \tag{3-9}$$

The solution obtained through this expression locally minimises joint velocities norms along the motion. This method shall be from now on referred to with the abbreviation LSV (for Least Squares on Velocities).

A second method shall be chosen that directly minimises kinetic energy of the manipulator. A simple local method that achieves such a result can be obtained from (3-8), by using the inertia matrix as a weight matrix.

$$\dot{\boldsymbol{q}} = \boldsymbol{B}^{-1}\boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^{T})^{-1}\boldsymbol{v} \tag{5-1}$$

This method is mentioned in literature as a simple way to use the weight matrix [24]. However, its local scope means that kinetic energy may not be minimised overall. Still, it can sometimes be effective, especially on very short trajectories since, as it is straightforward to figure out, this kind of solution performs better when the trajectory can be followed with limited excursion of the first joint, as it is the one with the highest inertial properties. It shall from now on be referred to as LMKE (Local Minimisation of Kinetic Energy).

Free-flying manipulators, on the other hand, are mostly controlled through reaction minimisation algorithms: that is, transmission of torque form the manipulator to the base is minimised (often reduced to 0) along the manipulator motion. Many methods have been mentioned in the literature review, but one that is efficient and easy to implement is presented by Cocuzza et al. [97] and exploit the derivative of the general solution (3-12), again repeated here for convenience:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\mathbf{0}} \tag{3-12}$$

Which originates the expression:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{+}\left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right) + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\ddot{\boldsymbol{q}}_{\mathbf{0}} \tag{5-2}$$

Considering the torques to be defined as per (15), and neglecting the gravity term:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{5-3}$$

Where $\boldsymbol{M}(\boldsymbol{q})$ is the joint-dependent inertia matrix and $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the term that comprises Coriolis and centrifugal forces, it is possible to locally minimise torques transmitted to the base as a Least Squares problem with Equality constraints (LSE) in the $\ddot{\boldsymbol{q}}$ unknown, originating the expression:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{+}\left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right) + [\boldsymbol{M}(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})]^{+}\left[\boldsymbol{M}\boldsymbol{J}^{+}\left(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right) + \boldsymbol{n}\right] \tag{5-4}$$

This provides the third and last solution method to be compared and shall from now on be referred as LSE (Least Square with Equality constraints). It should be noticed that, differently from other inverse kinematics schemes hereby mentioned, this one outputs accelerations. This has a consequence in that joint velocities are generally different than zero at the end of the trajectory.

Of these three methods, first two are very related to the specific topic, because they are both attempts to minimise the energy cost of the manipulator motion, while the third one is used to solve a different problem in the framework of redundant manipulators local planners. All of them are relevant for the comparison, for different reasons: Moore-Penrose pseudoinverse is the most used method for local motion planning, local kinetic energy minimisation is the most direct way to minimise energy consumption, and reaction minimisation is the method of choice to control free-floating manipulators. The comparison for the latter one will indeed be based on less data than the first two, as it features a reduced workspace compared to them, allowing for less trajectories to be simulated.

5.3 Fixed-base manipulator workspace analysis description

5.3.1 Qualitative analysis description

Before moving into a quantitative analysis, a qualitative analysis is performed on the workspace. For non-redundant manipulator, each point in the cartesian space can only be reached with a very limited number of joint configurations, thus allowing to clearly map end-effector position with kinematic singularities. This is not the case for a redundant manipulator, since its peculiar characteristic is that infinite configurations exist that are able to achieve a given end effector position. From this, a natural question arises if patterns in the reachability and energetic cost of moving the end effector to specific parts of the workspace arise. The qualitative analysis hereby performed tries to answer this question.

In order to highlight the energetic characteristics of the motion, kinetic energy is of course the most relevant parameter. However, its integral is slowly varying on trajectories with constant EE velocity, and does not allow to evidence phenomena that are very limited in time, such as velocity spikes that take place when the manipulator is close to a singularity. Thus, for the purpose of this analysis, another index is used as well, that is more apt at visually analysing energy patterns in presence of singularities. Thus, a performance index that is used in this workspace analysis is the integral of the power absolute value along the trajectory:

$$\int_0^L \left|\frac{dE}{dt}\right| dl \tag{5-5}$$

This allows to consider power spent both to accelerate and slow down the manipulator end-effector. Its variability in correspondence of changes in velocity and acceleration of the manipulator joints can easily evidence energy-expensive patterns. This index will be from now on called *total energy*, and it will be exploited in the reminder of this chapter with the main purpose to evidence singularities.

The qualitative analysis will be illustrated in the reminder of the chapter by means of contour plots of kinetic energy, total energy and manipulability of the manipulator (see below for a reminder about manipulability formulation). The manipulator is conventionally consider to be fixed to the base in the origin of each graph, with its end effector being at the other tip.

5.3.2 Quantitative analysis description

Kinetic energy is in any case a function of velocities, and for such reason energetic performance is hard to capture with a static index. Furthermore, none of the existing indexes seems specifically fit for the purpose (in fact, only dynamic manipulability, shown in equation (3-17), features some resemblance with the kinetic energy expression). The approach that is followed here is thus to observe what linear combination of kinematic indexes is more closely related to kinetic energy. This approach allows to understand which factors influence the kinetic energy the most, and get a better understanding of the complexity of the problem under examination. To this goal, values of kinetic energy and each of the kinematic indexes are logged at each time step of each trajectory, and their correlations are analysed.

In order to perform such analysis, an interesting mathematical instrument is provided by statistics, which typically assesses if two events are related with each other through study of their correlation. Correlation between two variables X and Y can be expressed as:

$$corr(X,Y) = \frac{E[(X-\mu_X)(Y-\ \mu_Y)]}{\sigma_X\sigma_Y} \tag{5-6}$$

In this expression, E is the expected value operator, μ_X and μ_Y are the expected values of the two variables, and σ_X and σ_Y are their standard deviations. In the context of this work, it makes sense to ask if this concept can be extended to a set of variables, and it indeed can, exploiting a concept called *canonical correlation*, which was first introduced by Hotelling [153]. Given two sets of variables $X = (X_1, X_2, \dots ,\ X_n)$ and $Y = (Y_1, Y_2, \dots, Y_n)$, canonical correlation analysis aims to find two real vectors *a* and *b* that maximise the correlation:

$$\rho = corr(a^T X, b^T Y) \tag{5-7}$$

How ρ, *a* and *b* are calculated is beyond the scope of this thesis: it is sufficient to say that $a^T X$ and $b^T Y$ are called *canonical variables*, while *a* and *b* are the *canonical factors*. In the case of correlating manipulator kinematic indexes with kinetic energy only, vector *b* would of course be reduced to only one coefficient.

In the workspace analysis hereby presented, canonical variables factors have been calculated for the following set of kinematic indexes:

- Manipulability
- Dynamic manipulability
- Jacobian Condition number
- Worst-case velocity index
- Spectral radius of the inertia matrix

The latter one is not usually considered a kinematic index of robotic manipulators, but has been added because of the importance inertia matrix has in the computation of kinetic energy. Spectral radius of a matrix *A* is defined as:

$$\rho(A) = max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\} \tag{5-8}$$

Where $|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|$ are the absolute values of the eigenvalues of the matrix. Including it in the analysis allows to take into account a rough measure of the amount of inertia that the manipulator is facing during its motion. The other kinematic indexes are extensively described in next section.

Not all these variables are expected to have a sensible influence on the amount of kinetic energy that the manipulator reaches along the motion. For this reason, canonical correlation analysis shall be performed with subsets of the considered kinematic indexes as well. This would allow to only identify those indexes that are mostly related with kinetic energy. Considering *KI* to be a *j x 1* vector of kinematic indexes correlated with kinetic energy, and *b* its corresponding canonical factors, the function of kinematics indexes with the highest correlation with kinetic energy is defined to be:

$$C_{kin} = b^T \, KI \tag{5-9}$$

A question that might be asked is if such resulting combination of indexes can be somehow used to improve the search of kinetic energy optima. Its use in such a way is however not possible in a direct way, as the values of the kinematic indexes will not, in any case, incorporate complete information about the manipulator joints velocity. This topic will be discussed in more detail in next chapter, where results from this analysis will be tested as part of a local algorithm. It may however be possible to use the results presented in this chapter within a more complex heuristic search method, which will be subject of further work.

5.3.3 Kinematic indexes description

Many indexes have been developed to assess the freedom of movement and how far away current manipulator configuration is from singularities. Manipulability and dynamic manipulability have already been mentioned in the Mathematical Background chapter, but others do exist. To the author's knowledge, first index ever used to assess the Jacobian performance is the condition number, which is in general a performance index for matrix transformations in general. A discussion about its use for robotic manipulators was presented by Sainsbury and Craig [154]. The condition number of a mathematical function is defined as how much the output value of the function can change for a small change in the input. This is basically a measure of how errors in input produce errors in output. For the Jacobian of a robotic manipulator, the relationship between input and output variation is:

$$\frac{\|\boldsymbol{\Delta x}\|}{\|\boldsymbol{x}\|} = k\frac{\|\boldsymbol{\Delta q}\|}{\|\boldsymbol{q}\|} \tag{5-10}$$

When the manipulator is in a singular configuration, a big change in the joints velocities corresponds to a small change in the position of the end effector. Thus, a high condition number clearly indicates proximity to singularities. The Jacobian condition number can be calculated as:

$$k = \|\boldsymbol{J}\|\|\boldsymbol{J}^{-1}\| \tag{5-11}$$

On the opposite, the condition number of the inverse matrix reveals a singularity when it equals zero. A formula to compute it is:

$$k^{-1} = \frac{\sigma_{min}}{\sigma_{max}} \tag{5-12}$$

Where σ_{min} and σ_{max} are respectively the smallest and largest singular value of the Jacobian. Clearly, when one of the singular values equals 0, the matrix is singular. Being σ_{min} the smallest singular value, and being singular values positive semidefinite, k^{-1} must be 0 as well. This is related with the already presented concept of manipulability ellipsoid, introduced in the Mathematical Background chapter in expression (3-13). The reader may recall that, when manipulability equals zero, the manipulator is singular, and the volume of the manipulability ellipsoid equals zero as well. This is directly related to singular values [14], as an alternative formula to express manipulability is:

$$w = \prod_{i=1}^{n} \sigma_i \tag{5-13}$$

Where σ_i is the *i*-th singular value of the Jacobian. The singular values are proportional to the length of the ellipsoid axes, and one of them approaching zero means that any end-effector velocity with a component in that direction will result in a singularity.

It is interesting to observe that both feature limitations when used for performance prediction: k^{-1} is efficient in individuating singularities, but it doesn't say anything about the singular values apart from the smallest one. The case for manipulability is even worse, as it may increase by increasing any σ_i, which may however have no effect on getting further from an actual singularity. A more apt index to identify what's the worst performance attainable for a robotic manipulator in a certain configuration has been developed by Olds [155]. This index identifies in which direction the surface of the ellipsoid is closer to its centre, and can be expressed as:

$$\mu_{worst} = \frac{1}{\max\limits_{1<i<n} \left\| \boldsymbol{J}_i^{-1} \right\|_2} \tag{5-14}$$

Where $\boldsymbol{J}_i^{-1}$ is the *i*-th row of $\boldsymbol{J}^{-1}$. This quantity is called the *Worst Case Velocity Index*, and it defines the end-effector velocity attainable in the less favourable direction with current manipulability ellipsoid. More complex indexes have been developed as well [25], mostly for design purposes rather than for kinematic optimal control.

5.4 Fixed-base simulations and results

5.4.1 Fixed- base simulation setup

In order to compute kinetic energy and the afore-mentioned indexes in the whole workspace, a straight-path workspace approach has been followed. That is: the usual 3 DoF manipulator (redundant in the plane) has been initialised with a starting end-effector position has been chosen, $\boldsymbol{x_{init}} = [0.4678; -0.000]$, and the end-effector has then been moved along rectilinear trajectories along a number of directions trying to cover as much of the workspace as possible. Each trajectory has been considered to have reached the workspace boundary when one of the joint velocities or the end-effector error have been deemed too high. The threshold has been set to be 2 rad/s for velocities, and 0.03 m for end effector error. It should be noticed that a high end-effector error means the algorithm doesn’t work, so the latter condition has never been reached for fixed-base. The simulation has been repeated for five initial joint configurations, all of them featuring the same end-effector initial position. Such initial joint configurations have been chosen keeping the end-effector fixed and choosing five equidistant positions for the first joint, covering its whole possible range without changing the end-effector position. Two of them are such that the second and third joint are aligned, all the values are reported in Table 5-1.

Table 5-1 Initial configurations for workspace analysis

Initial configuration #	Joint 1 position	Joint 2 position	Joint 3 position
1	0.4009	-0.6233	0

2	0.2004	0.0374	0.6474
3	0	0.3269	-0.7541
4	-0.2014	0.6015	-0.6484
5	-0.4028	0.6233	0

Considering this 5 starting position and the three algorithms that have been selected to the part of the study, this gives a total of 15 simulated workspaces..

Simulations are performed with a simulator based on the general architecture presented earlier in the thesis, in Figure 3-2. Two components have been added to the algorithm: the first input is now provided by a Workspace Explorer component, while a Workspace limit observer is used to stop the simulation of each trajectory.

The algorithm now works as follows:

1. A set of directions is required as input from the user.
2. The Workspace Explorer sets the first direction to explore.
3. The Kinematic solver moves the End Effector along the desired trajectory.
4. Aforesaid kinematic indexes are calculated along the motion.
5. A Workspace Limit Observer checks if joints velocity limit or end effector position error are violated. If yes, the kinematics solver quits the simulation and the final length of the trajectory is stored.
6. The Workspace Explorer picks another direction to explore till all pre-set directions are explored.

Since every trajectory is independent from the others, this workspace exploring algorithm can be run using parallel computing to explore several directions at the same time, which speeds up calculation times and allows for a very high angular resolution of the workspace. Trajectories used for this kind of analysis are not only rectilinear, but they also feature constant end-effector speed. This is not fully realistic, as it produces an acceleration and peak on the first point of the trajectory. However, a different velocity profile would result in different parts of the workspace having different end-effector velocities, which would influence the results of workspace limits and kinetic energy, making it hard to compare results from different zones of the workspace itself.

The directions explored for each initial joint configuration and each algorithm are 512, equally spaced at π/256, which is this the angular resolution of the analysis, resulting in 7260 simulated trajectories. The end effector velocity is 0.05 m/s which, considering a time steps of 0.001s, gives a radial resolution of $5 * 10^{-5}$ m. Considering that the kinematic indexes and kinetic energy are measured at every single time step, this provides an amount of data in the range of millions for every initial joint configuration.

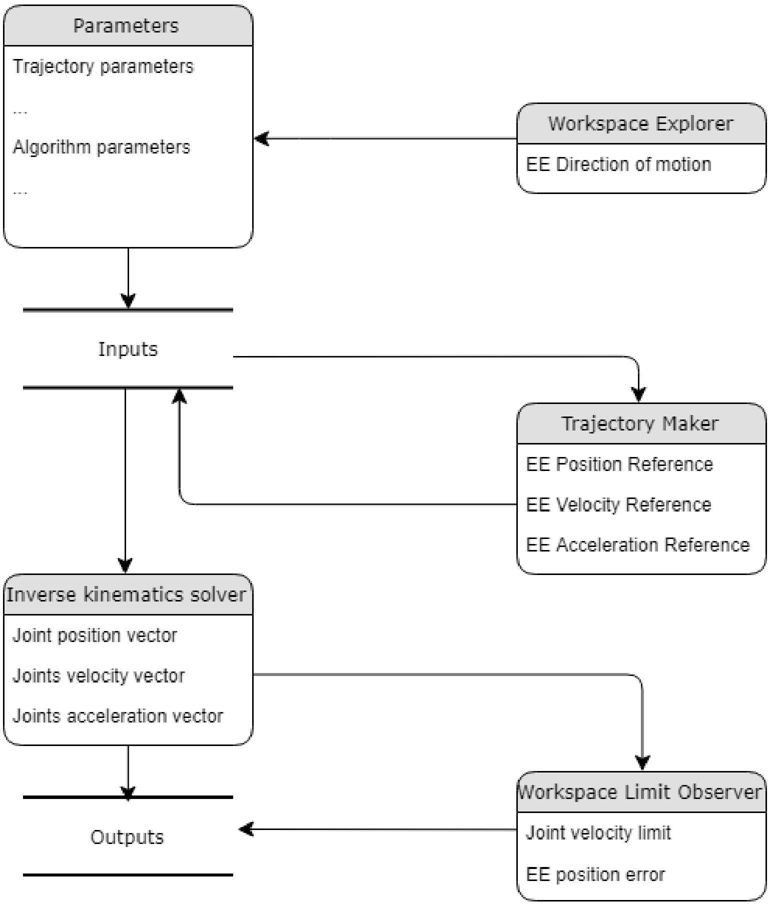


Figure 5-1 Simulator layout for workspace analysis

5.4.2 Fixed - base results

5.4.2.1 Qualitative analysis

Before observing results from canonical correlation analysis, it is interesting to have a look at Figure 5-2 and Figure *5-3*, which show contour lines for total energy for initial configuration 1 respectively for LSV and LMKE.

The distance between every contour line is 0.0004 J. Several peculiarities are worth noticing in these figures. First of them is that some preferential directions may be evidenced, and they are roughly perpendicular to the last joint. This does not vary depending with the algorithm, which allows to hypothesize that such directions depend on some intrinsic characteristic of the joint configuration rather than on the computational method chosen. Observation of similar patterns are confirmed for other configurations examined as well, as a further example Figure 5-4 and Figure 5-5 show total energy for configuration 2.

Another interesting peculiarity to observe are the wide differences in workspaces limits and position of singularities depending on initial joint configurations and algorithms, although not completely surprising, as the problem hereby examined is inherently nonlinear. LSE algorithm features the smallest workspace size (LSE workspace for configuration 1 is shown in Figure 5-6 as example), which is something to keep in mind for free-floating robots, as this is a clearly limiting factor for this solution method. LSV and LMKE, although they feature much bigger workspaces, are also affected by variability: the only common trait is that the robotic manipulator often has issues to complete longer trajectories (those passing through the central region of the workspace), which are more often affected by singularities. The limits of local algorithm when it comes to long trajectories have already been evidenced in literature [10].

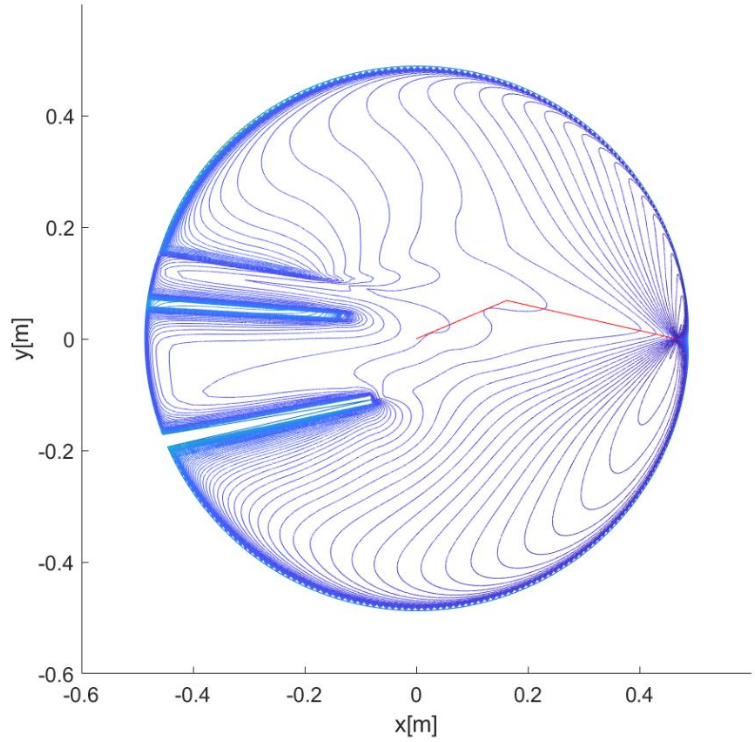


Figure 5-2 Total energy for configuration 1 solved with LSV

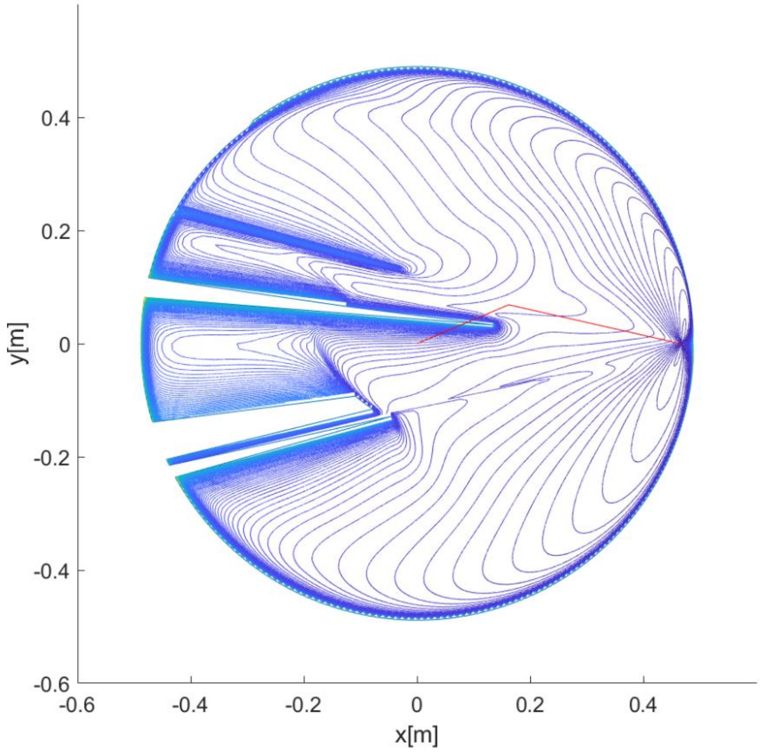


Figure 5-3 Total energy for configuration 1 solved with LMKE

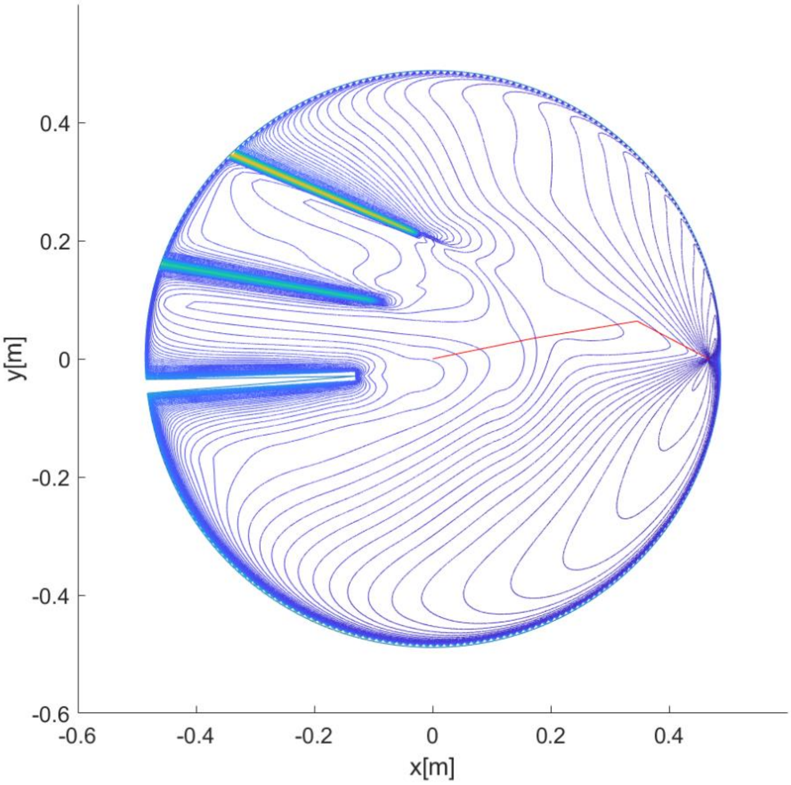


Figure 5-4 Total energy for configuration 2 solved with LSV

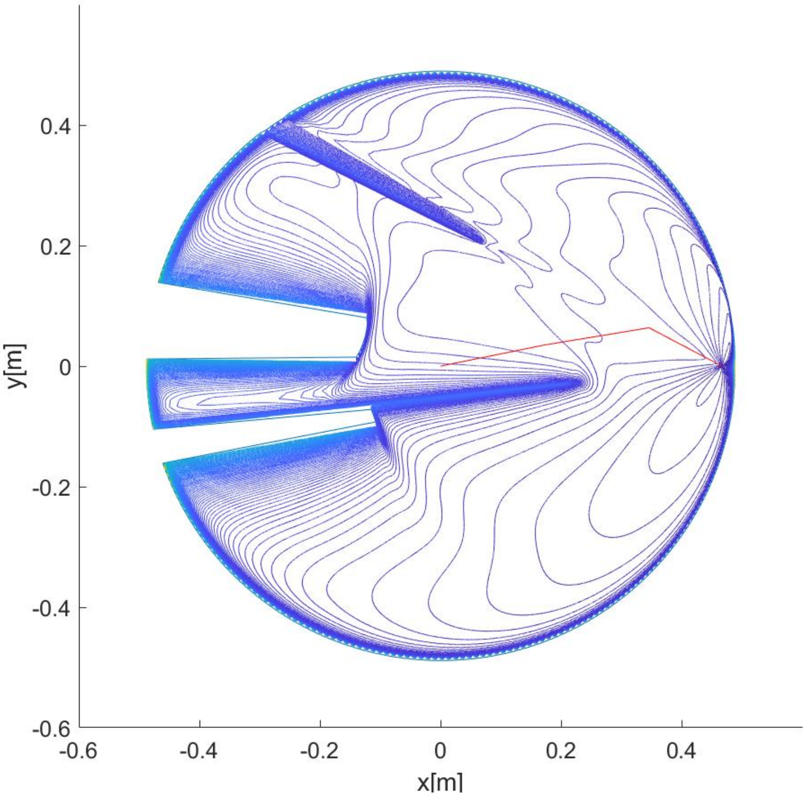


Figure 5-5 Total energy for configuration 2 solved with LMKE

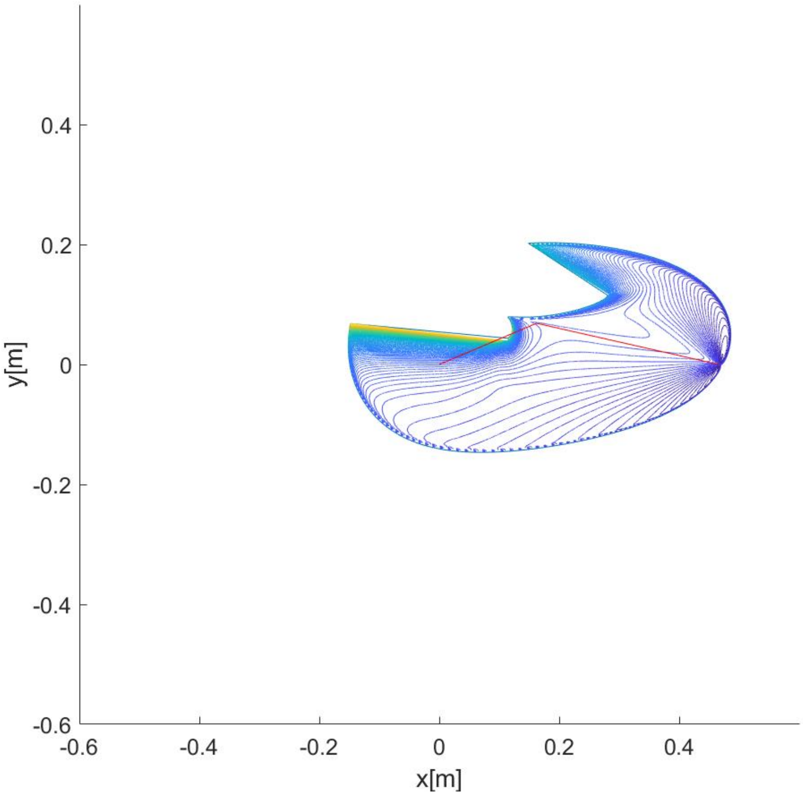


Figure 5-6 Total energy for configuration 1 solved with LSE

Singularities, in particular, appear in figures as points where the contour lines get very close to each other and brighter in colour, or where the workspace ends way before the manipulator has reached its full extension. They are not evenly distributed and there's apparently no pattern in them. For example, it can be observed that for LSV with starting joint configuration 3, a large portion of the workspace is not reachable due to the presence of singularities, as shown in Figure 5-7. One more, all important element to observe is that, while LMKE algorithm is explicitly based on minimising kinetic energy, although locally, it does not feature any special improvement over the LSV solutions over full trajectories, as can be easily observed by comparing the total energy figures for configurations 1 and 2.

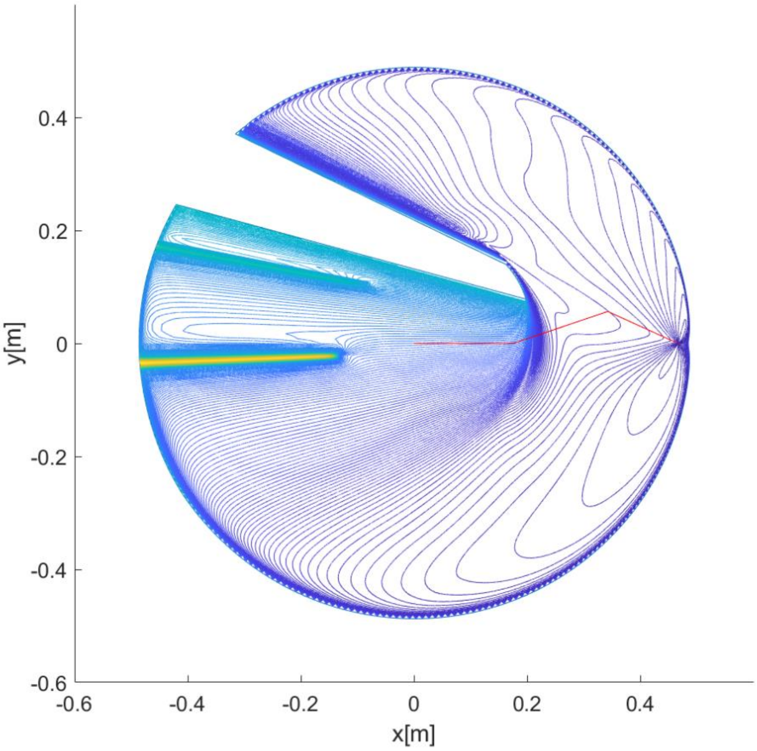


Figure 5-7 Total energy for configuration 3 solved with LSV

The explanation for this lies in the nature of the LMKE algorithm, which is simply based on weighting the joints depending on their inertia. This means it will use the outermost joint most, and resort to the ones closer to the base only when this is really needed. However, the reachability of the robot is of course limited unless the base joint is extensively moved as well. Due to the local scope of the algorithm, it is not possible to have any clue in advance when some extensive motion of the base joint will be needed to complete the trajectory. So it may happen that, at some point, a strong (and energetically expensive) adjustment of the base joint is needed in order to follow the trajectory. Generally, all algorithms presented so far are negatively affected by their local scope, which indeed is in most cases a very limiting condition.

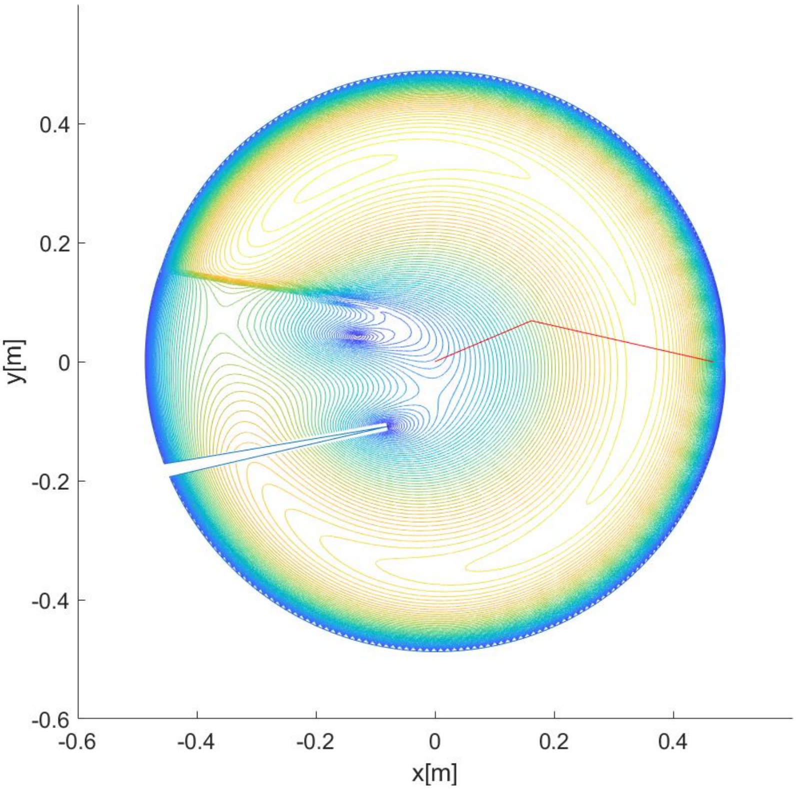


Figure 5-8 Manipulability for configuration 1 solved with LSV

Figure 5-8 illustrates manipulability contour lines for configuration 1 solved with LSV, with a distance of 0.001 between them. A structure similar to the one of non-redundant manipulators, as shown in the literature review chapter, is quite apparent: the centre and the external region of the workspace feature the lowest values (darker colours), while the intermediate region is the one with higher freedom of movement (brighter colours). Kinetic energy contour lines, in Figure 5-9, provide less interesting information, as the manipulator is moving very slowly, and kinetic energy generally increases in a noticeable way only in the neighbourhood of singularities. However, comparing the total energy and the manipulability, there's a first hint that they're not strongly related, as contour lines are completely different. The following paragraphs will show that canonical correlation analysis also confirms this, one of the most used kinematic indexes doesn't actually give any significant insight on how energetically viable it is to reach a certain point in the workspace.

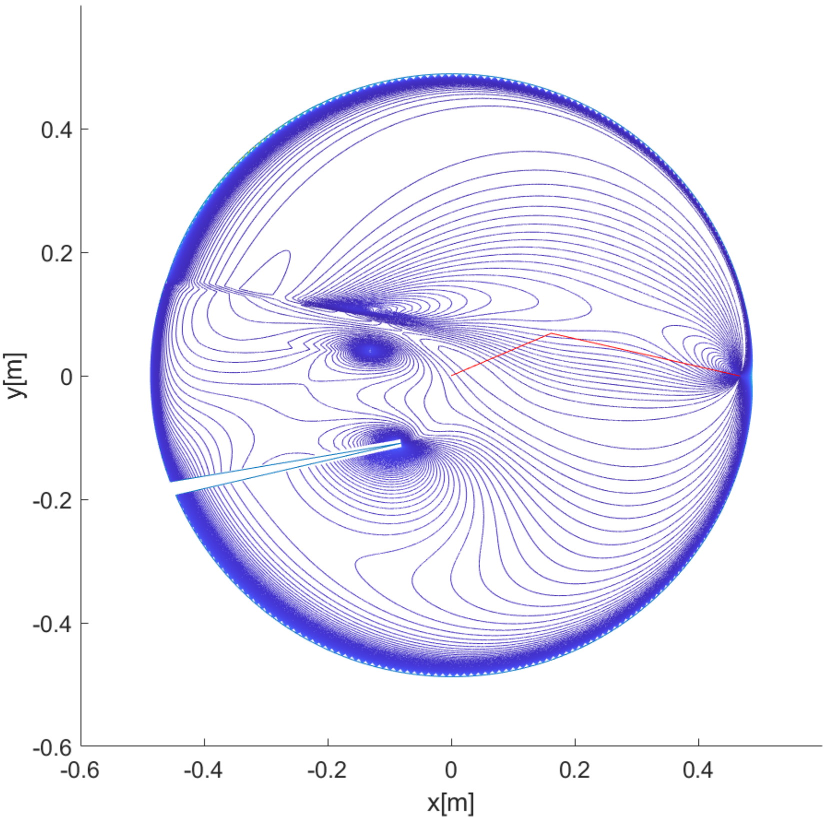


Figure 5-9 Kinetic energy for configuration 1 solved with LSV

5.4.2.2 Quantitative analysis

Before observing results from canonical correlation analysis, it may be interesting to have a look at Table 5-2, Table 5-3 and Table 5-4, which show the correlation between kinetic energy and each kinematic index for different positions and algorithms, respectively LSE, LSV and LMKE. The first difference can be drawn between LSE and LSV/LMKE: the first one shows sensibly lower correlation values for all variables. This can be explained by noticing that LSE is the only algorithm of the three that is acceleration based rather than velocity based. For LSV and LMKE, joint velocities are weighted by a mathematical formulation mainly involving the Jacobian matrix, which originates most of the kinematic indexes involved in the analysis. This provides a joint velocity vector that is more closely related with such matrix. For LSE, the computational method is completely different and, as such, the joint velocities are more responsible of the variation of kinetic energy than the Jacobian.

Table 5-2 Correlations of kinematic indexes with kinetic energy for LSE algorithm

LSE	Man.	Dyn. Man.	Cond. Number	Inertia Matrix spectral radius	Worst case velocity index
1	54.51%	53.85%	45.10%	53.73v	53.42%
2	53.63%	58.36%	45.81%	50.76%	51.83%
3	53.36%	58.79%	45.42%	50.84%	51.62%
4	52.16%	56.33%	49.79%	52.08%	51.04%
5	55.18%	54.54%	45.91%	54.40%	54.12%

Table 5-3 Correlations of kinematic indexes with kinetic energy for LSV algorithm

LSV	Man.	Dyn. Man.	Cond. Number	Inertia Matrix spectral radius	Worst case velocity index
1	34.06%	70.37%	71.57%	55.18%	30.05%
2	32.25%	73.00%	63.10%	47.05%	31.81%
3	36.17%	75.77%	50.19%	50.27%	38.64%
4	50.80%	71.15%	62.24%	61.07%	51.05%
5	34.05%	70.36%	71.55%	55.15%	30.04%

Table 5-4 Correlations of kinematic indexes with kinetic energy for LMKE algorithm

LMKE	Man.	Dyn. Man.	Cond. Number	Inertia Matrix spectral radius	Worst case velocity index
1	21.37%	68.44%	72.24%	41.03%	18.13%
2	20.32%	70.49%	72.49%	37.38%	16.91%
3	24.65%	72.10%	69.32%	39.43%	21.86%
4	25.39%	71.63%	68.43%	40.68%	23.00%
5	21.43%	68.36%	72.23%	41.19%	18.17%

More in detail, the LSE shows correlations in the range 45%-58% for all variables. The most correlated one with kinetic energy is indeed the dynamic manipulability, showing a correlation in the range 53.8%-58.8%, while the lowest one is the condition number, showing values between 45.1% and 49.8%. All the other kinematic indexes are instead correlated with values around 50%. This is generally too low to consider them relevant for the design of a kinematic planning algorithm.

When it comes to Moore-Penrose pseudoinverse solution (LSV), the difference is remarkable, in that the correlation with dynamic manipulability is in the range 70.3-75.8%, with the condition number being also noticeably correlated, in the range 50.2%-71.6%. The lower boundary of the range shows a correlation that is too small to be significant. However, looking at previously shown Table 5-3, it gives an insight on the condition number behaviour respect to kinetic energy when the LSV algorithm is used. When the workspace size is reduced (i.e. when a high number of singularities is present), the correlation between kinetic energy and condition number decreases. For example, initial joint configuration number 3 solved with LSV algorithm features a reduced workspace due to a high number of singularities on the top left side, as previously shown in Figure 5-7, and correlation is low. Manipulability, on the other hand, is not strongly correlated with kinetic energy for LSV solutions, with the correlation value being in the range 32.2%-50.8%. Manipulability has sometimes been proposed as a good index to base singularities avoidance algorithms ([12]), and it can indeed prove its worth for this purpose. It is however not the most significant index for kinetic energy minimisation. The other two variables examined, inertia matrix spectral radius, and worst-case velocity index, feature correlations respectively in the range 47.1%-61.1% (with most values around 50%), and 30-51%.

The correlations between configuration-dependent indexes for LMKE is slightly different, with dynamic manipulability being in the range 68.4% – 72.1% and Jacobian condition number in the range 68.4% -72.5%. The former is slightly less correlated than in the LSV case, while the latter is more correlated. On the other hand, manipulability feature correlation in the range 20.3%-25.4%, while the range is 37.9%-41.1% for the inertia matrix spectral radius, and 16.9%-23% for the worst case velocity index, showing that in the LMKE case the only really correlated indexes are dynamic manipulability and Jacobian condition number.

At this point, it is clear canonical correlation analysis requires some extra attention to be put on the linear combination of dynamic manipulability and Jacobian condition number. Correlation coefficients for the case with only dynamic manipulability and Jacobian condition number, and for the one with the whole set of the five static indexes are presented for the cases LSV and LMKE respectively in Table 5-5 and Table *5-6*, for each initial configuration. In the LSV case, the difference spans from 1.4% to 4.3%, while for LMKE it is in the range from 5.6% to 5.8%, showing more uniformity, possibly because of the lower number of singularities. Canonical coefficients are harder to interpret than correlation: their values can be observed in Table 5-7 and Table *5-8* for the whole set of configurations with LSV and LMKE. Results for LSV show no uniformity, apart from the fact that dynamic manipulability coefficient always stand out as the highest one. Apart from this, no other patterns are recognisable (coefficients for all other kinematic indexes do not even feature same sign). This changes for LMKE, for which again dynamics manipulability is the highest scoring variable, which can be expected due to its dependency on inertia matrix and Jacobian, which resembles kinetic energy formulation. The coefficients for all the other indexes are generally more stable than with LSV case, featuring the same sign for all configurations and, apart from manipulability, also the same order of magnitude.

Table 5-5 Canonical correlation of dynamic manipulability and condition number with kinetic energy

	1	**2**	**3**	**4**	**5**
LSV	73.55%	73.97%	75.93%	73.38%	73.54%
LMKE	73.05%	74.45%	74.50%	73.75%	73.02%

Table 5-6 Canonical correlation of the full set of indexes with kinetic energy

	1	**2**	**3**	**4**	**5**
LSV	75.83%	77.84%	77.36%	77.70%	75.83%
LMKE	78.54%	80.12%	80.08%	79.48%	78.49%

Table 5-7 Canonical coefficients for LSV

LSV	**1**	**2**	**3**	**4**	**5**
Man.	-9.3401	3.1711	-17.7220	-45.8844	-9.2247
Dyn. Man.	82.7259	100.5888	106.7686	52.3402	82.9785
k	0.0408	0.0676	-0.0026	0.0608	0.0405
$\rho(A)$	-6.8033	-7.7683	2.8284	5.0497	-6.8595
μ_{worst}	-0.0816	-1.2814	0.4824	15.0330	-0.1262

Table 5-8 Canonical coefficients for LMKE

LMKE	**1**	**2**	**3**	**4**	**5**
Dyn. man.	-20.2199	-2.4697	-9.6742	-17.5337	-20.0768
Man.	46.2659	42.1089	57.8107	58.1065	46.0321
k	0.1474	0.1694	0.1265	0.1223	0.1489
μ_{worst}	-6.1832	-8.4395	-6.8359	-6.6226	-6.2377
$\rho(A)$	5.6048	1.2729	2.0837	5.0609	5.6025

At this point, some hypothesis can be formulated about the best approximator for kinetic energy with a linear combination of configuration dependent indexes. It seems sensible not to use results from LSE, as the objective of the algorithm is not related to the minimisation of kinetic energy and correlation coefficients are pretty low, meaning it is probably not particularly helpful. LSV gets closer to the objective, by featuring higher correlations. Still, it features two issues: its coefficients are quite variable depending on the initial configuration, making a choice hard, and its objective is optimising the norm of joint velocities rather than kinetic energy. LMKE is not only more related, as the cost function is kinetic energy, but it also features more uniformity in correlations.

A local approximator for kinetic energy, as described by equation (5-9), can thus be developed averaging the canonical coefficients with LMKE algorithm for all the starting configurations. Since most of the coefficients are do not show high correlations with kinetic energy, only dynamic manipulability and condition number are part of the index, as the other variables do not seem to be relevantly correlated. The formula used is:

$$KI = 31.0116 * dyn.m. + \quad 0.1234 * k \tag{5-15}$$

This function represents a configuration-dependent index that is highly correlated with kinetic energy, and it is supposed to convey as much information about kinetic energy as possible. However, in next chapter, it will be used as an approximation of kinetic energy within an optimisation algorithm, showing that local functions depending solely on the manipulator joints configuration hardly have any capability to predict the behaviour of a dynamic quantity such as kinetic energy.

5.5 Free-floating manipulators analysis

5.5.1 Scope of the analysis

The fixed-base case computation is useful to observe the peculiarities of the workspaces generated by each algorithm. Particularly, the LSE method, which is the closest one to those actually used on free-floating manipulators ([109]), generates smaller workspaces than the others. This limit has already been observed in literature [92]. Furthermore, it is to be observed that the base attitude must be kept stable to respect its stability requirements. This enforces and extra-constraint on the angular velocity of the base spacecraft, and it will be shown that indeed the results are almost the same with all the three algorithms presented so far, due to the fact that the extra degree of redundancy is lost.

On the other hand, Vafa and Dubowski [92] also observe that, if the base attitude is actively controlled, the workspace of a free floating manipulator can be increased, although it won’t reach the size of a fixed-base one (or, more precisely, it will tend to the size of a fixed-base one as the moments of inertia of the base increase). This is also very interesting for the purpose of optimisation algorithms, as the use of an attitude actuator could not only increase the workspace, but provide an extra degree of freedom, which in turn would allow to optimise a cost function on top of kinematic constraints. The analysis that is performed here for free-floating manipulators is aimed at assessing how the workspace is changed by the inclusion of an attitude actuator, a reaction wheel, among the actuator controlled degrees of freedom. This is relevant for the research questions asked in this thesis as an extra degree of freedom provided by the ACS allows to solve the IK problem with redundancy resolution methods that are used for fixed-base manipulators, provided that the Jacobian is updated according to the new configuration. In order to perform this analysis, two steps are performed:

1. An extra constraint is added to the Jacobian, enforcing the base rotation rate to be equal to 0 with an extra row. It is then shown that there is no difference between minimization of manipulator momentum transmitted to the base, and enforcing a constraint on the base while using velocity-based inverse kinematics schemes.

2. A simple ACS constituted by a reaction wheel has been added to the Jacobian, constituting an extra column. This restores the redundancy to the manipulator, allowing to make use of any optimisation algorithm same as for fixed-base manipulators. In this case, a simple local optimisation has been used for demonstration purposes, but solution methods as the ones developed in this thesis can be used as well.

The scope of the analysis is thus to evidence how the ACS allows to retain freedom of motion for a free-floating redundant manipulator beyond the workspace that can normally be accessed through reaction minimizing solutions. It is thus argued that energy minimizing solutions exploiting concurrent control of the manipulator and the ACS bear their own advantages for operating space manipulators over reaction minimising solutions. The main tool used for this analysis is manipulability contour plots, which will be used as a measure of overall manipulator freedom of movement.

5.5.2 Simulation setup

The free-floating problem is much more mathematically complex than the fixed-base one, as the Jacobian and Inertia Matrix feature an increased degree of complexity.

For a planar manipulator such as the one used here, the base spacecraft has only one angular velocity, which is here identified with the symbol $\dot{\varphi}$, coherently with φ being used for the spacecraft angular position as mentioned in the chapter dedicated to mathematical background. In order to actively control it, a further row is added to the Jacobian. The direct kinematics problem then results into:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{bmatrix}=\begin{bmatrix}J_{11} & J_{12} & J_{13}\\ J_{21} & J_{22} & J_{23}\\ J_{31} & J_{32} & J_{33}\end{bmatrix}\begin{bmatrix}\dot{q_1}\\ \dot{q_2}\\ \dot{q_3}\end{bmatrix} \tag{5-16}$$

Where the Jacobian is no more redundant and the last row has been obtained by rearranging the conservation of angular momentum as per (3-26). The full terms have been omitted due to their complex formulation. A similar procedure is then performed by adding a reaction wheel with fixed inertia, resulting in:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{bmatrix}=\begin{bmatrix}J_{11} & J_{12} & J_{13} & J_{14}\\ J_{21} & J_{22} & J_{23} & J_{24}\\ J_{31} & J_{32} & J_{33} & J_{34}\end{bmatrix}\begin{bmatrix}\dot{q_1}\\ \dot{q_2}\\ \dot{q_3}\\ \omega_{RW}\end{bmatrix} \tag{5-17}$$

Where ω_{RW} is the reaction wheel rotational rate. In what follows, LSE and LSV based examples use the Jacobian as per equation (5-16), while the LMKE examples are computed with Jacobian (5-17).

The rest of the simulation set up is similar to the one already introduced for the fixed-base case, but with two differences: although the same initial end-effector position has been used, only one initial configuration is sufficient to demonstrate the workspace differences with and without ACS, so only configuration number 4 has been used. The other difference is that the angular resolution has been reduced for this case, due to the increased computational complexity and the lack of need for a statistically relevant quantity of data. The chosen angular resolution is $\frac{\pi}{64}$. Manipulability contour lines will be used to show that the reaction wheel can help to retain the manipulator freedom of movement in areas of the reachable workspace where it would normally be absent. Workspaces without reaction wheel are presented first, with results with the reaction wheel following afterwards. In all cases, the end-effector is moved in rectilinear trajectories at a speed of 0.025 m/s.

5.5.3 Results

Figure 5-10 and Figure 5-11 respectively show the manipulability characteristics of the straight-path workspace of a free-floating manipulator without reaction wheel for LSE and LSV, when All workspaces are represented with manipulability contour lines at a distance of 8E-7. It is easy to see that the two results are almost the same. Since there is no redundancy involved, either the minimisation of the reactions, or equating to zero the rotation of the base spacecraft, exploit the extra degree of freedom, producing the same result. This is not true when a reaction wheel is used, which provides a further degree of freedom, leaving room for an optimisation task on top of the base spacecraft rotational constraint.

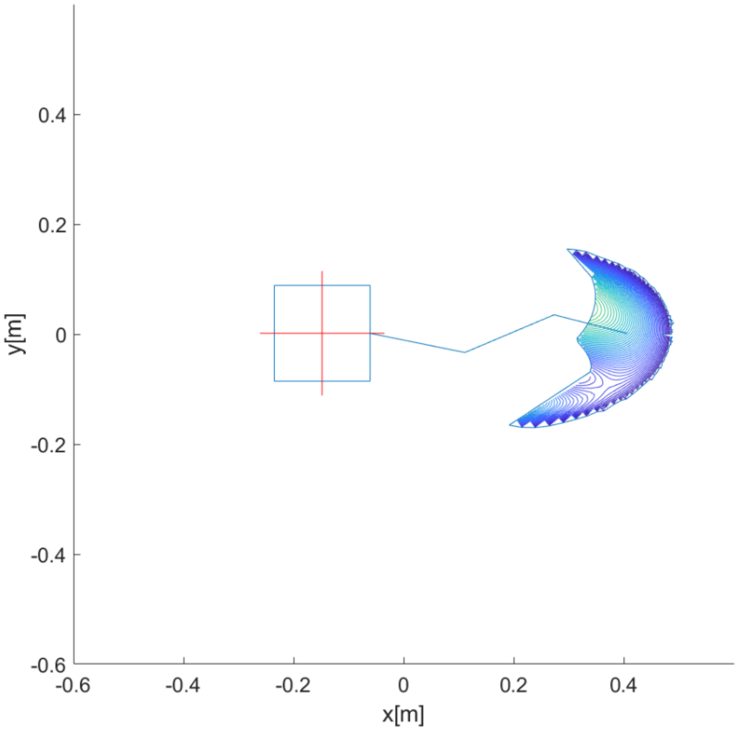


Figure 5-10 Free-floating manipulator workspace with LSE algorithm

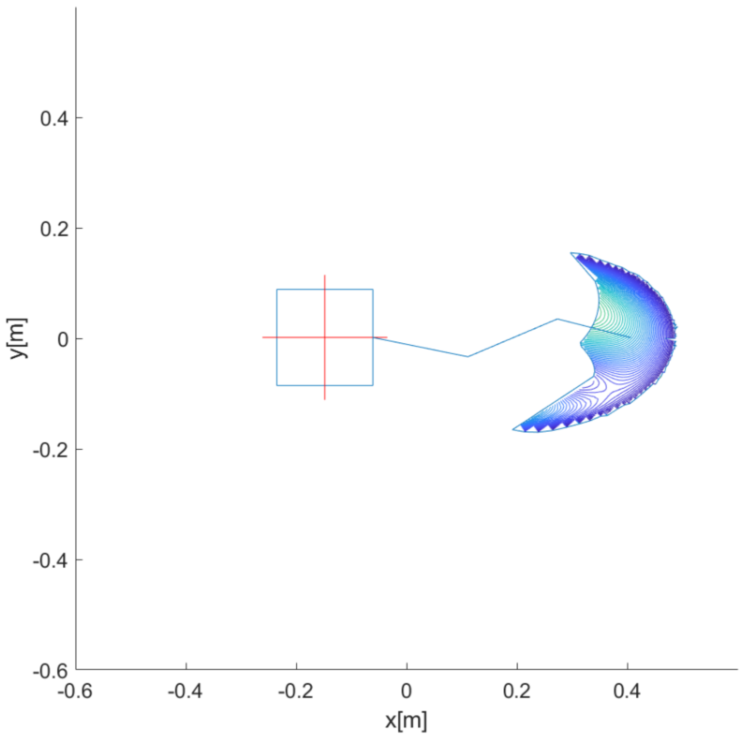


Figure 5-11 Free-floating manipulator workspace with LSV algorithm

The reaction wheel, as already seen, allows to insert a further column in the Jacobian of the manipulator. The momentum conservation equation constitutes a non-holonomic, non-integrable constraint for the motion. Such a constraint can usually require the base spacecraft to rotate in order to be satisfied, since momentum can be generated from the arm motion, which must be countered by the base. A reaction wheel can be used to counter the momentum instead of having the base rotate, allowing it to be kept stable. In fact, it can "store" some of the momentum by its rotation, which can be later discharged. One further workspace simulation has been performed with LMKE algorithm in order to show what changes this implies. The results, again with manipulability iso-curves distant 8E-07 from each other, is shown in Figure 5-12. It is possible to notice that this figure can be roughly divided in two zones: one where the manipulability iso-curves are very close, and thus manipulability changes a lot, and one where it is much less variable. The position of the reaction wheel does not appear in the Jacobian, and thus does not change the relationship between the joints velocities and the EE (or base) velocities, nor it introduces extra singularities. It is in fact trivial to observe that, for planar motion, the wheel inertia does not depend on its rotational position. From this, the conclusion is that there is an internal region, closer to the reaction free part of the workspace, where the motion is mostly achieved by the use of the manipulator actuators. Outside of this region, the manipulator is no more able to provide reaction-free motion, and the reaction wheel is mostly in charge for the accomplishment of the motion. In this case, the configuration of the robotic arm will still change, but not as much as in the reaction-free region, keeping manipulability more stable. If true, this somewhat confirms the results presented in literature.

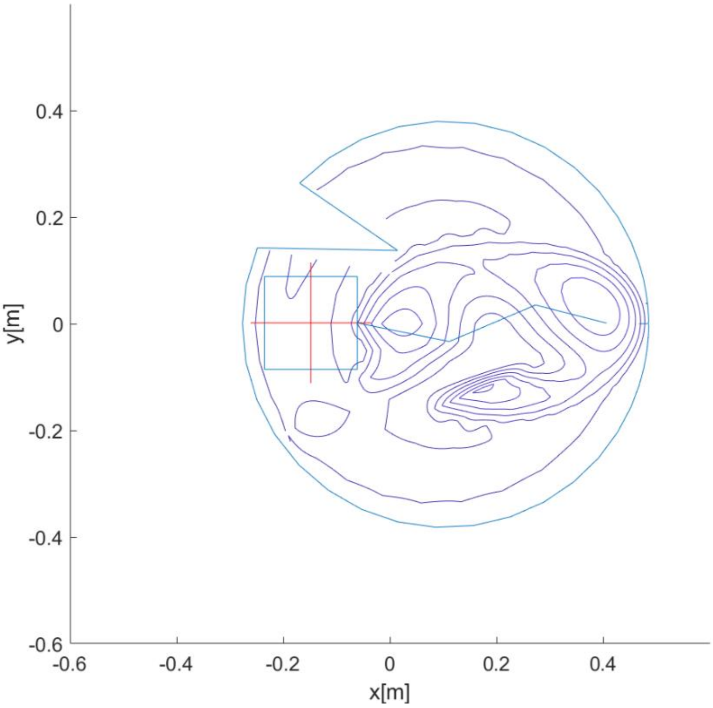


Figure 5-12 Free-floating manipulator workspace with reaction wheel with LMKE algorithm

The stroboscopic plot of a sample trajectory, computed with LMKE, is shown in Figure 5-13, with the joints and reaction wheel velocities being shown in Figure 5-14 and the manipulability in Figure 5-15. Colours for the joints are blue for the first joint, red for the second, and yellow for the third, while the reaction wheel is shown in purple. The reaction wheel is very expensive from the energetic point of view, and thus the kinetic energy locally minimising algorithm tries to avoid its use as much as possible if the motion can be accomplished without it. Once the limits of the reaction free workspaces are reached, around time 0.6s, the algorithm makes extensive use of the wheel in order to keep the base spacecraft in position. At this point, the manipulability is totally stable and almost doesn’t change. This result is aligned with existing literature [93] and shows that attitude control system can be used to increase the degree of redundancy of a free-floating manipulator and to implement optimisation methods on the motion planning of such manipulator. Incidentally, it is also shown that a local kinetic energy minimisation may imply higher expense later in the trajectory, supporting the case for a prediction-based minimisation algorithm such as the one developed in next chapter.

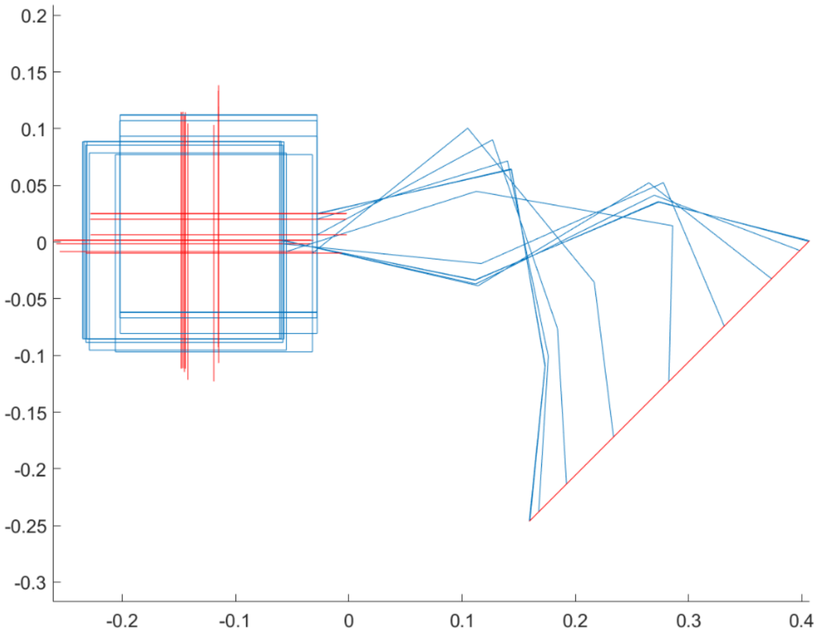


Figure 5-13 Stroboscopic plot of a sample free-floating trajectory

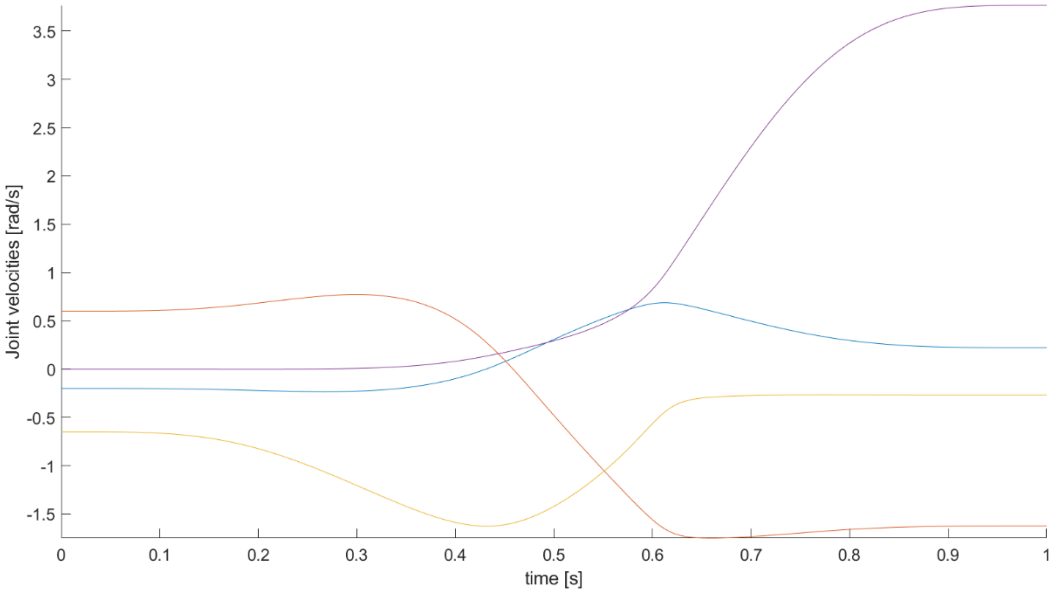


Figure 5-14 Joint velocities of a sample free-floating trajectory

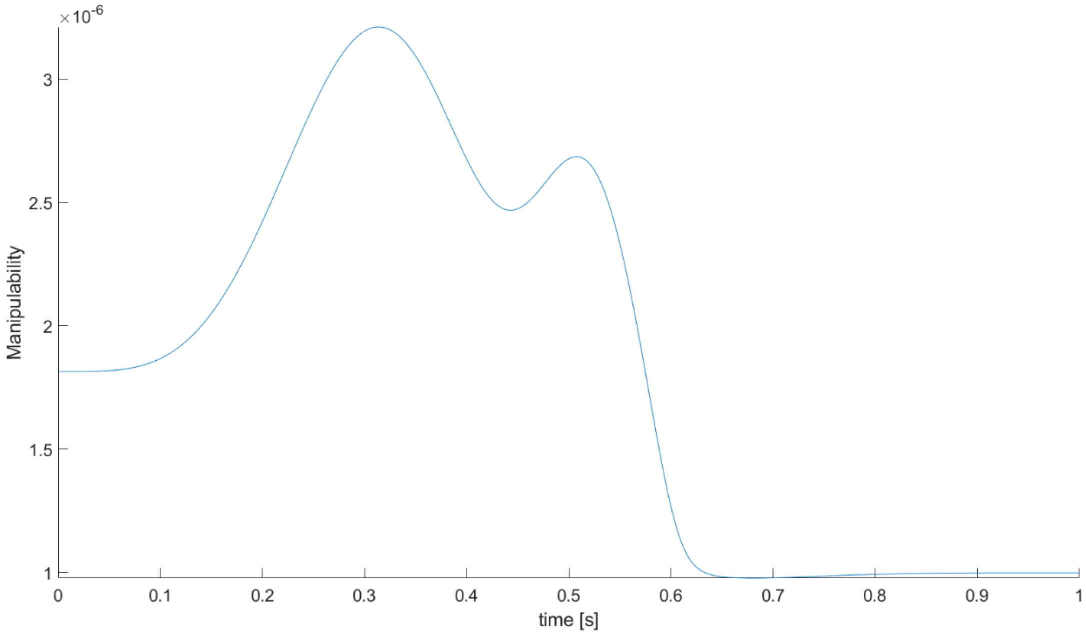


Figure 5-15 Manipulability of a sample free-floating trajectory

5.6 Conclusions

In this chapter, the workspace of a simple redundant manipulators has been analysed in a systematic way. First, kinematic indexes that are related to the configuration of a robotic manipulator have been introduced, alongside with their relationship with the manipulators' singularities. After this, canonical correlation has been introduced as an instrument to analyse the links between such indexes and kinetic energy, evidencing which kinematic indexes influence the kinetic energy of a manipulator most. An analysis has been performed on the workspaces of fixed-base manipulators, observing the complexity and nonlinearity of the relationship between manipulator configuration and energy. A linear combinations of dynamic manipulability and condition number has been evidenced to be the closest correlated function to kinetic energy. Such combination will be used in next chapter within a local algorithm to illustrate the fundamental incompleteness of knowledge regarding energy consumption when only local configuration-related information is considered. Finally, the extensibility of results to the free-floating case has been investigated, by adding a reaction wheel as an attitude control actuator to be used in conjunction with the manipulator joints. Such actuator has been controlled through the same Jacobian as the arm. The effect of the reaction wheel on the manipulator kinematics has been discussed and presented though simulation of the manipulator free-floating workspace.

Chapter 6. A local method based on the prediction of future kinetic energy integral

6.1 Introduction

In this chapter, a novel online inverse kinematics method for redundant manipulators is introduced and compared with three existing algorithms. Such a method can consistently reduce kinetic energy integral during the manipulator motion. The underlying principle of the algorithm is to estimate the direction of minimum kinetic energy expense by optimising a local approximation of the future kinetic energy integral. The prediction, which does not need to be updated at every time step, has been performed in two different ways: with the exact formula of kinetic energy to illustrate the validity of the approach in reducing kinetic energy, and with the kinetic-energy correlated function that has been derived in last chapter, to further show how kinetic energy results are unpredictable when the prediction is solely based on local variables. The prediction can be computed with a restricted number of parameters, resulting in a limited computational cost. As a result, the proposed method has enhanced singularity avoidance capability, alongside with well-behaved trajectories from the energetic point of view. Although the predictive algorithm does not involve Jacobian inversion in itself, its solution has been used as an extra task in a pseudoinverse-based inverse kinematics scheme. This provides precision tracking and allows to use the new method alongside existing pseudoinverse related theory.

As previously outlined, extra task formulations do exist to overcome singularities, which are the most catastrophic failures of local algorithms. Jacobian singularities imply that a certain end-effector velocity is not attainable with finite joint velocities, and of course such situation should be avoided in practical manipulators: much before reaching the singularity, high velocity and vibration may cause physical damage to the manipulator.

While this occurrence has long been mitigated by the use of specific singularity avoidance algorithms ([6], [11]), this only allows to prevent extremely suboptimal trajectories, but doesn't guarantee any optimality or near optimality from the energetic point of view. In order to obtain an energetically optimal trajectory, future energy cost has been here explicitly formulated, computed and taken into account. In order to compute its value as quickly as possible, the algorithm hereby presented relaxes the requirements for full optimality over the prediction horizon, choosing to limit the number of future path points taken into account and the frequency of updates of the prediction, which is not recomputed at every time step.

This chapter is organised as follows: first section presents an overview of the chapter and the methodology used for investigation, then the mathematical formulation of the new algorithm and the parameters of the optimisation are introduced. Afterwards, numerical results to validate the algorithm are presented and discussed. The chapter is completed with an analysis of the new algorithm's main parameters and main drivers in their choice and setting.

6.2 Mathematical formulation of the new predictive algorithm

The new algorithms hereby presented are both based on the possibility, widely exploited in literature to superimpose a further constraint to a pseudoinverse-based solution through the following formulation, already mentioned in previous chapters:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{q}}_{\boldsymbol{0}} \tag{3-12}$$

The homogeneous solution $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$ is in this case used to add a further term based on an integral cost, such as the ones typically used in optimal control problems [139], comprising the integral of squared end effector error along the trajectory plus the kinetic energy integral along the trajectory. In the case where the whole length of the trajectory is optimised at once, a widely used expression which has already mentioned before in this document:

$$C(\boldsymbol{q}, \boldsymbol{u}, t) = \int_{t_0}^{t_{fin}} \left[\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right)^T \boldsymbol{Q}\left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_{\boldsymbol{ref}}\right) + \ \boldsymbol{u}^T \boldsymbol{R}\,\boldsymbol{u}\right] dt \tag{3-32}$$

Final time t_{fin} doesn't however need to be the time when the manipulator motion is completed. It is possible to define a time horizon h, much shorter than the whole motion time. In this case, rather than optimising a control cost over the whole trajectory, it would only be optimised over a time window that extends from present time t_0 to future time $t_0 + h$. Furthermore, the quadratic control cost in the general case can be any motion-related function. The discrete formulation used for the PMKE algorithm developed by the author is:

$$C(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=t_0}^{t_0+h} [\left(\boldsymbol{x}_{\boldsymbol{i}}(\boldsymbol{q}_{\boldsymbol{i}}) - \boldsymbol{x}_{\boldsymbol{i,ref}}\right)^T \boldsymbol{Q}_i \left(\boldsymbol{x}_{\boldsymbol{i}}(\boldsymbol{q}_{\boldsymbol{i}}) - \boldsymbol{x}_{\boldsymbol{i,ref}}\right) + \ G(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)\ ]\Delta t \tag{6-1}$$

$\boldsymbol{x}_{\boldsymbol{i,ref}}$ is the reference end effector positions at time i, $\boldsymbol{x}_{\boldsymbol{i}}$ the actual end effector positions at time i, $\boldsymbol{k}_{\boldsymbol{i,err}}$ is the error weight matrix at time i, , $\dot{\boldsymbol{q}}_{\boldsymbol{i}}$ the joint velocities at time i, Δt is the discrete time step and $G(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ is a control cost function. For finite horizon optimisation, as in the case presented here, t_{fin} is dropped in favour of a closer time $t + h$, where *h* is a finite time horizon.

Computing such a cost for a real-time tracking algorithm is expensive and time consuming. For this reason, the algorithm hereby proposed exploits the pseudoinverse solution to track the trajectory instead. The integral cost is used instead to provide a velocity vector $\dot{\boldsymbol{q}}_{\boldsymbol{prediction}}$ pointing to the direction where kinetic energy integral is expected to feature the lowest value for the chosen trajectory. This velocity vector is used as an extra task added to the motion through the Jacobian null-space as per (3-12).

The steps of the algorithm are as follows:

1. A horizon *h* is selected.

2. A prediction step size ΔT is selected. It is a multiple of the pseudoinverse algorithm step size Δt and it is generally much larger.

3. An update interval size I is selected. It is a multiple of the pseudoinverse step size Δt, but it is generally smaller than ΔT.

4. Expression (74) is optimised with sampling times $t = t_0 + \Delta T, \dots, t_0 + h\Delta T$. Any deterministic optimisation algorithm can be used, for example any gradient descent method. This provides a prediction for a future window that is defined here as prediction window $T_p = h * \Delta T$.

5. The solution of the optimisation at point 4 outputs a set of h joint configurations, for the *h* sampling times $t = t_0 + \Delta T, \dots, t_0 + h\Delta T$. The set of these points is from now on referred as $\{\boldsymbol{q}_{\boldsymbol{prediction}}\}$. A specific point in the set is referred with its sampling time as a subscript, with a notation such as $\boldsymbol{q}_{\boldsymbol{\Delta T}}$.

6. A 4-th order spline interpolation from $\boldsymbol{q_0}$ to $\boldsymbol{q_{h\Delta T}}$, passing from all the points in the set $\{\boldsymbol{q_{prediction}}\}$ is computed. Endpoint conditions of the interpolation are set so that $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ at $\boldsymbol{q_0}$ are set equal to the ones reached by the manipulator at time t_0, and are set to zero at $\boldsymbol{q_{h\Delta T}}$. This ensures the resulting spline won't have large slopes in the interval of interest. The resulting spline is from now on referred as $\boldsymbol{q_{prediction}} = \boldsymbol{f}(t)$.

7. For each step *i* in the update interval I, the trajectory is computed with pseudoinverse method as per expression (3-12), using the error between previous $\boldsymbol{q_{i-1}}$ and the computed value of $\boldsymbol{q_{prediction}}$ as a secondary task.

$$\dot{\boldsymbol{q}}_{\boldsymbol{secondary}} = \frac{\boldsymbol{q_{prediction}} - \boldsymbol{q_{i-1}}}{dt} \tag{6-2}$$

8. Once I time has passed, steps 4 to 6 are repeated for the new update interval, using the last time step of previous interval as t_0.

9. Steps 4-8 are repeated till the amount of time before the end of the trajectory is T_p.

10. Steps left at this point are computed with the last computed $\boldsymbol{q_{prediction}}$.

Two version of the algorithm are presented here, differing in the control cost $G(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$. The first one has been called Predictive Minimisation of Kinetic Energy (PMKE), and it uses kinetic energy as control cost, while the second one, defined as Predictive minimization of kinematic indexes correlated with kinetic energy (PMCI), is different in that it uses the approximation of kinetic energy developed in last chapter as control cost. The first formulation is the main research contribution of this chapter, while the second one is provided for comparison, in order to show the difficulties in improving the kinetic energy integral of a trajectory by only using local information from the manipulator configuration. Such difficulties persist even when trying to optimize locally defined variables in a predictive way, as the PMCI algorithm does.

Thus, the formulation of PMKE control cost is, as usual with kinetic energy:

$$\frac{1}{2}\dot{\boldsymbol{q}}_{\boldsymbol{\imath}}{}^{T}\boldsymbol{W}(\boldsymbol{q}_{i})\dot{\boldsymbol{q}}_{\boldsymbol{\imath}} \tag{6-3}$$

Where $\boldsymbol{W}(\boldsymbol{q}_i)$ is the configuration dependent inertia matrix. The formulation of the control cost based on correlated indexes is instead as per formula (*5-15*):

$$KI\ =\ 31.0116\ *dyn.m.+\ \ 0.1234\ *\ k \tag{5-15}$$

This formulation will be used, in lieu of kinetic energy, to assess if a control cost only dependent on characteristics of the manipulator configuration can be used instead of a cost containing velocities, such as kinetic energy.

6.3 Validation

6.3.1 Validation strategy

It has already been explained in previous chapters that local algorithms are harder to evaluate than global algorithms because they are not built to find the best optimum, but rather to find a reasonably good optimum in a reasonably short time. Furthermore, for nonlinear problem like inverse kinematics, a local algorithm might be good for a specific set of trajectories and be inadequate for another. This happens because finding the best solution at a given time step might be detrimental for the performance at future time steps, and most of the times there's no precise way to tell in advance.

Five different algorithms are compared in this chapter, with the classic LSV used as a datum, and the other ones reported in terms of variation respect to LSV. The first three are the same ones that have been used in the workspace analysis, and the last two have been presented within this chapter.

1. Moore Penrose pseudoinverse (LSV).
2. Local minimization of kinetic energy (LMKE).
3. Local minimization of the reactions (LSE).
4. Predictive minimisation of kinetic energy (PMKE).
5. Predictive minimization of kinematic indexes correlated with kinetic energy (PMCI).

It has been highlighted that an analysis based on the results on a big sample of trajectories is necessary, since the behaviour of local algorithms might vary a lot depending on the trajectory, and thus good (or bad) results on a few simulations do not allow to make any assumption about the overall quality of a specific solution method.

In order to provide data for a complete analysis, an initial end-effector position has been chosen very close to the boundary of the workspace, and simulations have been performed in every direction, and starting from different joint configurations (all of them with the same end-effector position), in a similar fashion to what has been done to analyse workspaces. Indeed, starting joint configurations that will be used here are the same, as per Table 6-1.

Table 6-1 Initial configurations for local algorithm analysis

Configuration #	Joint 1 position	Joint 2 position	Joint 3 position
1	0.4009	-0.6233	0
2	0.2004	0.0374	0.6474
3	0	0.3269	-0.7541
4	-0.2014	0.6015	-0.6484
5	-0.4028	0.6233	0

Differently from the method used for the workspaces, however, simulations performed here feature a more realistic end-effector trajectory, as per formula (3-34), as it was done for the global planner. This allows to have continuous and smooth end-effector velocity and acceleration along the whole trajectory, comprising initial and final point. For each of the initial configuration shown in table, the end-effector motion has been simulated in three different directions. One of them has been set to pass through the origin, featuring thus an angle of *π* (opposite direction as the x axis), while the other two are separated from the x axis by angles of *1/8π* and -1*/8π*, featuring thus directions of *7/8π* and *9/8π* respect to initial end effector configuration. Trajectories have been simulated of three different lengths: 0.10m, 0.40m, and 0.70m. This choice is meant to highlight the behaviour of the case study algorithms different lengths of trajectories. Circular trajectories with radii of 0.15m and 0.25m have also been simulated for each starting configuration.

For all these simulations, kinetic energy has been computed and compared, without including singular trajectories. All the results have been compared considering the average performance of the algorithms on specific subsets (e.g. circular trajectories or rectilinear trajectories of a specific length).

6.3.2 Parameters description

The full list of parameters to be involved in the algorithm are inverse kinematics time step size *Δt*, predictive optimisation time step size *ΔT*, horizon *h*, update interval *I*. This section explains their meaning and the value that has been chosen for the simulations performed for the purpose of this work. A sensitivity analysis will be presented later in the chapter. It is to be noticed that some of these parameters are necessarily different between the two versions of the predictive algorithm.

Inverse kinematics time step size *Δt*: this parameter has been set to 0.001s. This is a standard value for inverse kinematics problems, widely used in research and industrial environments.

Prediction time step size *ΔT* and *horizon h*: these altogether determine the prediction window T_p, which in turn determines the ability of the algorithm to be able to take into account future issues that could otherwise undermine the quality of the solution. The two parameters must be balanced considering that, for a fixed T_p, having a bigger *h* and a smaller *ΔT* allows for better resolution of the prediction, making it less likely for the algorithm to pursue local improvements over global ones. However, it also implies longer computational times. The horizon h doesn't really influence the prediction for PMCI, since it is based on variables that only involve joint displacements. Thus, optimising them means finding the lowest value at the end of the prediction time T_p, without considering intermediary steps. This means that for PMCI, h can be set to 1, and $\Delta T = T_p$. Things get more complex with PMKE, which involves joint velocities between the steps, thus requiring to consider a horizon h > 1, in order to limit the chance of unexpectedly high velocities within the prediction window. This makes things slightly more complex, however for the robotic manipulator hereby presented satisfactory results are reached for an *h* value of 2. Thus, it has been limited to this value to avoid computational complexity. This means that, for PMKE, $\Delta T = T_p/2$. This leaves the question open about how to choose T_p. Increasing it doesn't change the computational complexity, which is only influenced by *h*, however a prediction too far in the future might not have the best influence on current path point, since only 1 or 2 path points are actually included in its computation, meaning that those in between are not considered. If their number grows too large, the prediction isn't realistic anymore. Furthermore, it is hard to establish a proper T_p because a value that is good for certain trajectories doesn't necessarily fit others: some of them might benefit of a longer prediction window, while some others might prefer a short one, which makes it difficult to develop a tuning strategy. For the simulations presented this thesis, the value of the prediction window has been set to 20% of the simulation time. This means that, for trajectories with simulation time of 1s, the algorithm will include an expectation of the value of the future control cost for up to 0.2s for PMKE and 0.1s for PMCI, corresponding to respectively 200 and 100 inverse kinematics time steps. This means that ΔT equals to 0.1s for both PMKE and PMCI. This value rests on the assumption that no energy wasting phenomenon will take place in the first 0.1s of the trajectory, which sounds reasonable considering that the end-effector velocity is very low in the first part of the trajectory, and that only a very small distance is covered during it. For the rest of the trajectory, the optimisation algorithm works as planned, computing a rough prediction of the control cost integral, and considering it as a secondary input of inverse kinematics through null-space projection. Further discussion about the choice of these parameters can be found later, in the section Parameters Drivers.

Update interval *I*: the update interval is the variable that decides how often the prediction of the control cost is updated. It should be ideally set as big as the time step *Δt*, but this would slow down the algorithm much beyond what's acceptable for real time computation. Real time computation on MATLAB and on the test machine used takes 0.0449s on average each time the prediction is updated for PMKE, and 0.0155 for PMCI. An acceptable time is thus 0.050s. Improved versions of the algorithm or superior test machines might allow to reduce these values even further.

Their values have been set as per Table 6-2.

Table 6-2 Parameters used for the local algorithm

Parameter	*Δt [s]*	*ΔT [s]*	*H*	*I [s]*
PMKE	0.001	0.1	2	0.05
PMCI	0.001	0.1	1	0.05

6.3.3 Simulation results

Since the number of simulations is very high, their results are compared through tables featuring average values, rather than by figures for every trajectory. Tables are presented in ascending order respect to trajectory length. Trajectories are divided in rectilinear and circular, and a different table is presented for the two sets. Each line of the table represents a set of trajectories with the same length, and it shows the mean of kinetic energy integral, alongside with its difference in term of percentage with the LSV case, which is used as reference. The percentage of singular trajectories for each algorithm is also highlighted. Trajectories have been considered singular if power exceeded 100W at any point of the trajectory, or if the inverse kinematics numerical solver failed straightaway. Occasionally, especially for the LSE algorithm, all trajectories of a certain length are singular. In this case statistical variables are omitted.

Table 6-3 Local algorithm results for rectilinear trajectories

Length	LSV	LSE	LMKE	PMKE	PMCI
0.10m	0.0797	0.0069 (-13.97%)	0.0061 (-23.68%)	0.0058 (-26.99%)	0.0098 (+23.18%)
% singular trajectories	0%	0%	0%	0%	0%
0.40m	0.0950	0.1061 (+11.70%)	0.088913 (-6.42%)	0.0688 (-27.59%)	0.1198 (+16.59%)
% singular trajectories	13.33%	40%	0%	0%	6.67%
0.70m	0.3680	-	0.3327 (-9,59%)	0.2933 (-20.31%)	0.3610 (-1.89%)
	20%	100%	26.67%	0%	33.33%

The results for rectilinear trajectories, shown in Table 6-3, vary depending on their length: for the shortest ones, with a length of 0.10m, LSV performance is surpassed by all algorithms except for PMCI both from the point of view of kinetic energy integral minimization. LMKE features the best performance among algorithms solely based on pseudoinverse techniques, with -23.68% for kinetic energy integral, while LSE sits between LSV and LSE, with an improvement of -13.97%. PMKE stands out as the best algorithm for kinetic energy integral minimization, with -26.99%, showing a slight superiority even to LMKE. PMCI shows difficulties in providing an optimal solution, with +23.18% for kinetic energy. This can be explained by the fact that the algorithm optimises a cost function solely based on joint configuration, as opposed to joint configuration and velocities. This means that it will have a tendency to reach the best configuration to minimize its cost function, no matter how much joints velocity will have to be increased to reach it. The results on the shortest set of trajectory show that, although a local minimisation of kinetic energy such as the one pursued by LMKE algorithm is effective in minimizing kinetic energy over the whole length of the trajectory, the PMKE algorithm can still produce improvements over it by taking into account the future energetic cost. The PMCI algorithm, on the other hand, suffers from not including velocities in its control cost, and shows that direct optimization of configuration-related values is too simple to provide results on a more than local scale.

In the set of trajectories with a length of 0.40m, LSV algorithm shows one further flaw in that 13.33% of the sample is singular. Furthermore, the results show that LSE algorithm has lost most of its energetic effectiveness, exceeding LSV kinetic energy integral values by 11.70%. This happens despite the fact that 40% of the LSE trajectories are singular, thus reducing the sample to the 60% most energy-effective trajectories. This is also the last set of LSE trajectories, since those in the set with a length of 0.70m are all singular. It can be concluded that reaction minimization, while having its merits for the control of space manipulators, does not provide good performances from the energetic point of view. LMKE, at this point, shows a reduced effectivity too, although it still has an edge over LSV, with an improvement of 6.42%. Although these numbers seem quite small, it should be observed that LMKE features no singular trajectory while LSV, as already mentioned, has a singularity percentage of 13.33% in this sample. PMKE is the clear winner for this set, featuring no singularities, and still surpassing LSV by a margin of 27.59%. Once more, PMCI does not feature satisfying results, with a variation of the kinetic energy integral average value of +16.59%. Furthermore, it does feature a 6.67% of singular trajectories, confirming that a cost function based on information about manipulator configuration only is not suitable for the problem under examination.

The last rectilinear set, featuring a length of 0.70m, is the most problematic one for local inverse kinematics based methods: the increased length means that probability to incur into a kinematic singularity is higher, and in fact 20% of the trajectories solved with LSV and

26.67% of the trajectories solved with LMKE are singular, while LSE method didn't output any feasible solution for this set, producing a 100% failure rate. LMKE is still more successful than LSV at minimising kinetic energy integral, although it should be kept in mind that it also resulted in a higher number of singular trajectories: it features a kinetic energy integral average reduction of 9.59%. Although an improvement in kinetic energy integral is visible, the larger amount of singular trajectories means that the different is possibly negligible. PMKE is now a clear winner, with an average reduction of the kinetic energy integral as high as 20.31%. Furthermore, the algorithm produces no singularities. In longer trajectories, the prediction shows its full capability, allowing to complete them with strong energy savings compared to traditional algorithms, and decreased risk of singularity. PMCI doesn't really compare in this case, with a decrease of -1.89% in kinetic energy integral cost compared to LSV. Furthermore, it also features 33.33% of singular trajectories, the highest value among all algorithms except for LSE. The results from rectilinear trajectories show that correlated indexes don't compare with actual predictive minimisation of kinetic energy integral, while PMKE outperforms existing methods on all sets of trajectories.

Table 6-4 Local algorithm results for circular trajectories

Diameter	**LSV**	**LSE**	**LMKE**	**PMKE**	**PMCI**
0,10m	0.0664	0.0586 (-11.80%)	0.0437 (-34.08%)	0.0413 (-37,72%)	0.0667 (+0.45%)
% singular trajectories	0%	60%	0%	0%	0%
0,25m	0.3991	-	0.3092 (-22.53%)	0.3197 (-19.90%)	0.4120 (+3.23%)
% singular trajectories	20%	100%	40%	0%	20%
0,40m	-	-	-	0.4422 -	-
% singular trajectories	100%	100%	100%	80%	100%

As shown in Table 6-4, some differences in behaviour can be evidenced on circular trajectories. It can be observed that these trajectories more issues with singularities, possibly due to their increased length compared to rectilinear ones, which increases the chances of a singularity along the way (longest circular trajectory is 1.2566m, 79.5% more than longest rectilinear trajectory at 0.70m). In fact, the shortest sample is the only one with complete sets of trajectories, except for LSE which has a 60% singularity rate. Those featuring smaller diameters taking advantage from both LMKE and PMKE, as they feature a size sufficiently small to never necessitate any major motion of the first joint (the one that would cause the highest kinetic energy increase). In a similar fashion to the rectilinear case, LSV is outperformed by every other algorithm from the energetic point of view, with PMKE being again a clear winner in this case (-37.72%). LMKE is however a close second, with -34.08%, while LSE follows at some distance (-11.80%). It should however be observed that LSE doesn’t really compete due to high number of singularities it features (60%). PMCI does not compete either, in that it is not able to anticipate increases in kinetic energy, resulting in almost the same performance of LSV, apart from a small increase (+0.45%).

Looking at the subset with 0.25m as diameter, LSE is unable to provide any solution, while LSV and LMKE also feature singularities, respectively on 20% and 40% of trajectories. PMCI as well features 20% of singular trajectories like LSV, continuing the trend of mimicking its performance on circular trajectories. The overall results in this case are similar to the previous set, with LMKE featuring the best results (-22.53%), and PMKE being a close second with a kinetic energy integral reduction of -19.97% compared to LSV. However, PMKE performance is not affected by singularities, differently from the other algorithm, which makes its performance more reliable (in fact, considering the singular trajectory, even LMKE would be much worse). PMCI is once again close to LSV in behaviour, featuring increased kinetic energy integral by +3.23%, following the datum very closely.

The final set of trajectories that have been simulated is composed by circular trajectories with a diameter of 0.40m. All solutions obtained with algorithms without prediction terms (LSV, LSE, LMKE) are singular, thus comparison between LSV algorithm and other solutions is no longer possible. PMKE features 80% of singular trajectories (4 out of 5) and PMCI features 100% singular trajectories too. This result suggests too that PMKE is more reliable than traditional inverse kinematics algorithm when it comes to singularity free solutions for motion planning problems, although it is less reliable than with circles of smaller diameters.

Table 6-5 Local algorithm relative changes compared to LS, for all trajectories

Type of trajectory	LSV	LSE	LMKE	PMKE	PMCI
Rectilinear	-	+9.71%	-9.19%	-21.89%	+2.26%
% singular trajectories	11.11%	60%	8.89%	0%	13.33%
Circular	-	-11.80%	-26.37%	-22.78%	+2.84%
% singular trajectories	40%	86.6%	46.67%	26.67%	40.00%

Table 6-5 shows results based on the complete set of 90 inverse kinematics problems and 450 simulations: the percentages shown are the average changes in value between each of the subsets that compose the rows of previous result tables. The two versions of the newly introduced algorithm provided very different results: while PMKE showed to be able to both reduce energy and avoid singularities, PMCI was by far less trustworthy and its performance has been very close to a plain LSV. For rectilinear trajectories, PMKE averaged -21.89% reduction in kinetic energy integral, much better than second best, LMKE, with -9.19%

reduction of the kinetic energy integral, with 8.89% of solutions being singular. Results with circular trajectories are more mixed, with LMKE showing a better performance in kinetic energy integral reduction compared to PMKE (the two are respectively sitting at -26.37% and -22.78%), but with more singular solutions (46.67% vs 26.67%, mostly concentrated on bigger circles). Overall, PMKE features a superior ability to reduce energy on rectilinear trajectories, while also avoiding singularities. Results with PMCI show instead that kinematic indexes aren't a good indicator for kinetic energy consumption at all, or at least that more sophistication is needed to turn them into a viable tool for online optimisation of kinetic energy. The numbers for this algorithm are an increase of +2.26% for kinetic energy integral for rectilinear trajectories, and +2.84% for circular trajectories, resulting in values overall very similar to the LSV, with a comparable number of singular solutions. These results show the inefficiency of local approximators in properly estimating a dynamic variable such as kinetic energy. It has been already been commented for the shortest set of trajectory that, since PMCI optimizes a quantity that is solely configuration-dependent, it actively searches for a minimum which might be considerably far from the current manipulator configuration, resulting in an excessive kinetic energy expense to reach it. More generally, it can be said that it does not provide a solution that is much distant from LSV, and in doing so it causes an overheard in kinetic energy integral value.

6.4 Observations on the setting of parameters

Previous section showed that PMKE algorithm is a viable alternative to traditional inverse kinematics algorithm, due to its ability to reduce kinetic energy integral of a robotic manipulator and reduce risks related to singularities. This leaves however the question open about its parameters and what's the best way to set them. For the results hereby presented, a combination of the parameters that could give reasonably good results without excessively increasing computational times has been chosen, however this is not the only possible choice.

Generally speaking, tuning the parameters of the algorithm without excessively reducing the computational speed requires a trade-off: it can either be chosen to have frequent updates of the predicted kinetic energy value, requiring to perform the optimisation more often, or improving the quality of the prediction itself. The choice of having more frequent updates lead to reducing the updated interval *I*, while the choice of improving the quality of the prediction is more subtle, as the prediction is driven by three different parameters, the prediction step time *ΔT* and the horizon *h*. If *ΔT* is increased while keeping *h* fixed, the prediction will span more in future, but an excessive increase in *ΔT* may cause the prediction to become imprecise, as it won't be able to capture phenomena that require a faster time step than *ΔT* to be observed. Furthermore, with a bigger *ΔT*, effective use of the spline interpolation could become a problem, as the longer *ΔT*, the more it would get far from proper tracking, resulting in excessive velocities for the secondary task of the pseudoinverse solution. On the other hand, increasing *h* also means that the prediction will span for more time in future, without interpolation risks related to an excessive increase of the time step. However, the computational time will become larger, as increasing *h* also increases the number of parameters involved in the simulation.

In order to explore these possibilities, solutions obtained with PMKE and different update intervals I are compared with a global solution. This allows to check the difference with the best possible solution. The result with an update step of 0.025s is presented in Figure 6-1 (energy figures) and Figure 6-2 (kinematic figures), while results for globally optimal solution (with time step reduced to 0.01s) are presented in Figure 6-3 for energy figures, and Figure 6-4 for kinematic figures.

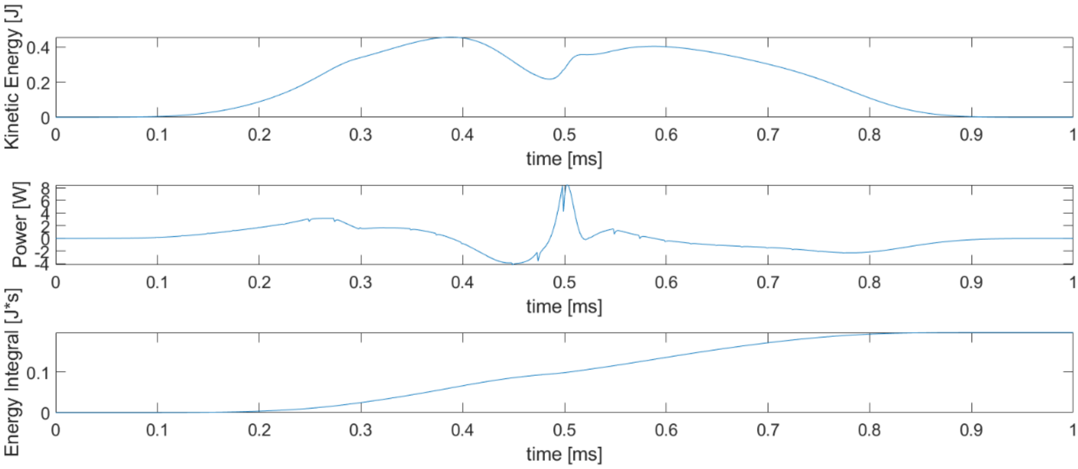


Figure 6-1 Energy figures for PMKE solution

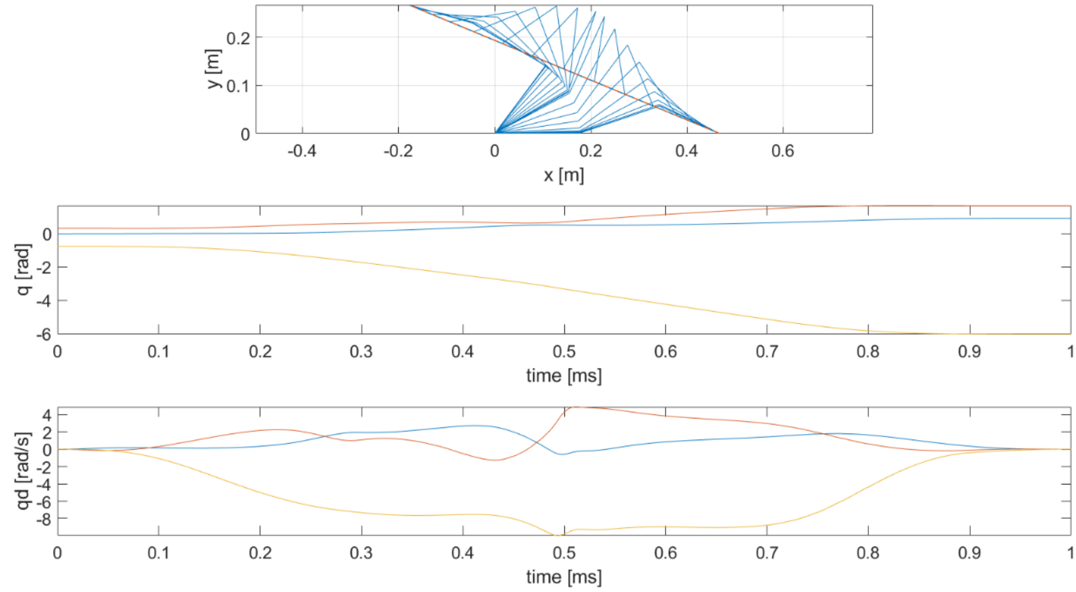


Figure 6-2 Joint figures for PMKE solution

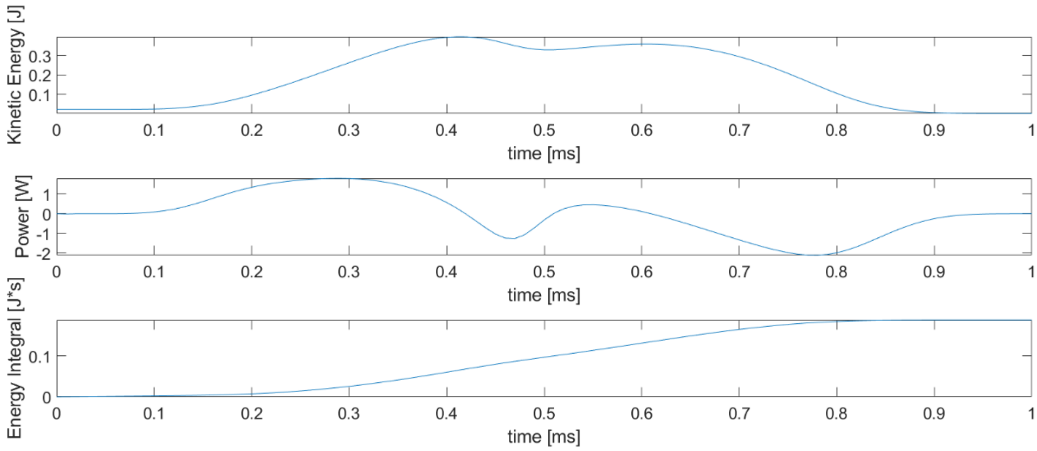


Figure 6-3 Energy figures for globally optimal solution

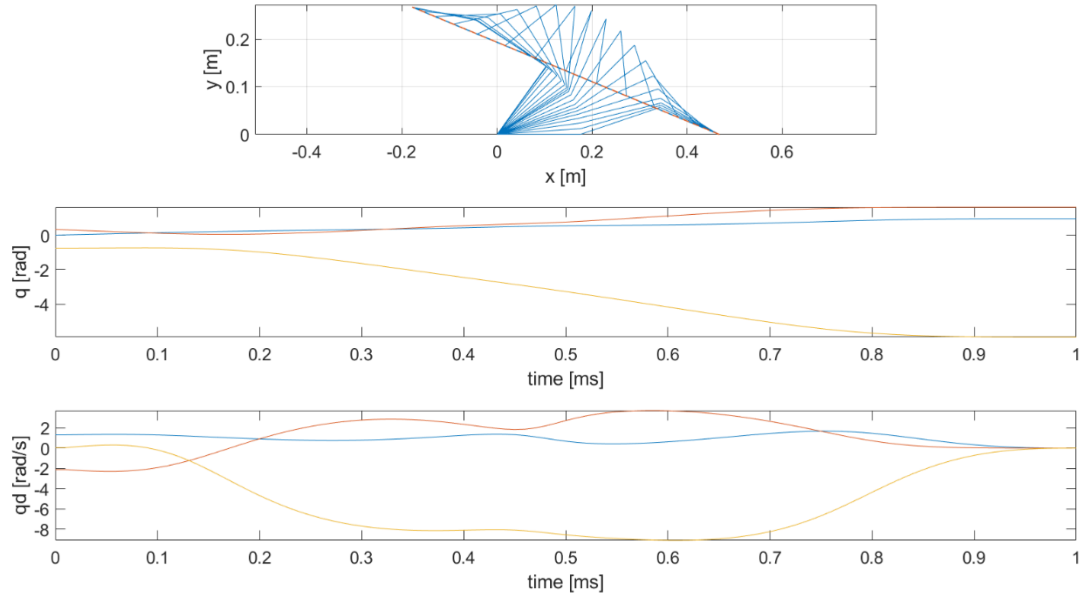


Figure 6-4 Kinematic figures for globally optimal solution

As mentioned by Nedungadi et al. [15], when boundary conditions are imposed on the initial joint configuration, initial velocities must be left free if final conditions are specified as well (in this case, joints velocity at final time must be equal to zero) , which explains why the global solution, for this case, has non-zero initial velocities. The two solutions output particularly similar kinetic energy integral values in this case (although this can't of course be a general rule, due to the prediction window limited size), being them 0.1852 J for the global solution and 0.1974 J from PMKE, with an increase with PMKE of only +6.59%. It can overall be observed that MKPE performs a similar task as a global algorithm, but on a reduced window: it introduces an extra-velocity at the current time step, that will allow for kinetic energy integral reduction over the whole trajectory. The same trajectory is computed again with different update intervals *I* (an with the filter in place again), and their result is transcribed in Table 6-6, alongside with percentage difference from the global algorithm: it can be observed that the changes are very small: from an update interval size of 0.025s to 0.050s, the loss in terms of cost function value is only 0.43%, and when the update interval goes from 0.025s to 0.075s, the percentage loss reaches 2.64%. This suggests that, for applications for which it is possible to introduce a higher delay, the best strategy is to refine the prediction rather than reduce the update interval, which is not the key parameter for the performance of the algorithm.

Table 6-6 Influence of update interval I

Update Interval [s]	*Global*	*0.025*	*0.050*	*0.075*
Kinetic energy Integral [J]	0.1852	0.1974 (+6.59%)	0.1982 (+7.02%)	0.2023 (+9.23%)

Table 6-7 Influence of horizon h

Horizon size	*2*	*3*	*4*	*5*
Kinetic energy Integral [Js]	0.2108	0.1994 (-5.41%)	0.1953 (-7.35%)	0.1928 (-8,54%)
Trajectory computational time [s]	2.579	4.659	7.935	13.029

Looking at different values of the horizon, increasing it seems instead a profitable strategy: Table 6-7 shows kinetic energy integral values for the same trajectory, with different value of the horizon h, alongside with computational times of the optimisation on a laptop featuring an Intel i7 9th Generation processor with 32GB RAM. The prediction time step size

ΔT has been kept constant to 0.05s, thus resulting in extended prediction window T_p. It is easy to notice that higher values of the horizon allow to compute trajectories with lower energetic cost, while however also requiring a increase in computational time (for comparison, the trajectories used for the validation required computational times in the range 1.7-1.9s on the same machine). One further thing to notice is that the biggest decrease can be observed between a horizon value of 2 and a horizon value of 3 (-5.41%). Further increases of the horizon value bear minor improvements: going from 3 to 4 produces a further decrease of 1.92%, while going from 4 to 5 achieves just a 1.19% reduction of kinetic energy integral. This suggests that the optimal size of the prediction can strike a balance between computational time and performance, since at some point any improvement in the horizon will produce high increases in computational time in exchange for modest performance gains. The exact balance might possibly depends on the scale of the manipulator: that is, phenomena that increase the kinetic energy of the manipulator do not involve the whole trajectory at once, but rather a fraction of it. The maximum size of such fraction, measured as a length, is the optimal size of the prediction window of the PMKE algorithm. This length needs to be translated into a time to be input into the algorithm as parameters T_p, ΔT and h. In order to do so, it has been considered that the most challenging trajectories that have been simulated in the thesis are indeed the longer ones of 0.70m and above. During the algorithm tuning, it has been seen that the prediction on them will be less effective if the part of the trajectory used for the prediction is shorter than 0.15m. This is consistent with the workspace analysis presented in previous chapter, which shows energy dissipating phenomena (singularities) influence their neighbourhood up to distances of the order of 1E-1m. Having a fixed h for computational time reasons, this makes the prediction time step size dependent on the total time to perform the motion T with a formulation such as:

$$\Delta T = \left\lfloor \frac{T}{10} \right\rfloor \tag{6-4}$$

Where the notation $\lfloor x \rfloor$ denotes the floor of x. This leaves the question open about how to individuate such window for more complex manipulators. Hypotheses in this sense could be Monte Carlo methods, or reinforcement learning. Possible future developments about this topic and all the others touched in this thesis will be further explained in next and last chapter, which constitutes the conclusions of this work.

6.5 Conclusions

This chapter presented a comparison between three inverse kinematics available in literature, and an innovative algorithm based on the computation of an approximate prediction of kinetic energy along the desired end effector trajectory. This has been presented in two different versions. One, PMKE, is based on the approximate prediction of kinetic energy, while the other, PMCI, is based on the approximate prediction of a linear combination of kinematic indexes correlated with kinetic energy. While PMKE proved more successful than traditional algorithms in minimising kinetic energy, PMCI has served the purpose of showing that locally defined variables are not a reliable predictor of kinetic energy of a robotic manipulator. Further studies are required to extend the PMKE to more complex manipulators, and to develop reliable techniques to compute the optimal prediction window size. The concept of the PMCI should instead be substituted by more efficient ways to incorporate the local information obtained from kinematic indexes into energy-saving optimization methods, perhaps as a tool to select suitable initial conditions.

Chapter 7. Conclusions and further developments

7.1 Introduction

The inverse kinematics problem of redundant manipulator is a problem that has always captured the interest of robotics researchers, which in turn have provided a wealth of solutions for a vast array of cost functions and constraints. Despite being a well established and well understood field of research, with many practical robotics controllers featuring well-working inverse kinematics algorithms, many questions still remain open about how to provide optimal solutions that reduce operational costs and alleviate power requirements, while incorporating all the different operational constraints that arise from practical manipulators. This chapter focuses on how the research questions were addressed, how the results in this thesis contribute to the field of knowledge of inverse kinematics of robotic manipulators, what limitations exist in proposed solutions, and how these can be addressed in the future. The chapter is organised as follows: first part reviews the research questions and the way they have been answered, second part focuses on the contributions to knowledge, and the third and final part reviews directions of future development of this work.

7.2 Key research findings

This thesis has presented an investigation on the inverse kinematics problem, with the two distinct aim to provide novel solutions to the problem both globally and locally. At this point, it is possible to go back to the research questions asked in chapter 1, and review how they have been answered by this document.

1. **What considerations are necessary to solve the inverse kinematics problem for redundant manipulators end-effector trajectory tracking through mathematical optimisation of energy-related integral cost functions?**

This is the first question that has been answered. Particularly, it has been addressed with the literature review and Chapter 3, dedicated to the mathematical background of this thesis. Current state-of-the-art has analysed and a number of key findings have been observed. When necessary, these key findings have been further highlighted in the relevant chapters discussing the results (4, 5 or 6).

Particularly, it has been observed that many proposed solutions of the IK problem of redundant manipulators aim at minimising the energy used by the manipulator in some form, be it the squared norm of joints velocity, or more sophisticated methods involving kinetic energy or torques. Furthermore, it has been found that the IK problem is in truth two distinct problems, although related in nature, the local and the global problem. Starting from this, literature review has investigated the nature of both problems, evidencing how the most relevant characteristic of the problem is its nonlinearity and non-existence of solutions in closed form, which leads to the necessity of using the so-called Jacobian matrix. A further complication for free-floating manipulators is that angular momentum must be preserved during the motion, resulting in a non-holonomic constraints that causes the base spacecraft attitude to vary during the manipulator motion, unless precise strategies are adopted to avoid it.

The Jacobian matrix has been identified to be an important tool for both local and global methods, yet it is a source of limitations due to it being prone to singularities, points where a desired velocity in the cartesian space translates into extremely high or infinitive velocity in the joints space. From the nature of the Jacobian, numerical problems arise both for local and global methods that have been used to tackle the IK problem. Such methods, for the local problem, involve the use of the Jacobian pseudoinverse as-is, while they incorporate it in frameworks based on calculus of variation for the global problem. They have been found to be affected in the first case by locality of the solution, which may or may not translate to a satisfying global result, and in the second case by several numerical complications related to the inversion of the Jacobian matrix, and to the necessity to solve a TPBVP.

Based on these findings, it has been observed that main challenges, apart from the nonlinearity of the problem, lie for local algorithms in the complexity of providing some prediction to the nonlocal outcome of the optimisation without adding an excessive burden of complexity to the algorithm involved. This drives the research for local algorithms to the direction of providing a suboptimal, approximated prediction of a global cost (such as the kinetic energy integral) as quickly as possible, so that the manipulator can be driven to the less consuming direction. For the global problem, the main points to take care of can instead be summarised in two points. The first one is the lack of attention in algorithms available in literature about the problem of finding the actual global optimum of the global problem, as opposed to find an optimal solution that isn't necessarily the best solution for the problem. The second one is the difficulty in handling complex constraints, especially nonlinear ones, like actuators' torque or power limits. These limits occur in real manipulators, since of course their actuators are limited in output torque and power they can produce, but are rarely or never addressed in research works. One more important point regarding global methods is the inherent difficulty in establishing initial guesses as input to the algorithm. For many methods, such as those based on the TPBVP, the quality of the solution largely depends on the quality of the initial guess.

Apart from this general considerations on the algorithms available in literature, chapter 3 of this thesis has been dedicated to the necessary mathematical tools necessary to tackle the problems presented in the thesis. The chapter includes a detailed analysis of the kinematics and dynamics of redundant manipulators, an overview of the methods used to solve redundant manipulator inverse kinematics, a description of most important nonlinear optimisation heuristics, and explanation of the manipulator models used and the trajectories analysed in the reminder of the thesis.

A further dive into general considerations on the inverse kinematics problem for redundant manipulators has been presented in chapter 5, where an analysis of a redundant manipulator's workspace has been presented. The content of the chapter further evidences the nonlinearity and complexity of the problem, through considerations on the so-called straight path workspace of a manipulator. This concept allows to see how the reachability and energetic cost of a manipulator's trajectory can vary depending on the algorithm used to solve the IK problem, and on the manipulator initial configuration.

2. **How can the limitations of conventional solutions of the problem mentioned in question 1 be tackled by applying optimisation techniques that have not been used this way before? And how can such techniques be applied to free-floating manipulators?**

This question has been answered in chapters from 4 to 6, which illustrate the solutions and techniques that have been developed for this thesis. In chapter 4, a novel solution to the global inverse kinematic problem has been presented, the Interpolation-based Global Kinematic Planner (IBGKP). This method of solution, differently from traditional ones, does not rely on Pontryagin maximum principle or on calculus of variations, and is focused instead of obtaining a good initial guess to solve the IK problem with a powerful constrained gradient descent method such as Sequential Quadratic Programming (SQP). To this goal, a heuristic based on multi-start method is presented, aimed at limiting its exploration to possible solutions that are compliant with EE trajectory constraints, as opposed to searching the whole joints space for solutions, which might result in initial guesses which result in completely different EE trajectories than what is required, and won't converge, or converge in extremely long times. Multi-start methods, to the author's knowledge, had never been used to solve the IK problem, and thanks to the vast array of existing gradient descent methods, they allow to avoid issues of conventional IK algorithms related to TPBVP and Jacobian-related numerical problems, while they also allow to include linear and nonlinear constraints in the process.

Chapter 5 provides a workspace analysis which investigates the limitations of existing algorithms more than proposing solutions based on novel optimisation techniques. However, within this chapter, considerations are presented about the possibility to use different IK solvers for free-floating redundant manipulators. The requirement of free floating manipulators IK algorithms to reduce the manipulator's reactions transmitted to the spacecraft is relaxed by including an ACS actuator (a reaction wheel) in the control algorithm, thus allowing the use of a wider variety of algorithms for free-floating manipulators.

Chapter 6 answers the question for local algorithms. It presents PMKE (Predictive Minimisation of Kinetic Energy), a novel local IK method based on a prediction of the joints motion that will cause the kinetic energy integral to grow less during the motion. While the prediction is based on a gradient descent optimisation, its resulting joints velocity vector is used as a secondary task within a Jacobian pseudoinverse framework, allowing it to be integrated with existing algorithm, PMKE, although still local in a wider sense, is no more limited to just one path point, and it is thus able to avoid some of the pitfalls caused by the locality of solutions. Its ability to move in the direction with the lowest kinetic energy integral allows for better singularity avoidance and for reduction of the integral up to 30% or more.

3. **Are there different limitations that come into play by applying different methods to the problem? How can they be tackled?**

Different kinds of limitation come into play depending on the problem under consideration. The main limit for the IBGKP presented in chapter 4 is certainly the so called *curse of dimensionality*, which means that the computational power and memory required to solve it can be exponential on the number of parameters of the optimisation, which in turn would cause not to have resources enough to converge. The curse of dimensionality has been countered by the fact that the IBGKP has been designed to be iterative. The joints trajectories are first optimised on a reduced subset of the path points, and the other path points are sequentially added by interpolating them with splines, and performing further steps of optimisation, till the complete global IK problem under examination is solved.

The local algorithm PMKE, presented in chapter 6, features a different kind of limitation in that a very precise prediction optimised at every time step would cause the algorithm to be unable to perform its task in real time. In order to overcome the issue, two methods have been adopted: the first one the prediction has been solely based on a coarse set of future path points, rather than all of them, resulting in a low number of parameters to optimise. The second one is that the prediction is not updated at every time step, but at rather large intervals of 25 or 50 timesteps, allowing for less optimisations of the prediction to be performed within the same trajectory. In order to avoid the discontinuities that this could cause, the prediction results are fitted with the local derivatives of the joints trajectory through interpolation with 4-th order splines.

7.3 Contributions to knowledge

This document presents a number of developments on the topic of inverse kinematics of redundant robotic manipulators. These resulted in a number of contributions to knowledge that are summarised in this section.

Limitations of local and global inverse kinematics methods A survey of literature has identified the shortcomings of the state-of-the-art of inverse kinematics methods. It has been subdivided according to the different categories of methods used to solve the problem, looking at both local and global methods, and both at fixed-base and free-floating manipulators. Furthermore, research regarding workspace characterisation of redundant manipulators has been reviewed. An analysis and discussion of the key findings for each subtopic has identified existing knowledge gaps and limitations in the field.

Interpolation-based Global Kinematic Planner A global algorithm has been proposed to solve the inverse kinematics problem of redundant manipulators. This algorithm has been developed aiming at overcoming the most relevant limitations of the existing global inverse kinematics methods. Most notably, the algorithm is not based on calculus of variation and optimises a function which does not include the Jacobian matrix pseudoinverse, avoiding the numerical problems and the limitations of both. It can find multiple optima of different cost functions, limiting the influence of the initial guess on its result, and it can incorporate linear and nonlinear constraints. This latter property allows it to solve bi-objective optimisation problems by the ε-constraint method, which allows to individuate the Pareto front even when it is non-convex, differently from the weighting method normally used in robotics.

Workspace analysis A workspace analysis has been performed aiming at observing the energetic behaviour of different algorithms over the whole size of the workspace of a robotic manipulator. It has been observed that no existing kinematic index has so far been successfully used to characterise the expected energetic performance of a robotic manipulator in a specific configuration. In order to overcome this issue, the analysis has relied on canonical correlation to individuate which linear combinations of kinematic indexes bear the closest relationship with kinetic energy, showing that a linear combination of dynamic manipulability and condition number feature the highest correlation with it. Alongside this quantitative analysis, some quantitative considerations have been made, regarding preferential directions of manipulators motion, and differences between the algorithms used.

Free-floating manipulator control with ACS It has been observed, in relationship to workspace size, that authors in literature pointed out how the concurrent use of a free-floating robotic arm with the ACS of the base spacecraft can enhance the size of the workspace. This has been practically being performed through the use of a generalised Jacobian incorporating a reaction wheel as part of the actuators controlled concurrently with the manipulators joints. The resulting algorithm has been shown capable to retain manipulability, exploiting the fact that the reaction wheel is able to absorb angular moment in excess. Thus, a practical method to increase the size of the workspace of a free-floating manipulator and to add it further degrees of redundancy exploiting the ACS has been presented.

Predictive minimization of Kinetic energy The last contribution to knowledge of this thesis is a local algorithm based on the predictive minimisation of kinetic energy. This algorithm exploits an estimation of the direction of minimum kinetic energy expense by optimising a local approximation of the future kinetic energy integral. The prediction does not need to be updated at every time step and can be computed with a restricted number of parameters, resulting in a limited computational cost. As a result, the proposed method has reduced energetic cost and enhanced singularity avoidance capability.

7.4 Limitations

This thesis covers a number of developments in the field of inverse kinematics of redundant robotic manipulators. Although these developments solve some of the issues presented in the literature review of this thesis, inverse kinematics is a complex field, and even more so when redundant manipulators are involved, and many limitations are still left to be overcome.

Chapter 4 introduced a novel global inverse kinematics algorithm, the Interpolation-based Inverse Kinematic Planner. Such algorithm is able to solve several different optimization problems, comprising bi-objective optimization problems, with a variety of linear and nonlinear constraints. It has been demonstrated with a three DOF planar robotic manipulator, resulting in an optimisation problem featuring 303 parameters. This has been helpful in providing a first proof of the algorithm capability, and has provided a number of solutions that are easy to visualize compared to three-dimensional ones. At current stage, the algorithm has however not been tested on more complex robotic arms, which would be challenging from the point of view of computational time especially. The algorithm is iterative, which helped in reducing computational times on the cases under examination, it must however be investigated on more complex problems. Another possible issue coming from increased complexity might be that the actual global optimum requires an extremely high number of candidate solutions to be individuated, or that the global optimum of the partial problem solved at the first steps of the interpolation does not correspond to the global optimum of the complete problem. Both these issues can be partially offset with a higher number of processor cores, aiming at optimising more candidate solutions at the same time. But, although computational power is often helpful for optimisation problems, and particularly for those based on Random Search methods, specific behaviour of the algorithm on more complex problems must be investigated first, before drawing any conclusion. A further limitation to be observed for the IBGKP is that the only friction model that has been used is the viscous friction model. The ability to find a global solution must be tested with Coulomb friction and other more realistic models as well.

Chapter 5 focuses on a workspace analysis. Such analysis has provided some understanding of the complexity of the problem of energetic minimization of redundant manipulators. Particularly, it has been shown that no specific kinematic index features a direct relationship with the kinetic energy. A closer relationship has been obtained by the use of canonical correlation, showing that the a linear combination of dynamic manipulability and condition number feature a correlation with kinetic energy in the range of 80%. This insight has however no direct application in finding energy-saving inverse kinematics solutions so far. It can possibly be used within an optimization heuristic to accelerate the individuation of a solution, but no practical method has been developed so far, especially because dynamic manipulability is prone to reach infinitive value close to singularity, just as much as the Jacobian matrix does. A further development of the chapter has been a practical inverse kinematic method to move a free-floating manipulator concurrently with the ACS of the base spacecraft, resulting in increased manipulability and extended workspace. Such motion could however be delicate to reproduce in practice, as any error in the motion of the wheel could result in attitude disruption. This should be simulated or investigated further with a realistic dynamic model.

Chapter 6 presents PMKE, a local inverse kinematics method for the minimization of kinetic energy integral along a predefined trajectory. The method, based on the optimisation of a prediction of the kinetic energy integral, has shown good capability in reducing the kinetic energy integral of a three DOF planar manipulator, it however still presents some limitations to be addressed. Most noticeably, the simplicity of the robotic arm has again been useful in proving the effectiveness of the algorithm on a simple simulation setup, it however also constitutes a limitation in that the new method must be tested on more complex manipulators. The main issue with a higher number of DOFs would probably be that the number of parameters involved in the kinetic energy prediction would increase, slowing it down. Ad-hoc solutions would need to be developed to speed up the computation. Another issue with PMKE are the tuning of its parameters, which depends on the specific manipulator, and requires trial-and-error and simulations to be performed appropriately.

7.5 Applicability to practical robotics

The contributions to knowledge provided by this thesis are mostly focused on the field of theoretical inverse kinematics, coherently with the original scope of providing novel solutions to the local and global inverse kinematics problems. In order to exploit such solutions for practical robotics, and especially in the field of free-floating manipulators, it is necessary to consider what the real operational constraints are for the field of space robotics. To this end, it must be first observed that, although each mission features its own peculiarities, a natural subdivision can be made between Low Earth Orbit (LEO) and Geostationary Orbit (GEO) missions. Former ones are characterised by altitudes below

1000km and orbital periods in the range of 60-100 minutes, while latter ones feature satellites that are meant to orbit over a specific point on Earth during their whole operational life, resulting in altitudes around 36000km and orbital periods of 24 hours. All the space manipulators used in practice (Space Shuttle and ISS ones) have been deployed in LEO, and have been used for satellite deployment and sometimes maintenance of expensive assets, such as ISS [156] and the Hubble Telescope [157], while most of the planned debris removal concepts in literature are aimed at being operated in LEO (e.g. it is worth observing that first in-orbit experiment of debris removal, ETS-VII, was performed in LEO [79], with an average altitude of 550 km). While mission concepts for space manipulators to be deployed in GEO have also been developed [158], they have been much more heavily focused on asset inspection, and maintenance, often aiming at the substitution of modular components, a concept inspired to the Orbital Replacement Units (ORU) used on the ISS. The reason for such a scope is that geostationary orbit is extremely expensive to reach, which in turn means that any extension of operational life of existing geostationary satellites save considerable costs by postponing the moment when they need to be substituted.

Such subdivision results in different operational requirements: systems deployed in LEO are designed to capture objects moving with high speed, and missions with the goal to capture non-collaborating space debris are expected in the near future. The nature of such targets implies uncertainties on their position and dynamic parameters. Indeed, a vast body of literature has been devoted to techniques to reduce such uncertainties (e.g. [159]–[162]). In this context, it is hard to imagine the possibility to use a global algorithm, since the end-effector trajectory shall follow the motion of the target, which might feature knowledge errors to be reduced in real-time based on sensors measurements, and thus it is most likely not fully known in advance. In such a scenario, motion planning must be effectuated quickly and updated based on new information. This suggests the use of a local planner, which is more able to react to the changes of the target motion, and is able to compute an IK solution once the target motion (i.e. desired end-effector trajectory) becomes known. In the case of the PMKE presented in this thesis, the motion can be updated at intervals corresponding to the parameter *l* of the algorithm, inferior to 0.1s in the examples presented, which makes it suitable for such operational conditions. There are of course constraints on the computational time that can be afforded for the prediction of the kinetic energy integral: ideally, it should be recomputed every time an update of the target motion (and thus end-effector motion) becomes available to the IK solver. Space manipulators however usually feature slower motion than their ground-based counterparts, and literature presents times in the range of tens or hundreds of seconds (e.g. [104]) to complete a trajectory. This allows for tracking with larger time-steps then the 0.001s used in the simulations presented in Chapter 6, while an on-board implementation would be in C rather than in Matlab, resulting in faster execution times.

The global algorithm presented in the thesis, IBGKP, is more inclined to direct practical use, because it can more readily include dynamic considerations and, more importantly, physical limits that occur with real manipulators. Its use in space is however limited to missions where the target and its motion are well understood in advance, as the full end-effector trajectory is a required input for the optimization. Such missions can be mainly individuated in maintenance missions, such as some of the ones carried out on the ISS (in the case of ORU replacement, the manipulation targets are usually placed on the ISS itself, making them very well known) and the ones planned in future missions to extend the life of geostationary satellites. In such case, the target is known and collaborative, and the end-effector task required to carry out the maintenance mission can thus be planned offline and optimized as required.

7.6 Concluding remarks and future perspectives

This thesis addressed some of the open challenges of the inverse kinematics problem of redundant manipulators through novel algorithms based on optimisation techniques. Most specifically, this research presented solutions to obtain energy-efficient solutions for inverse kinematics, considering both the global and the local inverse kinematics problem. The two problems presented different issues, which lie mostly in numerical complexity and difficulties in imposing realistic constraints for the global methods, and in hardships in obtaining a globally well-behaved trajectory when solving the IK problem locally for local algorithms.

According to this goals, the outputs of this work are algorithms and numerical methods that advance the state-of-the-art. Most notably, a global algorithm has been presented, and its capability demonstrated, to compute globally optimal solutions of different cost functions with nonlinear constraints. Local results of this work feature instead a first workspace analysis from the energetic point of view, an energy-optimal solution method for free-floating manipulators featuring concurrent control of the robotic arm and the base spacecraft ACS, and a local algorithm able to produce a consistent reduction of kinetic energy integral along an end-effector trajectory. All new algorithms have been validated against existing algorithms available in literature to prove their effectiveness, and the global algorithm has been used to solve problems that have not been solved before in robotics literature to the author’s knowledge (torque and power constrained, bi-objective optimisation).

Future research in this field can proceed in several interesting directions. Some have been mentioned when talking about the limitations: the most obvious future development is an extension of the work to manipulators with a higher number of degrees of freedom and testing on real manipulators. The manipulator used in this work has been chosen because it allows to focus on the novel algorithms rather than on the implementation but, although some practical manipulators with a limited amount of degrees of freedom exist, it is clear that extending the results to more complex problem would greatly increase the usefulness of the results of this work.

While both the global and the local methods presented would benefit from an investigation on higher numbers of DOF, and from testing on a real physical manipulator, the challenges arising for the two would be of a different kind: the global algorithm presents from this point of view a higher level of maturity, in that it can already respect all the limitations (joints, velocity, torques) of a real manipulator. It is however to test if cases exist when the optimal solution on the complete problem does not fall close to the optimal solution on a subset of parameters, and how the algorithm would behave in such cases. A further limitation that could arise from manipulators with increased complexity is computational time. Regarding this, it is important to observe that the algorithm, so far, has only been coded in Matlab with an off-the-shelf implementation of SQP. An ad-hoc adaptation of the SQP method, or even just a lower level C/C++ implementation, might reduce the time and cost required for convergence. Further important improvements for the global algorithm presented in this thesis lie in the direction of obstacle avoidance: since the method has proven effective in respecting nonlinear constraints, a natural question arises if it can tackle obstacles in the environment. Further questions arise from this, regarding the best implementation choice to represent obstacles as constraints, and to represent distance between the manipulator and the obstacles. Surely, such a development would in any case require several degrees of redundancy to be able to move around said obstacles.

The chapters dedicated to local algorithms analysis and development show many possible developments, which in most cases come directly from their local nature. Most noticeably, the workspace analysis presented in this work shows limits of local minimization of kinetic energy and presents some insights about the relationship between manipulator configuration and kinetic energy. This is however not directly usable, so far, for improving robotic inverse kinematics planning, as the attempt to use PMCI in Chapter 6 has shown. More investigation is required in this sense, to see if the results regarding kinematic indexes and their canonical correlation can be used to support energy optimisation heuristics for redundant robotic manipulators. Chapter 5 also presented a solution specific to free-floating manipulators. Coordinating actuators of different nature such as reaction wheels and robotic joints actuators can be challenging, and experimental work with free-floating manipulators trajectory tracking is required to show the feasibility of the result in practice.

Finally, the PMKE algorithm shows very promising results for online kinetic energy integral minimization. It also would benefit from future developments with a higher number of DOF and testing on a real manipulator, in this case however a more efficient implementation would be capital for the success of the experiment, as the algorithm currently failed in being real-time by a fraction of second. This can be expected in a test implementation in Matlab, which is of a different nature than implementations on real manipulators, a test with a speed-oriented implementation is however a necessity for future developments. One more future goal of the algorithm is to develop a version that effectively optimizes torque integral. This is more challenging than kinetic energy, as the torque has more granular dependency of the time step, rendering the method used for the prediction much less effective. This leads to the main goal in the future development of the PMKE, which is to integrate the spline interpolation in the optimisation, in order to obtain a result that is well behaved not only at the path points where the prediction is computed, but on the spline interpolation connecting them as well.

The reader can see that, despite having offered solutions for some of the open issues of the inverse kinematics problem of redundant manipulator, many points still remain open. This is due to the nonlinear, difficult nature of the inverse kinematics problem which, although being one of the first problems having been investigated in robotics, remains particularly challenging to these days, with new developments being published every year. Despite it seems so obvious for us humans to move in an efficient way and without violating our physical limits, it seems that the road to have our machines moving as efficiently as us, or more efficiently than us, still has not led us to a final destination.

References

[1] D. R. Tobergte and S. Curtis, *Handbook of Robotics*, vol. 53, no. 9. 2013.

[2] C. Sallaberger, “Canadian Space Robotic Activities,” *Acta Astronaut.*, vol. 4, no. 10, pp. 239–246, 1997.

[3] N. Sato and Y. Wakabayashi, “JEMRMS Design Features and Topics from Testing,” in *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS*, 2001, pp. 1–7.

[4] F. Didot, M. Oort, J. Kouwen, and P. Verzijden, “The ERA System : Control Architecture and Performances Results,” in *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS*, 2001, pp. 1–7.

[5] D. Whitney, “Resolved Motion Rate Control of Manipulators and Human Prostheses,” *IEEE Trans. Man Mach. Syst.*, vol. 10, no. 2, pp. 47–53, 1969, doi: 10.1109/TMMS.1969.299896.

[6] Y. Nakamura, H. Hanafusa, and Tsuneo Yoshikawa, “Task-Priority Based Redundancy Control or Robot Manipulators,” *Int. J. Rob. Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[7] E. Papadopoulos and S. Dubowsky, “Dynamic Singularities in Free- Floating Space Manipulators,” *Trans. ASME*, vol. 115, no. 1, pp. 44–52, 1993.

[8] Y. Nakamura and H. Hanafusa, “Optimal Redundancy Control of Robot Manipulators,” *Int. J. Rob. Res.*, vol. 6, no. 1, pp. 32–42, 1987.

[9] D. P. Martin, J. Baillieul, and J. M. Hollerbach, “Resolution of Kinematic Redundancy Using Optimisation Techniques,” *IEEE Trans. Robot.*, vol. 5, no. 4, pp. 529–533, 1989.

[10] D. N. Nenchev, “Redundancy resolution through local optimization: A review,” *J. Robot. Syst.*, vol. 6, no. 6, pp. 769–798, 1989, doi: 10.1002/rob.4620060607.

[11] C. W. Wampler, “Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods,” no. 1, pp. 93–101, 1986.

[12] T. Yoshikawa, “Manipulability and redundancy control of robotic mechanisms,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 1985, pp. 1004–1009.

[13] E. Staffetti, H. Bruyninckx, and J. De Schutter, “On the invariance of manipulability indices,” *Adv. Robot Kinemat.*, no. 1, pp. 57–66, 2002, doi: 10.1.1.18.2222.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer-Verlag London Limited, 2009.

[15] A. Nedungadi and K. Kazerouinian, “A local solution with global characteristics for the joint torque optimization of a redundant manipulator,” *J. Robot. Syst.*, vol. 6, no. 5, pp. 631–654, 1989, doi: 10.1002/rob.4620060508.

[16] K. S. K. Suh and J. Hollerbach, “Local versus global torque optimization of redundant manipulators,” *Proceedings. 1987 IEEE Int. Conf. Robot. Autom.*, vol. 4, pp. 0–5, 1987, doi: 10.1109/ROBOT.1987.1087955.

[17] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, “Predictive Inverse Kinematics for Redundant Manipulators with Task Scaling and Kinematic Constraints,” *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 278–285, 2019, doi: 10.1109/TRO.2018.2871439.

[18] A. Hirakawa, “Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion,” *Robot. Autom. 1997.*, no. April, pp. 5–10, 1997, [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=619323.

[19] Y. Umetani and K. Yoshida, “Resolved motion rate control of space manipulators with generalized Jacobian matrix,” *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 2–12, 1989, doi: 10.1109/70.34766.

[20] C. G. Atkeson, C. H. An, and J. M. Hollerbach, “Estimation of Inertial Parameters of Manipulator Loads and Links,” *Int. J. Rob. Res.*, vol. 5, no. 3, pp. 101–119, 1986.

[21] P. Huang, W. Xu, B. Liang, and Y. Xu, “Configuration control of space robots for impact minimization,” *2006 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2006*, pp. 357–362, 2006, doi: 10.1109/ROBIO.2006.340202.

[22] E. Shintaku, “Minimum energy trajectory for an underwater manipulator and its simple planning method by using a genetic algorithm,” *Adv. Robot.*, vol. 13, no. 2, pp. 115–138, 1998, doi: 10.1163/156855399X00171.

[23] E. Ferrentino, A. Della Cioppa, A. Marcelli, and P. Chiacchio, “An Evolutionary Approach to Time-Optimal Control of Robotic Manipulators,” *J. Intell. Robot. Syst. Theory Appl.*, pp. 245–260, 2019, doi: 10.1007/s10846-019-01116-9.

[24] J. M. Hollerbach and K. C. Suh, “Redundancy Resolution of Manipulators through Torque Optimization,” *IEEE J. Robot. Autom.*, vol. 3, no. 4, pp. 308–316, 1987, doi: 10.1109/JRA.1987.1087111.

[25] C. Gosselin and J. Angeles, “A Global Performance Index for the Kinematic Optimization of Robotic Manipulators,” *J. Mech. Des.*, vol. 113, no. September 1991, p. 220, 1991, doi: 10.1115/1.2912772.

[26] Y. Umetani and K. Yoshida, “Workspace and Manipulability Analysis of Space Manipulator,” *Trans. Soc. Instrum. Control Eng.*, vol. E-1, no. 1, pp. 1–8, 2001, doi: 10.9746/sicetr1965.26.188.

[27] K. Yoshida, “Practical Coordination Control Between Satellite Attitude and Manipulator Reaction Dynamics Based on Computed Momentum Concept,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 3, pp. 1578–1585, 1994, doi: 10.1109/IROS.1994.407645.

[28] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics

technologies for on-orbit servicing," *Prog. Aerosp. Sci.*, vol. 68, pp. 1–26, 2014, doi: 10.1016/j.paerosci.2014.03.002.

[29] C. A. Klein and C. H. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 2, pp. 245–250, 1983, doi: 10.1109/TSMC.1983.6313123.

[30] J. Baillieul, J. M. Hollerbach, and R. Brockett, "Programming and control of kinematically redundant manipulators," in *Proceedings of 23rd Conference on Decision and Control*, 1984, pp. 768–774.

[31] C. Chevallereau and W. Khalil, "A new method for the solution of the inverse kinematics of redundant robots," in *Proceedings. IEEE International Conference on Robotics and Automation*, 1988, vol. 4, pp. 37–42.

[32] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," *2008 8th IEEE-RAS Int. Conf. Humanoid Robot. Humanoids 2008*, pp. 22–27, 2008, doi: 10.1109/ICHR.2008.4755926.

[33] L. Sciavicco and B. Siciliano, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators," *IEEE J. Robot. Autom.*, vol. 4, no. 4, pp. 403–410, 1988, doi: 10.1109/56.804.

[34] K. Kazerouinian and Z. Wang, "Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators," *Int. J. Rob. Res.*, no. 1984, pp. 3–12, 1987.

[35] M. Vukobratovic and M. Kircanski, "A Dynamic Approach to Nominal Trajectory Synthesis for Redundant Manipulators," vol. 103, no. 4, pp. 580–586, 1984.

[36] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," 2016.

[37] D. Schinstock, T. Faddis, and R. Greenway, "Robust Inverse Kinematics Using Damped Least Squares With Dynamic Weighting," *… Intell. Robot. Field, Factory, …*, pp. 861–869, 1994, doi: doi:10.2514/6.1994-1299.

[38] A. A. Maciejewski and C. A. Klein, "Numerical Filtering of the Operation of Robotic Manipulators through Kinematically Singular Configurations." pp. 527–552, 1988.

[39] M. Kelemen, I. Virgala, T. Lipták, L. Miková, F. Filakovský, and V. Bulej, "A novel approach for a inverse kinematics solution of a redundant manipulator," *Appl. Sci.*, vol. 8, no. 11, 2018, doi: 10.3390/app8112229.

[40] J. Woolfrey, W. Lu, and D. Liu, "A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces," *J. Intell. Robot. Syst. Theory Appl.*, vol. 96, no. 1, pp. 3–16, 2019, doi: 10.1007/s10846-018-0964-8.

[41] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 785–792, 2011, doi: 10.1109/TRO.2011.2142450.

[42] F.-T. Cheng, R.-J. Sheu, and T.-H. Chen, "The Improved Compact QP Method For Resolving Manipulator Redundancy," *IEEE Trans. Syst. Man Cybern.*, vol. 25, no. 11,

pp. 1521–1530, 1995.

[43] W. S. Tang, S. Member, J. Wang, and S. Member, “Two Recurrent Neural Networks for Local Joint Torque Optimization of Kinematically Redundant,” vol. 30, no. 1, pp. 120–128, 2000.

[44] Y. Zhang and J. Wang, “A dual neural network for constrained joint torque optimization of kinematically redundant manipulators,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 32, no. 5, pp. 654–662, 2002, doi: 10.1109/TSMCB.2002.1033184.

[45] Y. Zhang, S. S. Ge, and T. H. Lee, “A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 34, no. 5, pp. 2126–2132, 2004, doi: 10.1109/TSMCB.2004.830347.

[46] T. Rybus, K. Seweryn, and J. Z. Sasiadek, “Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (NMPC),” *J. Intell. Robot. Syst.*, pp. 491–509, 2016, doi: 10.1007/s10846-016-0396-2.

[47] C. Schuetz, T. Buschmann, J. Baur, J. Pfaff, and H. Ulbrich, “Predictive Online Inverse Kinematics for Redundant Manipulators,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5056–5061, doi: 10.1109/ICRA.2014.6907600.

[48] O. L. Mangasarian, “Sufficient Conditions for the Optimal Control of Nonlinear Systems,” *SIAM J. Control*, vol. 4, no. 1, pp. 139–152, 1966, doi: 10.1137/0304013.

[49] M. Galicki, “Time-Optimal Controls of Kinematically Redundant Manipulators with Geometric Constraints,” *IEEE Trans. Robot.*, vol. 16, no. 1, pp. 89–93, 2000.

[50] S. Ma and M. Watanabe, “Time Optimal Path-Tracking Control of Kinematically Redundant Manipulators,” *JSME Int. J.*, vol. 47, no. 2, pp. 582–590, 2004.

[51] Z. L. Zhou and C. C. Nguyen, “Globally Optimal Trajectory Planning for Redundant Manipulators using State Space Augmentation Method,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 19, no. 1, pp. 105–117, 1997, doi: 10.1023/A:1007905817998.

[52] Y. Halevi, E. Carpanzano, G. Montalbano, and G. Halevi, Yoram and Carpanzano, Emanuele and Montalbano, “Minimum Energy Control of Redundant Linear Manipulators,” *J. Dyn. Syst. Meas. Control*, vol. 136, no. September 2014, pp. 1–8, 2014, doi: 10.1115/1.4027419.

[53] Z. Shiller and S. Dubowsky, “On Computing the Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles,” *IEEE Trans. Robot. an Autom.*, vol. 7, no. 6, pp. 785–797, 1991.

[54] S. F. P. Saramago and V. Steffen Jr, “Trajectory Modeling of Robot Manipulators in the Presence of Obstacles,” *Jounal Optim. Theory Appl.*, vol. 110, no. 1, pp. 17–34, 2001.

[55] A. Reiter, A. Muller, and H. Gattringer, “On Higher Order Inverse Kinematics Methods in Time-Optimal Trajectory Planning for Kinematically Redundant Manipulators,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 4, pp. 1681–1690, 2018, doi:

10.1109/TII.2018.2792002.

[56] A. Reiter, H. Gattringer, and A. Müller, "Redundancy resolution in minimum-time path tracking of robotic manipulators," *ICINCO 2016 - Proc. 13th Int. Conf. Informatics Control. Autom. Robot.*, vol. 2, no. Icinco, pp. 61–68, 2016, doi: 10.5220/0005975800610068.

[57] Richard Bellman, "On the Theory of Dynamic Programming," *Proc. N. A. S.*, vol. 38, pp. 716–719, 1952.

[58] E. Ferrentino and P. Chiacchio, "A Topological Approach to Globally-Optimal Redundancy Resolution with Dynamic Programming," in *ROMANSY 22, Robot Design, Dynamics and Control*, Springer International Publishing, 2019, pp. 77–85.

[59] A. Guigue, M. Ahmadi, R. Langlois, and M. J. Hayes, "Pareto optimality and multiobjective trajectory planning for a 7-DOF redundant manipulator," *IEEE Trans. Robot.*, vol. 26, no. 6, pp. 1094–1099, 2010, doi: 10.1109/TRO.2010.2068650.

[60] A. P. Pashkevich, A. B. Dolgui, and O. A. Chumakov, "Multiobjective optimization of robot motion for laser cutting applications," *Int. J. Comput. Integr. Manuf.*, vol. 17, no. 2, pp. 171–183, 2004, doi: 10.1080/0951192031000078202.

[61] A. Dolgui and A. Pashkevich, "Manipulator motion planning for high-speed robotic laser cutting," *Int. J. Prod. Res.*, vol. 47, no. 20, pp. 5691–5715, 2009, doi: 10.1080/00207540802070967.

[62] J. Gao, A. Pashkevich, and S. Caro, "Optimization of the robot and positioner motion in a redundant fiber placement workcell," *Mech. Mach. Theory*, vol. 114, pp. 1339–1351, 2017, doi: 10.1016/j.mechmachtheory.2017.04.009.

[63] G. Field and Y. Stepanenko, "Iterative Dynamic Programming : An Approach to Minimum Energy Trajectory Planning for Robotic Manipulators," in *Proceedings of the 1996 IEEE Intemational Conference on Robotics and Automation*, 1996, pp. 2755–2760.

[64] J. Nurmi and J. Mattila, "Global energy-optimal redundancy resolution of hydraulic manipulators: Experimental results for a forestry manipulator," *Energies*, vol. 10, no. 5, 2017, doi: 10.3390/en10050647.

[65] Y. Davidor, *Genetic Algorithms and Robotics: A heuristic strategy for optimization*. World Scientific Publishing Company, 1991.

[66] B. McAvoy, B. Sangolola, and Z. Szabad, "Optimal trajectory generation for redundant planar manipulators," in *2000 IEEE International Conference on Systems, Man & Cyberbetics*, 2002, pp. 3241–3246, doi: 10.1109/icsmc.2000.886503.

[67] L. Tian and C. Collins, "Motion planning for redundant manipulators using a floating point genetic algorithm," *J. Intell. Robot. Syst. Theory Appl.*, vol. 38, no. 3–4, pp. 297–312, 2003, doi: 10.1023/B:JINT.0000004973.29102.33.

[68] A. A. Ata and T. R. Myo, "Collision-free trajectory planning for manipulators using generalized pattern search," *Int. J. Simul. Model.*, vol. 5, no. 4, pp. 145–154, 2006, doi: 10.2507/IJSIMM05(4)2.074.

[69] R. Saravanan and S. Ramabalan, “Evolutionary Minimum Cost Trajectory Planning for Industrial Robots,” *J. Intell. Robot. Syst.*, vol. 52, no. 1, pp. 45–77, 2008, doi: 10.1007/s10846-008-9202-0.

[70] S. Stevo, I. Sekaj, and M. Dekan, “Optimization of Energy in Robotic arm using Genetic Algorithm,” in *19th World Congress The International Federation of Automatic Control*, 2014.

[71] C. Hansen, J. Kotlarski, and T. Ortmaier, “Experimental validation of advanced minimum energy robot trajectory optimization,” *2013 16th Int. Conf. Adv. Robot. ICAR 2013*, no. 1, pp. 1–8, 2013, doi: 10.1109/ICAR.2013.6766463.

[72] M. Jin, Q. Liu, B. Wang, and H. Liu, “an Efficient and accurate Inverse Kinematics for 7-DOF Redundant Manipulators Based on a Hybrid of analytical and Numerical Method,” *IEEE Access*, vol. 8, pp. 16316–16330, 2020, doi: 10.1109/aCCESS.2020.2966768.

[73] A. D’Souza, S. Vijayakumar, and S. Schaal, “Learning Inverse Kinematics,” in *International Conference on Intelligence in Robotics and Autonomous Systems (IROS)*, 2001, pp. 1–8.

[74] D. Berenson, P. Abbeel, and K. Goldberg, “A robot path planning framework that learns from experience,” *2012 IEEE Int. Conf. Robot. Autom.*, pp. 3671–3678, 2012, doi: 10.1109/ICRA.2012.6224742.

[75] K. Hauser, “Learning the Problem-Optimum Map: Analysis and Application to Global Optimization in Robotics,” no. Rlp, pp. 1–12, 2016, doi: 10.1109/TRO.2016.2623345.

[76] R. Raja, A. Dutta, and B. Dasgupta, “Learning framework for inverse kinematics of a highly redundant mobile manipulator,” *Rob. Auton. Syst.*, vol. 120, p. 103245, 2019, doi: 10.1016/j.robot.2019.07.015.

[77] S. Greaves, K. Boyle, N. Doshewnek, C. Engineer, and S. Engineer, “Orbiter Boom Sensor System and Shuttle Return to Flight: Operations Analyses,” in *AIAA Guidance, Navigation, and Control Conference*, 2005, pp. 1–7.

[78] R. L. Ticker, F. Cepollina, and B. B. Reed, “NASA’s In-Space Robotic Servicing,” 2015, pp. 1–8.

[79] M. Oda, K. Kibe, and F. Yamagata, “ETS-VII, space robot in-orbit experiment satellite,” *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, no. April 1996, pp. 739–744, 1997, doi: 10.1109/ROBOT.1996.503862.

[80] K. Yoshida, H. Nakanishi, N. Inaba, H. Ueno, and M. Oda, “Contact Dynamics and Control Strategy Based on Impedance Matching for Robotic Capture of a Non-cooperative Satellite,” in *Proc. 15th CISM-IFToMM Symp. On Robot Design, Dynamics and Control-Romansy*, 2004.

[81] R. B. Friend, “Orbital Express Programm Summary And Mission Overview,” *SPIE Def. Secur. Symp.*, vol. 6958, no. 2008, pp. 1–11, doi: 10.1117/12.783792.

[82] A. B. Bosse *et al.*, “SUMO : spacecraft for the universal modification of orbits,” 2004, vol. 5419, pp. 36–46, doi: 10.1117/12.547714.

[83] K. A. Harris, C. G. Henshaw, J. A. Lennon, W. E. Purdy, F. A. Tasker, and W. S. Vincent, "FREND : Pushing the Envelope of Space Robotics," *NRL Rev. - Sp. Res.*, pp. 239–241, 2008.

[84] T. J. Debus and S. P. Dougherty, "Overview and Performance of the Front-End Robotics Enabling Near-Term Demonstration (FREND) Robotic Arm," in *AIAA infotech@ aerospace conference*, 2009.

[85] H. A. Thronson, D. Akin, J. Grunsfeld, and D. Lester, "The Evolution and Promise of Robotic In-Space Servicing," in *AIAA SPACE Conference & Exposition*, 2009.

[86] D. Barnhart *et al.*, "Phoenix Program Status - 2013," in *AIAA SPACE conference & exposition*, 2013, doi: 10.2514/6.2013-5341.

[87] T. E. Rumford, "Autonomous Rendezvous Technology (DART) Project Summary," in *SPIE's SPace Systems Technology and Operations Conference*, 2003.

[88] B. Sommer, "Automation and Robotics in the German Space Program – Unmanned on-orbit servicing (OOS) & the TECSAS mission," 2004, doi: 10.2514/6.IAC-04-IAA.3.6.2.03.

[89] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "DEOS – The German Robotics Approach to Secure and De-Orbit Malfunctioned Satellites from Low Earth Orbits DEOS – A Robotic Servicing Mission," in *i-SAIRAS*, 2010, pp. 244–251.

[90] Y. Masutani, F. Miyazaki, and S. Arimoto, "Sensory feedback control for space manipulators," *Proceedings, 1989 Int. Conf. Robot. Autom.*, pp. 1346–1351, 1989, doi: 10.1109/ROBOT.1989.100167.

[91] Z. Vafa and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach," *Proceedings. 1987 IEEE Int. Conf. Robot. Autom.*, vol. 4, pp. 579–585, 1987, doi: 10.1109/ROBOT.1987.1088032.

[92] Z. Vafa and S. Dubowsky, "The Kinematics and Dynamics of Space Manipulators : The Virtual Manipulator," *Int. J. Rob. Res.*, vol. 9, no. 3, pp. 3–21, 1990, doi: 10.1177/027836499000900401.

[93] Z. Vafa and S. Dubowskr, "On the Dynamics of Space Manipulators Using the Virtual Manipulator , with Applications to Path Planning," in *Space Robotics: Dynamics and Control*, Kluwer Academic Publishers, 1993, pp. 45–76.

[94] S. Dubowsky and M. A. Torres, "Path planning for space manipulators to minimize spacecraft attitude disturbance," in *Pmedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2522–2528.

[95] F. Caccavale and B. Siciliano, "Kinematic control of redundant free-floating robotic systems," *Adv. Robot.*, vol. 15, no. 4, pp. 429–448, 2001, doi: 10.1163/156855301750398347.

[96] D. N. Nenchev and K. Yoshida, "Analysis , design and control of free-flying space robots using fixed-attitude-restricted Jacobian Matrix," *Robot. Res. Fifth Intenraitonal Symp.*, no. June, pp. 251–258, 1991.

[97] S. Cocuzza, I. Pretto, and S. Debei, “Least-Squares-Based Reaction Control of Space Manipulators,” *J. Guid. Control. Dyn.*, vol. 35, no. 3, pp. 976–986, 2012, doi: 10.2514/1.45874.

[98] M. Sabatini, P. Gasbarri, and G. B. Palmerini, “Coordinated control of a space manipulator tested by means of an air bearing free floating platform,” *Acta Astronaut.*, vol. 139, pp. 296–305, 2017, doi: 10.1016/j.actaastro.2017.07.015.

[99] M. De Stefano, H. Mishra, R. Balachandran, R. Lampariello, C. Ott, and C. Secchi, “Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1319–1326, 2019, doi: 10.1109/LRA.2019.2895420.

[100] S. Pandey and S. K. Agrawal, “Path Planning of Free-Floating Prismatic-Jointed Manipulators,” pp. 127–140, 1997.

[101] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, and M. Uchiyama, “Reaction Null-Space Control of Flexible Structure Mounted Manipulator System,” *IEEE Trans. Robot. an Autom.*, vol. 15, no. 6, pp. 1011–1023, 1999.

[102] D. N. Nenchev, K. Yoshida, and M. Uchiyama, “Reaction Null-space Based Control of Flexible Structure Mounted Manipulator Systems,” in *Proceedings of the 35th IEEE Decision and Control*, 1996, pp. 4118–3123, doi: 10.1109/CDC.1996.577417.

[103] P. Piersigilli, I. Sharf, and A. K. Misra, “Reactionless capture of a satellite by a two degree-of-freedom manipulator,” *Acta Astronaut.*, vol. 66, no. 1–2, pp. 183–192, 2010, doi: 10.1016/j.actaastro.2009.05.015.

[104] A. Pisculli, L. Felicetti, P. Gasbarri, G. B. Palmerini, and M. Sabatini, “A reaction-null/Jacobian transpose control strategy with gravity gradient compensation for on-orbit space manipulators,” *Aerosp. Sci. Technol.*, vol. 38, pp. 30–40, 2014, doi: 10.1016/j.ast.2014.07.012.

[105] W. Xu, B. Liang, C. Li, Y. Xu, and W. Qiang, “Path Planning of Free-Floating Robot in Cartesian Space Using Direct Kinematics,” *Int. J. Adv. Robot. Syst.*, vol. 4, no. 1, pp. 17–26, 1994.

[106] K. Yamada, “Arm path planning for a space robot,” in *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robot an Systems*, 1993, pp. 2049–2055.

[107] K. Yamada, S. Yoshikawa, and Y. Fujita, “Arm path planning of a space robot with angular momentum,” *Adv. Robot.*, vol. 9, no. 6, pp. 693–709, 1994, doi: 10.1163/156855395X00364.

[108] T. Suzuki and Y. Nakamura, “Planning Spiral Motion of Nonholonomic Space Robots,” in *IEEE international conferences on robotics and automation*, 1996, no. 1, pp. 718–725, doi: 10.1109/ROBOT.1996.503859.

[109] K. Yoshida, K. Hashizume, and S. Abiko, “Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm,” *IEEE Int. Conf. Robot. Autom.*, vol. 1, pp. 441–446, 2001, doi: 10.1109/ROBOT.2001.932590.

[110] Y. Nakamura and R. Mukherjee, “Exploiting Nonholonomic Redundancy of Free-

Flying Space Robots," *IEEE Trans. Robot. Autom.*, vol. 9, no. 4, pp. 499–506, 1993, doi: 10.1109/70.246062.

[111] E. Papadopoulos, I. Tortopidis, and K. Nanos, "Smooth planning for free-floating space robots using polynomials," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2005, no. 10, pp. 4272–4277, 2005, doi: 10.1109/ROBOT.2005.1570777.

[112] I. Tortopidis and E. Papadopoulos, "On point-to-point motion planning for underactuated space manipulator systems," *Rob. Auton. Syst.*, vol. 55, pp. 122–131, 2007, doi: 10.1016/j.robot.2006.07.003.

[113] J. Franch, S. K. Agrawal, S. Oh, and A. Fattah, "Design of Differentially Flat Planar Space Robots : A Step Forward in their Planning and Control," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 3–4, no. October, pp. 3053–3058, 2003, doi: 10.1109/IROS.2003.1249625.

[114] S. K. Agrawal *et al.*, "A Differentially Flat Open-Chain Space Robot with Arbitrarily Oriented Joint Axes and Two Momentum Wheels at the Base," *IEEE Trans. Automat. Contr.*, vol. 54, no. 9, pp. 2185–2191, 2009.

[115] W. Xu, Y. Liu, B. Liang, Y. Xu, C. Li, and W. Qiang, "Non-holonomic Path Planning of a Free-Floating Space Robotic System Using Genetic Algorithms," *Adv. Robot.*, vol. 22, no. 4, pp. 451–476, 2008, doi: 10.1163/156855308X294680.

[116] W. Xu, C. Li, B. Liang, Y. Xu, Y. Liu, and W. Qiang, "Target berthing and base reorientation of free-floating space robotic system after capturing," *Acta Astronaut.*, vol. 64, no. 2–3, pp. 109–126, 2009, doi: 10.1016/j.actaastro.2008.07.010.

[117] A. Kumar and K. J. Waldron, "The Workspaces of a Mechanical Manipulator," *J. Mech. Des.*, vol. 103, pp. 665–672, 1981.

[118] E. J. Haug, C.-M. Luh, F. A. ADkins, and J.-Y. Wang, "Numerical Algorithms for Mapping Boundaries of Manipulator Workspaces," *Adv. Des. Autom.*, vol. 2, no. 69, pp. 447–459, 1994.

[119] Y. Cao, K. Lu, X. Li, and Y. Zang, "Accurate Numerical Methods for Computing 2D and 3D Robot Workspace," *Int. J. Adv. Robot. Syst.*, vol. 8, no. 6, pp. 1–13, 2011, doi: 10.5772/45686.

[120] Yisheng Guan, K. Yokoi, and X. Zhang, "Numerical Methods for Reachable Space Generation of Humanoid Robots," *Int. J. Rob. Res.*, vol. 27, no. 8, pp. 935–950, 2008.

[121] O. Bohigas, M. Manubend, and L. Ros, "A Complete Method for Workspace Boundary Determination on General Structure Manipulators," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 993–1006, 2012, doi: 10.1109/TRO.2012.2196311.

[122] O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, and J. M. Porta, "A general method for the numerical computation of manipulator singularity sets," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 340–351, 2014, doi: 10.1109/TRO.2013.2283416.

[123] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing Robot Workspace Structure : Representing Robot Capabilities," pp. 3229–3236, 2007.

[124] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority Based Redundancy

Control of Robot Manipulators," *Int. J. Rob. Res.*, vol. 6, no. 1, pp. 32–42, 1987, doi: 10.1177/027836498700600103.

[125] P. Baerlocher and R. Boulic, "Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 1, no. October, pp. 323–329, 1998, doi: 10.1109/IROS.1998.724639.

[126] Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Trans. Robot. an Autom.*, vol. 5, no. 3, pp. 2–12, 1989.

[127] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *J. Appl. Mech.*, pp. 215–221, 1955.

[128] J. J. Craig, *Introduction to Robotics*, Third Edit. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[129] S. Boyd and L. Vandenberghe, *Convex Optimization*, Seventh Ed. 2009.

[130] J. Pearl, *Heuristics: Intelligent search strategies for computer problem solving*. 1984.

[131] A. P. Castaño, *Practical Artificial Intelligence*. Apress Media, 2018.

[132] M. Melanie, *An Introduction to Genetic Algorithms*, Fifth Edit. Cambride, MAssachusetts: MIT Press, 1999.

[133] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes - The Art of Scientific Computing*, Third Edit. Cambridge: Cambridge University Press, 2007.

[134] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti, "Scatter Search and Local NLP Solvers : A Multistart Framework for Global Optimization Scatter Search and Local NLP Solvers : A Multistart Framework for Global Optimization," *Informs J. Comput.*, vol. 19, no. 3, pp. 328–340, 2007, doi: 10.2139/ssrn.886559.

[135] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimisation*, Second Edi. Springer International Publishing, 2009.

[136] A. Wernli and G. Cook, "Suboptimal Control for the Nonlinear Quadratic Regulator Problem," *Automatica*, vol. 11, pp. 75–84, 1975.

[137] J. EIckhoff, *Simulating Spacecraft Systems*. Springer-Verlag Berlin Heidelberg, 2009.

[138] S. Cocuzza, I. Pretto, and S. Debei, "a Constrained Least Squares Approach for Reaction Torque Control in Spacecraft / Manipulator Systems," *Int. Astronaut. Congr.*, 2009.

[139] D. J. Bender and A. J. Laub, "The Linear-Quadratic Optimal Regulator for Descriptor Systems," *IEEE Trans. Autom. Control*, vol. 32, no. 8, 1987.

[140] T. Çimen and S. P. Banks, "Nonlinear optimal tracking control with application to super-tankers for autopilot design," *Automatica*, vol. 40, no. 11, pp. 1845–1863, 2004, doi: 10.1016/j.automatica.2004.05.015.

[141] S. P. Banks and K. Dinesh, "Approximate Optimal Control and Stability of Nonlinear

Finite- and Infinite-Dimensional Systems," *Ann. ofOperations Res.*, vol. 98, pp. 19–44, 2000.

[142] D. Guo and Y. Zhang, "Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1203–1217, 2014, doi: 10.1109/TASE.2014.2346490.

[143] F. J. Solis and R. J.-B. Wets, "Minimization by Random Search Techniques," *Math. Oper. Res.*, vol. 6, no. 1, pp. 19–30, 1981.

[144] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numer.*, pp. 1–52, 1996.

[145] R. Martì, "Multi-Start Methods," in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds. Boston, MA: International Series in Operations Research & Management Science, vol 57. Springer, 2003.

[146] B. J. Martin and James E., "Minimum-Effort Motions for Open-Chain Manipulators with Task-Dependent End-Effector Constraints," *Int. J. Rob. Res.*, vol. 18, no. 2, pp. 213–224, 1999.

[147] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Trans. Syst. Man Cybern.*, vol. 7, no. 12, pp. 868–871, 1977.

[148] R. J. Beckman, W. J. Conover, and M. D. McKay, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[149] D. F. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Math. Comput.*, vol. 24, no. 111, pp. 647–656, 1970.

[150] B. W. Choi, J. H. Won, and M. J. Chung, "Optimal Redundancy Resolution of a Kinematically Redundant Manipulator for a Cyclic Task," vol. 9, no. 4, pp. 481–503, 1991.

[151] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Struct. Optim.*, vol. 14, no. 1, pp. 63–69, 1997, doi: 10.1007/BF01197559.

[152] K. Chircop and D. Zammit-Mangion, "On Epsilon-Constraint Based Methods for the Generation of Pareto Frontiers," *J. Mech. Eng. Autom.*, vol. 3, no. 5, pp. 279–289, 2013.

[153] H. Hotelling, "Relations between two sets of variates," *Am. Math. Soc. Inst. Math. Stat.*, pp. 321–377, 1935.

[154] J.K. Salisbury and J. Craig, "Articulated hands: Force control and kinematics issues," *Int. J. Robot. Reseorh, Vol. 1, No. 1.*, pp. 4–17, 1982, [Online]. Available: http://journals.sagepub.com/doi/pdf/10.1177/027836498200100102.

[155] K. C. Olds, "Global Indices for Kinematic and Force Transmission Performance in Parallel Robots," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 494–500, 2015, doi:

10.1109/TRO.2015.2398632.

[156] L. P. Patterson, “On-Orbit Maintenance Operations Strategy for the International Space Station - Concept and Implementation,” pp. 1–9.

[157] D. King, “Hubble robotic servicing: Stepping stone for future exploration missions,” *1st Sp. Explor. Conf. Contin. Voyag. Discov.*, vol. 1, no. February 2005, pp. 246–257, 2005, doi: 10.2514/6.2005-2524.

[158] W. Xu, B. Liang, B. Li, and Y. Xu, “A universal on-orbit servicing system used in the geostationary orbit,” *Adv. Sp. Res.*, vol. 48, no. 1, pp. 95–119, 2011, doi: 10.1016/j.asr.2011.02.012.

[159] F. Aghili and K. Parsa, “An adaptive vision system for guidance of a robotic manipulator to capture a tumbling satellite with unknown dynamics,” *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 3064–3071, 2008, doi: 10.1109/IROS.2008.4650758.

[160] F. Aghili, “A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics,” *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 634–649, 2012, doi: 10.1109/TRO.2011.2179581.

[161] J. P. Alepuz, M. R. Emami, and J. Pomares, “Direct image-based visual servoing of free-floating space manipulators,” *Aerosp. Sci. Technol.*, vol. 55, pp. 1–9, 2016, doi: 10.1016/j.ast.2016.05.012.

[162] N. W. Oumer, G. Panin, Q. Mülbauer, and A. Tseneklidou, “Vision-based localization for on-orbit servicing of a partially cooperative satellite,” *Acta Astronaut.*, vol. 117, pp. 19–37, 2015, doi: 10.1016/j.actaastro.2015.07.025.

APPENDIX A. Code samples

A.1 Interpolation-Based Global Kinematic Planner

The script below is the one used to generate the candidate solutions for the IBGKP. Its input are an *ingress* vector containing EE trajectory and its derivatives, a set of initial conditions *initial_cond_matrix*, and the number of path points *qsize*. The script computes all the $n_{candidates}$ and saves them in a *startmatrix*, orders them from the best one to the worst one, saves the best one in *ptmatrix*, and discards *startmatrix*.

```
        x = ingressi.x_d;
        xd = ingressi.xd_d;
        xdd = ingressi.xdd_d;
        times = ingressi.times;
        dt = times(2)-times(1);
        q0 = initial_cond_matrix;

        samplesize = (3*qsize)^2;

        ptmatrix = zeros(bestsolutionsnumber,3*qsize);

        stddev = mean(std(results.qd));

        functionvalue = zeros(samplesize,1);

        initcondmatrix = [normrnd(0,stddev,samplesize,qsize)
normrnd(0,stddev,samplesize,qsize) normrnd(0,stddev,samplesize,qsize)];

        loc_qd = results.qd;

        random_weights = zeros(3,3,samplesize);

        random_eig = rand(1,3,samplesize);
        for rr = 1:samplesize
            loc_random_eig = random_eig(1,:,samplesize);
            density = rand;
            random_weights(:,:,rr) = full(sprandsym(3, density,
loc_random_eig));
        end

        for k = 1:samplesize
                q0loc = q0;
                localinitcondition = initcondmatrix(k,:);
                localstart = startmatrix(k,:);
                qstart = q0loc(randi(size(initial_cond_matrix,1)),:);
                J = jac_parfor(qstart,robot);
                E = eye(3)-pinv(J)*J;
                v = E*[localinitcondition(1); localinitcondition(qsize+1);
...
                localinitcondition(2*qsize+1)];
                localstart(1) = qstart(1)+ v(1)*dt;
                localstart(qsize+1) = qstart(2)+ v(2)*dt;
                localstart(2*qsize+1) = qstart(3) + v(3)*dt;

            for kk = 2:min(qsize,50)
                J = jac_parfor([localstart(1); localstart(qsize+1);
localstart(2*qsize+1)],robot);
                E = eye(3)-pinv(J)*J;
                v = E*[localinitcondition(kk); localinitcondition(qsize+kk);
...
                localinitcondition(2*qsize+kk)];
                localstart(1) = localstart(1) + v(1)*0.01*dt;
                localstart(qsize+1) = localstart(qsize+1) + v(2)*dt;
                localstart(2*qsize+1) = localstart(2*qsize+1) + v(3)*dt;
            end

            startmatrix(k,:) = localstart;
        end

        for u = 2:qsize

            for k = 1:samplesize

                localstart = startmatrix(k,:);
                local_algo = algo;
                J = jac_parfor([localstart(u-1); localstart(qsize+u-1);
localstart(2*qsize+u-1)],robot);


                    J_plus = wpinv(random_weights(:,:,k),J);
                    v = J_plus*xd(u,:)';

                localstart(u) = localstart(u-1) + v(1)*dt;
                localstart(qsize+u) = localstart(qsize+u-1) + v(2)*dt;
                localstart(2*qsize+u) = localstart(2*qsize+u-1) + v(3)*dt;
                v_prec_loc = v;
                startmatrix(k,:) = localstart;

            end

        end

        switch cost_function
            case 'VEL'
                fun =
@(q,x,dt,q0,qsize,robot)objectivefunctionMS_vel_norm_with_limits(q,x,dt,q0,q
size,robot,joint_limit_for_ranking, vel_limit);
            case 'KIN'
                fun =
@(q,x,dt,q0,qsize,robot)objectivefunctionMS_with_limits_trj_constr(q,x,dt,q0
,qsize,robot,joint_limit_for_ranking, vel_limit);
            case 'DYN'
                fun =
@(q,x,dt,q0,qsize,robot)objectivefunctionMS_dyn_with_limits(q,x,dt,q0,qsize,
robot,joint_limit_for_ranking, vel_limit, friction_C);
        end

        for i = 1:samplesize

            functionvalue(i) = fun(startmatrix(i,:),x,dt,q0,qsize,robot);

        end
        tic
        [~,min_id] = sort(functionvalue,1,'ascend');
        toc

        min_id = min_id(1:bestsolutionsnumber);
        disp(functionvalue(min_id(1)));

        for k = 1:bestsolutionsnumber

            ptmatrix(k,:) = startmatrix(min_id(k),:);

        end

        clear startmatrix;
```

A.2 Workspace analysis

Workspace analysis has been performed through the use of parallel computing. The following script computes every trajectory by using the function `avvia_simulazione_H` and stores the results in the columns of a series matrices every time (one per each variable of interest). When the matrices columns are not completely filled by the results (because the trajectory was interrupted earlier by a singularity), they are filled with NaN ("Not a Number") which excludes them from any further analysis.

```
parfor thetastep=1:searchspacesize

    theta = thetastep*pi/256;
    length = 2;
    feasible = 1;
    segmentenergyposition = 1; %indicates which energy value to store next
    res = results;
    res.qd = [0 0 0]; %initialize velocities
    localingr = ingr;
    localingr.r = length/2;
    localingr.T = 4*localingr.r/v_max;
    localingr.theta = theta; %write angle
    [res, why(thetastep), finallength] = avvia_simulazione_H(res,localingr,
v_max);
    th_ln_lims(thetastep,:) = [theta, finallength]; %- v_max*dt)];
    res.en = [res.en; NaN(energysize-size(res.en,1),1)];
    res.manipulability = [res.manipulability; NaN(energysize-
size(res.manipulability,1),1)];
    res.en_tot = [res.en_tot; NaN(energysize-size(res.en_tot,1),1)];
    res.dyn_manipulability = [res.dyn_manipulability; NaN(energysize-
size(res.dyn_manipulability,1),1)];
    res.power = [res.power; NaN(energysize-size(res.power,1),1)];
    res.power_abs = [res.power_abs; NaN(energysize-
size(res.power_abs,1),1)];
    res.B_spectral_radius = [res.B_spectral_radius; NaN(energysize-
size(res.B_spectral_radius,1),1)];
    res.Jcondnumb = [res.Jcondnumb; NaN(energysize-
size(res.Jcondnumb,1),1)];
```

```
    res.worstcasevelindex = [res.worstcasevelindex; NaN(energysize-
size(res.worstcasevelindex,1),1)];
    res.kin_pot = [res.kin_pot; NaN(energysize-size(res.kin_pot,1),1)];
    kin_energymatrix(:, thetastep) = res.en;
    manipulabilitymatrix(:, thetastep) = res.manipulability;
    energymatrix(:, thetastep) = res.en_tot;
    dyn_manipulabilitymatrix(:, thetastep) = res.dyn_manipulability;
    powermatrix(:, thetastep) = res.power;
    power_absmatrix(:, thetastep) = res.power_abs;
    B_spectral_radiusmatrix(:, thetastep) = res.B_spectral_radius;
    Jcondnumbmatrix(:, thetastep) = res.Jcondnumb;
    worstcasevelindexmatrix(:, thetastep) = res.worstcasevelindex;
    kin_potmatrix(:, thetastep) = res.kin_pot;

end%parfor
```

A.3 PMKE

The following script executes the inverse kinematics according to the PMKE algorithm, with *i* being the number of path points. This piece of script is set for a predictive optimisation time step size *ΔT* of 100 time steps (0.1s), a horizon *h* of 2 and interval *I* of 50 time steps (0.05s).

```
for i=2:size(xdd(:,1))

            step_size = 100;
            vel = [0 0 0]';
            horizon = 2;
            lim = size(xdd(:,1),1)-step_size*horizon-1;

                if ((mod(i,50) == 0) && i <lim) || (i == 2)
                    t_interp=clock;

                    q_fit = 
optimizationtask1(q_prec,x,dt,i,step_size,horizon,q_fit,robot,qd_prec);
                   pp1 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size 
i+2*step_size i+2*step_size]*dt,[q_prec(1) qd_prec(1) qdd_prec(1) q_fit(1) 
q_fit(2) 0 0]));
                    pp2 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size 
i+2*step_size i+2*step_size]*dt,[q_prec(2) qd_prec(2) qdd_prec(2) q_fit(3) 
q_fit(4) 0 0]));
                     pp3 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size 
i+2*step_size i+2*step_size]*dt,[q_prec(3) qd_prec(3) qdd_prec(3) q_fit(5) 
q_fit(6) 0 0]));
                    kk = 0;
                elseif (mod(i,step_size) == 0) && i == lim
                    q_fit = 
optimizationtask1(q_prec,x,dt,i,step_size,horizon,q_fit,robot,old_vel);
                    pp1 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size 
i+2*step_size i+2*step_size]*dt,[q_prec(1) qd_prec(1) qdd_prec(1) q_fit(1) 
q_fit(2) 0 0]));
                    pp2 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size 
i+2*step_size i+2*step_size]*dt,[q_prec(2) qd_prec(2) qdd_prec(2) q_fit(3) 
q_fit(4) 0 0]));
```

```
                    pp3 = (spapi(4,[i-1 i-1 i-1 i+step_size i+2*step_size
i+2*step_size i+2*step_size]*dt,[q_prec(3) qd_prec(3) qdd_prec(3) q_fit(5)
q_fit(6) 0 0]));

                    kk = 0;
               end
                if i < 1001
                    vel(1) = (fnval(pp1,i*dt)-q_prec(1))/dt;
                    vel(2) = (fnval(pp2,i*dt)-q_prec(2))/dt;
                    vel(3) = (fnval(pp3,i*dt)-q_prec(3))/dt;
                    kk = kk + 1;

                end
            J = jac(q_prec);
            qd_att = pinv(J)*xd(i,:)' + ((eye(3)-pinv(J)*J))*(vel);
            qdd_att = (qd_att-qd_prec)/dt;

%save results

end %for
```