UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de Doctorat

AUTOMATIZACIÓ AVANÇADA I ROBÒTICA

Tesi Doctoral

GRASP PLANNING UNDER TASK-SPECIFIC CONTACT CONSTRAINTS

Carlos J. Rosales Gallegos

Directors:
Lluís Ros Giralt and Raúl Suárez Feijóo

Institut d'Organizació i Control de Sistemes Industrials

Novembre de 2012

Universitat Politècnica de Catalunya

Programa de doctorat:
Automatizació Avançada i Robòtica

Aquesta tesi ha estat realitzada a:
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Institut d'Organizació i Control de Sistemes Industrials, UPC

Dirigida per:
Lluís Ros Giralt and Raúl Suárez Feijóo

Realitzada per:
Carlos J. Rosales Gallegos

.

To the memory of Ignacio Fresnedo Lourerio (1972-2011),
a good friend and Mechanical Engineer at heart.

GRASP PLANNING UNDER TASK-SPECIFIC CONTACT CONSTRAINTS

Carlos J. Rosales Gallegos

Abstract

Several aspects have to be addressed before realizing the dream of a robotic hand-arm system with human-like capabilities, ranging from the consolidation of a proper mechatronic design, to the development of precise, lightweight sensors and actuators, to the efficient planning and control of the articular forces and motions required for interaction with the environment. This thesis provides solution algorithms for a main problem within the latter aspect, known as the *grasp planning* problem: Given a robotic system formed by a multifinger hand attached to an arm, and an object to be grasped, both with a known geometry and location in 3-space, determine how the hand-arm system should be moved without colliding with itself or with the environment, in order to firmly grasp the object in a suitable way.

Central to our algorithms is the explicit enforcement of a pre-speceified set of hand-object contact constraints in the grasp configuration, imposed by the particular manipulation task to be performed with the object. This is a distinguishing feature from other grasp planning algorithms given in the literature, where a means of ensuring precise hand-object contact locations in the resulting grasp is usually not provided. These conventional algorithms are fast, and nicely suited for planning grasps for pick-an-place operations with the object, but not for planning grasps required for a specific manipulation of the object, like those necessary for holding a pen, a pair of scissors, or a jeweler's screwdriver, for instance, when writing, cutting a paper, or turning a screw, respectively. To be able to generate such highly-selective grasps, we assume that a number of surface regions on the hand are to be placed in contact with a number of corresponding regions on the object, and enforce the fulfilment of such constraints on the obtained solutions from the very beginning, in addition to the usual constraints of grasp restrainability, manipulability and collision avoidance.

The proposed algorithms can be applied to robotic hands of arbitrary structure, possibly considering compliance in the joints and the contacts if desired, and they can accommodate general patch-patch contact constraints, instead of more restrictive contact types occasionally considered in the literature. It is worth noting, also, that while common force-closure or manipulability indices are used to assess the quality

of grasps, no particular assumption is made on the mathematical properties of the quality index to be used, so that any quality criterion can be accommodated in principle. The algorithms have been tested and validated on numerous situations involving real mechanical hands and typical objects, and find applications in classical or emerging contexts like service robotics, telemedicine, space exploration, prosthetics, manipulation in hazardous environments, or human-robot interaction in general.

Acknowledgements

I would like to express my gratitude to my advisors Lluís Ros and Raúl Suárez for their invaluable guidance. Also, to Josep M. Porta and Jan Rosell for their unconditional support, and for leading the projects of such wonderful frameworks, the CUIK and Kautham suites, in which this thesis relies on. I would like to thank Federico Thomas who, together with Raúl, came with the idea of joining the worlds of grasping and kinematics that resulted in this PhD research. I am thankful to Antonio Bicchi and Marco Gabiccini for letting me accomplish a fruitful research stay in Pisa; Chapter 5 is a joint work with them. I'd like to highlight the work of Alexander Perez, a friend, a co-author, and coder of a large part of the Kautham suite; Montserrat Manubens, Léonard Jaillet, Oriol Bohigas, friends and contributors to the CUIK suite; Patrick Grosch, a friend who provided some of the 3D models used here as well as great ideas; and Leopold Palomo, a friend who introduced me to the open source world and provided technical support. Thus, I would like to acknowledge the work of the open source and free software communities since most of the content of this document has been generated using freely-available tools, such as Ubuntu, Debian, LaTeX, Inkscape, GIMP, Geomview, Blender, among many others, freely-available 3D models, such as the SketchUp hand-arm model used in Figure 1.3 shared by Daniel Murray, and freely-available LaTex templates, such as the PhD thesis style used for this document shared by Adolfo Rodriguez, all of them free as in free beer, and some as in freedom. I would like to acknowledge the reviewing work of numerous researchers who have improved this work with their valuable comments on publications. I'd like to thank my relatives for their support. And last, but not least for sure, I am more than pleased to have met very nice friends and colleagues at the IOC and IRI institutes, as well as countless friends aside who collabarated in different ways and means to the development of this work.

Carlos J. Rosales Gallegos
Barcelona, Catalunya
November 7, 2012

This work has been partially supported by the Spanish Ministry of Education and Science through the contracts:

DPI2007-63665 PROA: Grasping and Repositioning of Objects using Anthropomorphic Dexterous Manipulation: Analytic and Learning Approaches (Leader: R. Suárez).

DPI2007-60858 Analysis and Motion Planning of Complex Robotic Systems (Leader: L. Ros).

DPI2010-15446 MUMA: Multi-hand Systems for Complex Robotized Manipulation Tasks (Leader: R. Suárez).

DPI2010-18449 CUIK++: An Extension of Branch-and-Prune Techniques for Motion Analysis and Synthesis of Complex Robotic Systems (Leader: L. Ros).

Also, by the "Comunitat de Treball dels Pirineus" under contract 2006ITT-10004, and by the Interdepartamental Research Center "E. Piaggio" from Pisa.

Contents

Figures

Tables

Part I

Preliminaries

1 Introduction

The human hand, the most versatile end-effector provided by Nature, can perform such a variety of motions and subtle adjustments that mimicking its abilities mechanically has been qualified as the Holy Grail of robotic end-effectors [9]. This chapter outlines the main motivations underlying the pursuit of such abilities, delimits particular objectives and scope of this Ph.D. work, and summarizes the structure of the dissertation.

1.1 Motivation

When Karel Čapek first introduced the term in the 1920's, robots were conceived as human-like machines. However, the first industrial robots that appeared a few decades later were still far from exhibiting anthropomorphic features or dexterous manipulation abilities. These manipulators allowed more flexibility than hard automation machines for large-scale production, and were able to execute a variety of tasks by programming them, but they still required costly end-effector retoolings for each specific task, and could not accurately perform precise movements of the payload, such as those that might be required in an assembly or fine manipulation task. In the early 1980's, increasing needs of more flexible tools in the industry made robot designers think of more versatile grippers, giving rise to the appearance of the first multifingered end-effectors inspired by the human hand. Since a robot hand has many degrees of freedom, it provides a way to grasp a large class of objects with a single end-effector. Also, it is usually smaller than the arm to which it is attached, which allows improving the overall accuracy of the robot.

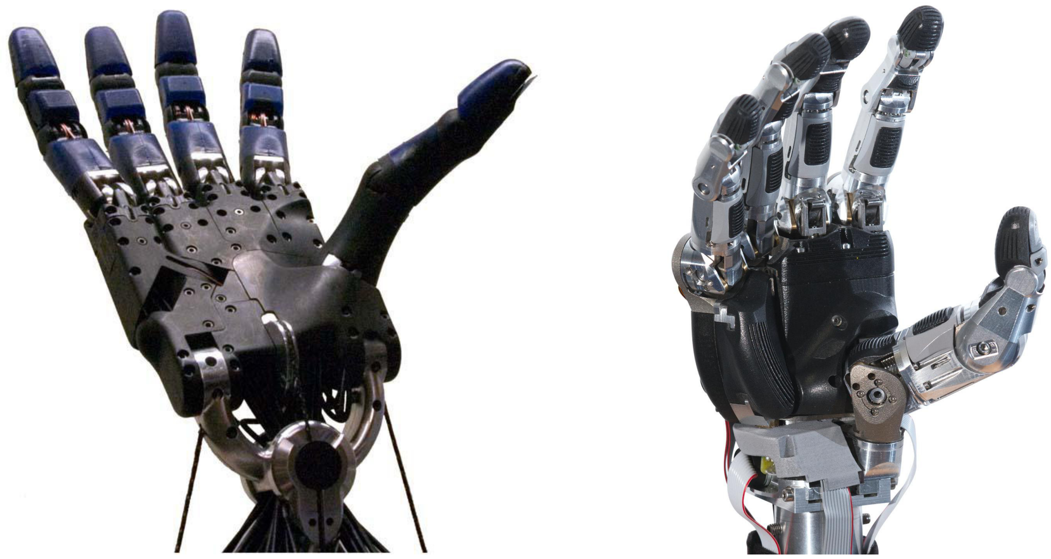


Figure 1.1: Commercially available robotic hands by Shadow Robot Company [107] (left) and Schunk GmbH & Co. KG [106] (right).

The price of using a mechanical hand is the complexity of the overall system. Truly anthropomorphic designs involve from four to five fingers, and up to forty actuators to provide agonist/antagonist motions (Figure 1.1), which complicates the kinematic and dynamic analyses of the system substantially. Since a hand is in contact with the object being manipulated, the kinematics and dynamics of a mechanical system with contact constraints must be analysed, and the increased number of degrees of freedom complicates the planning of a feasible grasp to perform a given task. It is for all of these reasons that an anthropomorphic hand may not be the answer to every manipulation problem. The use of end-effectors tailored to particular tasks can effectively cover many needs in areas such as manufacturing and materials transport. Alternative end-effectors, like classical grippers or underactuated hands, or innovative ones based on the jamming of granular material, can achieve stable grasps using a simpler design (Figure 1.2, top), whereas a similar grasp using a fully articulated hand typically requires the use of feedback control and might, eventually, result in an overall decrease of the grasp stability. Despite these difficulties, however, it seems clear that the vision of ubiquitous robotic assistants, whether in home, the industry, or in space, will not be realized without the ability to grasp and dexterously manipulate usual objects in human environments, for which fully-actuated hands seem to provide the ideal solution (Figure 1.2, bottom). Driven by the potential benefits of such hands, progress towards building hands for practical applications is underway, and the first commercial hands, such as the Shadow [107] and Schunk [106] hands, have recently come to light, allowing to picture out an increasing number of applications in the future, in classical or emerging contexts like service robotics, telemedicine, space exploration, prosthetis, manipulation in hazardous environments, or human-robot interaction.

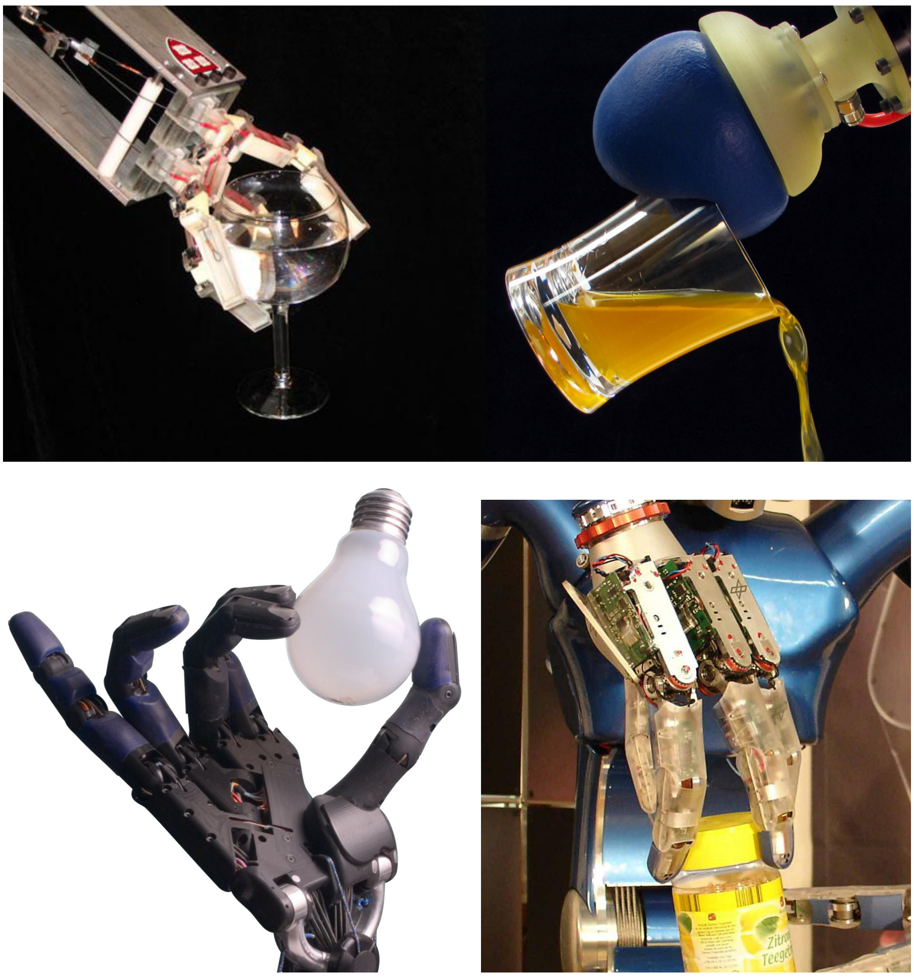


Figure 1.2: Alternative versus anthropomorphic designs. The underactuated hand from the Yale GRAB lab [29] (top-left) and the Universal Robotic Gripper [12] (top-right) perform stable grasps of different glasses using a single actuator. Thanks to their multiple acturators, the Shadow hand [107] (bottom-left) and the Schunk hand [106] (bottom-right) are able to perform more complex grasps, like those requiring the screwing of various objects.

1.2 Objectives and scope

Several aspects have to be addressed before realizing the dream of a robotic hand with human-like capabilities, ranging from the consolidation of a proper mechatronic design, to the development of precise, lightweight sensors and actuators, to the efficient planning and control of the finger forces and motions required for interaction with the envrionment. The aim of this thesis is to provide suitable solutions to a main problem within the latter aspect, known as the *grasp planning* problem: Given a robotic system formed by a multifinger hand attached to an arm, and an object to be grasped, both in a known geometry and location in 3-space, determine how should the hand-arm system be moved without colliding with itself or with the environment, in order to firmly grasp the object in a suitable way, allowing to perform a specific manipulation task (Figure 3.1).

The solution to the grasp planning problem will take the following hypotheses and requirements into account. First, note that many of the objects and tools used on everyday activities are designed to be grasped and manipulated in particular ways. Consider, for instance, how a pen, a scalpel, or a jeweler's screwdriver are held, to properly write on a paper, cut a tissue or turn a screw, respectively. To perform these tasks with a robotic hand, the grasp planner must be able to place the hand in contact with the object at *specific* locations, allowing the manipulation to properly occur. In accordance to such requirement, we will assume that a given set of contact constraints is to be satisfied between predefined regions on the object and hand surfaces, including both the finger and palm surfaces in the latter case. This is a distinguishing feature from other existing algorithms in the literature, such as those embedded in the GraspIt! suite [20, 37, 64] for example, where constraints of such kind are not explicitly enforced (Figure 1.4). Second, while no particular assumption will be made on how the contact constraints are derived—they could be inferred from learned experience through examples [77], automatically generated from contact synthesis algorithms [70, 88] and known patterns of static prehension [47], or directly informed by a human, for instance—these constraints will be assumed to be general patch-patch contact constraints between the hand and object surfaces, instead of the more restrictive contact constraints assumed in the literature, so as to be able to accommodate the endless possible shapes of objects that may be encountered in practice. Third, since robotic

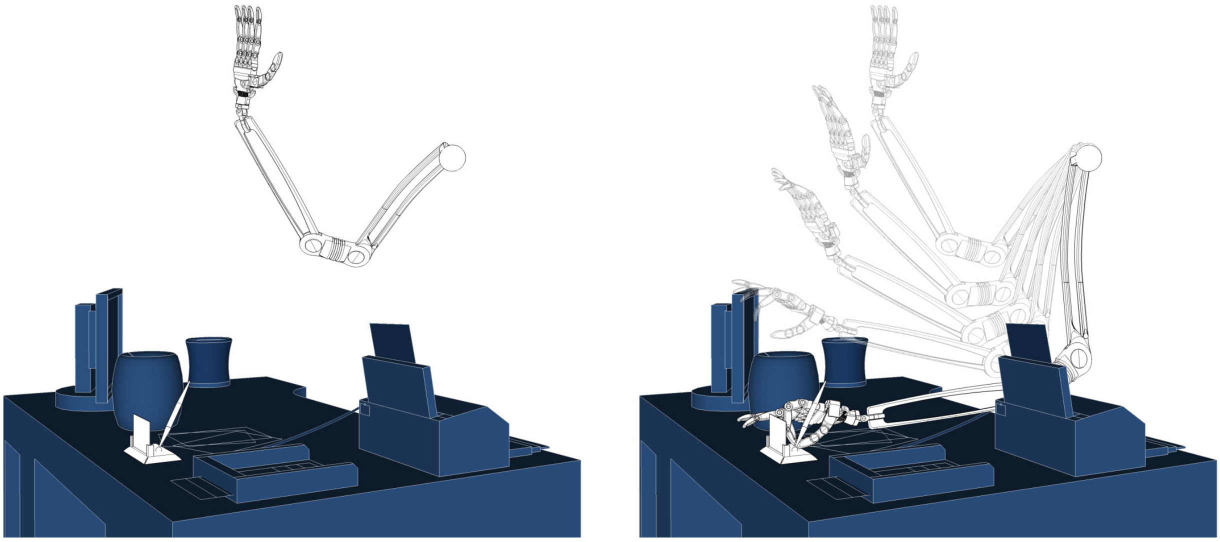


Figure 1.3: Grasp planning: Given a robotic hand-arm system and an object (left), determine how should the system be moved in order to grasp the object in a suitable way, avoiding collision with itself or with the environment (right).

end-effectors may adopt a variety of designs, with a range of kinematic topologies, link geometries, and actuation schemes, we will require our algorithms to be of sufficient generality to be able to apply them to any particular instance of such designs. This will entail adopting general numerical techniques for kinematic constraint solving in their implementation, since they are the only ones that have proved to be powerful enough to solve non-linear systems of equations of the kind that arise. Fourth, we will assume that accurate geometric models of the object, the hand-arm system, and the environment are available, so that the planning of motion paths can safely be made using them.

A divide-and-conquer strategy will be used to overcome the complexity of the grasp planning problem, breaking it up into the following subproblems:

1. **The grasp synthesis problem:** Determine a grasp configuration satisfying a number of hand-object contact constraints, while guaranteeing the manipulability and restrainability of the object at the same time.

2. **The approach path planning problem:** Determine a joint-space path allowing the hand-arm system to place the hand in the previously-obtained grasp configuration, without colliding with itself or with the environment.

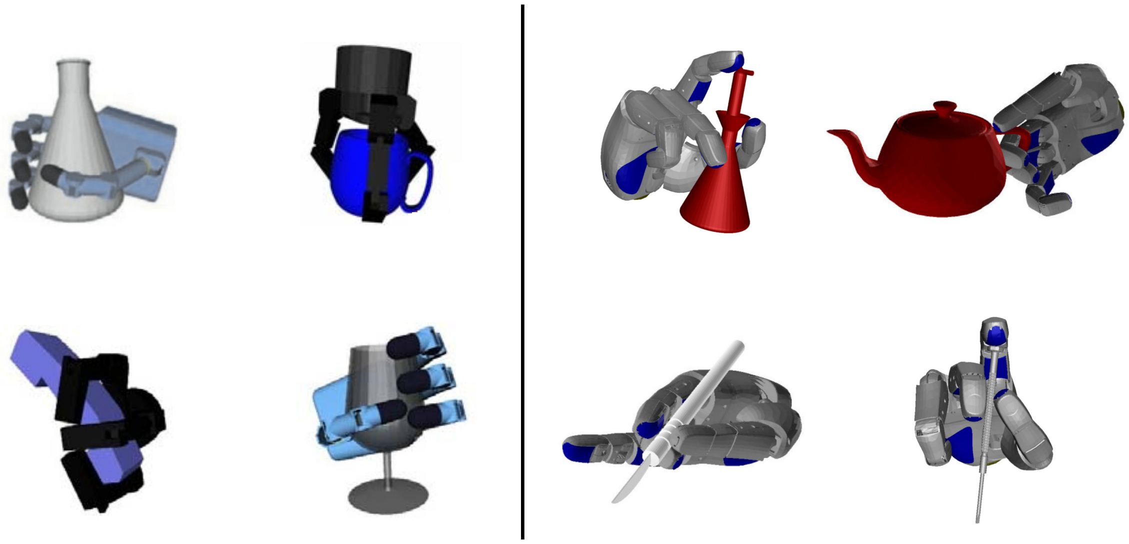


Figure 1.4: While the GraspIt! suite [20, 37, 64] is nicely suited to generate random grasps able to hold an object (left), it cannot produce specific grasps allowing a particular manipulation of the object (right). This PhD work aims at solving the grasp planning problem in the latter context.

A solution to the grasp planning problem will be obtained by devising modules able to solve the previous two subproblems. The grasp synthesis module provides a hand configuration in contact with the object, which can be fed to the approach path planning module to generate a goal configuration for the overall hand-arm system, using standard inverse kinematics techniques for the supporting arm. Once such a goal configuration is available, the latter module can start the search for a joint-space path connecting the current configuration of the system to the mentioned goal configuration.

1.3 Outlook at the dissertation

The dissertation is structured into four parts, which can be outlined as follows:

- **Part I** provides the motivation, objectives and scope of the work (Chapter 1), and encloses a short account on the state of the art on the two topics related to this thesis (Chapter 2).
- **Part II** presents the solution given to the grasp synthesis problem. Initially, an algorithm is developed for solving a relaxed version of the problem. Namely,

the computation of a configuration of the hand-object system that respects all assembly constraints imposed by the hand joints, and all hand-object contact constraints considered (Chapter 3). Next, a second algorithm is developed that, departing from the grasp configuration computed by the previous algorithm, is able to explore the manifold of hand configurations in contact with the object, to find those that are the maximally restrained and manipulable. The technique is general enough to cope with usual force-closure or manipulability metrics, combinations of such metrics, or any other metric that might be defined (Chapter 4). Finally, the fact that both the joints and the hand-object contacts are actually non-rigid is taken into account, and a way to synthesize grasp configurations where such joints and contacts behave according to given compliance models is introduced (Chapter 5). The technique is general enough to tackle the kinematic and restrainability constraints at once, and only requires to be initialized with an estimation of the solution, which can be computed using the techniques developed in the preceding chapters.

- **Part III** presents the solution given to the approach path planning problem. Firstly, a meaningful way to reduce the dimension of the hand configuration space is proposed. The approach draws inspiration from how the human hand moves in the free space to lessen the finger freedom by coordinating the joint movements, and biases the wrist orientation towards the goal by imposing relational positioning constraints (Chapter 6). This reduction is later on exploited to develop an algorithm that solves the approach path planning problem in an adaptive way, adjusting the configuration space dimension where needed, according to the local obstacle complexity encountered at each step (Chapter 7).
- **Part IV**, finally, draws the main conclusions of the thesis, summarizes its contributions, and enumerates points deserving further attention (Chapter 8).

1.4 Glossary

For the sake of clarity, we next define basic terminology used throughout the manuscript.

Hand: Following Salisbury and Craig [103], a hand is defined as a collection of serial kinematic chains acting as fingers, connected to a common base acting as a palm, intended to provide an end-effector of similar versatility to that of the human hand. The number of fingers must be at least three to grant sufficient stability. The fingers must have at least two links acting as phalanges, articulated through lower-pair joints. The number of joints must be at least three per finger, with at least three actuators per finger. The joint distribution must allow the finger to produce abduction/adduction and flexion/extension motions.

Task-specific contact constraint: A contact constraint that must necessarily be fulfilled in order to manipulate the object in a specific way. Such constraints are assumed to be given, and take the form of general patch-patch contact constraints, so that a surface patch defined on the hand is enforced to be in contact with a corresponding patch defined on the object, with the normal vectors to the patches aligned. The hand patches can be defined anywhere on the finger or palm surfaces.

Configuration of the hand-object system: A description of the position and orientation of all bodies involved in the hand and the object, respecting the joint assembly constraints of the hand only. In such a configuration, the hand may or may not be in contact with the object.

Grasp: A particular configuration of the hand-object system in which the hand is in contact with the object, but not necessarily satisfying any of the task-specific contact constraints assumed.

Kinematically-feasible grasp: A grasp that satisfies all of the imposed task-specific contact constraints.

Restrained grasp: A grasp that is able to firmly hold an object, and to control the internal forces applied on the object. A restrained grasp is said to be force- or form-closed, depending on whether such ability relies on frictional contacts or not, respectively. Provided that the number of contacts is at least the minimum required to hold the object, a restrained grasp is obtained when all forces applied on the object are in equilibrium, and in agreement with the frictional model assumed, if any. Various metrics can be defined to assess how far is a restrained grasp from loosing this property.

Manipulable grasp: A grasp that is able to perform controllable object motions along arbitrary directions of its pose space. Again, various metrics can be defined to assess the quality of a manipulable grasp.

Arm: A serial kinematic chain acting as a carrying device for a robotic hand. The arm must be able to move the hand throughout large portions of its pose space, that is, both in position and orientation. Six-revolute arms will be considered for concreteness, but the presented techniques can be applied to general serial arms.

Approach Path: A path defined in the joint space of the hand-arm system, connecting two points corresponding to initial and final configurations of the system. The final configuration is assumed to be one in which the hand is directly in contact with the object, forming a grasp, or very close to realizing such contact, forming a so-called pre-grasp.

2
Related Work

This chapter provides a short account of previous work related to the problems of grasp synthesis and approach path planning dealt with in this Thesis. Along the way, the reader is pointed to related sections of the manuscript where such techniques play an important role, or where an enhancement or alternative to them is given.

2.1 Grasp synthesis

The earliest studies on grasping can be traced back to the 19th century, when Reuleux defined the notions of form- and force-closure grasp [86] in the context fixturing and machine design. A restrained grasp is one that is able to counteract external disturbances applied on the object, and it is force- or form- closed, depending on whether such ability relies on frictional contacts or not, respectively (Figure 2.1). In the frictionless case, Reuleaux [86] and Somov [110] showed that a minimum of four (resp., seven) contact points were necessary to achieve closure of 2D (resp., 3D) objects. About a century later, with the arrival of robot hands, these results were revisited [28, 66], and Markenscoff et al. [62] proved that, under very general conditions, four and seven contact points were also sufficient to respectively generate 2D and 3D form-closed grasps [62]. They also showed that when Coulomb friction is taken into account, three fingers are sufficient in the 2D case, and four in the 3D case. Although these results did not provide constructive procedures for synthesizing proper contact points, they laid the

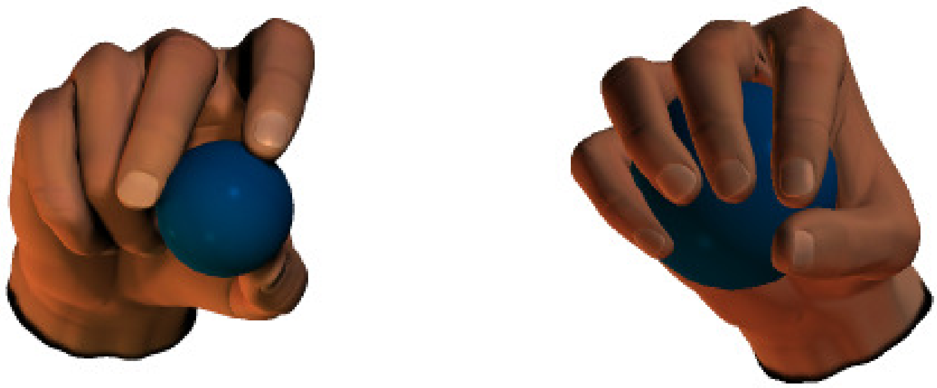


Figure 2.1: Whereas a force-closed grasp relies on friction (left), a form-closed grasp relies on shape to restrain the object (right). [Images courtesy of the Human Grasping Database [31].]

foundations for finding them. From that point, the external wrench applied to the object was related to the forces applied at the contacts through the so-called grasp matrix, and qualitative tests were provided to determine whether a given set of wrenches restrained an object [102], which naturally led to quantitative tests [33, 89, 116]. By using these tests, numerous approaches have been developed for synthesizing contact points in 2D [22, 58, 70, 73] and 3D [10, 56, 77, 78, 87] objects.

In parallel, another issue considered has been the determination of the contact forces and joint torques required to optimally balance a given external force applied on the object [4, 53, 124]. Since the contact forces are related to the joint torques through the hand Jacobian, by adjusting such torques it is possible to achieve a set of contact forces that lie inside the friction cones of the contact points. Assuming that the overall configuration of the hand-object system is known, both the grasp matrix and the hand Jacobian are known, and by linearizing the cone constraints, it is possible to rapidly determine which joint torques allow to grasp the object with maximum stability, by resorting to Linear Programming techniques [18, 49]. Solutions that avoid the latter linearization have been given as well, but at the expense of using more costly non-Linear Programming techniques [13, 16, 122, 125]. The connections of the problem to the force analysis of walking robots [72, 120], multi-arm manipulation systems [54], or tensegrity frameworks [46], have also been investigated.

All of the works just mentioned provide useful insights into key aspects of grasp planning, but the resulting algorithms are of limited applicability in practice, because the hand configuration in contact with the object is either neglected, despite being fundamental to the force-closure concept adapted to grasping [5], or assumed to be given, but never synthesized in any case. Recently, a more comprehensive approach to the problem has been attempted, which emphasizes the computation of hand configurations in contact with the object from the very beginning, in addition to searching for an optimal grasp [20]. The main difficulty in this case is that the set of hand configurations in contact with the object is a complex manifold, implicitly defined by a system of non-linear equations that expresses all joint-assembly and contact constraints involved in the hand-object system. To avoid this complexity, Ciocarlie and Allen [20] initially relax the contact constraints, resulting in a search space that coincides with the configuration space of the hand. Typically, this space is of a high dimension, but meaningful simplifications derived from the use of hand postural synergies [104] can be introduced to narrow the search to a subspace explorable in reasonable times, using simulated annealing. The hand configurations obtained, however, are not exactly in contact with the object and, thus, they must be evaluated with pre-grasp quality indices, which involves purely geometric features only. Unfortunately, a grasp qualified as good using these indices does not always result in a high-quality grasp once the contact with the object is finally enforced using local techniques, or by simply closing the fingers until contact is achieved. The final hand-object contacts, moreover, are not necessarily suitable for a specific task and, hence, the technique is nicely suited to generate random grasps able to hold an object, but not those grasps allowing a particular manipulation of the object.

To generate a specific grasp for a given task, both the joint-assembly constraints of the hand and the contact constraints imposed by the task need to be enforced simultaneously. The problem to be solved to this end can be stated as follows. Given a number of regions on the surface of the hand, and a number of corresponding regions on the surface of the object, determine how the hand should be configured relative to the object, so that each hand region establishes contact with its corresponding object region. This problem has mostly been addressed with local search methods to date. Examples of such methods include those proposed by Borst et al. [11], who cast the problem into one of unconstrained optimization, where the constraints are enforced using penalty terms in the objective function, Gorce and Rezzoug [38], who rely on a neural network to learn the finger inverse kinematics, and on reinforcement learning to optimize the pose of the hand, and Rosell et al. [95], who propose an iterative method to compute joint displacements that maximally reduce the distance from the fingertips to the desired contact points. Such methods are usually fast and return solutions in many cases, but their convergence is not always guaranteed, even if a solution exists. Some of the methods, moreover, require a sufficiently-good initial estimation of the solution [11, 95], which might not always be available.

A way around such limitations will be worked out in Chapter 3, by proposing a new algorithm of guaranteed convergence; i.e., one that always provides a hand configuration satisfying the required contact constraints whenever one exists. This algorithm does not require an initial estimation of the solution and can, in fact, solve a superclass of the configuration problems dealt in [11, 38, 95], because all contact constraints considered in such works can be seen as particular cases of more general ones tractable herein.

Regardless of the adopted method to compute a grasp, it is worth noting that the grasp is usually not optimized in terms of any quality criterion, so that a final optimzation process is needed to obtain a suitable high-quality grasp. Implementing such a process is not trivial though, since trying to optimize the grasp in a generate-and-test fashion is computatinally too expensive, and local optimization methods like [75] are likely to get trapped into local optima because, except in simpler cases [57, 111], grasp quality indices present local extrema. A grasp optimization procedure that avoids such drawbacks will be proposed in Chapter 4, based on tracing a set of relevant grasps exhaustively using a higher-dimension continuation technique. A rigid-grasp model is assumed in such a procedure, but a way to take grasp compliance into account will be given in Chapter 5.

2.2 Approach path planning

When planning the motions of an articulated system, the common practice is to compute some representation of the collision-free regions of the configuration space of the system, using e.g. cell-decompositions, potential field methods, or roadmap techiques, and then search for a valid path between the start and goal configurations using this representation. The application of such techniques to the approach path planning problem is not straightforward however, because of the high number of joints involved in an anthropomorphic hand-arm system, and the fact that the goal configuration is either very close to the obstacle region, or directly on its border. A main trend since the early attacks to the problem, thus, has been to simplify the problem in some way or another, either by using simpler hands or restricted actuation schemes, or by assuming favorable uncluttered environments. Nevertheless, the increase of computational power over the years, and the appearance of succesful probabilistic roadmap methods, has allowed to progressively drop some of these simplifications, gradually increasing the range of solvable problem instances (Figure 2.2).

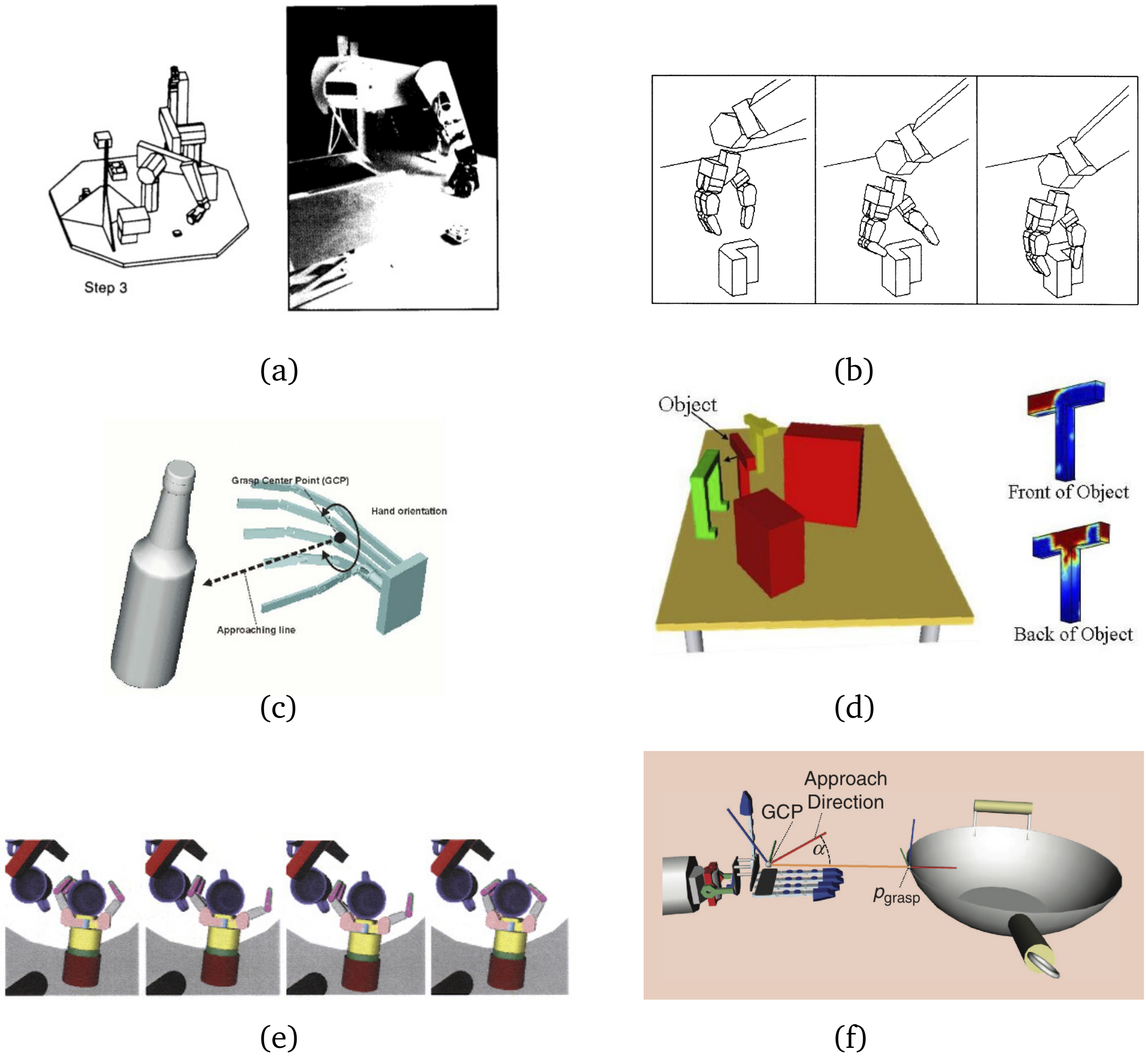
One of the earliest works on the problem was that by Lozano-Perez et al. [61] on the Handey system, which involved a classical one-degree-of-freedom gripper mounted on a six-revolute arm, aimed at performing pick-and-place operations for assembly tasks. In this system, the planning of an approach path to the object was subdivided into two-stages. First, using a cell-decomposition method, a path involving gross motions of the arm was computed during a global stage in order to bring the gripper in close proximity to the object. Then, a simple path taking the system to the final grasp was generated during a local stage, using a potential field method. Although several simplifying assumptions were made in both stages —including planar-faced objects, motions involving only three joints, and a relatively uncluttered environment around the object— the Handey system was taking many aspects of the problem into consideration, and many subsequent works are still reminiscent of the two-stage strategy adopted. The work in [61] was extended by Pollard [76] later on, replacing the gripper by the more complex, three-fingered Salisbury hand. The planning of the path was done under similar assumptions, using cell-decomposition and potential field methods for the global and local stages, but more involved grasp synthesis methods were introduced to account for the increased complexity and reconfiguration capacity of the newer hand. The grasp configuration was synthesized by choosing proper contact points for the fingertips, together with a feasible pose for the wrist heuristically, and an attempt to connect this configuration to an intermediate configuration at a given distance from the object was then made using a straight-line path possibly deformed locally, so that the latter configuration could be easily reached by the global planner.

The local problem of computing a motion path from a pre-grasp configuration to one in contact with the object was considered by Kragic et al. [50]. In this work, the pre-grasp configuration is provided by a human guiding the hand wrist, and the GraspIt! system is used to evaluate whether a closing of the fingers can yield a sufficiently good grasp of the object. The objects are again plane-faced, and, strictly speaking, there is no planning of the approach path to the object, since the pre-grasp configuration is informed, and the closing of the fingers is deterministic. However, Morales et al. [67] extended this approach to account for more complex objects and hands, alleviating the need of informing manually the system with pre-grasp configurations. In their case, the pre-grasp configuration is automatically selected from a database of previously-computed pre-grasps, and the final approach path to the object, taken as well from the database, follows a straight-line pattern similar to the one used by Pollard [76], but allowing the rotation of the wrist about the path if necessary. In simulation, the planner iteratively translates the hand along the path, and possibly rotates it, checking along the way whether the resulting hand configuration can properly reach the object by closing the fingers.

Aiming to deal with more cluttered environments, Berenson et al. [2] propose to rank all possible approach paths a priori before trying them, using a score function that quantifies the promise of the path to yield a high-quality, collision-free grasp in the end. The paths considered are identical to those tried by Morales et al. [67] but, once one of them is found to be valid, the system is able to plan a motion from the start configuration of the hand-arm system, to the pre-grasp configuration determined, using bi-directional Rapidly-exploring Random Trees (RRTs). The hand configuration is kept constant during such planning in order to grow the trees in six-dimensional joint spaces only. Berenson and Srinivasa [3] present a refinement of the technique in which all degrees of freedom of the hand are allowed to move in a coordinated way during the exploration of the final approach path, adding one additional degree of freedom that increases the chances of succeeding along the path.

A slight twist to these approaches has been recently introduced by Vahrenkamp et al. [118], in an attempt to integrate the grasp synthesis and approach path planning phases to the largest possible extent. In their work, a global planner iteratively grows an RRT from the initial configuration of the hand-arm system, selecting one of its nodes randomly from time to time for exploration towards the object. This exploration, which is similar to that in previous approaches, consists in trying a linear path until a threshold position around the object is reached, then closing the fingers until contact is established, and finally evaluating the grasp using typical grasp quality metrics.

Despite the significant advances to the problem, three principal weaknesses can be identified in all previous works. First, note that although all methods may perform well in their particular contexts of application, there is little or no control on whether the



(a) (b)

(c) (d)

(e) (f)

Figure 2.2: Grasp planners of increasing complexity proposed over the years. (a) The Handey system by Lozano-Perez et al. [61], which assumes a classical gripper. (b) The extension considered by Pollard [76], which changes the gripper by the Salisbury hand and considers possibly-deformed straight-line approach paths. (c) Morales et al. [67] introduces rotations about such paths. (d) Berenson et al. [2] rank the paths a priori before trying them, according to a score function considering grasp quality and potential collisions. (e) Berenson and Srinivasa [3] add finger-coordinated motions to increase the chances of finding a collision-free path. (f) Vahrenkamp et al. [118] integrate grasp synthesis and approach path planning under a randomized approach.

final grasp configuration will be suitable to a particular manipulation task involving the object. In most grasp planning systems the hand is simply closed from a pre-grasp configuration, so that it is difficult to predict in which precise object regions will the fingers finally land. Second, the overall grasp planner is usually a combination of global and local planners whose domain of application is heuristically defined with rule-of-thumb conditions that seem artificial to the problem. Third, the hand-arm system is usually set to move under simplified motion paths only, which restricts the range of possible movements, and hence the ability to find proper solutions in complex situations with cluttered environments.

An alternative planner overcomming such limitations is proposed in Chapters 6 and 7, based on the idea of introducing principal motion directions and wrist orientation constraints. As it will be shown, such a planner is able to drive the hand-arm system to a pre-grasp or a final kinematically-feasible grasp by increasing or decreasing the mobility of the system as needed, according to the local complexity of the obstacle region encountered.

Part II

Grasp Synthesis

3
Finding Grasp Configurations

This chapter presents a method to compute a kinematically-feasible grasp, i.e., one that satisfies a given collection of hand-object contact constraints. In contrast to previous algorithms given for the same purpose, the one presented here allows specifying such contact constraints between free-form surface patches on the hand and object surfaces, and always returns a solution whenever one exists. The method is based on formulating the problem as a system of polynomial equations, and then exploiting the special form of the equations to isolate the solutions, using a numerical technique based on linear relaxations. Although explained for anthropomorphic hands, the approach is general, in the sense that it can be applied to any grasping mechanism with lower-pair joints. The approach can also accommodate as many hand-object contacts as required, involving any of the hand links, including the palm.

Given a number of regions on the surface of the hand, and a number of corresponding regions on the surface of the object, our goal is to determine how should the hand be configured relative to the object so that each hand region establishes contact on its corresponding object region (Figure 3.1). To see the constraints involved into such a problem, we first describe the kinematic structure of common anthropomorphic hands, and how the hand-object contact constraints are assumed to be mathematically specified. We then show that, reflecting such constraints, the problem can be posed as one of solving a system of polynomial equations, for which a complete method able to find a solution, if one exists, is proposed. We finally illustrate the performance of this method on the particular case of the Schunk Anthropomorphic hand (SA) on various tasks requiring an object to be grasped in a special way. The method will be useful in Chapter 4 to synthesize an initial grasp from which to start an optimization process, aimed at obtaining high-quality, restrained and manipulable grasps.

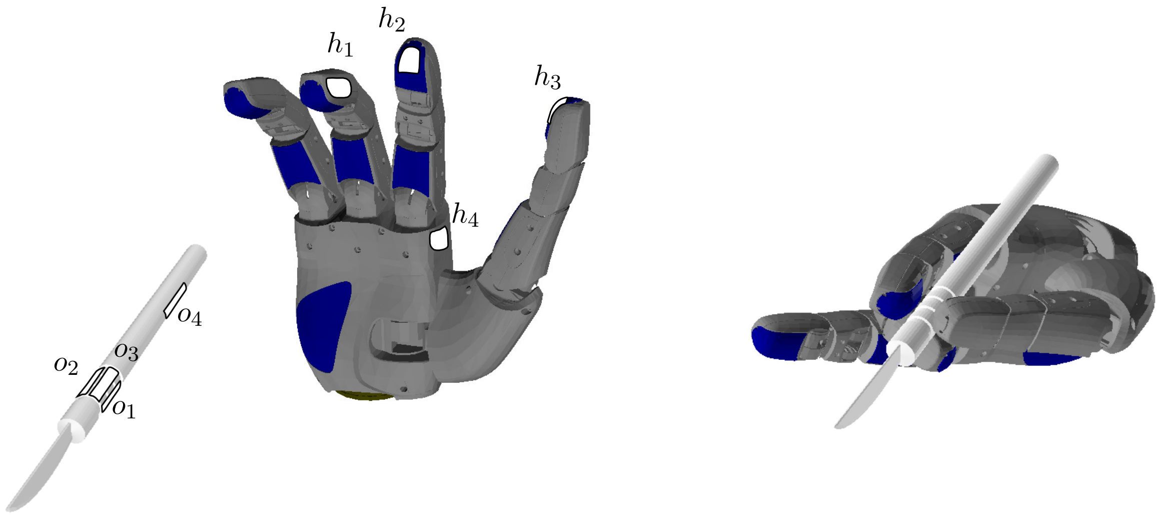


Figure 3.1: A typical grasp for a scalpel can be specified by requiring the contact of regions $h_1, \ldots, h_4$ of the hand, with regions $o_1, \ldots, o_4$ on the object (left). The problem is to determine how should the hand be configured relative to the object, in order to bring the hand regions into contact with their corresponding object regions (right).

3.1 The hand-object system

3.1.1 Structure of an anthropomorphic hand

Although each anthropomorphic hand follows a particular design, all hands are in general made up of a palm and several fingers, one of them acting as the thumb. Usually, all fingers are aligned with each other and with the palm, except the thumb, which is mounted asymmetrically so that it can push against the other fingers. Each finger is composed of several phalanges, usually articulated through revolute (R) or universal (U) joints, whose freedom may be actuated, underactuated, or coupled to those of other joints. Mechanical limitations usually exist, that constrain these joints to take values within prescribed ranges.

Many finger designs follow an URR structure or slight variations of it (Figure 3.2). This structure closely mimics that of the human finger [69]. It mounts a universal joint at the finger base, to model the metacarpophalangeal joint, and two additional revolute joints, to model the proximal and distal interphalangeal joints. The axis of the U joint that is fixed to the palm is responsible for abduction/adduction movements, and the remaining axes, which are usually parallel, are responsible for flexion/extension movements of the finger. The thumb structure is more diverse and controversial [36, 119]. Designs are found where the thumb adopts the same structure as that of the remaining fingers, which facilitates the construction of the hand, like in the SA hand, for instance. Other designs either decrease or increase the mobility of the thumb, by removing or adding joints with respect to the basic URR design. In all cases, however, the tip of the thumb is allowed to face all other fingertips, so as to be able to grasp and manipulate objects under stable prehensions.

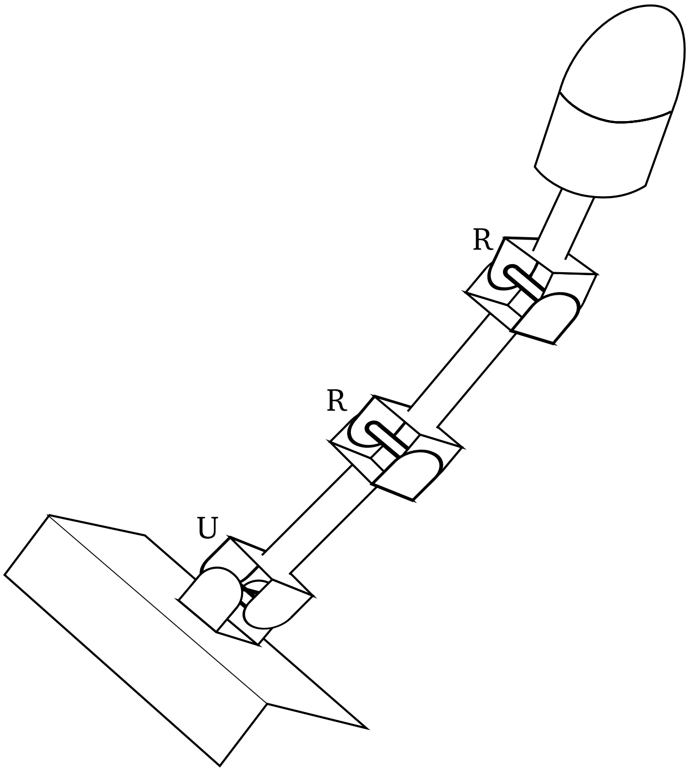


Figure 3.2: The URR structure of a finger.

A summary of representative hand designs adopted during the last decade is provided in Table 3.1. In order to reduce the number of actuators, note that many hands have coupled degrees of freedom. The coupling of two joints A and B is indicated as $\overset{\frown}{AB}$ in Table 3.1, meaning that a rotation about an axis of A produces a proportional rotation about an axis of B. On a $\overset{\frown}{UR}$ coupling only the parallel axes are coupled. Such couplings can be implemented in software or in hardware, with their respective advantages and disadvantages. In the first case, the idea is to reduce the computations in the planning/control stages without losing dexterity, since the coupling can easily be switched off if further dexterity is required. Implementing the couplings in hardware has the benefit pf reducing the number of actuators but at the cost of compromising dexterity [34, 82]. A natural way to implement these couplings in any of the two cases is through the use of postural synergies [104]. Such synergies are obtained by studying human grasps quantitatively, and picking up the most useful coupling patterns for a given set of grasps of common objects. It is worth noting that this is a different concept from real underactuation [9].

	#Actuated	Finger designs				
Hand	**d.o.f.**	Little	Ring	Middle	Index	Thumb
DIST hand [1998]	16	-	URR			
Robonaut hand [1999]	12	$\text{R}\overset{\frown}{\text{R}}\overset{\frown}{\text{R}}\text{R}$		$\text{U}\overset{\frown}{\text{R}}\text{R}$		$\overset{\frown}{\text{U}}\text{R}$
LMS hand [2001]	16	-	URR			
Ultralight Anthropom. hand [2001]	10	$\text{R}\overset{\frown}{\text{R}}\text{R}$				$\overset{\frown}{\text{U}}\overset{\frown}{\text{R}}\text{R}$
GIFU II hand [2002]	16	$\text{U}\overset{\frown}{\text{R}}\text{R}$				URR
Shadow Robot hand [2003]	18	$\text{RU}\overset{\frown}{\text{R}}\text{R}$	$\text{U}\overset{\frown}{\text{R}}\text{R}$			UUR
DLR II hand [2004]	13	-	$\text{U}\overset{\frown}{\text{R}}\text{R}$			$\text{RU}\overset{\frown}{\text{R}}\text{R}$
UBH 3 hand [2004]	20	URR				
MA-I hand [2005]	16	-	URR			
SA hand [2012]	13	-	$\text{U}\overset{\frown}{\text{R}}\text{R}$			$\text{RU}\overset{\frown}{\text{R}}\text{R}$
Twendy-One hand [2009]	13	-	$\text{U}\overset{\frown}{\text{R}}\text{R}$			RUR

Table 3.1: Representative hand designs adopted during the last decade.

3.1.2 Contact constraint specification

The contact constraints are assumed to be given as a collection of pairs (h_c, o_c), for $c = 1, \ldots, b$ contacts, where h_c and o_c are two-dimensional regions on the hand and object surfaces, respectively. The constraint (h_c, o_c) is meant to require the contact of h_c and o_c at some point, with the normals to h_c and o_c aligned at such point, to avoid the interpenetration of the regions (Figure 3.3).

By convention, h_c and o_c are assumed to be given as polynomial patches. That is, it is assumed that a polynomial function of the form

$$\mathbf{p} = \mathbf{p}(u, v), \tag{3.1}$$

is given for each region, providing the parametric coordinates $\mathbf{p} = (p_x, p_y, p_z)$ of a point P in the region, in terms of some scalar parameters u and v, bound to lie within the interval $[0, 1]$. To properly align the normals of h_c and o_c, the parameterization $\mathbf{p}(u, v)$ is supposed to be non-degenerate, in the sense that, if $\mathbf{p}_u$ and $\mathbf{p}_v$ are the partial derivatives of $\mathbf{p}(u, v)$ with respect to u and v, then the normal vector to the patch, defined as

$$\mathbf{n} = \mathbf{p}_u \times \mathbf{p}_v, \tag{3.2}$$

never vanishes for $(u, v) \in [0, 1] \times [0, 1]$.

For ease of explanation, $\mathbf{p}(u, v)$ will adopt the form of a standard Bézier patch of some given degree $M \times N$,

$$\mathbf{p}(u, v) = \sum_{i=0}^{M} \sum_{j=0}^{N} \mathbf{b}_{i,j} \cdot B_{i,M}(u) \cdot B_{j,N}(v), \tag{3.3}$$

where $\mathbf{b}_{i,j}$ denote the Bézier control points of the patch, and $B_{i,j}(x) = \binom{j}{i} x^i (1 - x)^{j-i}$ is the ith Bernstein polynomial of degree j. Note that any polynomial paramaterization $\mathbf{p}(u, v)$ can be converted into such form, by using an appropriate change of basis [30].

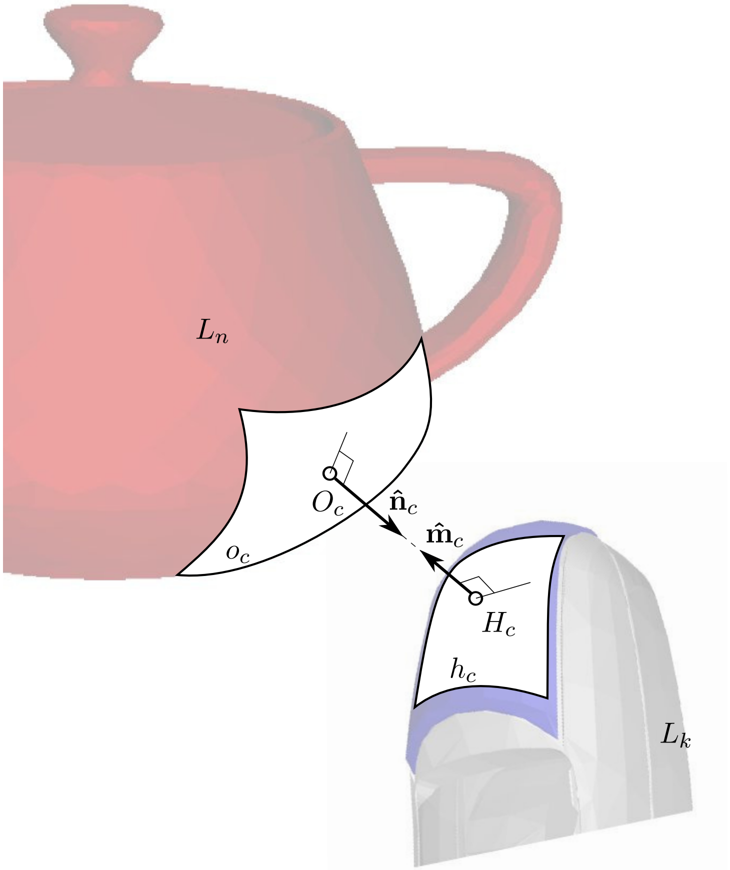


Figure 3.3: Elements intervening in a contact constraint (h_c, o_c). The constraint is satisfied when points $O_c \in o_c$ and $H_c \in h_c$ coincide, with the normals on such points aligned.

3.2 Kinematic equations

A kinematically-feasible grasp can be formulated as a number of constraints that the poses of the hand and object links must fulfill. This section formulates such constraints mathematically, following the methodology proposed by Porta et al. [80]. Once gathered together, the constraints form a system of polynomial equations characterizing all possible solutions of the problem. The special structure of this system will be beneficial to solve the problem numerically, as it will be shown in Section 3.3.

3.2.1 Link constraints

It will be convenient to label the hand and object links as $L_0, L_1, \ldots, L_n$, where L_0 is the palm link, $L_1, \ldots, L_{n-1}$ are the various phalange links, and L_n is the object link. The joints of the hand will also be labelled for reference, as $J_1, \ldots, J_m$.

Each link L_l, $l = 0, \ldots, n$, will be furnished with a local reference frame $\mathcal{F}_l$, and we will let the reference frame of the palm link, $\mathcal{F}_0$, to act as the absolute frame. Moreover, each frame will have an associated vector basis, and we will write $\mathbf{v}^{\mathcal{F}_l}$ to refer to the coordinates of vector $\mathbf{v}$, written in the basis of $\mathcal{F}_l$. Vectors with no superscript will either be expressed in the basis of the absolute frame, or in no particular frame, depending on the context.

With the previous notation, a *configuration* of the hand-object system will be an assignment of a *pose* $(\mathbf{r}_l, \mathbf{R}_l)$ to each link L_l, $l = 1, \ldots, n$, where $\mathbf{r}_l \in \mathbb{R}^3$ is the position of the origin of $\mathcal{F}_l$ with respect to $\mathcal{F}_0$, and $\mathbf{R}_l \in SE(3)$ is a 3×3 rotation matrix giving the orientation of $\mathcal{F}_l$ relative to $\mathcal{F}_0$. The elements of the rotation matrices are not independent, because if $\mathbf{R}_l$ has the form $(\hat{\mathbf{c}}_l, \hat{\mathbf{d}}_l, \hat{\mathbf{e}}_l)$, then it must be

$$\|\hat{\mathbf{c}}_l\|^2 = 1, \tag{3.4}$$

$$\|\hat{\mathbf{d}}_l\|^2 = 1, \tag{3.5}$$

$$\hat{\mathbf{c}}_l \cdot \hat{\mathbf{d}}_l = 0, \tag{3.6}$$

$$\hat{\mathbf{c}}_l \times \hat{\mathbf{d}}_l = \hat{\mathbf{e}}_l, \tag{3.7}$$

for $l = 1, \ldots, n$, in order for $\mathbf{R}_l$ to represent a valid rotation. Note that the joints, the contacts, and the mechanical limits impose additional constraints on the link poses. These constraints are next formulated explicitly.

3.2.2 Joint assembly constraints

Since most hand designs only resort to revolute or universal joints (Table 3.1), we focus on formulating the constraints imposed by such joints, but other joint types would be formulated in a similar way [80].

In terms of spatial constraints, the assembly of two links L_j and L_k, through a revolute joint J_i, is equivalent to imposing the coincidence of two points, P_i and Q_i, and the alignment of two unit vectors, $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$, respectively fixed to L_j and L_k (Figure 3.4a). These two points and vectors are chosen on the axis of the joint, and they

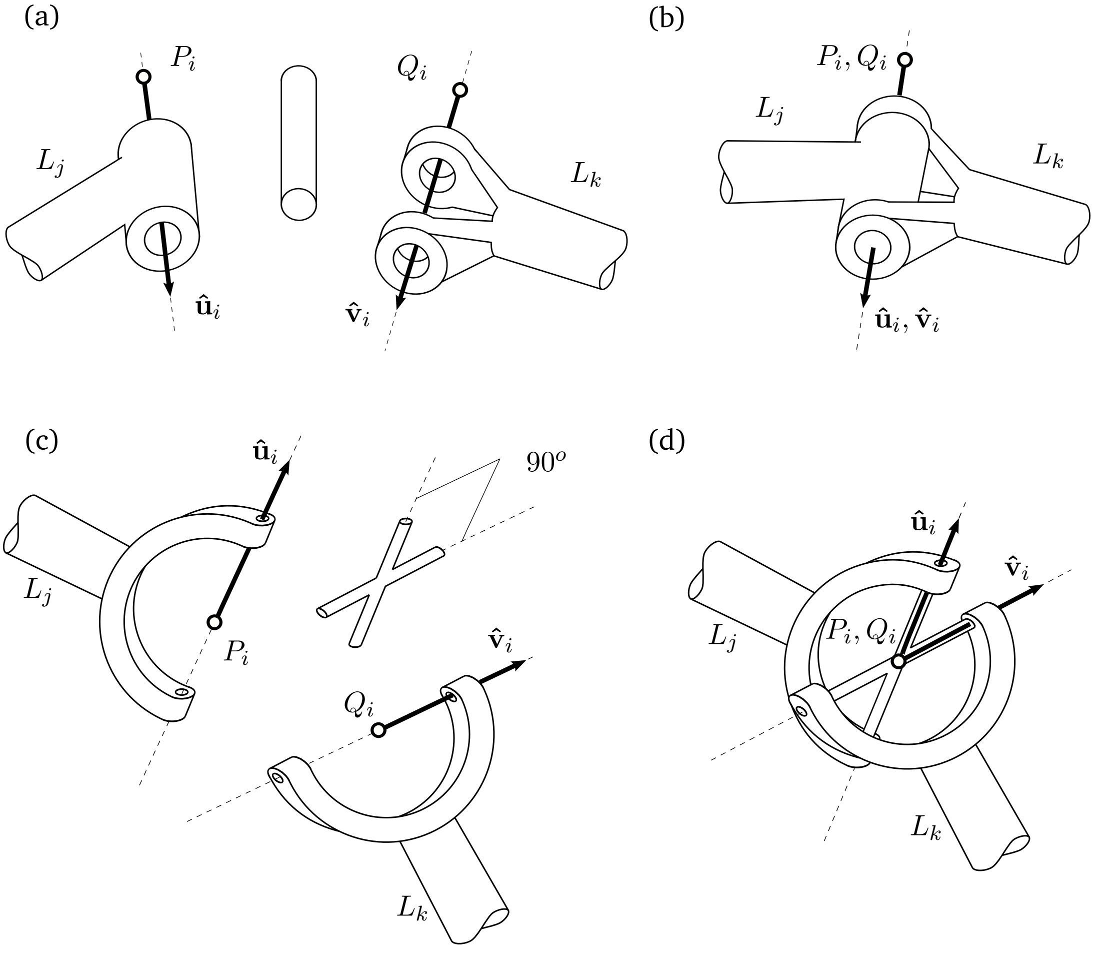


Figure 3.4: (a,b): The assembly of two links through a revolute joint is specified by imposing the coincidence of two points and the alignment of two vectors. (c,d): The assembly through a universal joint is specified by imposing the coincidence of two points and the orthogonality of two vectors. [Figure adapted from [80].]

coalesce into a single point and vector when the two links get assembled (Figure 3.4b). The coincidence and alignment conditions can be written, respectively, as

$$\mathbf{r}_j + \mathbf{R}_j\, \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k\, \mathbf{q}_i^{\mathcal{F}_k}, \tag{3.8}$$

$$\mathbf{R}_j\, \hat{\mathbf{u}}_i^{\mathcal{F}_j} = \mathbf{R}_k\, \hat{\mathbf{v}}_i^{\mathcal{F}_k}, \tag{3.9}$$

where $\mathbf{p}_i^{\mathcal{F}_j}$ and $\mathbf{q}_i^{\mathcal{F}_k}$ refer to the position vectors of P_i and Q_i in frames $\mathcal{F}_j$ and $\mathcal{F}_k$, respectively. The valid poses of the two links, hence, are those that fulfill Eqs. (3.8) and (3.9) simultaneously.

Similarly, if J_i is a universal joint, the valid poses of L_j and L_k are those that fulfill

$$\mathbf{r}_j + \mathbf{R}_j\, \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k\, \mathbf{q}_i^{\mathcal{F}_k}, \tag{3.10}$$

$$\mathbf{R}_j\, \hat{\mathbf{u}}_i^{\mathcal{F}_j} \cdot \mathbf{R}_k\, \hat{\mathbf{v}}_i^{\mathcal{F}_k} = 0, \tag{3.11}$$

where Eqs. (3.10) and (3.11) impose the coincidence of two points P_i and Q_i, and the orthogonality of two unit vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$, respectively fixed on L_j and L_k. The points are located on the center of the universal joint, on $\mathcal{F}_j$ and $\mathcal{F}_k$. The vectors are aligned with the axes of the joint on such frames (Figures 3.4c and 3.4d). Since vectors $\mathbf{p}_i^{\mathcal{F}_j}$, $\mathbf{q}_i^{\mathcal{F}_k}$, $\hat{\mathbf{u}}_i^{\mathcal{F}_j}$, and $\hat{\mathbf{v}}_i^{\mathcal{F}_k}$ are known a priori, the only unknowns in Eqs. (3.8)-(3.11) are the poses of the two links $(\mathbf{r}_j, \mathbf{R}_j)$ and $(\mathbf{r}_k, \mathbf{R}_k)$.

3.2.3 Joint limit constraints

For a revolute joint J_i incident to links L_j and L_k, the relative angle between L_j and L_k, denoted ϕ_i, is the angle between two unit vectors $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$ orthogonal to the axis of J_i, fixed in L_j and L_k, respectively. Usually, due to the existence of mechanical limits, ϕ_i can only take values within a prescribed interval which, using a proper location for $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$, can always be written in the form $[-\alpha_i, \alpha_i]$, with $\alpha_i \in [0, \pi]$. In our formulation, these limits can be taken into account by constraining the cosine of ϕ_i. For this, we define a new variable $c_i = \cos(\phi_i)$, and observe that the constraint $\phi_i \in [-\alpha_i, \alpha_i]$ is equivalent to the constraint $c_i \in [\cos\alpha_i, 1]$. Then we note that

$$c_i = \hat{\mathbf{a}}_i \cdot \hat{\mathbf{b}}_i, \tag{3.12}$$

where

$$\hat{\mathbf{a}}_i = \mathbf{R}_j \, \hat{\mathbf{a}}_i^{\mathcal{F}_j}, \tag{3.13}$$

$$\hat{\mathbf{b}}_i = \mathbf{R}_k \, \hat{\mathbf{b}}_i^{\mathcal{F}_k}. \tag{3.14}$$

Thus, to constrain ϕ_i to the range $[-\alpha_i, \alpha_i]$ it is only necessary to add Eqs. (3.12)-(3.14) to the system to be solved, taking into account that c_i can only take values in the range $[\cos\alpha_i, 1]$. Joint limits for a universal joint can be imposed in a similar way.

3.2.4 Contact constraints

Let us suppose that in the required grasp some hand link L_k is required to be in contact with the object link L_n, where the contact has to be established between given regions h_c and o_c defined on L_k and L_n, respectively (Figure 3.3). Let $H_c \in h_c$ and $O_c \in o_c$ be two points on such regions, with position vectors $\mathbf{h}_c^{\mathcal{F}_k}$ and $\mathbf{o}_c^{\mathcal{F}_n}$ relative to $\mathcal{F}_k$ and $\mathcal{F}_n$, respectively, and let $\hat{\mathbf{m}}_c$ and $\hat{\mathbf{n}}_c$ denote unit normal vectors to the link surface at such points. Then, the poses of L_k and L_n that bring the two regions in contact through H_c and O_c are those that fulfill

$$\mathbf{r}_k + \mathbf{R}_k \, \mathbf{h}_c^{\mathcal{F}_k} = \mathbf{r}_n + \mathbf{R}_n \, \mathbf{o}_c^{\mathcal{F}_n}, \tag{3.15}$$

$$\mathbf{R}_k \, \hat{\mathbf{m}}_c^{\mathcal{F}_k} = -\mathbf{R}_n \, \hat{\mathbf{n}}_c^{\mathcal{F}_n}, \tag{3.16}$$

where Eq. (3.15) imposes the coincidence of H_c and O_c, and Eq. (3.16) establishes the alignment of $\hat{\mathbf{m}}_c$ and $\hat{\mathbf{n}}_c$.

All vectors and matrices in Eq. (3.15) are unknowns. However, since H_c and O_c are bound to lie on h_c and o_c, the additional constraints

$$\mathbf{h}_c^{\mathcal{F}_k} = \mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c), \tag{3.17}$$

$$\mathbf{o}_c^{\mathcal{F}_n} = \mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c), \tag{3.18}$$

must be taken into account to properly formulate the contact, where $\mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c)$ and $\mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c)$ are parametric descriptions of regions h_c and o_c, given in the form of Eq. (3.3). Note that the Bézier control points of the patches $\mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c)$ and $\mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c)$ must be given in frames $\mathcal{F}_k$ and $\mathcal{F}_n$, respectively.

Analogously, the unit vectors $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ in Eq. (3.16) must also be related to the patch parameters. This relationship can be established by taking into account that, for a parametric patch $\mathbf{p}(u,v)$ of the form of Eq. (3.3), the normal vector $\mathbf{n}(u,v)$ defined by Eq. (3.2) can be written as

$$\mathbf{n}(u,v) = \sum_{i=0}^{2M-1} \sum_{j=0}^{2N-1} \mathbf{b}'_{i,j} \cdot B_{i,2M-1}(u) \cdot B_{j,2N-1}(v), \tag{3.19}$$

so that it can be thought of as a new Bézier patch, but now of degree $(2M-1)\times(2N-1)$. Explicit formulas for computing the control points $\mathbf{b}'_{i,j}$ in this expression, in terms of the control points $\mathbf{b}_{i,j}$ of $\mathbf{p}(u,v)$, are given by Yamaguchi [123]. Thus, $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ can be related to the patch parameters by defining two unnormalized vectors $\mathbf{m}_c^{\mathcal{F}_k}$ and $\mathbf{n}_c^{\mathcal{F}_n}$, and their norms μ_c and ν_c, placed in correspondence with $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ through the constraints

$$\mu_c^2 = \|\mathbf{m}_c^{\mathcal{F}_k}\|^2, \tag{3.20}$$

$$\nu_c^2 = \|\mathbf{n}_c^{\mathcal{F}_n}\|^2, \tag{3.21}$$

$$\mathbf{m}_c^{\mathcal{F}_k} = \mu_c \hat{\mathbf{m}}_c^{\mathcal{F}_k}, \tag{3.22}$$

$$\mathbf{n}_c^{\mathcal{F}_n} = \nu_c \hat{\mathbf{n}}_c^{\mathcal{F}_n}, \tag{3.23}$$

and setting the additional constraints

$$\mathbf{m}_c^{\mathcal{F}_k} = \mathbf{m}_c^{\mathcal{F}_k}(u_c, v_c), \tag{3.24}$$

$$\mathbf{n}_c^{\mathcal{F}_n} = \mathbf{n}_c^{\mathcal{F}_n}(s_c, t_c), \tag{3.25}$$

whose right-hand sides follow the form of Eq. (3.19).

3.2.5 Final system of equations

Summarizing, the final system of equations defining the possible grasp configurations will be formed by Eqs. (3.4)-(3.7) for each link, Eqs. (3.8) and (3.9) for each revolute joint, Eqs. (3.10) and (3.11) for each universal joint, equations of the form of (3.12)-(3.14) for each joint limit constraint, and Eqs. (3.15)-(3.18) and (3.20)-(3.25) for each contact constraint. Note that the variables involved in this system are:

- The pose variables $(\mathbf{r}_l, \mathbf{R}_l)$ corresponding to links L_l, $l = 1, \ldots, n$.
- The variables $\hat{\mathbf{a}}_i$, $\hat{\mathbf{b}}_i$, and c_i corresponding to the joint limit constraints on all joints J_i, $i = 1, \ldots, m$.
- The contact point coordinates $\mathbf{h}_c^{\mathcal{F}_k}$ and $\mathbf{o}_c^{\mathcal{F}_n}$, associated normal vectors $\mathbf{m}_c^{\mathcal{F}_k}$, $\mathbf{n}_c^{\mathcal{F}_n}$, $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$, $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$, vector norms μ_c and ν_c, and parameters u_c, v_c, s_c, and t_c, corresponding to all contact constraints (h_c, o_c), $c = 1, \ldots, b$.

It is worth mentioning that the $\mathbf{r}_l$ variables of this system can actually be eliminated through a process explained in detail in [80]. The elimination is based on the fact that, for a loop of links pairwise constrained by joint or contact constraints, Eqs. (3.8), (3.10), and (3.15) occurring along the loop can be substituted by an equivalent "loop-closure" equation which is their sum, which does not contain any of the $\mathbf{r}_l$ variables. This process simplifies the system, and can always be invoked if desired, but the numerical method that follows is equally applicable to both the original and the simplified system.

3.3 Numerical Solution

Let n_e and n_v be, respectively, the number equations and variables of the final system described in Section 3.2.5. This system can be compactly written as

$$\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}, \tag{3.26}$$

where $\mathbf{q} = (q_1, \ldots, q_{n_v})$ refers to a vector encompassing all of its variables, and $\mathbf{\Phi}$ is a polynomial function from $\mathbb{R}^{n_v}$ to $\mathbb{R}^{n_e}$ describing its equations. We now wish to develop a method to find a point $\mathbf{q}$ satysfying Eq. 3.26, whenever one exists.

Several strategies have been given to solve systems of equations of this kind. For example, algebraic-geometric methods, including those based on resultants and Gröbner bases, use variable elimination to reduce the initial system to a univariate polynomial, so that the roots of such polynomial, once backsubstituted into other equations, yield all solutions of the original system [23]. Continuation methods, in contrast, begin with an initial system whose solutions are known, and then transorm it gradually to the system whose solutions are sought, tracking all solution paths along the way [109].

In a third approach, branch-and-prune methods use approximate bounds of the solution set to rule out portions of the search space that contain no solution, reducing the initial domain as needed until a fine-enough approximation of the solution is obtained [17, 27, 63, 80, 85].

While algebraic-geometric and continuation methods are general, they have a number of limitations in practice. On the one hand, algebraic-geometric methods usually explode in complexity, may introduce extraneous roots, and can only be applied to relatively simple systems of equations. On the other hand, continuation methods should be implemented in multi-precision arithmetic to avoid numerical instabilities, leading to important memory requirements and, like elimination methods, they must compute all possible roots, even the complex ones, which are physically meaningless in our case, thus slowing the process substantially on systems with a small fraction of real roots. Branch-and-prune methods are also general, and present a number of advantages that make them preferable in our case: (1) Contrarily to many elimination methods, they do not require intuition-guided symbolic reductions, (2) they directly isolate the real roots, (3) they can be made numerically robust without resorting to extra-precision airthmetic, and (4) some of them can tackle under- and over-constrained problems without needing modifications. These are the main reasons that motivate the approach we adopt next, which belongs to this latter category. The approach, which is based on the one proposed in [80], entails expanding the equations to a standard form (Subsection 3.3.1) and then using a branch-and-prune method exploiting this form to isolate the solutions (Subsections 3.3.2 and 3.3.3).

3.3.1 Equation expansion

We distinguish two groups of equations in the final system $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. A first group encompassing Eqs. (3.17), (3.18), (3.24), and (3.25), whose polynomials follow the Bézier form of Eqs. (3.3) and (3.19), and a second group encompassing the remaining equations, whose polynomials only contain monomials of the form q_i, q_i^2 and q_iq_j. Note that all equations of the second group can be easily converted into linear form by introducing the changes of variables

$$p_i = q_i^2 \tag{3.27}$$

$$b_k = q_iq_j \tag{3.28}$$

for all q_i^2 and q_iq_j monomials occuring in them. After such changes, we obtain a new system of the form

$$\left.\begin{aligned}\mathbf{\Lambda}(\mathbf{x}) = \mathbf{0}\\ \mathbf{\Psi}(\mathbf{x}) = \mathbf{0}\end{aligned}\right\}, \tag{3.29}$$

where $\mathbf{x}$ is an n_x-dimensional vector encompassing all of the original q_i variables, and the newly-introduced p_i and b_k ones. Here, $\mathbf{\Lambda}(\mathbf{x}) = 0$ represents a collection of linear equations in $\mathbf{x}$, and $\mathbf{\Psi}(\mathbf{x}) = 0$ represents a collection of equations, each of which can only adopt one of these three forms:

$$x_k = x_i^2, \tag{3.30}$$

$$x_k = x_ix_j, \tag{3.31}$$

$$x_k = f(x_i, x_j). \tag{3.32}$$

While the first two forms correspond to the changes of variables in Eqs. (3.27) and (3.28), the latter form corresponds to the scalar components of Eqs. (3.17), (3.18), (3.24), and (3.25), so that $f(x_i, x_j)$ refers to a Bernstein-form polynomial of degrees d_i and d_j in x_i and x_j, respectively.

3.3.2 Equation solving

It can be seen that, under the used formulation, each variable x_i of $\mathbf{x}$ can only take values within a prescribed interval [80], so that the Cartesian product of all such intervals defines an initial n_x-dimensional box $\mathcal{B} \subset \mathbb{R}^{n_x}$ which bounds all solutions of Eqs. (3.29). The algorithm to isolate such solutions recursively applies two operations on $\mathcal{B}$: *box shrinking* and *box splitting*.

Using box shrinking, portions of $\mathcal{B}$ containing no solution are eliminated by narrowing some of its defining intervals. This process is repeated until either 1) the box is reduced to an empty set, in which case it contains no solution, or 2) the box is "sufficiently" small, in which case it is considered a *solution* box, or 3) the box cannot be "significantly" reduced, in which case it is bisected into two sub-boxes via box splitting (which simply bisects the box through its largest interval). To converge to all solutions, the whole process is recursively applied to the new sub-boxes, until one ends up with a collection of solution boxes, whose side lengths are below a given threshold σ.

As it turns out, this algorithm explores a binary tree of boxes, whose internal nodes correspond to boxes that have been split at some time, and whose leaves are either solution or empty boxes. By properly implementing the bookkeeping of boxes awaiting to be processed, this tree can be explored either in depth- or breadth-first order, the choice of order depending on whether one wishes to isolate just one solution, or the entire solution set.

Note that the algorithm is complete, in the sense that the solution boxes it returns include all solution points of Eqs. (3.29). Thus, the algorithm will always succeed in isolating a solution, whenever one exists, provided that a small-enough value of the σ parameter is used. Detailed properties of the algorithm, together with examples of its output, are given by Porta et al. [79, 80].

3.3.3 Box shrinking

We next see how a given sub-box $\mathcal{B}_c \subseteq \mathcal{B}$ can be reduced, discarding portions of the box that contain no solution. Observe that the solutions of Eqs. (3.29) lying within $\mathcal{B}_c \subseteq \mathcal{B}$ must lie on the linear variety defined by $\mathbf{\Lambda}(\mathbf{x}) = 0$. Thus, in principle, we might shrink $\mathcal{B}_c$ to the smallest possible box bounding this variety inside $\mathcal{B}_c$. The lower and upper limits of the shrunk box along dimension x_i, $i = 1, \ldots, n_x$, would respectively be found by solving the two linear programs

$$\textbf{LP1:}\ \text{Minimize } x_i,\ \text{ subject to: } \mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c,$$
$$\textbf{LP2:}\ \text{Maximize } x_i,\ \text{ subject to: } \mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c.$$

The sub-box $\mathcal{B}_c$ may be further reduced, however, because the solutions must also satisfy the equations $\mathbf{\Psi}(\mathbf{x}) = 0$. These equations can be taken into account by noting that for each equation it is possible to define a convex polytope that bounds the equation solutions within $\mathcal{B}_c$. Thus, to better delimit the solutions of the system, $\mathcal{B}_c$ can be safely reduced to the smallest possible box enclosing the intersection of $\mathbf{\Lambda}(\mathbf{x}) = 0$ and the polytopes of all equations in $\mathbf{\Psi}(\mathbf{x}) = 0$. This reduction can be implemented by representing the individual polytopes with linear inequalities, and adding such inequalities to the constraint set of the linear programs **LP1** and **LP2**. We next see how such polytopes can be derived, for each one of Eqs. (3.30)-(3.32). The notation $[l_i, u_i]$ will refer to the interval of $\mathcal{B}_c$ relative to x_i.

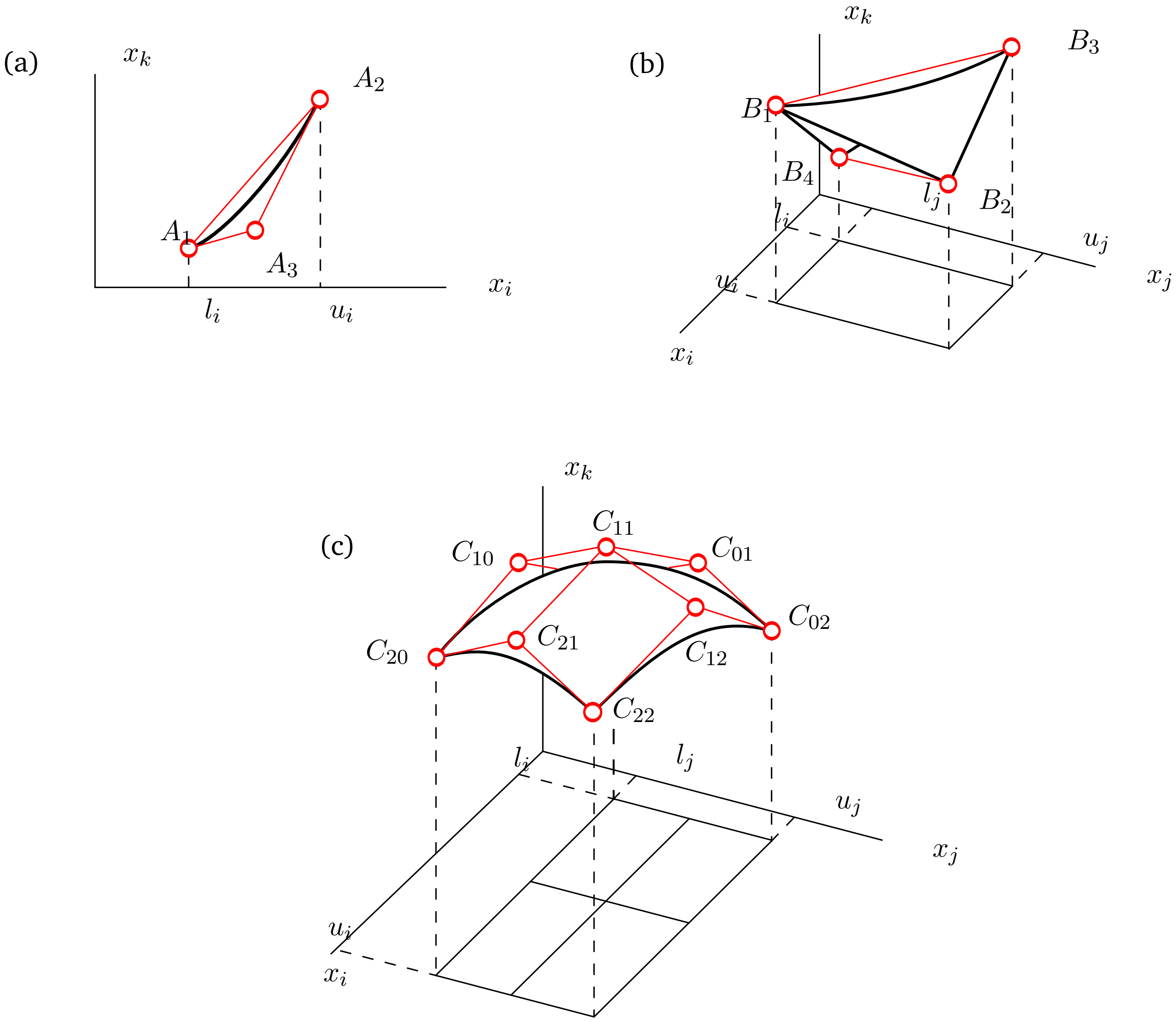


Figure 3.5: Polytope bounds within $\mathcal{B}_c$. (a) The points on $x_k = x_i^2$ are bound by the triangle $A_1A_2A_3$. (b) The points on $x_k = x_ix_j$ are bound by the tetrahedron $B_1B_2B_3B_4$. (c) The points on $x_k = f(x_i, x_k)$ are bound by the convex hull of the points C_{pq}. In this example, $f(x_i, x_k)$ is a Bernstein-form polynomial of degree two in x_i and x_j, so that the control points C_{pq} form a grid of size 3×3.

To derive a polytope for $x_k = x_i^2$, note that the portion of the parabola $x_k = x_i^2$ lying within $\mathcal{B}_c$ is bounded by the triangle $A_1A_2A_3$ in the x_i-x_k plane, where A_1 and A_2 are the points where the parabola intercepts the lines $x_i = l_i$ and $x_i = u_i$, and A_3 is the point where the tangent lines at A_1 and A_2 meet (Figure 3.5a). Thus, the polytope of $x_k = x_i^2$ is defined by the triangle $A_1A_2A_3$, which can be represented by three inequalities that correspond to the three edges of this triangle.

To derive a polytope for $x_k = x_i x_j$, we realize that the portion of the surface $x_k = x_i x_j$ included in $\mathcal{B}_c$ is bound by a tetrahedron $B_1 B_2 B_3 B_4$ in the x_i-x_j-x_k subspace, whose vertices B_i are obtained by lifting the four corners of the rectangle $[l_i, u_i] \times [l_j, u_j]$ vertically to the surface $x_k = x_i x_j$ (Figure 3.5b). Thus, the polytope of $x_k = x_i x_j$ is defined by the tetrahedron $B_1 B_2 B_3 B_4$, and can be represented by four inequalities, corresponding to the four faces of this tetrahedron. Finally, to derive a polytope for $x_k = f(x_i, x_j)$, we resort to the subdivision and convex-hull properties of Bernstein polynomials [30]. Using the subdivision property, on the one hand, $f(x_i, x_j)$ is written in the form

$$f(x_i, x_j) = \sum_{p=0}^{d_i} \sum_{q=0}^{d_j} b_{p,q} \cdot B_{p,d_i}(x_i) \cdot B_{q,d_j}(x_j),$$

where the scalars $b_{p,q}$ are the so-called control points of $f(x_i, x_j)$ relative to the interval $[l_i, u_i] \times [l_j, u_j]$. Using the convex-hull property, on the other hand, we know that the surface $x_k = f(x_i, x_j)$ must be contained inside the convex-hull of the 3D points C_{pq} with coordinates

$$\mathbf{c}_{pq} = \left(l_i + \frac{p}{d_i}(u_i - l_i), l_j + \frac{q}{d_j}(u_j - l_j), b_{p,q} \right), \tag{3.33}$$

for $p = 0, \ldots, d_i$, and $q = 0, \ldots, d_j$ (Figure 3.5c). This convex hull defines a polytope for equation $x_k = f(x_i, x_j)$, which can be encoded as a set of inequalities by resorting to an algorithm for convex-hull computations [1].

3.4 Test cases

The presented method has been implemented in C, extending the libraries of the CUIK suite [80]. This section illustrates the performance of the method under this implementation, on various test cases where an object needs to be grasped in a particular way, in order to fulfill a given task.

The tests involve the solution of various instances of the problem defined on a scalpel, a teapot, and a guitar, where each problem involves a number of regions to be placed in contact, imposed by the specific requirements of the task to be accomplished with the object (Table 3.2). In all test cases, the SA hand has been used to grasp the objects (Figure 3.6), but the presented methodology is equally applicable to any other hand. While the area of all contact regions defined on the hand is approximately 40%

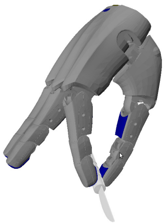
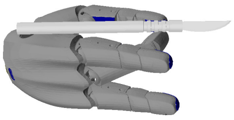
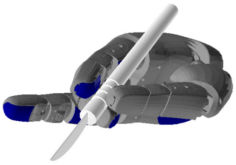
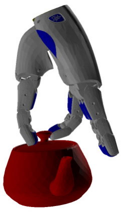
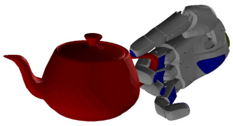
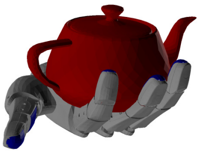
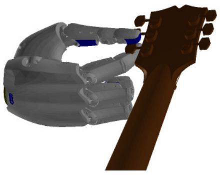
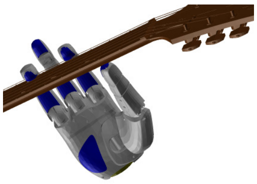
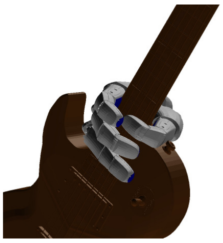
Number of contacts	**2 (slightly constrained)**	**3 (moderately constrained)**	**4 (highly constrained)**
SCALPEL tasks	Upholding	Handling	Incision
Task requirements	It requires picking the scalpel up using the index finger and the thumb.	It requires handling the scalpel delicately using the middle finger, the thumb, and the palm.	It requires a pencil-like grasp using two fingers, the thumb, and the palm.
n_v, n_e, d	$219, 209, 17$	$243, 235, 16$	$331, 324, 18$
CPU time $[s]$	106	255	418
Computed solution			
TEAPOT tasks	Lid lifting	Service	Transportation
Task requirements	The lid must be pulled up through its knob using the index finger and the thumb.	The hand is required to hold the teapot by its handle, placing the thumb on top, while the index and middle fingers embrace it.	The palm contacts the bottom of the teapot, while the fingers enclose it so that it does not slide out.
n_v, n_e, d	$219, 209, 17$	$288, 278, 19$	$312, 305, 17$
CPU time $[s]$	114	262	375
Computed solution			
GUITAR tasks	Tunning	Playing	Holding
Task requirements	It requires the hand to grasp a given key using the index finger and the thumb to tune the tension of the corresponding string.	The fingertips must be at specified strings and frets to perform a chord, while the thumb contacts the guitar neck.	It requires a whole-hand grasp on a specific region where the guitar can not be damaged while being transported.
n_v, n_e, d	$219, 209, 17$	$307, 298, 19$	$331, 324, 18$
CPU time $[s]$	68	229	664
Computed solution			

Table 3.2: Test cases and their computed solutions.

of the fingertip area (the dark patches on the upper limbs in Figure 3.6b), the area of the contact regions on the object varies from experiment to experiment, from 2% of the fingertip area on the teapot knob ("lid lifting" experiment), to 9000% of such area on the guitar neck ("playing" experiment).

We next explain how the equations of the hand can be set up, and later discuss the algorithm's performance on the mentioned tests.

3.4.1 Equations for the Schunk Anthropomorphic hand

The Schunk Anthropomorphic hand is composed of four identical fingers that follow the anthropomorphic structure illustrated in Figure 3.2. Three of these fingers are directly mounted on the palm, and act as ring, middle, and index fingers. The fourth finger is mounted on an intermediate link articulated with the palm through a revolute joint, which allows this finger to act as a thumb (Figure 3.6). The hand has a total of fourteen links (one palm and thirteen phalanges) and thirteen joints (nine revolute joints and four universal joints).

To set up the equations, the links of the hand are labelled as $L_0, \ldots, L_{13}$, as shown in Figure 3.6, and the joints as $J_1, \ldots, J_{13}$, letting J_i be the joint between L_{i-1} and L_i (for clarity, joint labels are not shown in Figure 3.6). Twenty-six points and unit vectors are then defined, that provide the positions and orientations of all rotation axes of the hand relative to the involved links. The points correspond to the centers of the universal joints and to the midpoints of the revolute joints. The vectors correspond to unit vectors aligned with the rotation axes of the joints. These points and vectors are displayed in Figure 3.6 and their coordinates are given in Table 3.3, in milimeters. All reference frames $\mathcal{F}_l$ are located with their origin in Q_l, so that $\mathbf{q}_l^{\mathcal{F}_l} = (0,0,0)$, for $l = 0, \ldots, 13$. The orientations of such frames can be deduced easily from the coordinates provided in Table 3.3. Taking into account these definitions, Eqs. (3.4)-(3.11) can readily be written for all links and joints involved.

To write down the equations of Section 3.2.3, the mechanical limits of the SA hand must be considered. Regarding the universal joints, the rotations about their $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$ axes are limited to the ranges $[-15^o, 15^o]$ and $[-4^o, 75^o]$, respectively. Regarding the revolute joints, all of them can only rotate in the range $[4^o, 75^o]$, except for the revolute joint at the base of the thumb, which is restricted to the range $[0^o, 90^o]$. The reference configuration corresponding to all rotation angles at zero is shown in Figure 3.6b.

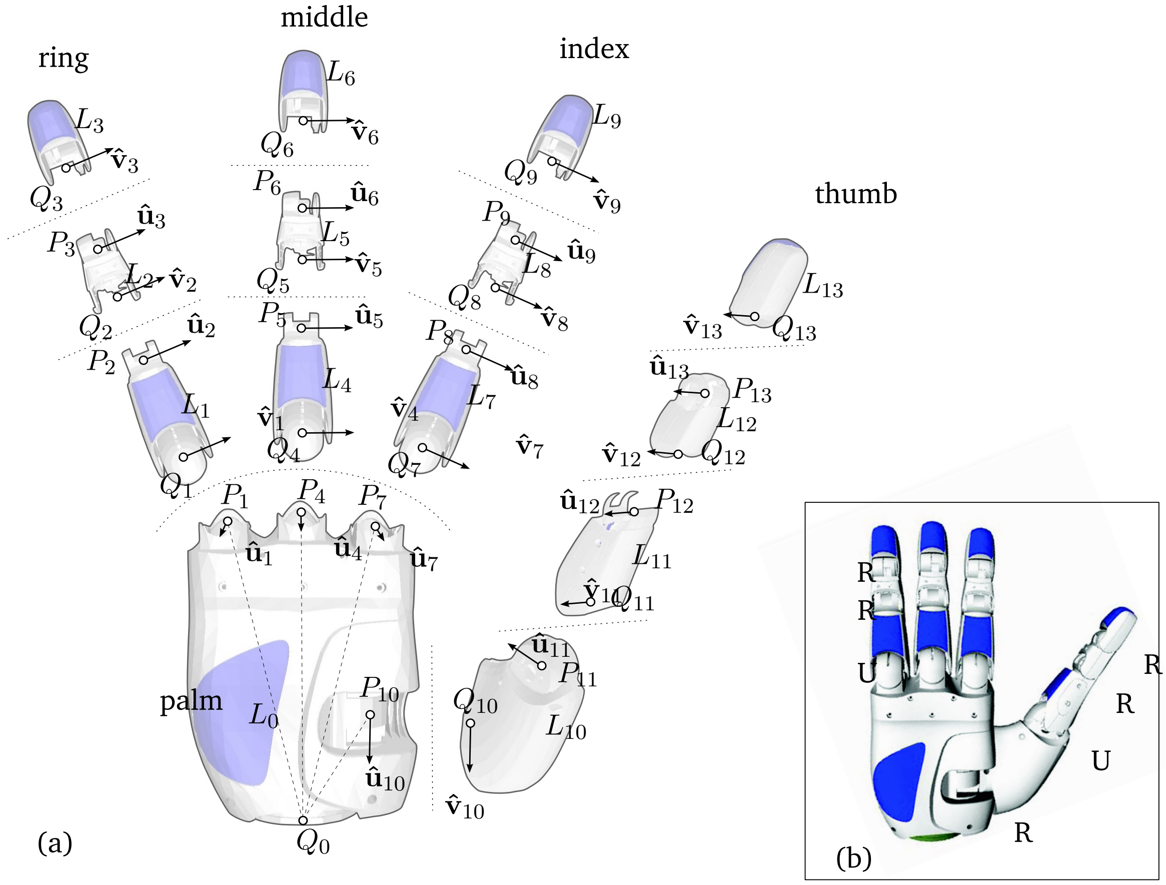


Figure 3.6: Geometric parameters (a) and reference configuration (b) of the Schunk Anthropomorphic Hand. The various joint types are indicated in (b).

Finally, it must be taken into account that not all joints of the SA hand are independently actuated. The two distal joints of each finger are coupled, so that when one of such joints is actuated, a rotation of the same angle about the other is produced. In the adopted formulation, the coupling of two rotation angles is simply imposed by equating the sine and cosine of such angles.

3.4.2 Computed solutions

A system of equations has to be solved for each task of Table 3.2, encompassing the equations of the SA hand, together with the contact equations that impose the specific requirements of the task. It must be noted that Eqs. (3.4)-(3.11) relative to fingers not in contact with the object can actually be removed from this system, because such fingers

Joint	**Ring**		**Middle**		**Index**		**Thumb**	
type	Par.	Value	Par.	Value	Par.	Value	Par.	Value
R	$\mathbf{p}_3^{\mathcal{F}_2}$	$(30,0,0)$	$\mathbf{p}_6^{\mathcal{F}_4}$	$(30,0,0)$	$\mathbf{p}_9^{\mathcal{F}_7}$	$(30,0,0)$	$\mathbf{p}_{13}^{\mathcal{F}_{12}}$	$(30,0,0)$
	$\hat{\mathbf{u}}_3^{\mathcal{F}_2}$	$(0,1,0)$	$\hat{\mathbf{u}}_6^{\mathcal{F}_5}$	$(0,1,0)$	$\hat{\mathbf{u}}_9^{\mathcal{F}_8}$	$(0,1,0)$	$\hat{\mathbf{u}}_{13}^{\mathcal{F}_{12}}$	$(0,1,0)$
	$\hat{\mathbf{v}}_3^{\mathcal{F}_3}$	$(0,1,0)$	$\hat{\mathbf{v}}_6^{\mathcal{F}_6}$	$(0,1,0)$	$\hat{\mathbf{v}}_9^{\mathcal{F}_9}$	$(0,1,0)$	$\hat{\mathbf{v}}_{13}^{\mathcal{F}_{13}}$	$(0,1,0)$
R	$\mathbf{p}_2^{\mathcal{F}_1}$	$(67.80,0,0)$	$\mathbf{p}_5^{\mathcal{F}_4}$	$(67.80,0,0)$	$\mathbf{p}_8^{\mathcal{F}_7}$	$(67.80,0,0)$	$\mathbf{p}_{12}^{\mathcal{F}_{11}}$	$(67.80,0,0)$
	$\hat{\mathbf{u}}_2^{\mathcal{F}_1}$	$(0,1,0)$	$\hat{\mathbf{u}}_5^{\mathcal{F}_4}$	$(0,1,0)$	$\hat{\mathbf{u}}_8^{\mathcal{F}_7}$	$(0,1,0)$	$\hat{\mathbf{u}}_{12}^{\mathcal{F}_{11}}$	$(0,1,0)$
	$\hat{\mathbf{v}}_2^{\mathcal{F}_2}$	$(0,1,0)$	$\hat{\mathbf{v}}_5^{\mathcal{F}_5}$	$(0,1,0)$	$\hat{\mathbf{v}}_8^{\mathcal{F}_8}$	$(0,1,0)$	$\hat{\mathbf{v}}_{12}^{\mathcal{F}_{12}}$	$(0,1,0)$
U	$\mathbf{p}_1^{\mathcal{F}_0}$	$(-4.30,-40.16,145.43)$	$\mathbf{p}_4^{\mathcal{F}_0}$	$(-4.30,0,145.43)$	$\mathbf{p}_7^{\mathcal{F}_0}$	$(-4.30,40.16,145.43)$	$\mathbf{p}_{11}^{\mathcal{F}_{10}}$	$(97,6,-87)$
	$\hat{\mathbf{u}}_1^{\mathcal{F}_0}$	$(1,0,0)$	$\hat{\mathbf{u}}_4^{\mathcal{F}_0}$	$(1,0,0)$	$\hat{\mathbf{u}}_7^{\mathcal{F}_0}$	$(1,0,0)$	$\hat{\mathbf{u}}_{11}^{\mathcal{F}_{10}}$	$(\cos 55^o, 0, \sin 55^o)$
	$\hat{\mathbf{v}}_1^{\mathcal{F}_1}$	$(0,1,0)$	$\hat{\mathbf{v}}_4^{\mathcal{F}_4}$	$(0,1,0)$	$\hat{\mathbf{v}}_7^{\mathcal{F}_7}$	$(0,1,0)$	$\hat{\mathbf{v}}_{11}^{\mathcal{F}_{11}}$	$(0,1,0)$
R							$\mathbf{p}_{10}^{\mathcal{F}_0}$	$(-3,27.10,0)$
							$\hat{\mathbf{u}}_{10}^{\mathcal{F}_0}$	$(0,0,-1)$
							$\hat{\mathbf{v}}_{10}^{\mathcal{F}_{10}}$	$(1,0,0)$

Table 3.3: Parameters of the Schunk Anthropomorphic hand.

do not intervene in any kinematic loop, and hence impose no loop-closure constraint on the overall system.

Table 3.2 provides the size of the equation system Eq. (3.26) to be solved in each case, in terms of the number of variables, n_v, and equations, n_e, it involves, and the dimension of its solution space, d, predicted as the number of variables minus the number of non-redundant equations. Note in this regard that Eq. (3.9) introduces equations that are redundant in terms of predicting such dimension, because $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$ are unit vectors, and it is sufficient to establish two out of the three components of Eq. (3.9) to determine the alignment of L_k relative to L_j. The third component of Eq. (3.9), however, is needed to remove a sign ambiguity in such alignment. Since a similar redundancy is introduced by Eq. (3.16), there will be as many redundant equations as the number of joints and contacts involved in the problem at hand.

As it can be observed from Table 3.2, typical problems yield solution spaces of rather high dimension. To avoid the curse of dimensionality as much as possible, and converge to one solution rapidly, the proposed algorithm must be set to explore in depth-first order (Subection 3.3.2). Running the algorithm in this order, we have obtained the hand-object configurations depicted, in the CPU times indicated in each case. All times reported correspond to a parallelized version of the algorithm, executed on a grid of 8

DELL Poweredge computers, equipped with two Intel Quadcore Xeon E5310 processors and a 4Gb RAM each one, using a threshold of $\sigma = 0.1$. Note that the cost of computing a solution increases with the number of contact constraints to be satisfied. This is because the size of the linear programs to be solved during box shrinking is proportional to the number of polytope inequalities introduced by such constraints, which increases the cost of each iteration of the algorithm (Subsection 3.3.3).

3.5 Summary

This chapter has presented a new method to compute kinematically-feasible grasps. When compared to other approaches for the same problem, the proposed method is always guaranteed to converge to a solution whenever one exists. Additional features of the method are the ability to deal with general region-to-region contact constraints, as opposed to point-to-region [11, 95] or point-to-point [38] constraints, and the possibility to define the regions as general Bézier patches, to better adapt to the shape of real surfaces on the hand and object considered. The proposed algorithm is based on formulating the problem as a system of polynomial equations of special form, and then exploiting such form to solve the equations, using an extended version of a recent method based on linear relaxations [80]. It must be noted that, whereas the algorithm in [80] can deal with lower-pair mechanisms of general structure, it cannot be directly applied to mechanical hands, because it is unable to cope with general contact constraints between free-form surfaces. Here, we have extended that algorithm to be able to specify such constraints between Bézier patches defined anywhere on the object or on the hand, and to solve the corresponding equations.

4

Optimizing Grasp Configurations

This chapter provides a method for optimizing the quality of a robotic grasp, subject to satisfying a number of hand-object contact constraints of the kind assumed in the preceding chapter. Due to the multi-modal nature of typical grasp quality measures, approaches that resort to local optimization methods are likely to get trapped into local extrema on such problem. An additional difficulty of the problem is the fact that the set of feasible grasps is a highly-dimensional manifold, implicitly defined by a system of non-linear equations. The proposed procedure finds a way around these issues by focusing the exploration on a relevant subset of grasps of lower dimension, and tracing this subset exhaustively using a higher-dimensional continuation technique. Using this technique, a detailed atlas of the subset is obtained, on which the highest-quality grasp according to any desired criterion, or a combination of criteria, can be readily identified.

Regardless of the method used to generate an initial configuration of the hand in contact with the object, the returned grasp may not be optimal in terms of any quality criterion, so that an optimization process is necessary to generate a suitable high-quality grasp satisfying the specified contacts (Figure 4.1). The grasp optimization procedure proposed here entails characterizing the manifold of kinematically-feasible grasps as a system of equations to be fulfilled (Section 4.1), then extending this system with meaningful equations to reduce the dimension of its solution set (Section 4.2), and finally performing an exhaustive search over a point grid discretizing such set at a desired resolution, to determine the highest-quality grasp attainable on the grid (Section 4.3). Test cases are provided that validate the approach on simple and complex robotic hands grasping objects with different geometries, under typical force-closure and manipulability indices (Section 4.4).

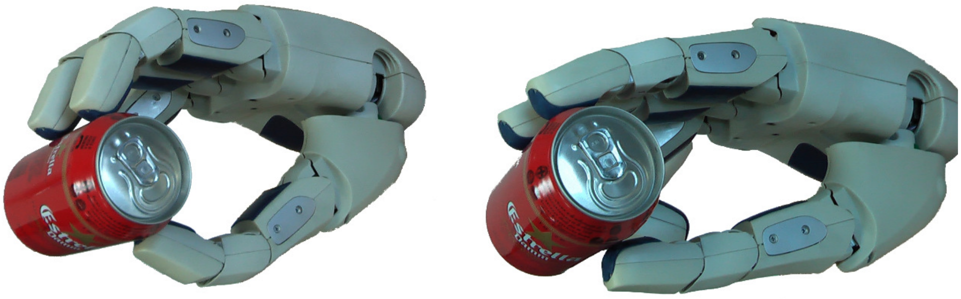


Figure 4.1: Two grasps of a can for drink service. While both grasps are force-closed and manipulable, the grasp on the left is preferable. The fingers are almost fully extended in the grasp on the right, limiting the possibility to move the can in one direction.

4.1 Kinematically-feasible grasps

Recall that, a kinematically-feasible grasp is a configuration of the hand-object system in which a number of regions of the hand h_i, for $i = 1, \ldots, b$, are in contact with corresponding regions o_i on the object. The regions and their pairings are pre-specified, and the contact between h_i and o_i is assumed to be established with a point $H_i \in h_i$ coinciding with another point $O_i \in o_i$, keeping aligned the surface normals at such points, $\hat{\mathbf{m}}_i$ and $\hat{\mathbf{n}}_i$, to avoid local hand-object inter-penetrations. We further assume that the hand joints are independently actuated or mechanically coupled, but do not consider the case of adaptive underactuated hands [9].

Independently of the particular formulation adopted, a grasp configuration can be represented by a vector $\mathbf{x} = (\mathbf{x}_{\mathrm{h}}^{\top}, \mathbf{x}_{\mathrm{o}}^{\top}, \mathbf{x}_{\mathrm{c}}^{\top})^{\top} \in \mathbb{R}^n$ of generalized coordinates, where $\mathbf{x}_{\mathrm{h}}$ and $\mathbf{x}_{\mathrm{o}}$ encompass the configuration variables of the hand and the object, respectively, and $\mathbf{x}_{\mathrm{c}}$ encompasses contact-related variables. Without loss of generality, we will assume that the absolute reference frame is attached to the palm of the hand, so that its pose variables do not intervene in $\mathbf{x}$.

The variables in $\mathbf{x}$ are subject to a number of constraints. A first set of equations,

$$\mathbf{H}(\mathbf{x}_{\mathrm{h}}) = \mathbf{0}, \tag{4.1}$$

enforces $\mathbf{x}_{\mathrm{h}} \in \mathbb{R}^h$ to be a valid hand configuration, i.e., one respecting the assembly constraints imposed by the joints, usually revolute or universal, on the various bodies they connect, that is, the palm and the several finger phalanges. Note that Eq. (4.1) is not necessary if the $\mathbf{x}_\text{h}$ coordinates are independent, as it happens for instance when choosing joint angles to represent a configuration [26]. In our case, however, we resort to the dependent coordinates defined in the previous chapter because they are equations of a simple structure, which has proved to be beneficial in the context of grasp synthesis (Chapter 3), and for the application of continuation techniques [121]. In particular, this formulation encodes the six degrees of freedom of a body with twelve variables, providing the position vector and the rotation matrix of a reference frame attached to the body. Therefore, in addition to the joint assembly equations, Eq. (4.1) includes constraints to enforce the 12-tuple of each body to be a member of $SE(3)$. Similarly, the spatial pose of the object is encoded by twelve variables, so that a second set of equations,

$$\mathbf{L}(\mathbf{x}_\text{o}) = \mathbf{0}, \tag{4.2}$$

constrains $\mathbf{x}_\text{o} \in \mathbb{R}^{12}$ to define a member of $SE(3)$. Finally, a third set of equations formulates the contact constraints between the hand and the object. As in Chapter 3, we assume that each contact region h_i on the hand is specified as a parametrized patch, i.e., as a smooth function of the form

$$\mathbf{h}_i = \mathbf{h}_i(u_i, v_i, \mathbf{x}_\text{h}), \tag{4.3}$$

providing the absolute coordinates of a point $\mathbf{h}_i = (x_i, y_i, z_i)$ in the patch, in terms of two bounded scalar parameters, u_i and v_i, and of the hand configuration $\mathbf{x}_\text{h}$. Analogously, the normal to any point in this patch is assumed to be given as a function

$$\hat{\mathbf{m}}_i = \mathbf{m}_i(u_i, v_i, \mathbf{x}_\text{h}). \tag{4.4}$$

In general, Eqs. (4.3) and (4.4) define two-dimensional regions described, for example, by Bézier patches [30], but they can be replaced by single-parameter curves or fixed points, if desired. The points and normals for the contact patches on the object can be defined in a similar way

$$\mathbf{o}_i = \mathbf{o}_i(s_i, t_i, \mathbf{x}_\text{o}), \tag{4.5}$$

$$\hat{\mathbf{n}}_i = \mathbf{n}_i(s_i, t_i, \mathbf{x}_\text{o}), \tag{4.6}$$

in terms of bounded parameters s_i and t_i, and the object pose $\mathrm{x_o}$. With the previous definitions, the contact of h_i with o_i can be expressed by setting

$$\mathbf{h}_i - \mathbf{o}_i = \mathbf{0}, \tag{4.7}$$

$$\hat{\mathbf{m}}_i + \hat{\mathbf{n}}_i = \mathbf{0}. \tag{4.8}$$

Thus, the vector $\mathrm{x_c}$ encompases the vectors $\mathbf{h}_i$, $\hat{\mathbf{m}}_i$, $\mathbf{o}_i$ and $\hat{\mathbf{n}}_i$, and the patch parameters u_i, v_i, s_i and t_i intervening in Eqs. (4.3) and (4.8), for $i = 1, \ldots, b$.

Each variable in $\mathrm{x} = (\mathrm{x_h^\top}, \mathrm{x_o^\top}, \mathrm{x_c^\top})^\top$ can only take values within a given range. For instance, the variables defining the orientation matrices in $\mathrm{x_h}$ and $\mathrm{x_o}$ take values within $[-1, 1]$. Also, the size of the hand provides interval bounds on the translation variables, and, without loss of generality, the patch parameters of the contact regions can be normalized to the $[0, 1]$ range [30]. Thus, the Cartesian product of such ranges defines a rectangular domain $\mathcal{D} \subset \mathbb{R}^n$ where the search for an optimal feasible grasp is to be confined.

In sum, the set of feasible grasps $\mathcal{F}$ encompasses the points $\mathrm{x} \in \mathcal{D}$ satisfying the system

$$\mathbf{F}(\mathrm{x}) = \mathbf{0} \tag{4.9}$$

formed by Eqs. (4.1) and (4.2), and Eqs. (4.3) to (4.8) for all contacts $i = 1, \ldots, b$.

The formulation in Chapter 3 guarantees that $\mathbf{F}(\mathrm{x})$ is differentiable and, to keep the presentation of the method as simple as possible, we assume that its Jacobian is full rank for all $\mathrm{x} \in \mathcal{F}$, which is the generic situation according to Sard's theorem of Analysis. Thus, $\mathcal{F}$ can be assumed to be a smooth manifold of dimension $t = n - f$, where f is the number of scalar equations in Eq. (4.9). Now, using Eq. (4.9), the method in Chapter 3 can be applied to determine an initial kinematically-feasible grasp $\mathrm{x}_1 \in \mathcal{F}$ from which to start the optimization process. We emphasize, however, that other grasp synthesis techniques could be used to compute x_1 [11, 21] since the optimization approach introduced in this chapter makes no assumption on how this configuration is obtained.

4.2 Relevant grasps

Although obtaining points on $\mathcal{F}$ is feasible, the dimension of $\mathcal{F}$ is very large for a realistic hand and contact model, which hinders the exhaustive exploration of this space independently of the methodology adopted. In the context of grasping, however, studies on the human behavior suggest that humans do not use all degrees of freedom of the hand independently, but in a coordinated way [104]. Following this idea, anthropomorphic hands are usually controlled using postural hand synergies [6, 104], where fewer degrees of freedom are used to account for the overall hand configuration space. By taking postural synergies into consideration, the search of a good grasp can be narrowed to a subset of relevant grasps $\mathcal{R} \subset \mathcal{F}$ of lower dimension, thus speeding up the overall optimization process.

In this work, we compute the postural synergies via linear dimension-reduction techniques on a predetermined set of hand configurations $\mathbf{X}_{\mathrm{h}} = \{\mathbf{x}_{\mathrm{h}}^1, \ldots, \mathbf{x}_{\mathrm{h}}^z\}$, where each $\mathbf{x}_{\mathrm{h}}^i$ is a value of $\mathbf{x}_{\mathrm{h}}$ satisfying Eq. (4.1). Let h be the number of components in $\mathbf{x}_{\mathrm{h}}$, $\bar{\mathbf{x}}_{\mathrm{h}}$ the average of the configurations in $\mathbf{X}_{\mathrm{h}}$, and $\mathbf{T}$ an $h \times z$ matrix whose i-th column is $\mathbf{x}_{\mathrm{h}}^i - \bar{\mathbf{x}}_{\mathrm{h}}$. The principal component analysis of $\mathbf{X}_{\mathrm{h}}$ can be performed by diagonalizing the covariance of $\mathbf{T}$,

$$\mathbf{T}\,\mathbf{T}^{\top} = \mathbf{E}\,\mathbf{S}^2\,\mathbf{E}^{\top}.$$

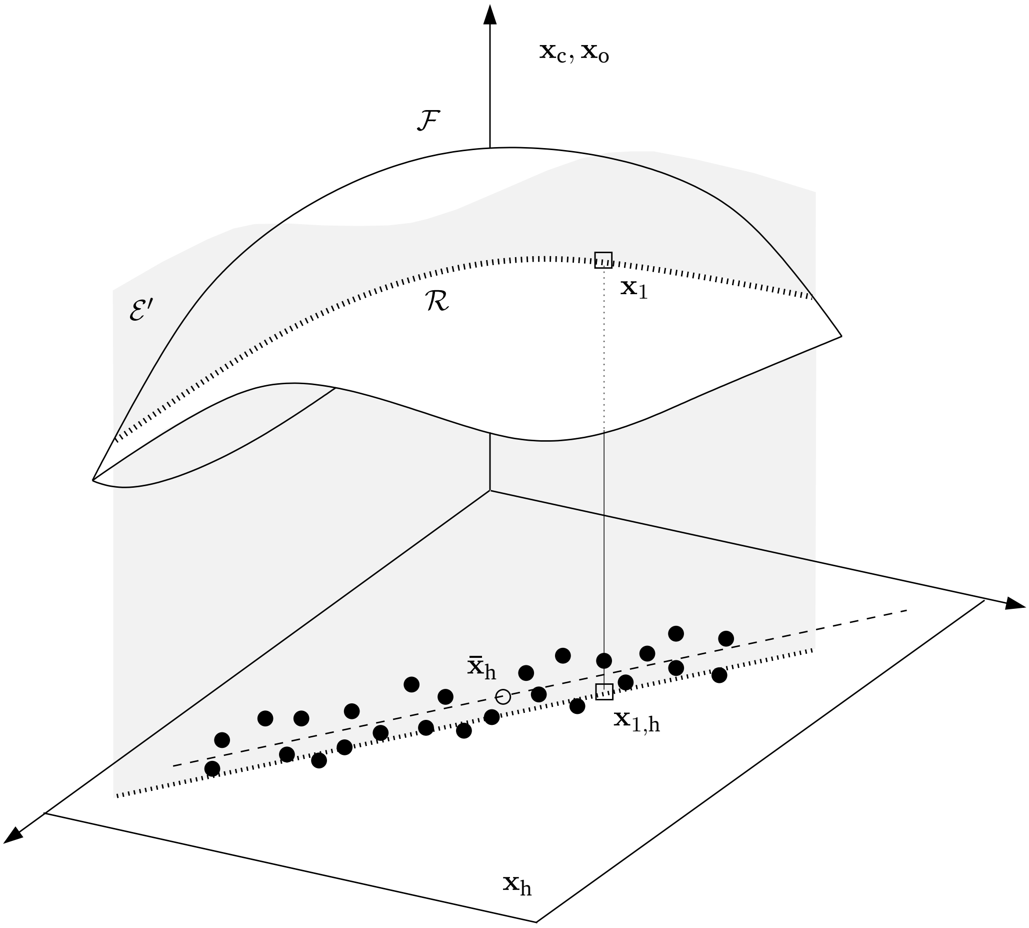
The $h \times h$ orthonormal matrix $\mathbf{E}$ gives the directions of variance of the data, and the diagonal matrix $\mathbf{S}^2$ is the variance in each one of these directions, sorted in decreasing magnitude. The linear variety through $\bar{\mathbf{x}}_{\mathrm{h}}$ generated by the first p columns of $\mathbf{E}$ defines the set $\mathcal{E}$ of the p postural synergies. The vectors in $\mathcal{E}$ have null components along the columns of $\mathbf{E}_s$, the matrix formed by the last $s = h - p$ columns of $\mathbf{E}$, so that $\mathcal{E}$ is the solution set of

$$\mathbf{E}_s^{\top}(\mathbf{x}_{\mathrm{h}} - \bar{\mathbf{x}}_{\mathrm{h}}) = \mathbf{0}. \tag{4.10}$$

This equation, together with Eq. (4.9), defines the system

$$\mathbf{R}(\mathbf{x}) = \mathbf{0}, \tag{4.11}$$

characterizing the non-linear set $\mathcal{R} = \mathcal{F} \cap \mathcal{E}$ of relevant grasps. In the generic case, the Jacobian of $\mathbf{R}(\mathbf{x})$ will also be full rank for all $\mathbf{x} \in \mathcal{R}$, so that, like $\mathcal{F}$, $\mathcal{R}$ can also be as-



$\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{o}$
$\mathcal{F}$
$\mathcal{E}'$
$\mathcal{R}$
$\mathbf{x}_1$
$\bar{\mathbf{x}}_\mathrm{h}$
$\mathbf{x}_{1,\mathrm{h}}$
$\mathbf{x}_\mathrm{h}$

Figure 4.2: Schematic representation of all the elements involved in the optimization framework presented in this chapter. See the text for details.

sumed to be a smooth manifold with no bifurcations, which allows adopting a simplified continuation strategy below, with no provisions for branch switching operations [41].

Since t is the dimension of $\mathcal{F}$, and assuming k is the desired dimension for $\mathcal{R}$, then we must set

$$s = t - k \tag{4.12}$$

in Eq. (4.10). For efficiency reasons, k must be small in the adopted continuation method (typically below 5), and s must be set accordingly. However, s must be smaller than h, which limits the amount of dimension reduction obtained by the use of postural synergies. As we will see, however, this is not an issue in practice, since the amount of dimension reduction to be introduced is moderate in all cases, due to the presence of the contact constraints. Actually, these constraints allow using a significantly smaller s than that used in existing approaches relying on postural synergies.

Nonetheless, the introduction of postural synergies might lead to an empty set $\mathcal{R}$ if the contact points are not reachable by the hand when constrained to be in $\mathcal{E}$. To guarantee that $\mathcal{R}$ is not empty, this set is redefined as $\mathcal{R} = \mathcal{F} \cap \mathcal{E}'$ hereafter, where $\mathcal{E}'$ is the solution set of

$$\mathbf{E}_s^\top(\mathbf{x}_\mathrm{h} - \mathbf{x}_{1,\mathrm{h}}) = \mathbf{0}, \tag{4.13}$$

with $\mathbf{x}_{1,\mathrm{h}}$ being the subvector of the initial grasp $\mathbf{x}_1$ containing the values of $\mathbf{x}_\mathrm{h}$. In general, the difference of using Eq. (4.13) instead of Eq. (4.10) is minor, since $\mathbf{x}_{1,\mathrm{h}}$ is usually close to the original set of principal hand motions, but this approximation ensures that the hand can always conform to the object surface because $\mathcal{R}$ will at least include the initial feasible grasp $\mathbf{x}_1$. Figure 4.2 summarizes the different elements involved in the approach. $\mathcal{F}$ is the set of feasible grasps defined in the space of $\mathbf{x}_\mathrm{h}$, $\mathbf{x}_\mathrm{o}$, and $\mathbf{x}_\mathrm{c}$. In this representation, the configurations in $\mathbf{X}_\mathrm{h}$ are shown as black dots, the white dot is their average, $\bar{\mathbf{x}}_\mathrm{h}$, and the original set of p principal hand motions is shown as a dashed line in the $\mathbf{x}_\mathrm{h}$ plane. The latter set is approximated by a line through $\mathbf{x}_{1,\mathrm{h}}$, shown dotted in the figure, which, when extended to the whole space, generates the linear variety $\mathcal{E}'$. Finally, the set $\mathcal{R} = \mathcal{F} \cap \mathcal{E}'$ is the space where the grasp optimization is to be performed. Note that $\mathcal{R}$ is one-dimensional in this schematic representation, but it is a k-dimensional set in general.

Typically, previous methods that use principal hand motions explore $\mathcal{E}$. Using local methods, then, they try to modify points on such space to yield configurations in contact with the object. However, the final configurations may not necessarily lie on the set $\mathcal{F}$ of grasps satisfying the contact constraints required for the task. In contrast, our method directly operates on the set $\mathcal{R}$, which is fully included in $\mathcal{F}$.

4.3 Grasp quality optimization

The search for an optimal grasp is performed by computing an atlas of the k-dimensional manifold $\mathcal{R}$ just defined, including the relevant grasps. Such an atlas provides a collection of charts, where each chart parametrizes a portion of $\mathcal{R}$, and this allows enumerating a representative collection of grasps in $\mathcal{R}$, on which any quality index can be evaluated to detect the optimal one.

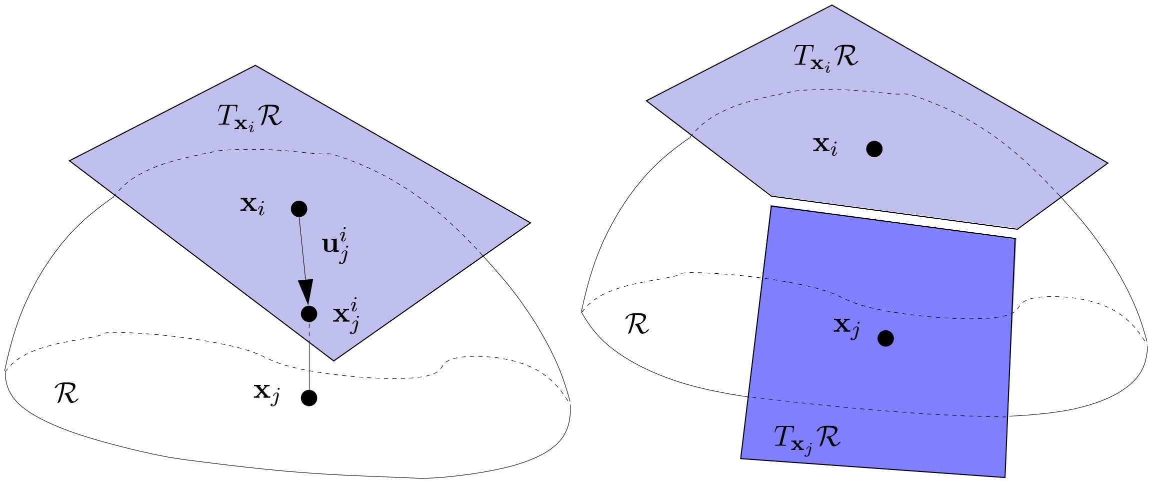


Figure 4.3: The higher-dimensional continuation method applied to a two-dimensional manifold $\mathcal{R}$ in 3D space. A point $\mathbf{x}_j \in \mathcal{R}$ can be obtained by orthogonally projecting a point $\mathbf{x}_j^i$ corresponding to a vector $\mathbf{u}_j^i \in T_{\mathbf{x}_i}\mathcal{R}$ (top). If a new chart is defined at $\mathbf{x}_j$, it must be properly coordinated with the chart at $\mathbf{x}_i$ so that their projections smoothly cover the manifold (bottom).

4.3.1 Tracing the manifold of relevant grasps

Formally, a chart $\mathcal{C}_i$ is a local map from a parameter domain $\mathcal{P}_i \subset \mathbb{R}^k$ to an open neighborhood around a given point $\mathbf{x}_i \in \mathcal{R}$, initially $\mathbf{x}_1$. The higher-dimensional continuation method proposed in [40] defines the map for chart $\mathcal{C}_i$ using $\mathbf{\Phi}_i$, an $n \times k$ orthonormal basis of $T_{\mathbf{x}_i}\mathcal{R}$, the k-dimensional tangent space of $\mathcal{R}$ at $\mathbf{x}_i$. The map is constructed by first selecting a k-dimensional vector of parameters $\mathbf{u}_j^i \in T_{\mathbf{x}_i}\mathcal{R}$ (Figure 4.3, top), and then using this vector to generate a point $\mathbf{x}_j^i \in \mathbb{R}^n$ in the neighborhood of $\mathbf{x}_i$ using

$$\mathbf{x}_j^i = \mathbf{x}_i + \mathbf{\Phi}_i\,\mathbf{u}_j^i. \tag{4.14}$$

Then, the point $\mathbf{x}_j \in \mathcal{R}$ that corresponds to the orthogonal projection of $\mathbf{x}_j^i$ on $\mathcal{R}$ is computed, by solving the system

$$\left.\begin{aligned} \mathbf{R}(\mathbf{x}_j) = \mathbf{0} \\ \mathbf{\Phi}^\top(\mathbf{x}_j - \mathbf{x}_j^i) = \mathbf{0} \end{aligned}\right\}, \tag{4.15}$$

using a Newton-Raphson method initialized at $\mathbf{x}_j^i$.

Each point on the manifold is the potential center of a new chart (Figure 4.3), and a method introduced by Henderson [40] can be used to select the chart centers, in order to ensure a good coverage of the manifold. Using his approach, the domain $\mathcal{P}_i$ of a chart $\mathcal{C}_i$ is initialized as a k-dimensional hypercube enclosing a ball $\mathcal{B}_i$ of radius r, where $\mathcal{P}_i$ and $\mathcal{B}_i$ are both defined in $T_{\mathbf{x}_i}\mathcal{R}$ as illustrated in Figure 4.4 (top). A vertex of $\mathcal{P}_i$ exterior to $\mathcal{B}_i$, with position vector $\mathbf{v}$, is then employed to generate a point $\mathbf{x}^i_j$, using Eq. (4.14) with

$$\mathbf{u}^i_j = \frac{\alpha}{\|\mathbf{v}\|}\,\mathbf{v}, \tag{4.16}$$

where α is initialized to the radius r. If the projection of $\mathbf{x}^i_j$ to $\mathcal{R}$ does not converge, or if the new chart $\mathcal{C}_j$ at $\mathbf{x}_j$ is too far or too different from $\mathcal{C}_i$, i.e., if

$$\|\mathbf{x}_j - \mathbf{x}^i_j\| > \epsilon, \tag{4.17}$$

or

$$\|\mathbf{\Phi}_i^\top \mathbf{\Phi}_j\| < 1 - \epsilon, \tag{4.18}$$

for a given threshold ϵ, then the new chart is discarded and a new attempt of chart

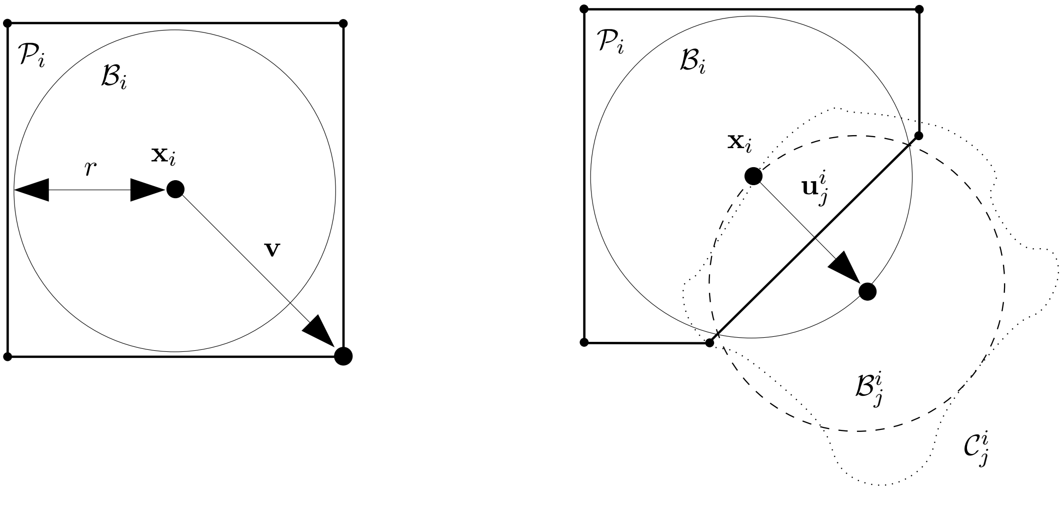


Figure 4.4: The process of chart construction. The domain $\mathcal{P}_i$ of chart $\mathcal{C}_i$ is initialized as a box in $T_{\mathbf{x}_i}\mathcal{R}$ circumscribing a ball of radius r centered in $\mathbf{x}_i$ (top). $\mathcal{P}_i$ is refined using a ball $\mathcal{B}^i_j$ that approximates $\mathcal{C}^i_j$, the projection on $\mathcal{C}_i$ of the part of the manifold covered by $\mathcal{C}_j$ (bottom).

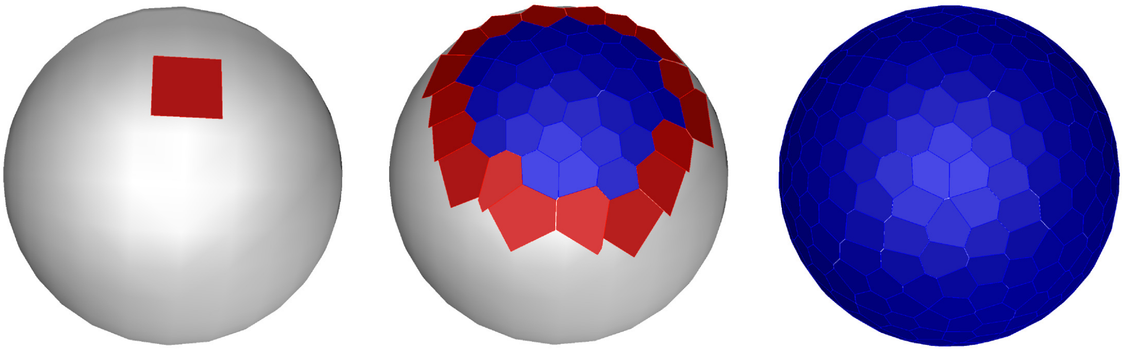


Figure 4.5: Three stages of the construction of an atlas over a sphere. Red and blue polygons represent charts under expansion and charts whose domain is already bounded, respectively.

generation is performed with a smaller α. This procedure adapts the size of the area covered by each chart to the local curvature of the manifold. When $\mathcal{C}_j$ is valid, it is used to refine $\mathcal{P}_i$ from the intersection between $\mathcal{B}_i$ and $\mathcal{C}_j^i$, the projection on $T_{\mathbf{x}_i}\mathcal{R}$ of the part of the manifold covered by $\mathcal{C}_j$. This projection is approximated by a ball $\mathcal{B}_j^i$ in $T_{\mathbf{x}_i}\mathcal{R}$, as shown in Figure 4.4 (bottom). The intersection of $\mathcal{B}_i$ and $\mathcal{B}_j^i$ defines a new face of $\mathcal{P}_i$ that eliminates some of its vertices (in particular the one given by $\mathbf{v}$) and generates new ones. Similarly, the polytope $\mathcal{P}_j$ associated with $\mathcal{C}_j$ is cropped using the projection of $\mathcal{C}_i$ into $\mathcal{C}_j$. When all vertices of the domain $\mathcal{P}_i$ are included in $\mathcal{B}_i$, the domain is said to be bounded. Moreover, charts whose center is out of the domain $\mathcal{D}$ are also considered bounded. If $\mathcal{R}$ has the manifold structure everywhere and r is properly set [40], then when all charts are bounded, the atlas fully covers the connected component of $\mathcal{R}$ containing the initial point $\mathbf{x}_1$ (Figure 4.5).

Algorithm 4.1 summarizes the atlas computation procedure. The algorithm receives as inputs the set $\mathbf{F}$ of equations implicitly defining $\mathcal{F}$, the set $\mathbf{X}_{\mathrm{h}}$ of representative hand configurations, the initial grasp $\mathbf{x}_1$, the dimension t of $\mathcal{F}$, the desired dimension k for $\mathcal{R}$, and the parameters r and ϵ used to construct the atlas. As output, the algorithm returns an atlas $\mathcal{A}$ of the component of $\mathcal{R}$ reachable from $\mathbf{x}_1$. The algorithm determines the postural synergies as described in Section 4.2 (line 1). The required number of constraints relative to such motions is then computed as a function of t and k (line 2) in order to obtain the set $\mathbf{R}$ of equations defining $\mathcal{R}$ (line 3). Then, $\mathcal{A}$ is initialized with a chart centered in $\mathbf{x}_1$ (line 4), and the construction of $\mathcal{A}$ proceeds while any of the

Algorithm 4.1: Computation of the atlas of $\mathcal{R}$

```
   AtlasComputation($\mathbf{F}, \mathbf{X}_\mathrm{h}, \mathbf{x}_1, t, k, r, \epsilon$)
   input  : The set $\mathbf{F}$ of equations , the set $\mathbf{X}_\mathrm{h}$ of representative
            hand configurations, the initial grasp $\mathbf{x}_1$, the
            dimension $t$ of $\mathcal{F}$, the desired dimension $k$ for $\mathcal{R}$, and
            the parameters $r$ and $\epsilon$ used to build the atlas.
   output: An atlas $\mathcal{A}$ of $\mathcal{R}$.
1  $\mathbf{E} \leftarrow$ PosturalSynergies($\mathbf{X}_\mathrm{h}$)
2  $s \leftarrow t - k$
3  $\mathbf{R} \leftarrow \mathbf{F} \cup \{\mathbf{E}_s(\mathbf{x}_\mathrm{h} - \mathbf{x}_{1,\mathrm{h}})\}$
4  $\mathcal{A} \leftarrow \{$GenerateChart($\mathbf{R}, \mathbf{x}_1, r$)$\}$
5  while not Bounded($\mathcal{A}$) do
6      $\mathcal{C}_i \leftarrow$NotBoundedChart($\mathcal{A}$)
7      $\alpha \leftarrow r$
8      $\mathbf{v} \leftarrow$ExpandibleVertex($\mathcal{C}_i$)
9      repeat
10         $\mathcal{C}_j \leftarrow$NewChart($\mathbf{R}, \mathcal{C}_i, \alpha, \mathbf{v}, r$)
11         $\alpha \leftarrow \alpha \cdot 0.9$
12     until not SimilarCharts($\mathcal{C}_i, \mathcal{C}_j, \epsilon$)
13     $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{C}_j\}$
14 return $\mathcal{A}$
```

charts can be extended (lines 5 to 13). The extension of a chart $\mathcal{C}_i$ starts by selecting a vertex of $\mathcal{P}_i$ not included in $\mathcal{B}_i$ (line 8). This vertex is used to generate a new chart $\mathcal{C}_j$ (line 10) using Eqs. (4.14) and (4.15) to determine its center. If the difference between the new chart and the previous one is too large according to Eqs. (4.17) and (4.18), chart generation is attempted closer to $\mathbf{x}_i$, i.e., with a smaller α (line 11). Otherwise, the new chart is added to $\mathcal{A}$, coordinating the chart with those already included in $\mathcal{A}$ (line 13). The computational cost of this algorithm is exponential in k and, therefore, it is only practical to compute an atlas on manifolds of moderate dimension, as it is the case of the manifold $\mathcal{R}$ herein considered.

4.3.2 Evaluating the quality of relevant grasps

Once the atlas is computed, we can identify the optimal grasp over $\mathcal{R}$ by considering an ordered set $\mathbf{Q}$ of quality indices. The indices in $\mathbf{Q}$ can be taken into account either in series or in parallel [113]. In the first case, only grasps with a minimum value for a given index are evaluated with the subsequent indices. In the second case, all indices are

Algorithm 4.2: The grasp optimization process.

```
   GraspOptimization($\mathcal{A}, d, \mathbf{Q}, \mathbf{l}, \mathbf{w}$)
   input : The atlas $\mathcal{A}$, the number $d$ of points to consider on
           each chart, the set $\mathbf{Q}$ of quality measures to be
           optimized, and the vectors $\mathbf{l}$ and $\mathbf{w}$ of lower bounds
           and weights for the quality measures, respectively.
   output: The optimal grasp $\mathbf{x}_g$.
1  $g \leftarrow -\infty$
2  $\mathbf{x}_g \leftarrow \varnothing$
3  forall the $\mathcal{C} \in \mathcal{A}$ do
4      $\mathcal{U} \leftarrow$ PointsOnChart($\mathcal{C}, d$)
5      forall the $\mathbf{u} \in \mathcal{U}$ do
6          $\mathbf{x} \leftarrow$ ChartToManifold($\mathcal{C}, \mathbf{u}$)
7          $\mathbf{q} \leftarrow \mathbf{Q}(\mathbf{x})$
8          if $\mathbf{q} \succeq \mathbf{l}$ then
9              $q \leftarrow \mathbf{w}^T \mathbf{q}$
10             if $q \geq g$ then
11                 $\mathbf{x}_g \leftarrow \mathbf{x}$
12                 $g \leftarrow q$
13 return $\mathbf{x}_g$
```

evaluated simultaneously and combined to produce a single measure, typically using a weighted sum. Algorithm 4.2 allows these two ways of considering the quality indices. The algorithm iterates over all charts in the atlas $\mathcal{A}$ (lines 3-12). For each chart $\mathcal{C}$ in $\mathcal{A}$, it generates a set $\mathcal{U}$ of d points on the tangent space associated with this chart (line 4). These points can be either computed on a regular grid or sampled randomly but, in any case, they all must lie inside the domain of $\mathcal{C}$. The number d of points to generate depends on the resolution at which the optimal grasp is required, and on the smoothness of the quality indices considered; i.e., the sharper the variations the denser the set of points. For each one of the points in $\mathcal{U}$, we obtain the corresponding point on $\mathcal{R}$ (line 6) using Eqs. (4.14) and (4.15). Then, for each point on $\mathcal{R}$ we evaluate the quality indices in **Q** (line 7). If the obtained values for the indices are all above the required thresholds in an element-wise comparison (line 8), we combine the indices (line 9), and then check whether the combined value is larger than that of the best grasp found up to the moment (lines 10-12). By setting the appropriate thresholds in $\mathbf{l}$ and using $\mathbf{w} = (0, \ldots, 0, 1)^\top$ we will obtain a serial evaluation of the indices in which the last index in **Q** will be optimized. A parallel evaluation can be obtained by using $\mathbf{l} = (-\infty, \ldots, -\infty)^\top$ and setting the desired values in $\mathbf{w}$. Mixed evaluation schemes can be obtained too, by adequately setting $\mathbf{l}$ and $\mathbf{w}$. By iterating over all charts and points, the optimal grasp over the computed points of $\mathcal{R}$ is finally identified and returned (line 13). In an extreme case, the algorithm could return no grasp if the quality indices for all of the considered grasps are below the given thresholds in $\mathbf{l}$. Note that if an optimal grasp is required with a finer precision, we only need to use Algorithm 4.2 with a larger value of d, and there is no need to recompute the atlas from scratch. In a multi-resolution optimization context, this refinement can be focused into the most promising areas previously identified. In any case, the overall cost of this algorithm is bilinear with the number of charts in the atlas, and with the number d of points considered for each chart.

4.4 Test cases

For clarity, we first exemplify the application of the method on a simple hand, and then summarize the results for the Schunk Anthropomorphic hand. All results correspond to an implementation in C of Algorithm 4.1, and in Matlab of Algorithm 4.2, running on an Intel Core 2 Duo processor at 3 GHz.

4.4.1 A planar hand

Figure 4.6 shows a planar hand with three fingers and two phalanges per finger holding an object composed of circles, where the OXY absolute reference frame is attached to the base of one of the fingers. The length and absolute orientation of the j-th phalanx of the i-th finger are given by the parameter $l_{i,j}$ and the unit vector $\hat{\mathbf{v}}_{i,j} \in \mathbb{R}^2$, respectively. Since the lengths are fixed, the configuration of the hand can be encoded in a simplified form in this case, by the vector $\mathbf{x}_\mathrm{h} = (\hat{\mathbf{v}}_{1,1}^\top, \ldots, \hat{\mathbf{v}}_{3,2}^\top)^\top$ subject to the constraints

$$\|\hat{\mathbf{v}}_{i,j}\|^2 = 1, \tag{4.19}$$

for all phalanges. Thus, Eq. (4.1) is the system formed by Eqs. (4.19). Since this system contains 6 equations in 12 variables, its solution set is of dimension 6, which agrees with the number of degrees of freedom of the hand. Note that $\mathbf{x}_\mathrm{h}$ only includes parameters for the orientation of the hand links, since their position vectors can be computed from the orientations and the constant length parameters [80].

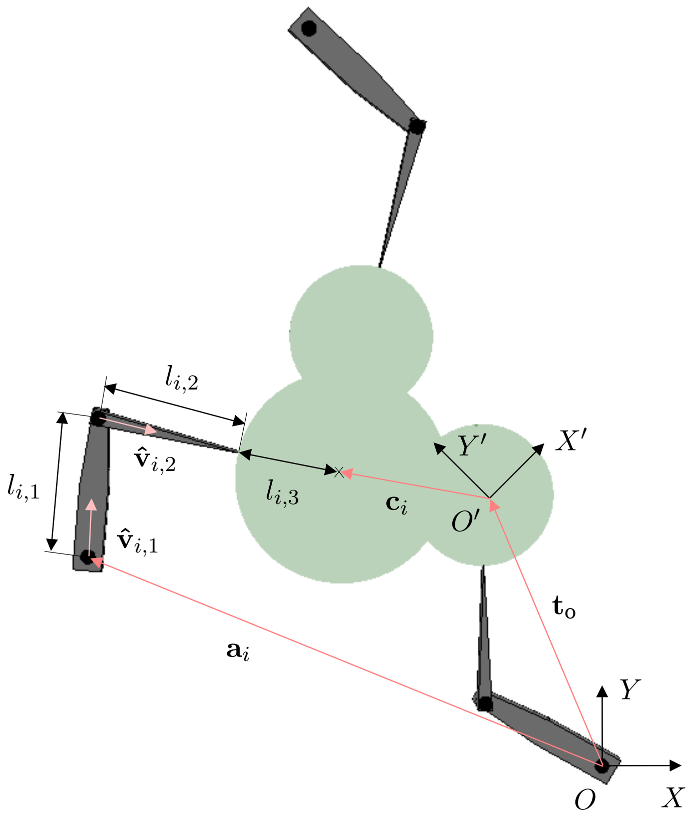


Figure 4.6: A simple planar hand with three fingers holding an object. The parameters are indicated for one finger only, but apply to the three fingers.

A local reference frame $O'X'Y'$ is attached to the object, whose pose in the absolute frame is given by $\mathbf{x}_\mathrm{o} = (\mathbf{t}_\mathrm{o}, \hat{\mathbf{v}}_\mathrm{o})$, where $\mathbf{t}_\mathrm{o} = (x_\mathrm{o}, y_\mathrm{o})^\top$ is the position vector of O' and $\hat{\mathbf{v}}_\mathrm{o} = (s_\mathrm{o}, c_\mathrm{o})^\top$ is a unit vector aligned with the X' axis. Then, Eq. (4.2) becomes

$$\|\hat{\mathbf{v}}_\mathrm{o}\|^2 = 1. \tag{4.20}$$

In this example, the contact regions in each fingertip reduce to a point and the explicit expression of Eq. (4.3) is

$$\mathbf{h}_i = \mathbf{a}_i + l_{i,1}\,\hat{\mathbf{v}}_{i,1} + l_{i,2}\,\hat{\mathbf{v}}_{i,2}, \tag{4.21}$$

where $\mathbf{a}_i$ is the position vector of the palm anchor point of finger i in the absolute frame, and Eq. (4.4) providing the associated normal is

$$\hat{\mathbf{m}}_i = \hat{\mathbf{v}}_{i,2}. \tag{4.22}$$

Moreover, the contact regions on the object are arcs of circumference given by a single parameter. Thus, for each one of such arcs, Eq. (4.5) boils down to the following expression

$$\mathbf{o}_i = \mathbf{t}_\mathrm{o} + \begin{bmatrix} c_\mathrm{o} & -s_\mathrm{o} \\ s_\mathrm{o} & c_\mathrm{o} \end{bmatrix} (\, \mathbf{c}_i + l_{i,3}\, \hat{\mathbf{w}}(u_i)), \tag{4.23}$$

where $\mathbf{c}_i$ is the center of the circumference in local coordinates of the object, $l_{i,3}$ is its radius, and

$$\hat{\mathbf{w}}(u_i) = \begin{bmatrix} \cos u_i \\ \sin u_i \end{bmatrix}, \tag{4.24}$$

with $u_i \in [a_i, b_i]$, is the angular range defining the arc for contact patch i. Similarly, Eq. (4.6) reduces to

$$\hat{\mathbf{n}}_i = \begin{bmatrix} c_\mathrm{o} & -s_\mathrm{o} \\ s_\mathrm{o} & c_\mathrm{o} \end{bmatrix} \hat{\mathbf{w}}(u_i), \tag{4.25}$$

in this case. Thus, Eq. (4.9) encompasses Eqs. (4.19) to (4.25), defining a set $\mathcal{F}$ of feasible grasps of dimension $t = 3$. The proposed optimization procedure can be directly applied to problems of this dimension. However, to complete the example and to facilitate the visualization of the results, it is better to reduce the dimension to obtain a set of relevant grasps $\mathcal{R}$ of dimension $k = 2$. Thus, according to Eq. (4.12), $s = 1$ linear constraints given by Eq. (4.13) must be added to Eq. (4.9), in order to get Eq. (4.11). In this example, the set $\mathbf{X}_\mathrm{h}$ used to reduce the dimensionality contains randomly generated hand configurations.

Figure 4.7 shows the results obtained when applying the proposed method to this example. Two complementary views of the computed atlas are depicted (left), together with the best and worst grasps found (center and right, respectively). The atlas was obtained using Algorithm 4.1 with $r = 0.125$ and $\epsilon = 0.4$ in about 0.1 seconds. It contains a total of 750 charts, whose polytopes $\mathcal{P}_i$ form the shown hexagonal-like mesh.

To optimize the grasp, the atlas was evaluated using Algorithm 4.2 under the force-closure quality index reported in [83] normalized to the range $[0, 1]$, obtaining the results shown in Figure 4.7, where green and red charts respectively correspond to configurations with a large and low value of the index. Thus, Algorithm 4.2 was called with **I** containing only this index in this case, and setting $\mathbf{l} = 0$ and $\mathbf{w} = 1$. The contacts were modeled as frictional point contacts with a friction coefficient of

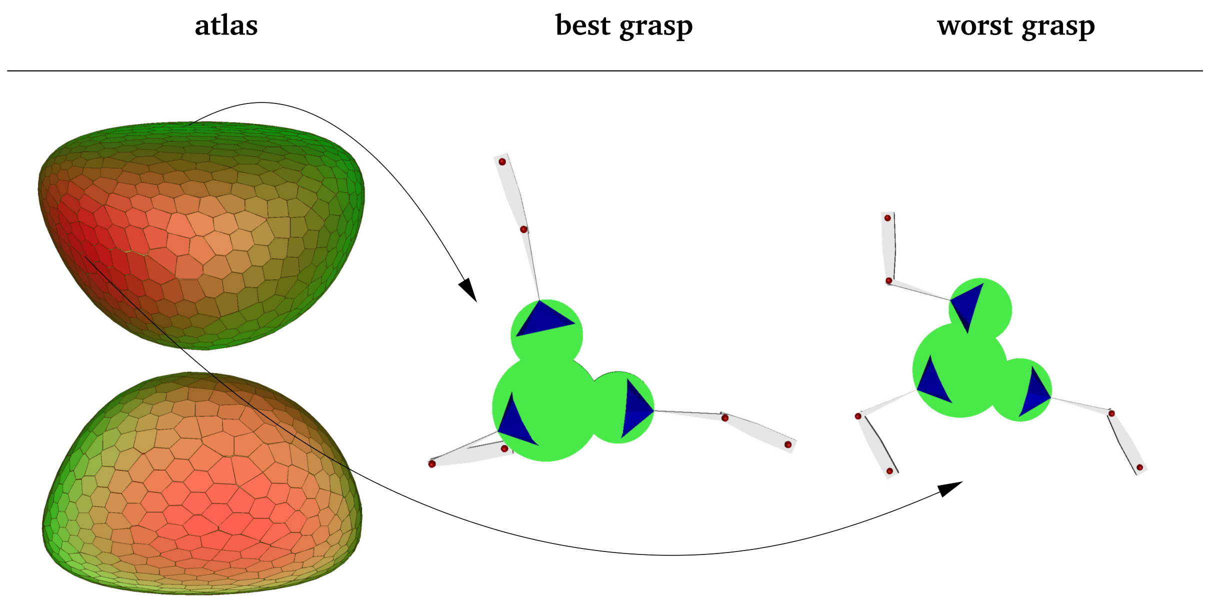


Figure 4.7: Two views of the atlas of the set $\mathcal{R}$ of relevant grasps on the planar hand example, together with the best and worst grasp configurations obtained, according to the force-closure quality index in [83]. The views have been obtained by projecting the atlas on three of the problem variables. Green and red points in the views correspond to grasps with large and low values of the index, respectively.

$\mu = 1$, resulting in the shown friction cones of 45°. Since in this case the quality index is smooth at the resolution of the computed atlas, we set $d = 1$ in Algorithm 4.2, which corresponds to evaluating one point of each chart only, e.g. its center point. Algorithm 4.2 evaluated the whole atlas in about 10 seconds in this situation. Note that if the kinematic constraints of the hand were neglected, the optimal force-closed grasp would have the contact normals equi-distributed in the plane, i.e., with angles of 120° between them [65]. However, this configuration is not reachable in our case, because it does not satisfy the joint-assembly and contact constraints of the hand-object system considered. The example, thus, emphasizes the relevance of taking into account both the kinematic and contact constraints when optimizing a given grasp, as proposed in this chapter.

To illustrate a case where the quality index exhibits multiple local optima over $\mathcal{R}$, the same atlas was evaluated according to the normalized inverse condition number of the manipulability ellipsoid proposed in [7], which is actually the ratio between the smallest and largest axis lengths of this ellipsoid. Algorithm 4.2 took 1.5 seconds in

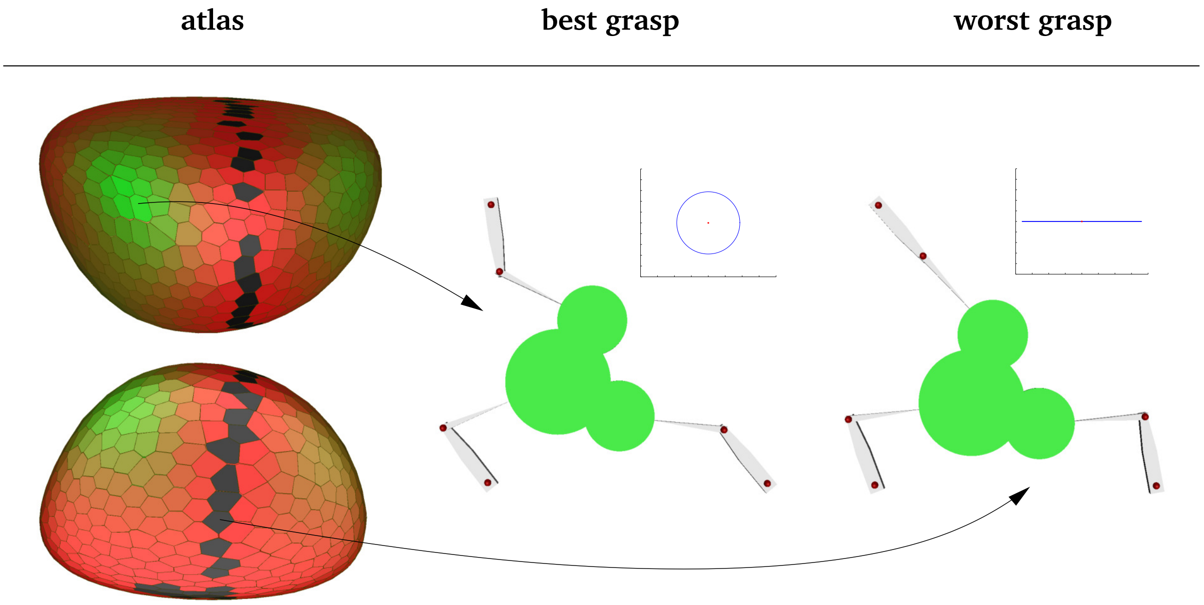
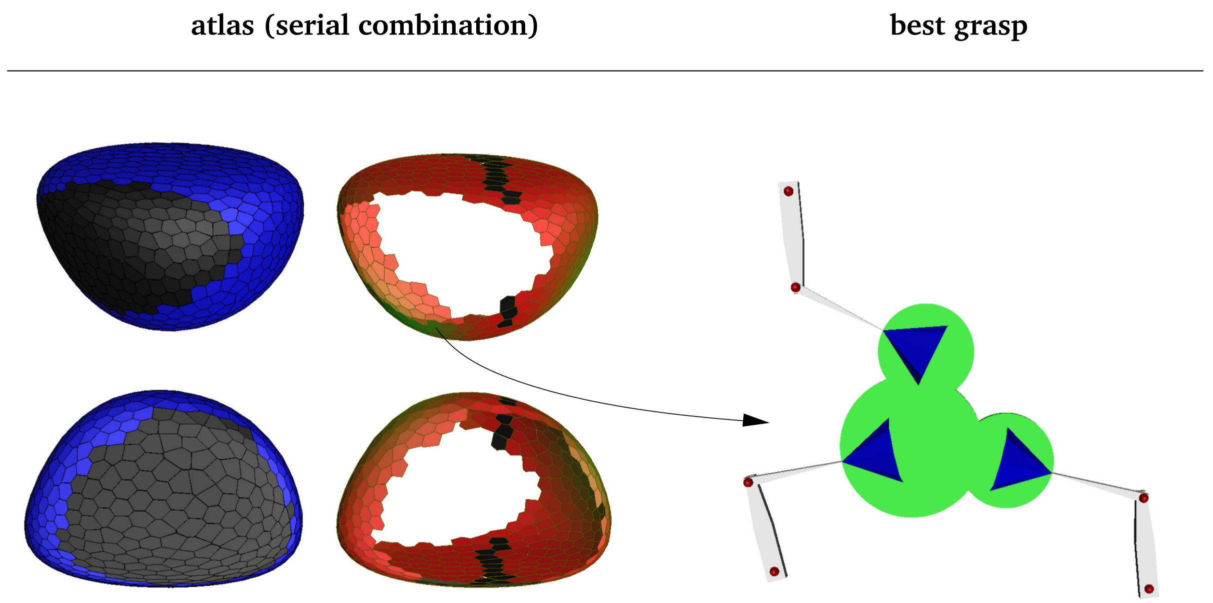


Figure 4.8: The same atlas of Figure 4.7, but now colored according to the manipulability index defined in [7]. Green, red points correspond to grasps with a large and low value of this index, respectively. Black corresponds to near-singular grasps. Note that, this measure yields three local optimal grasps.

this case, and produced the results shown in Figure 4.8, where the atlas views coincide with those in Figure 4.7, but are now colored according to the new index. In the figure, green and red correspond to grasps with large and low values of this index, and black corresponds to grasps where the index is below 10^{-3}. In such near-singular grasps, the manipulability ellipsoid flattens in at least one direction, meaning that the hand can hardly move the object along that direction while maintaining the required contacts. Figure 4.8 shows 2D projections of this ellipsoid revealing this fact, for the best and worst grasp configurations found. One of the fingers is fully extended in the worst configuration, which is in agreement with the inverse singularity condition of the 3-RRR parallel manipulator equivalent to this grasp [**?**]. Note from the figure that this quality index would pose difficulties to local optimization methods, since the index yields three local optimal grasps.

Finally note that, in a precision manipulation task, both the force-closure and the manipulability criteria may need to be taken into account. However, as it can be appreciated when comparing Figures 4.7 and 4.8, these two criteria are conflicting in this example. The global optimum in Figures 4.8, for example, corresponds to a point



atlas (serial combination) **best grasp**

Figure 4.9: The same views of the atlas of Figures. 4.7 and 4.8, but now evaluated under a serial approach that combines the force-closure and manipulability indices. The regions with a force-closure index above 0.2 are indicated in blue in the atlas. These regions are then evaluated according to the same manipulability index used in Figure 4.8 to select the best configuration on them.

with a low value of the force closure index in Figures 4.7. We use a serial evaluation approach, and optimize the manipulability index only for those grasps with a minimum value of the force-closure index. Figures 4.9 shows the result of such strategy, obtained by applying Algorithm 4.2 with $\mathbf{I}$= $(I_1, I_2)^\top$, where I_1 and I_2 are the force-closure index and the inverse condition number of the manipulability ellipsoid, respectively, and using $\mathbf{l} = (0.2, 0)^\top$ and $\mathbf{w} = (0, 1)^\top$. These thresholds are set so that about half of the charts are discarded using the force-closure index. Clearly, the optimal grasp would be different if the indices were considered in the reverse order or with different thresholds.

4.4.2 The Schunk Anthropomorphic hand

To validate the approach on a complex grasping device, we have applied it to the Schunk Anthropomorphic hand. Assuming that all joints are independently actuated, this hand has 13 degrees of freedom and, formulated according to Chapter 3, Eq. (4.1) is a hand system of 88 equations in 101 variables. The tests are performed using three objects:

(1) a can, (2) a jeweler's screwdriver, and (3) a Marquina oil bottle. In all cases, the grasps are to be performed using three fingers, and the complexity of the examples is determined by the dimension and the distribution of the contact patches.

In the case of the can, the contact patches at the fingertips are punctual. Moreover, the contact patch on the object for the thumb is reduced to a line along the can to avoid repeated solutions caused by the axial symmetry of the can, but the two other fingers are allowed to contact an identical cylindrical patch defined all over the surface of the can. Despite involving only one contact patch, the large extension of this patch makes this test case a hard one, because a large atlas will have to be computed. In the case of the screwdriver, a point on the index fingertip is set to be in contact with the flat head (to be able to apply forces that ensure a proper contact of the screwdriver with the screw), a point on the thumb is limited to move on a line along the screwdriver's body (to avoid symmetric solutions), and a curve on one side of the last phalanx of the third finger is constraint to contact the screwdriver's body at any point. Thus, this example illustrates the applicability of the method under different contact models. Finally, to properly dispense oil with the Marquina oil bottle, a point on the index finger must contact the top of the bottle along a curve, and points on the two other fingers must touch patches in the middle and bottom sections of the bottle, respectively. Therefore, the fingers contact the object on three disjoint patches with different sizes and orientations, which represents a general situation for the proposed approach.

After the contact constraints are imposed, Eq. (4.9) involves 128 equations in 136 variables (and hence $t = 8$) in the first example, 131 equations in 141 variables (and hence $t = 10$) in the second case, and 133 equations in 142 variables (and hence $t = 9$) in the third example. Thus, in order to obtain a set $\mathcal{R}$ of dimension $k = 2$, s must be set to 6, 8, and 7, respectively. By removing at most 8 postural synergies out of 13, we still retain more than 99% of the motion capability of the hand. In this case, the synergies are computed from a database of human hand configurations.The atlases for the three examples were obtained in 170, 180, and 18 seconds using Algorithm 4.1 with $r = 1$ and $\epsilon = 0.5$, and they include 4900, 4800, and 400 charts, respectively. These times agree with the fact that the stronger the contact constraints, the less the number of charts in the atlas, and thus, the faster the generation of such atlas.

Figure 4.10 shows the results of applying Algorithm 4.2 with $d = 1$ on the atlases obtained on the three examples, using the force-closure quality index with the same friction coefficient $\mu = 1$ used in the planar hand example. In this case, the spatial

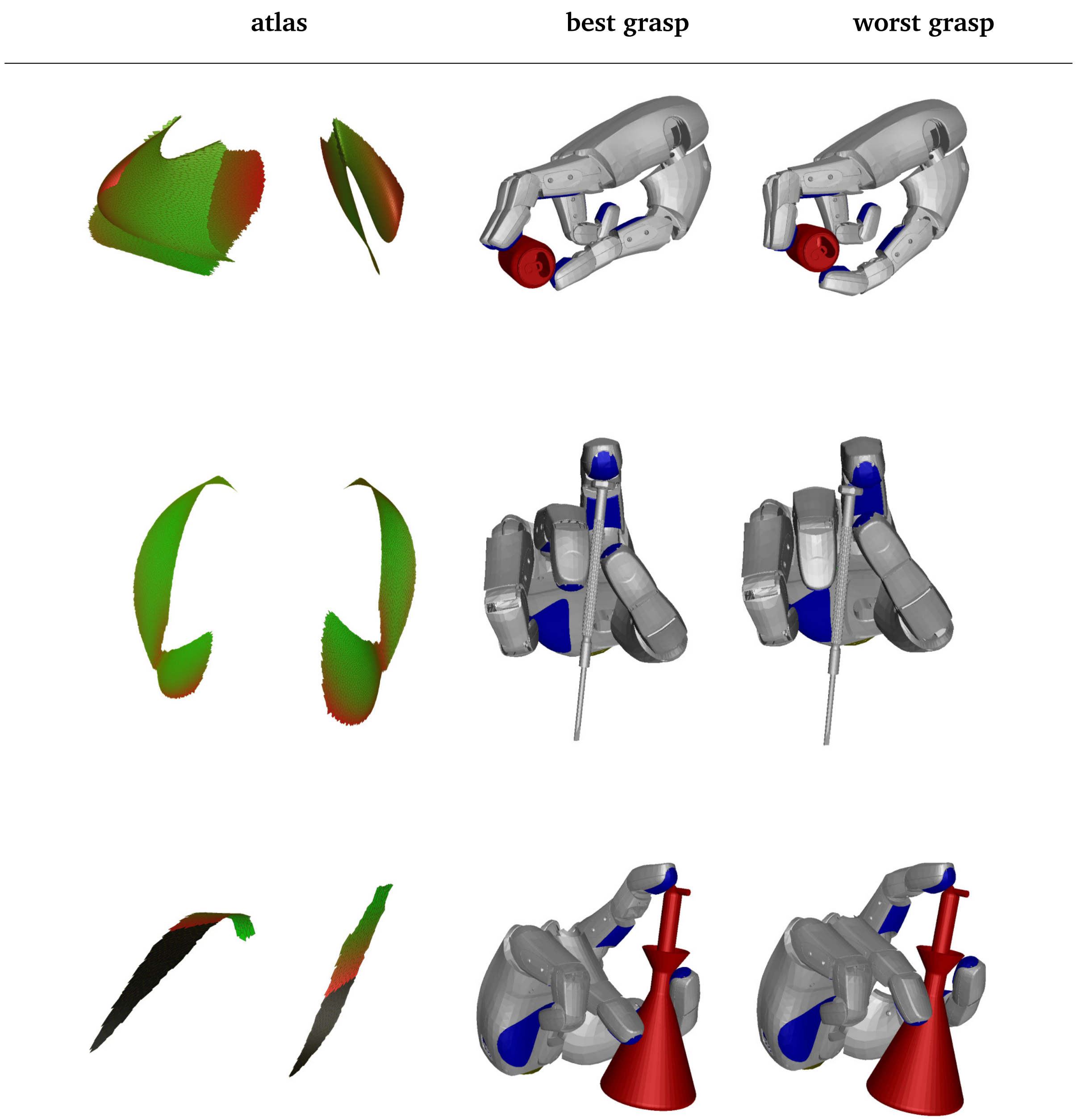


Figure 4.10: Optimization of the force-closure quality index described in [83] for the Schunk anthropomorphic hand grasping a can (top), a jewelry screwdriver (middle), and an oil drizzler (bottom), assuming punctual contacts with a friction coefficient of $\mu = 1$. Two views of the atlas obtained for each object are shown, where green and red points correspond to configurations with large and small values of the index, respectively, and black points correspond to non-force-closed grasps.

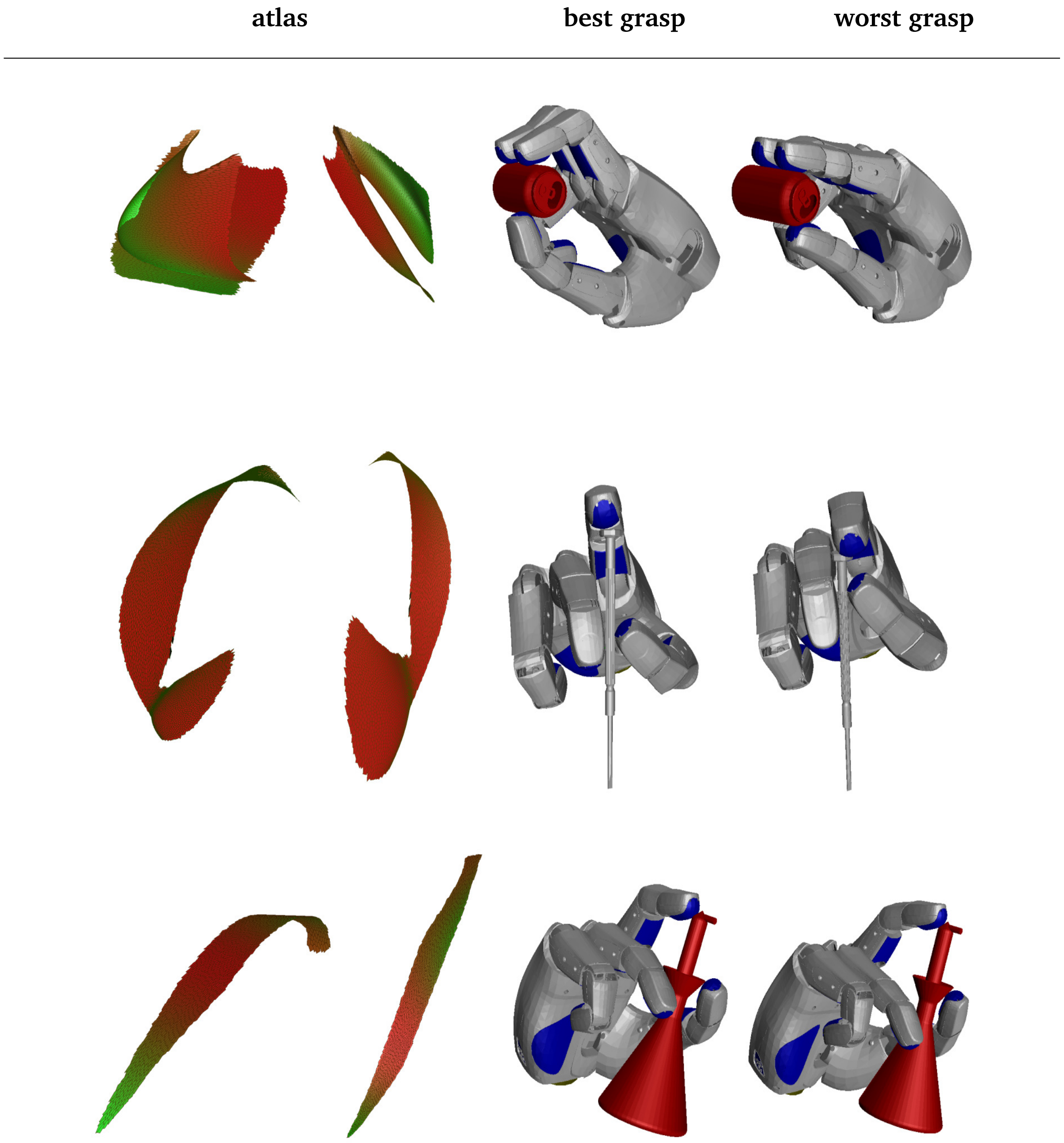


Figure 4.11: Optimization of the inverse of the condition number of the manipulability ellipsoid defined in [7], for the Schunk anthropomorphic hand grasping the same objects as in Figure 4.10. Two views of the atlas obtained for each object are shown, where green, red, and black points respectively correspond to grasps with a large, low, and null value of this index.

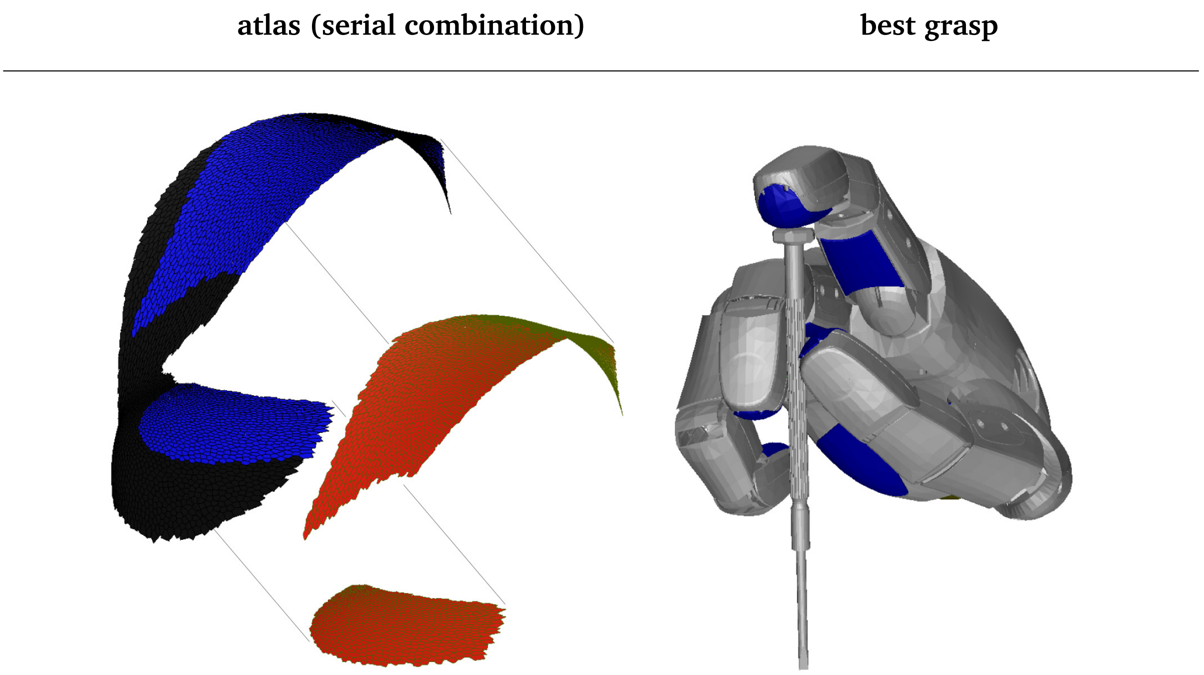


Figure 4.12: The best grasp found when optimizing the manipulability criterion on those grasps that exhibit a minimum value of the force-closure index (the blue regions indicated in the atlas).

friction cones are linearly approximated using 8 generators. The optimization for the three examples took 115, 120, and 9 seconds, respectively. These times basically scale with the number of charts in the atlas.

Figure 4.11 shows the results of applying again Algorithm 4.2 on the same three atlases, but now considering the inverse condition number of the manipulability ellipsoid, as done in Figure 4.9. The evaluation took 17, 16, and 1.5 seconds, respectively. In this case the execution of Algorithm 4.2 is faster because here we only have to solve a relatively small eigenvalue problem for each tested grasp.

In the oil bottle test case, we also optimized the manipulability criterion with $k = 3$, simply by considering one additional principal hand motion. In this case, the atlas includes 12800 charts and the optimization takes 1050 seconds. However, the optimal grasp returned is almost the same as that obtained with $k = 2$, which confirms that, in this example, 6 postural synergies are enough to capture most of the mobility of the hand.

Finally, we use the screwdriver example to illustrate the results of optimizing the grasp under a criterion combining two quality indices. In this case, we have computed the grasp that maximizes the manipulability criterion, subject to exhibit a normalized force-closure quality index equal to or larger than 0.675 (i.e., using $\mathbf{l} = (0.675, 0)^\top$ and $\mathbf{w} = (0, 1)^\top$ in Algorithm 4.2), obtaining the grasp shown in Figure 4.12. Note how removing the lower quality force-closed grasps results in two separate regions where to optimize the manipulability. Despite one of the regions yields better grasps than the other, a local optimizer might get stuck in the bad region, if not initialized in the region including the optimum grasp.

4.5 Summary

This chapter has proposed a procedure for optimizing the quality of a robotic grasp subject to satisfying the specified contact constraints established between the hand and the object. The procedure enforces the satisfaction of all kinematic and contact constraints of the hand-object system during the whole optimization process, and it is global, in the sense that it explores the whole set of relevant grasps attainable from a given point, determining the optimal grasp at the desired resolution, without getting trapped into local extrema. The method is also general, meaning that it can be applied to any hand structure, and to any desired set of quality indices. The efficiency of the method critically depends on the dimension of the traced manifold. In the case of grasps, however, postural synergies allow reducing the dimension of such manifold considerably. Actually, the proposed method can keep a large number of postural synergies (up to 7 out of 13 for the Schunk hand), while previous methods [20] use a smaller number of them (typically 2). This is because the method proposed here integrates the contact constraints a priori, which already introduces a large dimension reduction.

5

Accounting for Compliance in the Grasp

This chapter proposes a solution to the grasp synthesis problem in the case where compliance is present in both the hand and the object. In the two previous chapters, a grasp was obtained under the assumption that the contacting bodies and the actuation were rigid. However, there are cases in which small deformations occur when applying the required forces through the contacts, and the hand joints should accommodate for such deformations to keep the object firmly held. To tackle this issue, we present a kinetostatic formulation that models the compliance at the joints and the contacts by introducing springs. This provides a proper framework to find a kinematically feasible and restrained grasp by considering simultaneously the necessary constraints. As a consequence, a solution of the proposed system of equations results in a set of hand configurations that allow to execute the grasp using a position controller.

Several hands are designed with soft elements, with the purpose of increasing the performance of the grasp under different types of situations, such as uncertainty in the geometric models and the finger positioning, and the lack of precise contact force measurements (Figure 5.1). Even though the rigid body approach is good enough in many practical applications, considering the finger and contact compliances introduced by the various soft elements would yield a more accurate grasping model. A similar problem is found as well on the synthesis of stable configurations of tensegrities [46, 55], since having a structure supported by beams and cables results in a considerably elastic structure. The compliance in a grasp has been determined to be a function of servo compliance, structural compliance, changes in the grasp geometry, coupling among the different joints and fingers, and different contact types and changes in the contact locations, e.g. due to rolling or sliding [25]. Recent works account for these compliances to some extent [19, 44], but they either miss some of the necessary constraints, limit the approach to planar grasps, or use a reduced number of fingers.

This chapter proposes a more general solution to the synthesis of compliant grasps using a kinetostatic formulation inspired by the notion of *soft synergies* [6, 34, 81, 82]. The approach effectively introduces an elastic model of the hand whereby the *physical* hand is attracted towards a *reference* hand through a set of virtual springs at the joints, while being repelled by the object through springs at the contacts, which are also supposed to be compliant. A system of equations is proposed to account for the behaviour of such model, which implicitly enforces the necessary constraints to find a kinematically-feasible and restrained grasp in the presence of compliant elements. As a consequence, any solution to this system allows to execute the grasp using a position controller, because it provides the actuated-joint setpoints that keep the object held after the deformation of the hand-object system. The dimension of the problem is considerably large, so that the use of a local search method becomes inevitable to compute a solution. However, the method can be properly initialized using an estimation of the solution computed with the methods in Chapters3 and 4, for instance, in combination with joint torque determination methods [49].

5.1 Kinetostatic formulation of a grasp

The formulation involves the specification of three hand configurations, as shown in Figure 5.2. The outer hand configuration (black) accounts for the kinematic-feasibility constraints, i.e. whether the fingertips can actually reach the object surface given the kinematic structure of the hand. The inner hand configuration (blue) produces joint torques that attract the outer and yield an interaction with the object. Such interaction generates reaction forces that repel the hand. Thus, the middle hand configuration (green) is the actual grasp where the joint torques and contact forces are balanced accounting for restrainability.

5.1.1 Model description

For simplicity, in this chapter we assume *precision grasps*, i.e. that only one contact per finger at the fingertip is used. Precision grasps are most often used for dextrous manip-

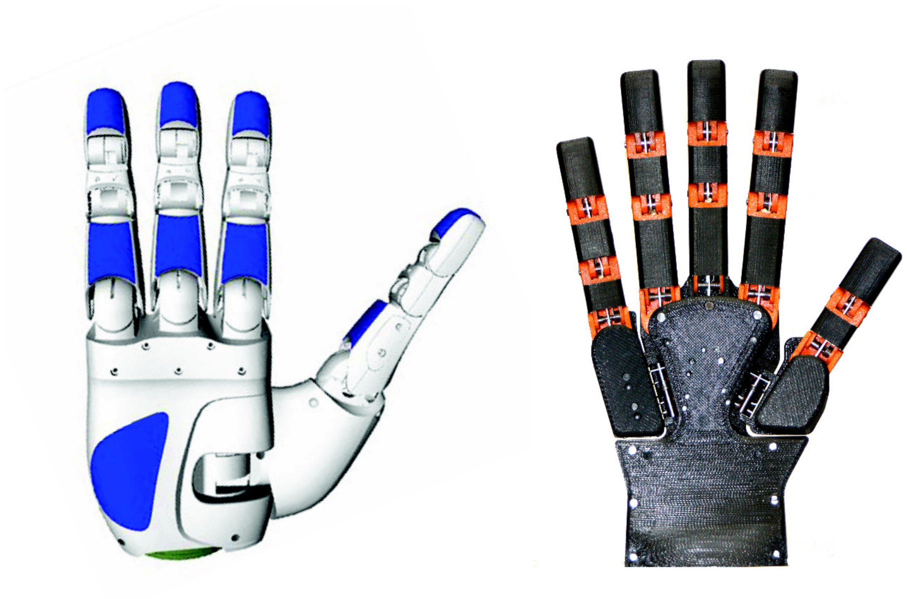


Figure 5.1: THE First hand [8] (right), developed at the Interdepartamental Research Center “E. Piaggio”, is cable-driven, which is considerably more compliant than a gear-driven system, such as that of the Schunk Anthropomorphic hand [106] (left), based on the DLR II hand.

ulation [24]. While the soft synergy model can easily account for inner-hand contacts and power grasps, the algorithmic complexity grows with the number of contacts to place and with the number of kinematic constraints.

A robotic hand is usually composed of several articulated fingers attached to a palm. The hand palm is positioned and oriented with respect to the world using the matrix $\mathbf{T}_H \in SE(3)$. The hand is composed of n fingers, each of them articulated through m_i revolute joints, for $i = 1, \ldots, n$, which sum up to $m = \sum_{i=1}^{n} m_i$ hand joints. The rotation angle of the j-th joint at the i-th finger is the joint value $q_{ij} \in \mathbb{S}$, where $\mathbb{S}$ denotes the angular nature of values. The phalanges are positioned and oriented with respect to the world using the homogeneous matrix $\mathbf{T}_{ij}$, which depends on the joint values, q_{ij}, for $j = 1, \ldots, m_i$ and $i = 1, \ldots, n$. Thus, by collecting all joint values in the vector $\mathbf{q} = (q_{11}, \ldots q_{ij} \ldots, q_{nm_n})$, a configuration of the hand is represented by the pair $(\mathbf{q}, \mathbf{T}_H) \in \mathbb{S}^m \times SE(3)$.

There are n given contact points on the hand, specifically at the fingertips. A reference frame, positioned and oriented using the matrix $\mathbf{X}_c \in SE(3)$, is placed at the contact point $\mathbf{x}_c \in \mathbb{R}^3$, for $c = 1, \ldots, n$. The outward normal vector with respect to

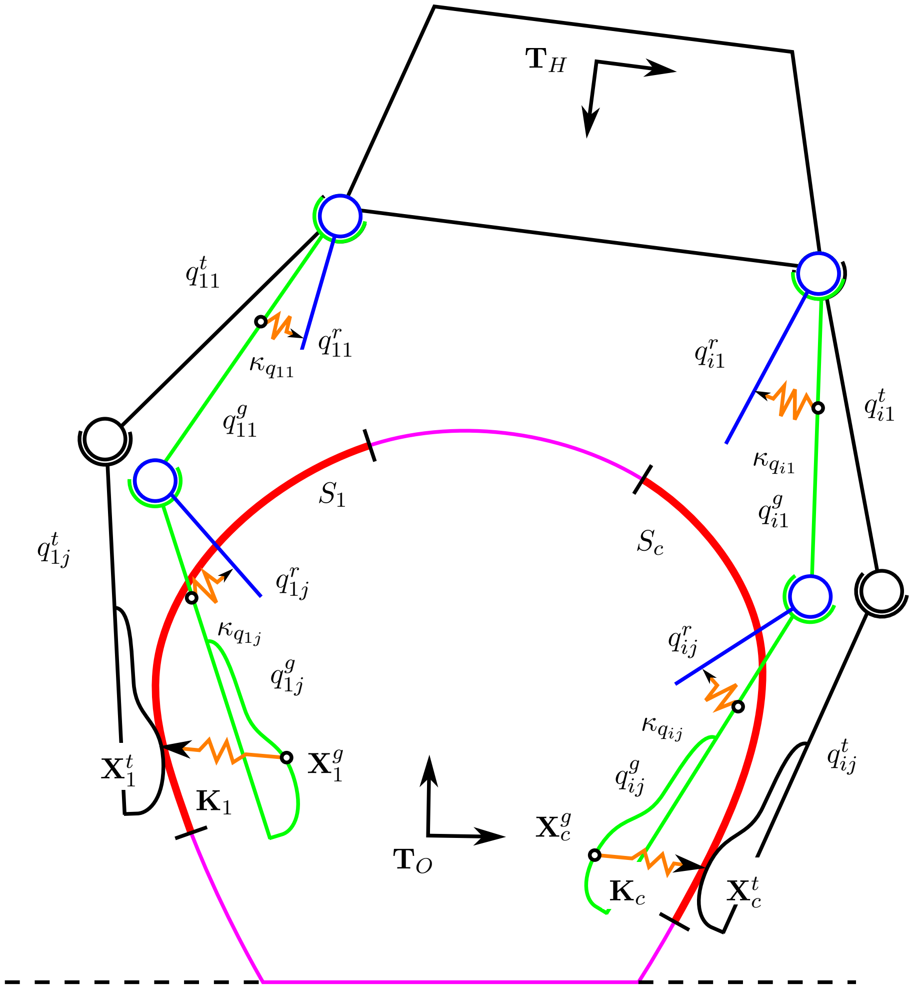


Figure 5.2: Kinetostatic model using springs at the joints and at the contact. The grasp is characterized by three hand configurations accounting for the reachability and restrainability of a grasp.

the fingertip surface at the contact point is denoted as $\hat{\mathbf{n}}_c$. Both the contact point and the contact normal depend on the hand configuration $(\mathbf{q}, \mathbf{T}_H)$. The number of contact locations, n, is assumed to be, at least, the minimum required to restrain the object according to the chosen contact model (e.g. $n = 7$ for non-frictional contacts spatial grasps (Section 2.1).

The object position and orientation with respect to the world is defined using the matrix $\mathbf{T}_O \in SE(3)$. Without loss of generality, the object frame represented by $\mathbf{T}_O$ is considered to be the absolute coordinate frame. On the object surface, there is a given contact region, S_c, for each contact point, $\mathbf{x}_c$, on the hand, defined by the

Symbol	Definition
$\mathbf{T}_H$	Hand reference frame
$\mathbf{T}_O$	Object reference frame
m	Number of joints
n	Number of fingers and contacts
q_{ij}	Value of the j-th joint at the i-th finger
$\kappa_{q_{ij}}$	Stiffness of torsional spring of the j-th joint at the i-th finger
$\mathbf{q}^t$	Touching joint configuration
$\mathbf{q}^g$	Grasping joint configuration
$\mathbf{q}^r$	Reference joint configuration
$\delta\mathbf{q}$	Joint displacement from $\mathbf{q}^g$ to $\mathbf{q}^r$
$\mathbf{X}_c^g$	Contact reference frame, with origin at the point $\mathbf{x}_c^g$
$\mathbf{X}_c^t$	Contact reference frame, with origin at the point $\mathbf{x}_c^t$
$\delta\mathbf{X}_c$	The displacement from $\mathbf{X}_c^t$ to $\mathbf{X}_c^g$
$\mathbf{K}_c$	Stiffness of the spatial spring at the c-th contact point
h_c	Number of linear springs used at the c-th contact point
S_c	Contact region on the object corresponding to the c-th contact point
$\boldsymbol{\gamma}_c$	Parameter that defines a point on S_c
a_c	Dimensionality of region S_c (point, curve, surface)
b_c	Orientation freedom at the c-th contact (normal alignment or not)
s_c	Contact point on the region S_c defined by $\boldsymbol{\gamma}_c$
$\hat{\mathrm{m}}_c$	Normal vector at the point s_c defined by $\boldsymbol{\gamma}_c$
$\mathbf{J}$	Hand Jacobian evaluated at the grasping configuration
$\mathbf{G}$	Grasp matrix evaluated at the grasping configuration
$\mathbf{K}_q$	Hand stiffness matrix
$\mathbf{K}$	Contact stiffness matrix

Table 5.1: Notation used in the kinetostatic model of a grasp.

parametrization $\mathrm{s}_c(\boldsymbol{\gamma}_c)$, where $\boldsymbol{\gamma}_c \in \mathbb{R}^{a_c}$, with $a_c = 0, 1, 2$, the vector of parameters defining a point, a curve or a surface, respectively. The parametric inward normal vector with respect to the object at the point s_c, is denoted as $\hat{\mathrm{m}}_c(\boldsymbol{\gamma}_c)$.

The contact forces and joint torques are modeled using a spatial spring at the contacts and a torsional spring at the joints with known stiffness constants $\mathbf{K}_c$ and $\kappa_{q_{ij}}$, respectively. The rest position for the contact springs is defined by the touching configuration of the hand, $(\mathbf{q}^t, \mathbf{T}_H)$, that makes the desired contact point, $\mathbf{x}_c^t$, reach the corresponding region, S_c, on the object. The reference configuration of the hand, $(\mathbf{q}^r, \mathbf{T}_H)$, pulls the fingers inside the object surface loading the springs at the joints. The grasping configuration, $(\mathbf{q}^g, \mathbf{T}_H)$, floats between those two, modifying the lenght of the joint springs when the contact frame is moved to $\mathbf{X}_c^g$, loading the contact springs and pushing the fingers out of the object back to $\mathbf{X}_c^t$. The list of symbols is summarized in Table 5.1.

5.1.2 Characterizing kinematically-feasible grasps

A grasp is kinematically-feasible when the touching configuration of the hand, $(\mathbf{q}^t, \mathbf{T}_H)$ makes the fingertips contact properly on the corresponding regions on the object, i.e. $\mathbf{x}_c^t \in S_c$. Thus, the contact constraint is written as

$$\|\mathbf{x}_c^t - \mathbf{s}_c(\gamma_c)\|_2 = 0. \tag{5.1}$$

In order to avoid interpenetrations of the fingertips into the object, the vectors normal to the fingertip surface, $\hat{\mathbf{n}}_c^t$, and to the object surface, $\hat{\mathbf{m}}_c$ at the contacting points are aligned by requiring

$$\hat{\mathbf{n}}_c^t \cdot \hat{\mathbf{m}}_c(\gamma_c) = 1. \tag{5.2}$$

And, the position vector $\mathbf{r}_{ij}$ of matrices $\mathbf{T}_{ij}$ are forced to be outside the object by requiring

$$\hat{\mathbf{m}}_c^{\mathsf{T}}(\mathbf{s}_c - \mathbf{r}_{ij}) > 0, \tag{5.3}$$

that is, a positive projection of the vector going from $\mathbf{r}_{ij}$ to $\mathbf{s}_c$ onto the diretion normal to the object at the contact point.

Finally, the joint values of real robotic hands are subject to mechanical limitations. Hence, the touching configuration must fulfill the preceding constraints while the joint values stay within the valid range

$$\mathbf{q}^l \leq \mathbf{q}^t \leq \mathbf{q}^u, \tag{5.4}$$

with $\mathbf{q}^l$ and $\mathbf{q}^u$ being the vectors of minimum and maximum values, respectively.

5.1.3 Characterizing restrained grasps

A grasp is restrained when the object motion due to external perturbations is prevented by applying valid contact forces according to the contact model and to the grasping configuration (*object equilibrium*), and such contact forces are balanced by applying joint torques according to the reference configuration (*hand equilibiurm*) [19]. It is worth noting that, the restrainability condition, together with the assumption that n is the minimum number of contacts required to restrain the object according to the contact model, yields a force-closure grasp as defined in [5].

Object equilibrium

Each contact force is modeled using a spatial springs conformed of h_c linear springs connecting the frames $\mathbf{X}_c^t$ and $\mathbf{X}_c^g$ at the contact (see Figure 5.3). Thus, we express the effect of these springs acting on the object, i.e. the c-th contact force, as the sum of all spring forces as $\mathbf{w}_c = [\ \hat{\mathbf{p}}_{1,c} \ \ \ldots \ \ \hat{\mathbf{p}}_{h_c,c}\ ]\boldsymbol{\lambda}_c$, where $\hat{\mathbf{p}}_{k,c} \in \mathbb{R}^6$ represents the supporting line of the k-th spring passing through the contact point $\mathbf{x}_c^g$, and $\boldsymbol{\lambda}_c = [\ \lambda_{1,c} \ \ \ldots \ \ \lambda_{h_c,c}\ ]^{\mathsf{T}}$ collects the force magnitudes of the springs obtained as $\lambda_{k,c} = -\kappa_{k,c} d_{k,c}$, where $d_{k,c}$ is the spring elongation and $\kappa_{k,c}$ the stiffness constant of the k-th spring. Thus, the magnitude of the contact force can be written as $\boldsymbol{\lambda}_c = -\tilde{\mathbf{K}}_c \mathbf{d}_c$ using the diagonal matrix $\tilde{\mathbf{K}}_c = \text{diag}(\kappa_1, \ldots, \kappa_{h_c})$ and $\mathbf{d}_c = [\ d_{1,c} \ \ \ldots \ \ d_{h_c,c}\ ]^{\mathsf{T}}$. Then, introducing the matrix $\mathbf{P}_c = [\ \hat{\mathbf{p}}_{1,c} \ \ \ldots \ \ \hat{\mathbf{p}}_{h_c,c}\ ]$, we express the c-th contact force as

$$\mathbf{w}_c = -\mathbf{P}_c \tilde{\mathbf{K}}_c \mathbf{d}_c. \tag{5.5}$$

The displacement that goes from $\mathbf{X}_c^t$ to $\mathbf{X}_c^g$ can be parametrized using six independent variables, known as the exponential coordinates [68], expressed as

$$\mathbf{X}_c^g = e^{(\delta \mathbf{X}_c)} \mathbf{X}_c^t, \tag{5.6}$$

where $e^{(\delta \mathbf{X}_c)}$ is the exponential map representing the relative finite rigid body displacement, $\delta \mathbf{X}_c \in se(3)$, between $\mathbf{X}_c^t$ and $\mathbf{X}_c^g$. The spring elongations are obtained by projecting the displacement onto the supporting lines of the springs as

$$\mathbf{d}_c = \mathbf{P}_c^{\mathsf{T}} \delta \mathbf{X}_c. \tag{5.7}$$

Substituting Eq. (5.7) in Eq. (5.5) yields the expression of the contact force as a function of the touching and grasping configuration,

$$\mathbf{w}_c = -\mathbf{K}_c \delta \mathbf{X}_c, \tag{5.8}$$

where $\mathbf{K}_c = \mathbf{P}_c \tilde{\mathbf{K}}_c \mathbf{P}_c^{\mathsf{T}}$ models the compliance of the c-th contact.

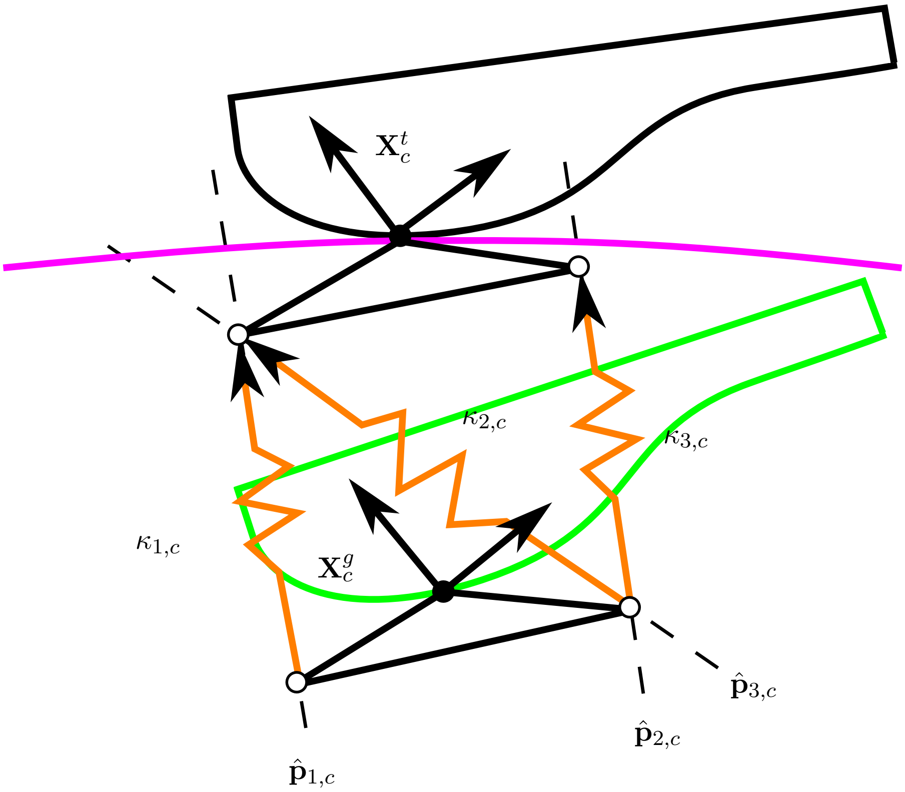


Figure 5.3: Representation of the spatial spring placed at the contact locations, using $h_c = 3$ linear springs. The triangles are drawn for clarity, however, the vertices are coincident with the contact point, making $\mathbf{p}_{1,c}$, $\mathbf{p}_{2,c}$ and $\mathbf{p}_{3,c}$, pass through the origin of the frame placed at the contact point at the touching (black) and the grasping (green) configuration, represented by $\mathbf{X}_c^t$ and $\mathbf{X}_c^g$, respectively. The rest lenght of the springs correspond to the configuration where $\mathbf{X}_c^g = \mathbf{X}_c^t$.

Since we are assuming that n is the minimum number of required contact points, we need to enforce the equilibrium of all contact forces, i.e. $\sum_{c=1}^{n} \mathbf{w}_c = 0$. Thus, considereing the matrix $\mathbf{G} = [\ \mathbf{P}_1 \ \ \dots \ \ \mathbf{P}_n\ ]$, the block diagonal matrix $\mathbf{K} = \text{blkdiag}(\tilde{\mathbf{K}}_1, \dots, \tilde{\mathbf{K}}_n)$, the block diagonal matrix $\mathbf{P} = \text{blkdiag}(\mathbf{P}_1, \dots, \mathbf{P}_n)$, and collecting all contact displacements in $\delta\mathbf{X} = [\ \delta\mathbf{X}_1^{\mathsf{T}} \ \ \dots \ \ \delta\mathbf{X}_n^{\mathsf{T}}\ ]^{\mathsf{T}}$, the object equilibrium can be expressed as

$$\mathbf{G}\mathbf{K}\mathbf{P}^{\mathsf{T}}\delta\mathbf{X} = \mathbf{0}. \tag{5.9}$$

Additionally, the contact forces must comply with the contact model. Typical contact models in grasping include: the point contact without friction (PC), point contact with friction (PCWF), and contact with a soft finger (SF) [83]. They can be implemented by considering springs only along the constrained directions, with $h_c = 1, 3, 4$, depending whether we use a PC, PCWF, or SF, respectively. Without loss of generality, the supporting lines of the springs are chosen such that $\hat{\mathbf{p}}_{1,c}$ indicates a translation along the inward normal direction, $\mathbf{p}_{2,c}$ and $\hat{\mathbf{p}}_{3,c}$ indicates translations along the tangent directions, and $\hat{\mathbf{p}}_{4,c}$ indicates a rotation about the normal direction, with respect to the object. Thus, the linear and torsional friction coefficients, μ_c and ν_c, define an additional constraint on the vector $\mathbf{w}_c = [\begin{array}{ccc} w_{1,c} & \dots & w_{6,c} \end{array}]$, which must belong to the generalized friction cone

$$\mathcal{C}_c = \{\mathbf{w}_c \in \mathbb{R}^6 | \|\mathbf{w}_c\|_\Delta \leq w_{1,c}\}, \tag{5.10}$$

where $\|\mathbf{w}_c\|_\Delta$ can take the form 0, $\frac{1}{\mu_c}\sqrt{w_{2,c}^2 + w_{3,c}^2}$ or $\frac{1}{\mu_c}\sqrt{w_{2,c}^2 + w_{3,c}^2} + \frac{1}{\nu_c}|w_{4,c}|$ depending on whether we use the PC, PCWF, or SF model, respectively, as proposed in [83].

Hand equilibrium

Each joint torque is modeled using a torsional spring connecting the grasping and the reference configuration at the joints. The resultant force due to the spring elongation is written as

$$\mathbf{w}_{ij} = \hat{\mathbf{z}}_{ij}\tau_{ij}, \tag{5.11}$$

where $\hat{\mathbf{z}}_{ij} \in \mathbb{R}^6$ represents the supporting line that coincides with the joint rotation axis at the grasping configuration, and τ_{ij} is the torque magnitude obtained as $\tau_{ij} = \kappa_{ij} d_{ij}$, where d_{ij} is the spring elongation. The joint displacement of the i-th finger is expressed as $\delta\mathbf{q}_i = \mathbf{q}_i^r - \mathbf{q}_i^g$. Thus, introducing the matrix $\mathbf{Z}_i = [\begin{array}{ccc} \hat{\mathbf{z}}_{i,1}^\mathsf{T} & \dots & \hat{\mathbf{z}}_{i,m_i}^\mathsf{T} \end{array}]^\mathsf{T}$, the joint torque magnitudes that result of applying a force $\mathbf{w}_c$ at the c th contact point is

$$\mathbf{K}_{q_i}\delta\mathbf{q_i} = \mathbf{Z}_i^\mathsf{T}\mathbf{w}_c, \tag{5.12}$$

where $\mathbf{K}_{q_i} = \text{diag}(\kappa_{q_{i1}}, \dots, \kappa_{q_{im_i}})$, for $c = 1, \dots, n$ and $i = c$ in turns, models the compliance of the i-th joint.

Finally, we must ensure that the contact forces are compensated by joint torques. Since the fingers are independent, the force applied at the c-th should be compensated by torques at the i-th finger, with $i = c$. Thus, introducing the block diagonal matrix $\mathbf{J} = \text{blkdiag}(\mathbf{Z}_1, \dots, \mathbf{Z}_n)$ and the vector $\mathbf{w} = [\begin{array}{ccc} \mathbf{w}_1^\mathsf{T} & \dots & \mathbf{w}_n^\mathsf{T} \end{array}]^\mathsf{T} = \mathbf{PKP}^\mathsf{T}\delta\mathbf{X}$, the hand equilibrium can be expressed as

$$\mathbf{K}_q \delta \mathbf{q} = \mathbf{J}^\mathsf{T} \mathbf{P} \mathbf{K} \mathbf{P}^\mathsf{T} \delta \mathbf{X}, \tag{5.13}$$

where $\mathbf{K}_q$ and $\delta \mathbf{q}$ consider all joints, ordered accordingly to the corresponding row of $\mathbf{J}^\mathsf{T}$.

In addition, the grasping configuration must be attainable by the actual hand, hence the joint value limitations are again applicable here as

$$\mathbf{q}^l \leq \mathbf{q}^g \leq \mathbf{q}^u. \tag{5.14}$$

Finally, the joint torques are subject to mechanical limitations as well, written as

$$|\mathbf{K}_q \delta \mathbf{q}| \leq \boldsymbol{\tau}^u, \tag{5.15}$$

with $\boldsymbol{\tau}^u$ being the vector of maximum torques that the joint motors can exert, and the absolute value and the inequality must be read component-wise. Even when $\mathbf{q}^r$ is not subject to the mechanical limitations, the fingers must push against the object, enforcing a minimum torque τ^l by including

$$|\mathbf{K}_q \delta \mathbf{q}| \geq \boldsymbol{\tau}^l. \tag{5.16}$$

5.1.4 System overview and dimension analysis

A grasp configuration $\mathbf{y} = (\mathbf{q}^r, \mathbf{q}^g, \mathbf{q}^t, \mathbf{T}_h, \boldsymbol{\gamma})$ is kinematically feasible and restrained when it fulfills Eqs. (5.1), (5.2), (5.9), and (5.13) collected in

$$\mathbf{M}_{\text{eq}}(\mathbf{y}) = \mathbf{0}, \tag{5.17}$$

and (5.3), (5.10), (5.15), and (5.16), transformed in less-than-equal inequalities and collected in

$$\mathbf{M}_{\text{ineq}}(\mathbf{y}) \leq \mathbf{0}, \tag{5.18}$$

while staying within the valid ranges defined by (5.4) and (5.14), for $c = 1, \ldots, n$ contacts and $j = 1, \ldots, m_i$ joints with $i = c$.

The number of variables, in this case, correspond with the internal degrees of freedom $n_v = 3m + \sum_{c=1}^{n} a_c + \sum_{c=1}^{n} b_c$, where m is the total sum of joints per hand configuration, a_c defining the whether the contact region is a surface, a curve or a point, and b_c being 1 or 3 depending whether the normal vector at the contact points are aligned or not. The number of algebraic constraints is $n_e = D(n-1) + m + D$, i.e. the constraint due to the $(n-1)$ independent loops in D-space, m equations due to the hand equilibrium constraint, D equations from the object equilibrium constraint, with D being 3 or 6 whether it is a planar or a spatial grasp, respectively. Assuming $a_c = a$ and $b_c = b$, for $c = 1, \ldots, n$, the dimension of the solution space is then $n_s = n_v - n_e = 2m + (a + b - D)n$. In general, this number is relatively high.

5.2 Solution strategy

There may be multiple solutions to the system given by Eqs. (5.17) and (5.18) due to high nonlinearities in the constraints, even for 0-dimensional solution spaces. Thus, we propose the minimization of the potential energy of the springs at the joints as the selection criterion, which is expressed by the objective function

$$\Psi(\mathbf{y}) = \frac{1}{2}\delta\mathbf{q}^{\mathsf{T}}\mathbf{K}_q\delta\mathbf{q}. \tag{5.19}$$

It is worth noting that, when the constraints are met, the substitution of Eq. (5.13) in Eq. (5.19), yields

$$\Psi'(\mathbf{y}) = \frac{1}{2}\delta\mathbf{q}^{\mathsf{T}}(\mathbf{J}^{\mathsf{T}}\mathbf{P}\mathbf{K}\mathbf{P}^{\mathsf{T}}\delta\mathbf{X}). \tag{5.20}$$

and, additionally, $\delta\mathbf{X} \approx \mathbf{J}(\mathbf{q}^g - \mathbf{q}^t)$ for small joint displacements. Thus, introducing the block diagonal matrix $\mathbf{K}' = \mathbf{P}\mathbf{K}\mathbf{P}^{\mathsf{T}}$, Eq. (5.20) becomes

$$\Psi''(\mathbf{y}) = \frac{1}{2}(\mathbf{q}^r - \mathbf{q}^g)^{\mathsf{T}}\mathbf{J}^{\mathsf{T}}\mathbf{K}'\mathbf{J}(\mathbf{q}^g - \mathbf{q}^t). \tag{5.21}$$

By comparing Eqs. (5.21) and (5.19), the criterion can be rewritten as

$$\Psi'''(\mathbf{y}) = \|\mathbf{K}_q - \mathbf{J}^{\mathsf{T}}\mathbf{K}'\mathbf{J}\|, \tag{5.22}$$

i.e. when the constraints are met and joint displacements are small, the criterion selects the configuration $\mathbf{y}$ in which the contact stiffness seen from the joints equals the joint stiffness.

The problem can be casted as: Given a hand with n articulated fingers to grasp an object, with a kinematic configuration defined by the pair $(\mathbf{q}, \mathbf{T}_h)$, a contact on the fingertip $\mathbf{X}_c$, its corresponding contact region on the object surface S_c, m joint spring stiffnesses κ_{ij} and nh_c contact spring stiffnesses $\kappa_{k,c}$, and friction coefficients μ_c and ν_c, find a configuration $\mathbf{y}$ that minimizes the objective function expressed by Eq. (5.22) subject to the constraints in Eqs. (5.17-5.18), (5.4), and (5.14). This non-linear optimization problem is in the form required by the MATLAB routine `fmincon`. We select the `SQP` algorithm due to its ability to work out of the solution manifold using a feasibility reformulation. This slows down the process, however it is desired when the method is starting and the configurations are far from satisfying the constraints [71].

5.3 Test cases

The method is illustrated here with three experiments. The first one consists of a simple planar hand grasping an ellipse, the second one consists of a complex robotic hand grasping an ellipsoid, and the third one is a an informed instance of the problem with a real execution. The details of each case are shown next.

5.3.1 A simple planar hand grasping an ellipse

In this example, we use a simple planar hand with $n = 2$ fingers, and $m_i = 2$ joints per finger, for a total of $m = 4$ joints. The object to be grasped is an ellipse, and the contact regions cover competely the ellipse boundary (Figure 5.4). The normal vectors are not aligned at the contact points, hence $b_c = 1$ (one extra internal degree of freedom due to free orientation of contacting link). The dimension of the solution space in this case is $n_s = 6$. The kinematic structure, spring constants and friction coefficients needed to write the grasp synthesis problem as stated in Section 5.2 are shown in Table 5.2. In this case, the initial guesses are randomly generated, but biased towards the center of the search range for each variable. The results using the proposed method from two different initial guesses are shown in Figure 5.4. The relation between the touching, grasping and reference configurations can be appreciated, for instance, in the left figure, where most of the work in the right finger is done by the outer joint, not the case for the left finger, as expected from their respective touching configurations and the need of pushing against each other. For a different touching configuration and the same

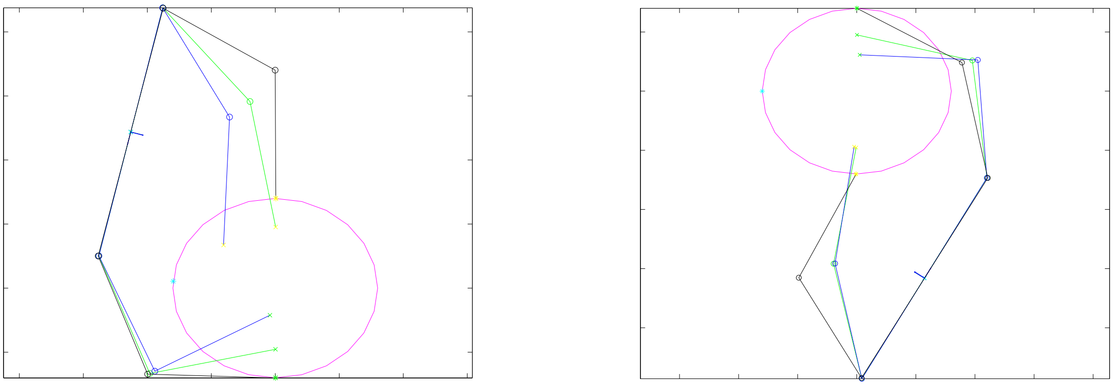


Figure 5.4: Two solutions obtained for a simple hand satisfying all constraints. The color code corresponds to that of Figure 5.2, black for the outer hand configuration, blue for the inner hand configuration, and green for the middle hand configuration.

pushing requirement, which is the second solution found, the grasping and reference configurations are different. Note, in this case, the little work made by the joints in the left finger, where most of the contact force is absorbed by the structure, that traduces into high Cartesian stiffness, requiring less load in the joint spring. Though the hand is symmetric and their parameters are symmetric, the solutions are not symmetrical due to the randomness for their initial guesses.

5.3.2 THE First hand grasping an ellipsoid

The second example employs a complex hand, in this case, using the paremeters of teh Schunk Anthorpomorphic shown in Figure 5.1(right). The grasp uses $n = 3$ fingers (three out of the four available), with $m_1 = m_2 = 3$ and $m_3 = 4$ joints, for a total of $m = 10$. The object is an ellipsoid, and the contact regions completely cover its surface (Figure 5.5). The normal vectors must be aligned, hence $b_c = 1$. The dimension of the solution space in this case is $n_s = 11$. The kinematic structure, spring constants and friction coefficients needed to write the grasp synthesis problem as stated in Section 5.2 are shown in Table 5.2. In this case, the initial guess was set by introducing the constraints sequentially, such that it was as close as possible from the solution manifold. The result using the proposed method is shown in Figure 5.5.

a) Kinematic structure and limit values	
Finger base reference frame	$T_{11} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$ $T_{21} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Phalanx length [cm]	All phalanges are of length 1
Joint limits [deg]	$\mathbf{q}^l = \begin{bmatrix} 0 & -90 & 0 & -90 \end{bmatrix}^\top,$ $\mathbf{q}^u = \begin{bmatrix} 90 & 45 & 90 & 45 \end{bmatrix}^\top$
Torque limits [Ncm]	$\boldsymbol{\tau}^{\min} = 1[\mathbf{1}]$, $\boldsymbol{\tau}^{\max} = 10[\mathbf{1}]$, where $[\mathbf{1}] \in \mathbb{R}^m$ is a vector containing ones
Contact point $\mathbf{x}_c, \forall c$ in local [cm]	$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
Regions $S_c, \forall c$ [cm]	$\mathbf{s} = \begin{bmatrix} 0.8\cos(\gamma_c) \\ 0.7\sin(\gamma_c) \end{bmatrix},$ with $\gamma_c \in [0, 2\pi]$
b) Coefficients	
Joint stiffness [N/rad]	$\mathbf{K}_q = \text{blkdiag}(1,1,1,1)$
Friction (PCWF) $\mu_c, \forall c$	$\mu = 0.5$
Contact stiffness $\tilde{\mathbf{K}}_c, \forall c$ [N/cm]	$\tilde{\mathbf{K}} = \text{blkdiag}(1,1)$

Table 5.2: Kinematic and elastic parameters for the planar hand.

5.3.3 Experiments with THE First hand grasping a ball

For this example, the five-fingered hand developed at the Pisa laboratory was used, namely THE First hand [8]. In this case, the hand was mounted on a fixed base, therefore, the reachability constraint was calculated in a decoupled way considering the workspace of the hand. The object was placed such that the fingers could reach it, and the constraint was verified. The restrainability constraint was then met using the solution strategy presented in Section 5.2. The results from the implementation are shown in Figure 5.6. The hand first make contact fulfilling the reachability constraint

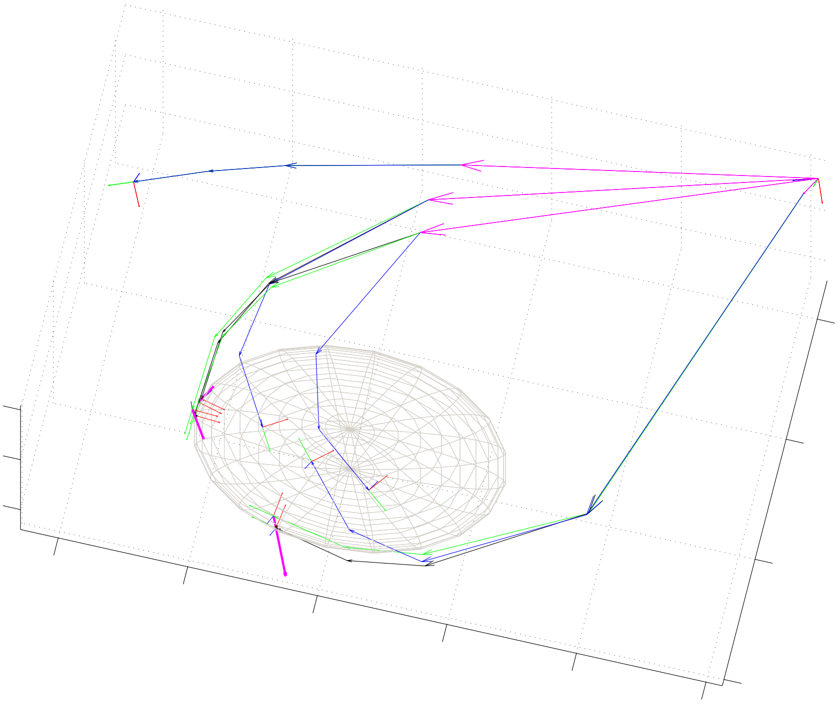


Figure 5.5: A solution obtained for the Schunk Anthropomorphic hand satisfying all the constraints performing the grasp with three fingers (thumb, index and middle). The color code corresponds to that of Figure 5.2, black for the outer hand configuration, blue for the inner hand configuration, and green for the middle hand configuration.

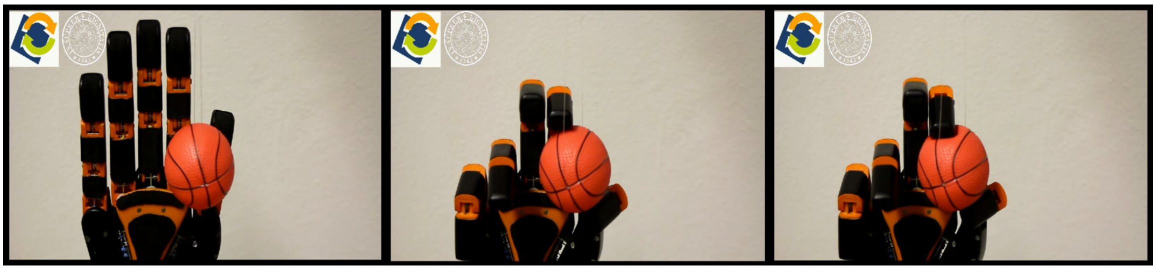


Figure 5.6: A solution obtained and implemented on THE First Hand [8]. The object is placed at the computed position (left). The fingertips make contact at the object surface satisfying the rechability constraint (middle). The fingers perform the squeezing action to firmly hold the object (right).

a) Kinematic structure and limit values	
Finger base reference frame	$T_{11} = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 27.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$ $T_{21} = \begin{bmatrix} 0 & 0 & 1 & -4.3 \\ 0.035 & -0.99 & 0 & 40.165 \\ 0.99 & 0.035 & 0 & 145.43 \\ 0 & 0 & 0 & 1 \end{bmatrix},$ $T_{31} = \begin{bmatrix} 0 & 0 & 1 & -4.3 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 145.43 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
DH parameters $(a_j, \alpha_j, d_j)_i$ [cm,rad,cm]	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\pi/2 & 0 \\ 67.8 & 0 & 0 \\ 30 & 0 & 0 \end{pmatrix},$ for $j = 1, 2, 3, 4$, for $i = 1, 2, 3$, and $q_{i3} = q_{i4}$
Joint limits [deg]	$\mathbf{q}^l = \begin{bmatrix} 0 & -15 & -4 & 4 & -15 & -4 & 4 & -15 & -4 & 4 \end{bmatrix}^\top,$ $\mathbf{q}^u = \begin{bmatrix} 90 & 15 & 75 & 75 & 15 & 75 & 75 & 15 & 75 & 75 \end{bmatrix}^\top$
Torque limits [Ncm]	$\boldsymbol{\tau}^{\min} = 10[\mathbf{1}]$, $\boldsymbol{\tau}^{\max} = 1000[\mathbf{1}]$, where $[\mathbf{1}] \in \mathbb{R}^m$ is a vector containing ones
Contact point $\mathbf{x}_c, \forall c$ in local [cm]	$\mathbf{x} = \begin{bmatrix} 0 & 29.5 & 0 \end{bmatrix}^\top$
Regions $S_c, \forall c$ [cm]	$\mathbf{s} = \begin{bmatrix} 60\cos(\gamma_{1,c})\sin(\gamma_{2,c}) \\ 40\sin(\gamma_{1,c})\sin(\gamma_{2,c}) \\ 20\cos(\gamma 2, c) \end{bmatrix},$ with $\gamma_{1,c} \in [0, 2\pi]$ and $\gamma_{2,c} \in [0, \pi]$
b) Coefficients	
Joint stiffness [N/rad]	$\mathbf{K}_q = 100[\text{blkdiag}(4, 3, 2, 1, 3, 2, 1, 3, 2, 1)]$
Friction (PCWF) $\mu_c, \forall c$	$\mu = 0.25$
Contact stiffness $\tilde{\mathbf{K}}_c, \forall c$ [N/cm]	$\tilde{\mathbf{K}} = \text{blkdiag}(5, 5, 5)$

Table 5.3: Kinematic and elastic parameters of THE First hand.

(Figure 5.6, middle) and then perform the squeezing action fulfilling the restrainability constraint (Figure 5.6, right), completing the grasp action.

5.4 Summary

The proposed approach tackles simultaneously the contact constraint, the object and the hand equilibrium issues that a grasp must satisfy in order to be reachable and restrained. This is obtained by introducing torsional springs modeling the joint compliance, and a set of linear springs for the contact interaction. This leads to a solution where all the variables ultimately employs configuration values, and therefore, the hand can be commanded to grasp the object using only a position controller.

The results show practicable solutions provided by the proposed method for illustrative examples, suggesting how to: (i) reach the specified regions on object with the fingertips, (ii) apply the forces in the directions allowed by the contact model within the friction constraints, and (iii) compensate such forces using the hand joints, i.e. the hand performs a reachable and restrained grasp of the object with the minimum effort of the joints.

Part III

Approach Path Planning

6 Reducing the Hand Configuration Space

A hand configuration is determined by all the joint values and the position and orientation of the wrist. Thus, the dimension of the hand configuration space is very high, typically over ten. A menaningful reduction is of great importance in order to apply motion planning techniques that work better in lower dimensions. Concerning the finger joints, the main consideration that supports the reduction in the finger configuration spaces is that the human hand has several joint movements that are not completely independent, that is, there are some joints that can be coordinated using some relation. A typical example is given by the last two joints of each finger, which, in general, can not be moved independently in the free space. In the same line, some other correlations can be found when the human hand motions are carefully analyzed. These correlations can be mapped to a mechanical hand in order to mimic human-like motions while exploring the relevant workspace. Concerning the wrist, the main consideration that supports the reduction is that the hand usually faces the object during the approach phase. Therefore, relational positioning techniques can be used to introduce proper constriaints in the wrist configuration space. This chapter provides the insights of both finger motion coordination and wrist orientation constraint.

6.1 Finger motion coordination

The main consideration that supports the reduction in the finger configuration spaces is that the human hand has several joint movements that are not completely independent, that is, there are some joints that can be coordinated using some relation. A typical example is given by the last two joints of each finger, which, in general, can not be moved independently in the free space. In the same line, some other correlations can be found when the human hand motions are carefully analyzed. These correlations can be mapped to a mechanical hand in order to mimic human-like motions while exploring the relevant workspace.

A relevant previous work in this line [104] uses an initial set of grasping configurations to find a bidimensional *grasp subspace*, i.e. a set of hand configurations used to grasp different objects. This subspace, already used in Chapter 4, has also been used in other works to look for grasping configurations [20, 117] as well. Dimensionality reduction techniques have also been used to synthesize human-like motion in graphic applications [101].

Unlike the works specifically oriented to synthesize a grasp, we use here an initial set of unconstrained general hand configurations in order to model the real hand workspace, that is, the self-collision free region of the hand configuration space $\mathcal{X}_\text{h}$.

The procedure consists of acquaring a number of samples of human hand postures using a sensorized glove, and then, mapping those samples to the mechanical hand configuration space, $\mathcal{X}_\text{h}$. The samples in $\mathcal{X}_\text{h}$ are analyzed using a Principal Component Analysis (PCA) to find the direction with largest dispersion, which is iteratively repeated considering orthogonal directions until a new base of $\mathcal{X}_\text{h}$ is generated. Then, by selecting the first p vectors of this base and properly choosing a bounding box $\mathcal{B}_\text{h}$ aligned with these vectors and centered at the mean value of the original set of points, a good bounded approximation of the hand workspace. This approximation is the subspace $\mathcal{E}$, defined by the Principal Motion Directions (PMDs), which represent ways to coordinate the fingers.

6.1.1 Experimental set-up

The experimental set-up involves: an anthropomorphic mechanical hand, a sensorized glove, and a hand/robot simulator connected with the real elements. The main relevant details about these elements are:

Mechanical hand: The hand is the Schunk Anthropomorphic hand (SA) [106], shown in Figure 6.1 (left) , which was built upon the DLR hand model [14]. It has three fingers with four joints plus the thumb with five joints, in all of them the distal and middle flexion joints are mechanically coupled, thus there are a total of 17 joints

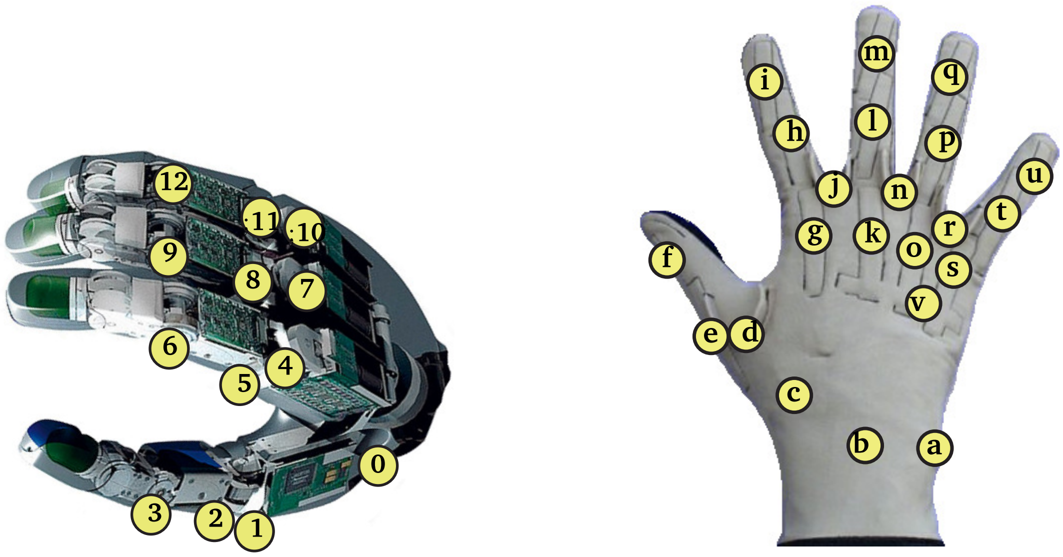


Figure 6.1: Mapping between a mechanical hand and a sensorized glove. The numbers indicate an independent joint on the SA hand (left). The letters indicate a sensor on the CyeberGlove (right)

with only 13 independent degrees of freedom. The extra degree of freedom of the thumb is called the “thumb base joint”, numbered with “0” in Figure 6.1 (left), and moves the whole thumb with respect to the palm. This is a particular joint because it has an independent controller and cannot be moved in a synchronized way with the other hand joints. This enforces to do some particular adjustments in the motion planner to allow real implementation in practice, as will be shown in Chapter 7.

Sensorized glove: The glove is the CyberGlove®, shown in Figure 6.1 (right). It is a fully instrumented glove that provides 22 joint-angle measurements using resistive bendsensing technology. It includes three flexion/extension sensors per finger, four abduction/adduction sensors between the fingers, a palm-arc sensor, and two sensors to measure the flexion and the abduction of the wrist.

Hand and robot simulator: The simulator is used to visualize the movements of the mechanical hand associated with the movements of the human operator hand, which are captured with the sensorized glove, as shown in Figure 6.2. The simulator detects collision between the involved elements. For more details, see [84].

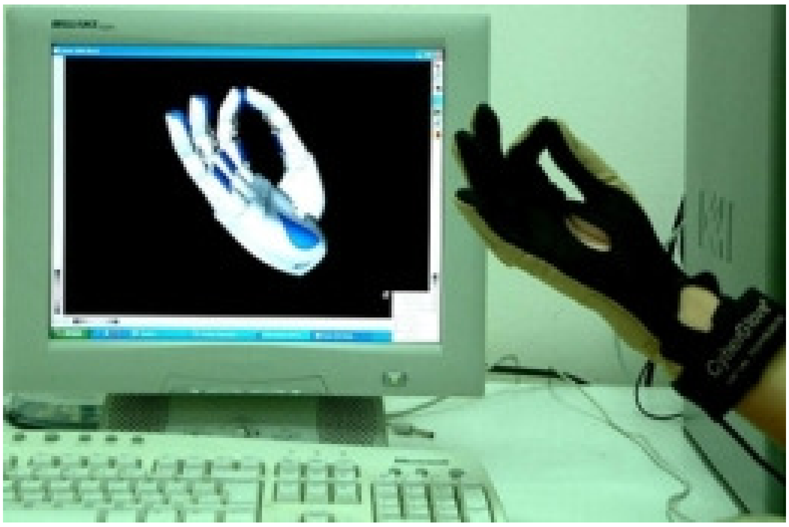


Figure 6.2: A human hand with a sensorized glove connected to the mechanical hand simulator.

The schema of the whole experimental set-up is illustrated in Figure 6.3, including the type of connection between the different elements.

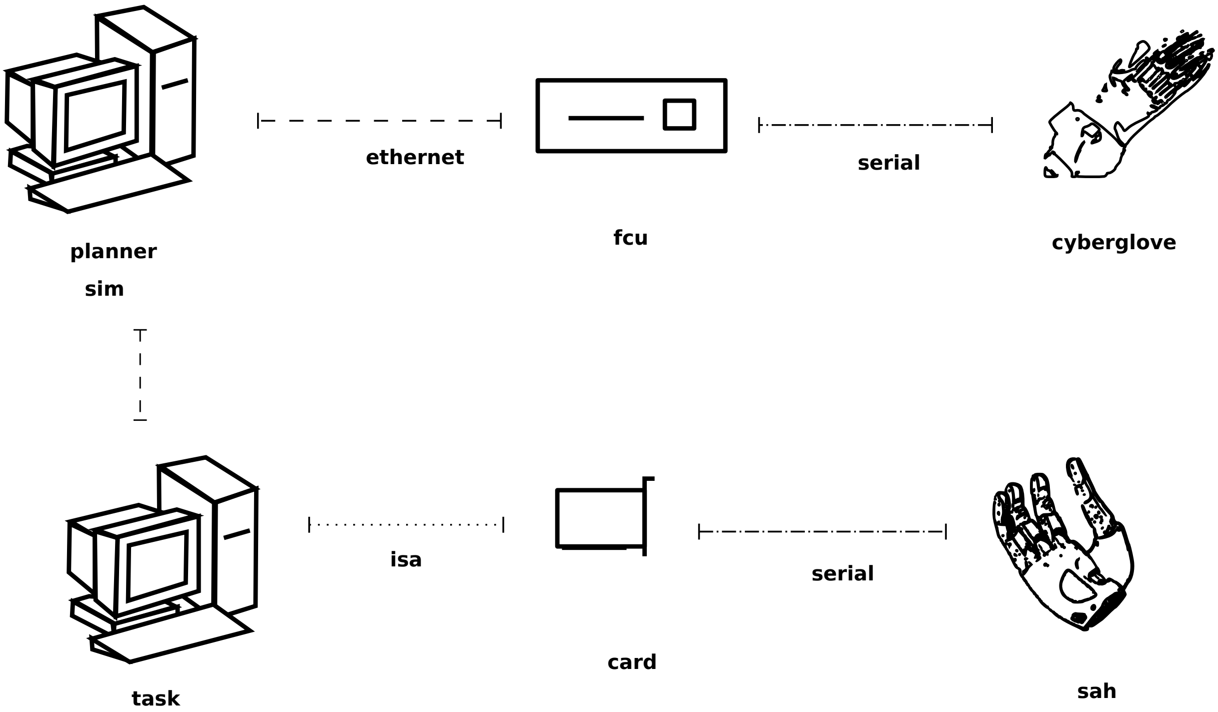


Figure 6.3: Schematic representation of the experimental setup.

6.1.2 Experimental protocol

The postures of a human operator hand are captured using the sensorized glove. The operator freely moves his/her hand in an unconstrained way, i.e. without performing any specific task, trying to cover the whole hand workspace. There is no guarantee that the operator actually covers the whole workspace, but in this way it is expected that he/she performs the most natural and evident hand movements, thus the most natural and evident hand postures are captured. The operator can have a continuous visual feedback of the mechanical hand postures associated with his/her hand postures by means of the hand simulator, as illustrated in Figure 6.2.

The mapping of the data obtained from the glove sensors to the joints of the SA hand is done considering the following issues (see Figure 6.1):

- The palm of the mechanical hand is rigid, therefore the palm arc sensor **v** and the wrist flexion and abduction sensors **b** and **a** are ignored.
- The mechanical hand lacks the little finger, therefore the sensors **u**, **t**, **s** and **r** are ignored.
- The mechanical hand has a one-to-one coupling between the medium and distal phalanx of each finger, therefore the distal phalanx sensors **i**, **m**, and **q** are ignored.
- The abduction is measured in a relative way in the glove, i.e. sensors **j** and **n** give the relative angle between the index and the middle fingers and between the middle and the ring fingers, respectively. Therefore, the mapping uses the middle finger as reference, i.e. the joint #7 is fixed to zero, and sensors **j** and **n** are directly associated to joints #4 and #10, respectively.
- The use of sensor **c** to control joint #1 produces a more natural motion of the SA hand than using sensor **d**, because sensor **d** measures the relative abduction between the thumb and the index. Therefore sensor **c** is used for both joints #0 and #1.

Then, only 11 values from the 22 sensors available in the glove are used in practice to command the joints of the SA hand. The complete mapping is shown in Table 6.1. Note that this mapping makes the motions of the SA hand to be defined with 11 independent parameters despite it has 13 actuated joints.

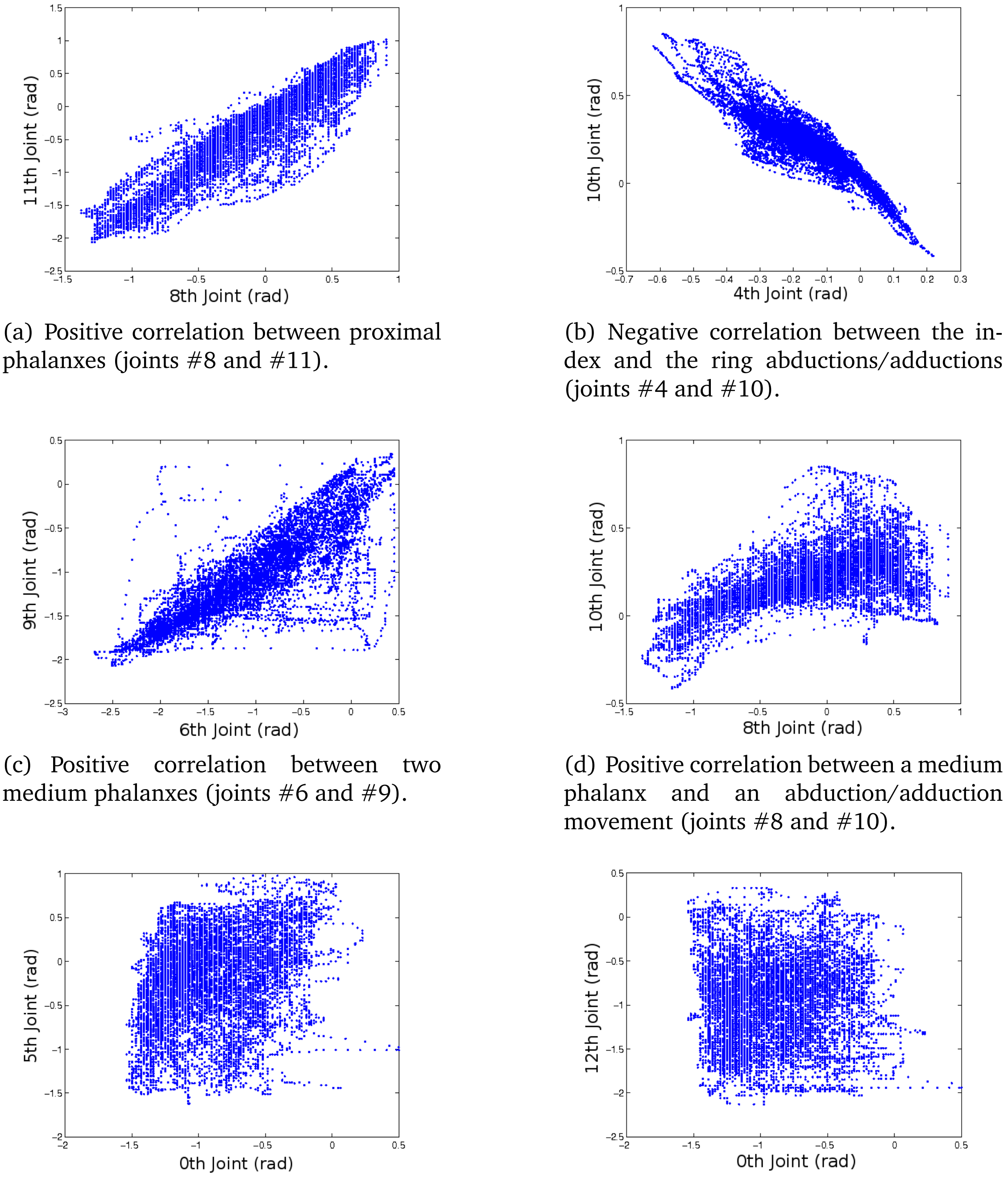
	Cyberglove Sensor	→	SA Hand Joint
Id.	Name	Id.	Name
c	thumb roll	0	thumb base
c	thumb roll	1	finger base (thumb)
e	thumb inner	2	proximal phalanx (thumb)
f	thumb outer	3	medium phalanx (thumb)
j	index abduction	4	finger base (index)
g	index inner	5	proximal phalanx (index)
h	index middle	6	medium phalanx (index)
-	medium abduction	7	finger base (medium)
k	medium inner	8	proximal phalanx (medium)
l	medium medium	9	medium phalanx (medium)
n	ring abduction	10	finger base (ring)
o	ring inner	11	proximal phalanx (ring)
p	ring medium	12	medium phalanx (ring)

Table 6.1: Correspondence between the CyberGlove sensors and the joints of the SA hand shown in Figure 6.1.

6.1.3 Data analysis

Dimensionality reduction of a feature set is a common preprocessing step used for pattern recognition and classification applications as well as in compression schemes. Principal Component Analysis (PCA) is often used in these fields to reduce multidimensional data sets to lower dimensions for their treatment [45], and it is also used as a tool in exploratory data analysis as well as for making predictive models. Basically, PCA involves the computation of the eigenvalue decomposition of a data covariance matrix or the singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The larger the eigenvalues or the singular values the larger the dispersion of the data along the corresponding eigenvector direction. This analysis allows the identification of the directions of the space where the samples have larger dispersion.

In this work, PCA is used to reduce the hand configuration space, $\mathcal{X}_\mathrm{h}$, of the mechanical hand to a more tractable space, $\mathcal{E}$, of smaller dimension, using for that purpose the data obtained from the hand postures of a human operator mapped to the mechanical hand, as described above. The dimension reduction is done based on the correlation



(a) Positive correlation between proximal phalanxes (joints #8 and #11).

(b) Negative correlation between the index and the ring abductions/adductions (joints #4 and #10).

(c) Positive correlation between two medium phalanxes (joints #6 and #9).

(d) Positive correlation between a medium phalanx and an abduction/adduction movement (joints #8 and #10).

(e) No correlation between the thumb base and the medium phalanx of the index (joints #0 and #5).

(f) No correlation between the thumb base and the medium phalanx of the ring finger (joints #0 and #12).

Figure 6.4: Correlation of joints in the hand configuration space.

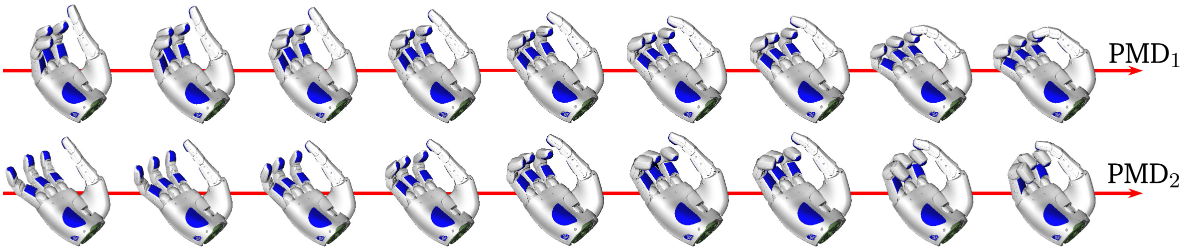


Figure 6.5: Configurations of the SA hand when it is moved along the first two PMDs.

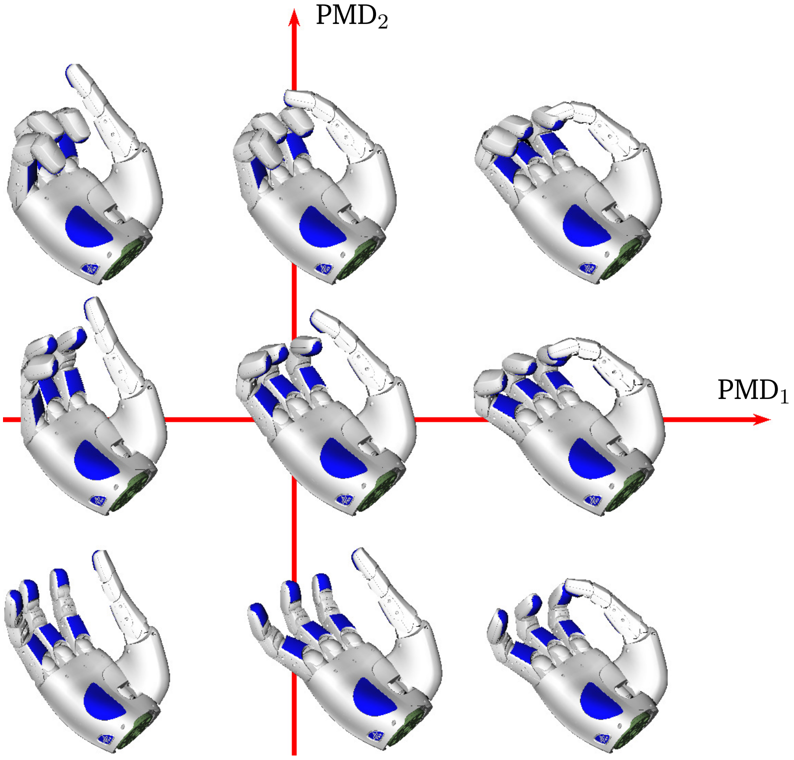


Figure 6.6: Configurations of the SA hand when it is moved along a combination of the first two PMDs.

that there exists between some joints of the mechanical hand when it follows the hand postures of the human operator. For instance, for a set of 13,500 hand postures captured with the sensorized glove, Figure 6.4 shows different examples of the obtained correlations between some particular pairs of joints.

From the captured data it can be seen that the position of joint #0 of the mechanical hand is rather independent of the other hand joints, of course, with the exception of joint #1 that is completely equivalent due to the selected mapping; two examples are given in Figures 6.4(e) and 6.4(f). This, together with the fact that joint #0 of the SA hand cannot be coordinately controlled with the other joints, motivates the selection of joint #0 to form part of a base of $\mathcal{E}$. The remaining directions of the base of $\mathcal{E}$ are obtained applying PCA to the samples of the mechanical hand, which are still defined by 11 independent parameters. The PCA returns a new base of the configuration space $\mathcal{X}_\mathrm{h}$, with the base vectors ordered according to the dispersion of the samples along each vector direction. The first vector indicates the direction of maximal dispersion of the samples. The directions indicated by these vectors in $\mathcal{X}_\mathrm{h}$ are called Principal Motion Directions (PMDs). In order to illustrate the variation of the hand configuration along the PMDs, Figure 6.5 shows the hand postures along the two first PMDs, and Figure 6.6 the postures resulting from their linear combination.

In this experimental dataset, the first PFM represents the 42.19% of the total variance, the first two components the 77.12%, and the first three components the 84.71%. The total accumulated variance as a function of the number of selected first PMDs is shown in Figure 6.7. Following these results, the use of four PMDs is considered enough

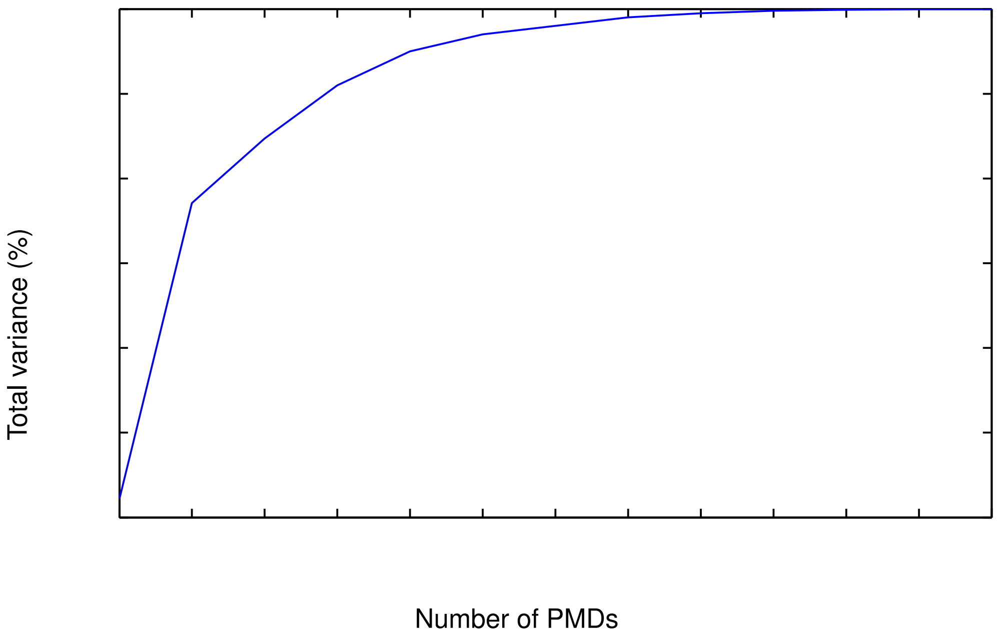


Figure 6.7: Total variance covered when using an increasing number of PMDs.

to represent, together with the thumb base, the subspace $\mathcal{E}$, which, therefore would be of dimension 5. The inclusion of the thumb base to define one of the dimensions of $\mathcal{E}$ is a particularity related with the SA hand and it does not reduce the generality of the approach. Note also that, any desired coordinated motion can be incorporated to define $\mathcal{E}$, such as the task motion direction proposed in [114], which joins the current and goal configuration with a linear segment in the configuration space.

6.2 Wrist orientation constraint

The main consideration that supports the reduction is that the hand usually faces the object during the approach phase. Therefore, relational positioning techniques can be used to introduce proper constraints in the wrist configuration space. The dimension reduction for the wrist motions can be achieved, thus, by considering virtual constraints. In constrained-based motion planning problems, the robot is forced to move in a submanifold with lower dimension than the original space. If sampling-based methods are used, the planning of collision-free movements of a constrained object does not require the sampling of the whole configuration space, but only the regions of the configuration space where the robot is allowed to move: its configuration submanifold. These submanifolds can be described in terms of geometric constraint sets by explicitly stating the relations that must hold between two or more geometric entities. Geometric constraint solvers introduce constraints between the relative position of rigid bodies using, for instance, distance and angles bewteen planes, lines and points [90], which then are used to find the map between constraint sets and configuration submanifolds.

The orientation constraint is to be imposed to the reference frame at the wrist, $\mathcal{F}$. The frame should be oriented such that the vector $\hat{\mathbf{u}}$ always points towards a point of interest on the object, P, which is defined using the wrist orientation at the goal hand configuration, as shown in Figure 6.8. This orientation constraint provides more realistic motions during the approach path, as the wrist is always facing the object during the approach phase and, typically, reduces post-processes such as prunning unnecessary branches in the path. The submanifold of $SE(3)$ that satisfies this orientation constraint is 4-dimensional, i.e. the motion of the palm is constrained to four degrees of freedom, three of translation defining the position, $O(x, y, z)$, of the origin of $\mathcal{F}$, and one defining the rotation of $\mathcal{F}$ about $\hat{\mathbf{u}}$ with an angle α, as illustrated in Figure 6.9.

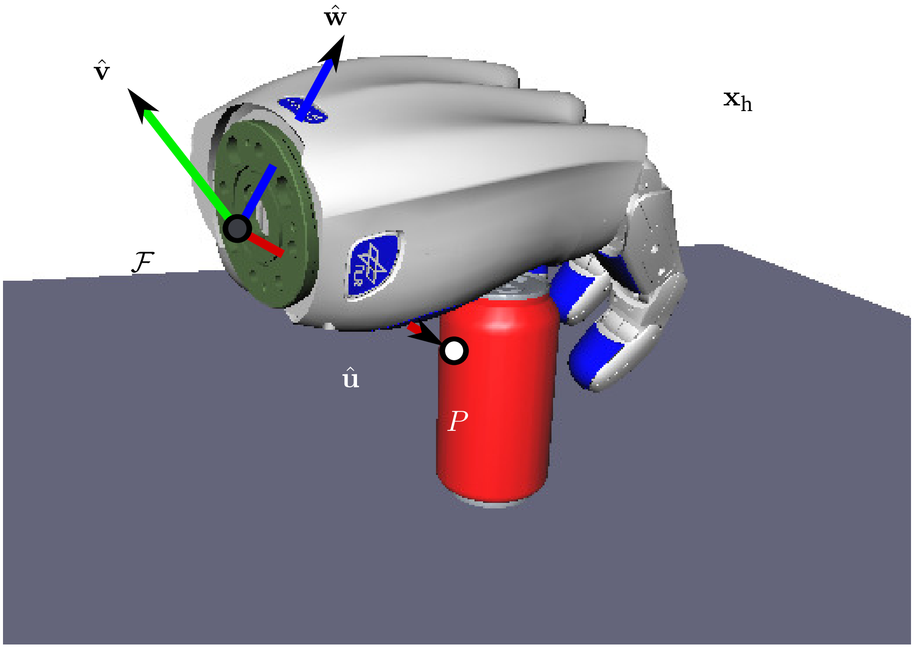


Figure 6.8: The position vector of the hand wrist at a desired configuration, $\mathbf{x}_\mathrm{h}$, is used to define point P, which is the point of interest for the constraint.

The mathematical expression describing such constraint can be stated as follows. With support in Figure 6.10, given a position, O, of the origin of $\mathcal{F}$, to satisfy the orientation constraint, the frame $\mathcal{F}$ must be rotated using the matrix

$$\mathbf{R}_1 = Rot(\beta, \hat{\mathbf{n}}), \tag{6.1}$$

where β is the angle between $\hat{\mathbf{u}}$ and $\hat{\mathbf{r}}$, the latter being the unitary vector pointing from O to P, and $\hat{\mathbf{n}}$ is the unitary vector normal to the plane defined by $\hat{\mathbf{u}}$ and $\hat{\mathbf{r}}$, obtained as

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{u}} \times \hat{\mathbf{r}}}{||\hat{\mathbf{u}} \times \hat{\mathbf{r}}||}. \tag{6.2}$$

In addition, the frame $\mathcal{F}$ is free to rotate any given angle α about $\hat{\mathbf{u}}$ using the matrix

$$\mathbf{R}_2 = Rot(\alpha, \hat{\mathbf{u}}). \tag{6.3}$$

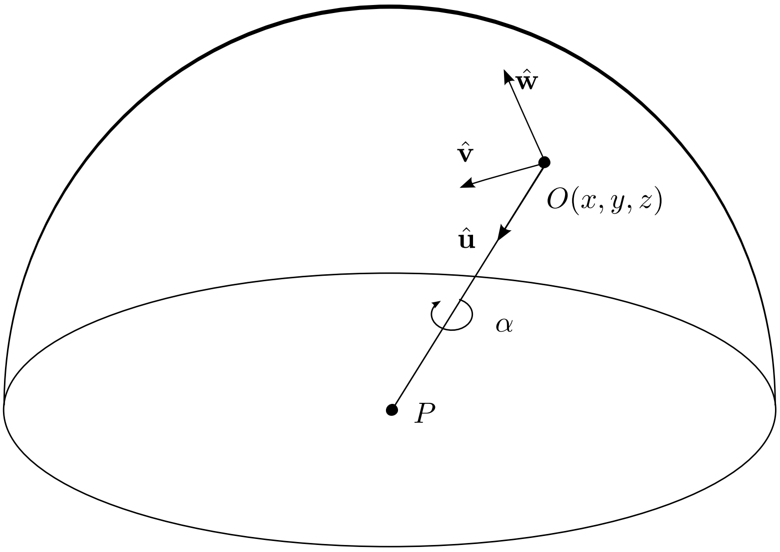


Figure 6.9: Parameters x, y, z and α of the 4-dimensional submanifold that satisfies the orientation constraint.

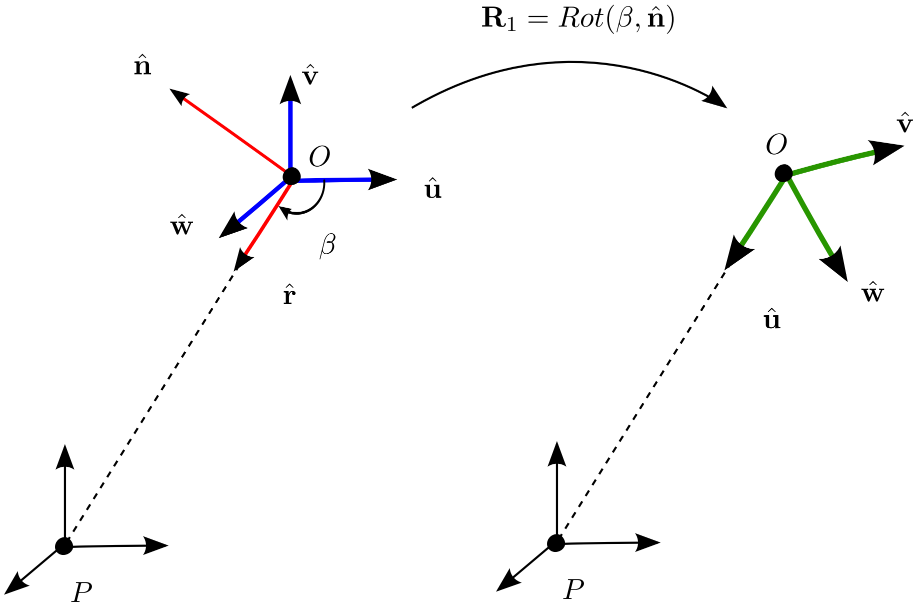


Figure 6.10: Rotation matrix $\mathbf{R}_1$ used to define the orientation constraint.

Finally, the homogeneous transformation that defines the position and orientation of $\mathcal{F}$ satisfying the orientation constraint is computed as

$$\mathbf{T} = \left(\begin{array}{ccc|c} & & & x \\ & \mathbf{R}_2\mathbf{R}_1 & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array}\right). \tag{6.4}$$

6.3 Summary

This chapter has presented a suitable way to reduce the dimension of the hand configuration space $\mathcal{X}_{\mathrm{h}}$, such that it can be used in the hand motion planning with a reasonable computational cost and human-likeliness during the approach path phase. The reduction of the finger configuration space is done by capturing the human hand workspace by means of a sensorized glove and then mapping the configurations to the mechanical hand workspace, where the most relevant finger motion coordinations can be identified using PCA. The wrist configuration is also constrained to belong to a configuration subspace of lower dimension. In this case, using a sphere-like constraint, the hand moves always facing the object, which is usually the case for humans too. Both reductions are meant to increase the efficiency when planning and, additionally, improve the quality of the randomly generated samples, as it will be shown in Chapter 7. It is worth noting that, even when the approach have been developed for the SA hand, the procedures are general enough to considere any other particular hand.

7

Finding Approach Paths

Planning the motion of a hand-arm system is a hard problem due to the high dimensionality of the combined system. This chapter deals with the problem of motion planning of such systems avoiding collisions and trying to mimic real human hand motions. The approach uses the results from the previous chapter to propose a solution with a compromise between motion optimality and planning complexity (time). The proposal is a PRM-based motion planner which is able to determine the motions of a hand-arm system towards any desired configuration, for instance, an optimal grasp from Part II, in other words, it provides a solution to the approach path planning problem for grasps satisfying the task-specific contact constraints.

In this chapter, we present a new approach for the motion planning of an anthropomorphic hand assembled on a robot arm. It is of particular interest for the hand-arm movement close to the goal configuration in a cluttered environment, that is, when the existence of a linear path for the arm considering a bounding volume for the hand is unlikely to be found. Therefore the movements must be planned in a meaningful space defined using the hand-arm degrees of freedom. Efficiency is sought by sampling hand configurations from lower dimensional subspaces, whose dimension is increased according to the difficulty of finding a collision-free approach path. Simultaneously, arm configurations are sampled using two strategies. The first one follows the classical approach, that is using a linear segment connecting the initial and the final arm configurations, but enhancing the path by allowing samples around it, as if a box were swept along it, that is, the resulting motion is a quasi-linear path inside a volume. The second one uses the wrist orientation constraints proposed in Chapter 6. In both cases, the configuration space of the arm is reduced.

Let $\mathcal{X} = \mathcal{X}_\mathrm{h} \times \mathcal{X}_\mathrm{a}$ be the configuration space of a hand-arm system, where $\mathcal{X}_\mathrm{h}$ and $\mathcal{X}_\mathrm{a}$ are the configuration spaces of the hand and of the manipulator arm, respectively. Then, the dimension of $\mathcal{X}$ is equal to the number of degrees of freedom of the hand plus the number of degrees of freedom of the arm. The problem to be solved is the following: given an initial hand-arm configuration $\mathrm{x}^\mathrm{i} \in \mathcal{X}$ and a final desired one $\mathrm{x}^\mathrm{f} \in \mathcal{X}$, find a collision free path in $\mathcal{X}$ from x^i to x^f, i.e. a collision free path for the hand-arm system.

The dimension of the search space, $\mathcal{X}$, is relatively large, and therefore conventional solutions require high computational times. In this context, the proposed approach is based on a reduction of the search space dimension proposed in Chapter 6, which is done by using a representative subspace $\mathcal{E}$ of the hand configuration space $\mathcal{X}_\mathrm{h}$ using Principal Motion Directions (PMDs), and then looking for continuous valid paths in the reduced subspace $\mathcal{R} = \mathcal{E} \times \mathcal{X}_\mathrm{a}$. Of course, there may be solutions in $\mathcal{X}$ not included in $\mathcal{R}$, thus, the selection of a proper $\mathcal{R}$ (i.e. a proper $\mathcal{E}$) is a relevant step in the proposed approach. On the other hand, if a solution is found in $\mathcal{R}$, for sure it is valid in $\mathcal{X}$. The following sections detail the sampling and interconnecting issues and the proposed general planning algorithm.

7.1 Sample generation

The basic features of the procedure to sample hand-arm configurations are listed below, and then the sampling algorithms are formally presented. The features are the following:

- A random sampling source is considered.
- Hand configurations are sampled from an axis-aligned box $\mathcal{B}_\mathrm{h}$ in the subspace $\mathcal{E}$ of dimension h, having side lenght $2\lambda_i$, where λ_i is a number proportional to the deviation of the data set in the ith PMD (Figure 7.1). Let:
  - $\mathbf{E} = (\hat{\mathrm{e}}_1, \hat{\mathrm{e}}_2, \ldots, \hat{\mathrm{e}}_h)^\top$ be a matrix with a base of $\mathcal{E}$ as columns,
  - $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_h)$ with $\epsilon_i \in [-\lambda_i, \lambda_i]$ be a sample obtained inside $\mathcal{B}_\mathrm{h}$,
  - $\bar{\mathrm{x}}_\mathrm{h}$ be the mean value of the data set used for the PCA analysis.

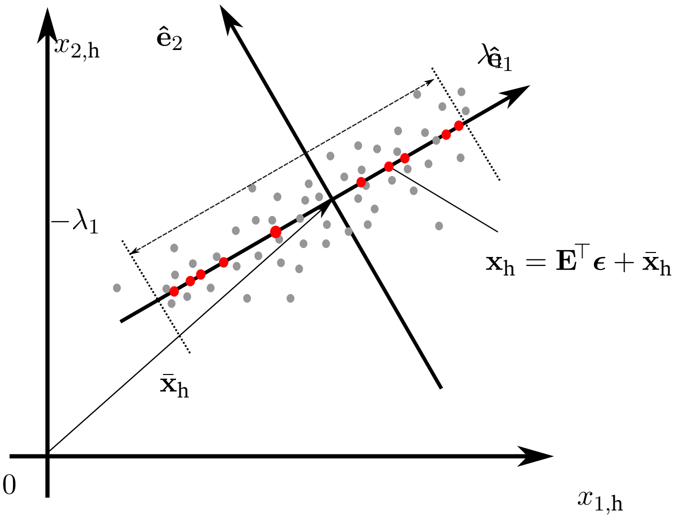

$x_{2,\mathrm{h}}$
$\hat{\mathrm{e}}_2$
λ_1
$\hat{\mathrm{e}}_1$
$-\lambda_1$
$\mathbf{x}_\mathrm{h} = \mathbf{E}^\top \boldsymbol{\epsilon} + \bar{\mathbf{x}}_\mathrm{h}$
$\bar{\mathbf{x}}_\mathrm{h}$
0
$x_{1,\mathrm{h}}$

Figure 7.1: A 2-dimensional example of a space $\mathcal{X}_\mathrm{h}$ conformed by the two joint values, $x_{1,\mathrm{h}}$ and $x_{2,\mathrm{h}}$, modelled with two PMDs, $\hat{\mathrm{e}}_1$ and $\hat{\mathrm{e}}_2$, obtained from the input dataset shown as gray points. The subspace $\mathcal{E}$ is 1-dimensional and defined by $\mathbf{E}^\top = (\hat{\mathrm{e}}_1)$. Samples are obtained from the sampling box $\mathcal{B}_\mathrm{h}$ that in this case is the segment $[-\lambda_1, \lambda_1]$ shown as bigger red dots on the $\hat{\mathrm{e}}_1$-axis.

Then, the joint values $\mathbf{x}_\mathrm{h}$ of the hand are obtained as:

$$\mathbf{x}_\mathrm{h} = \mathbf{E}^\top \boldsymbol{\epsilon} + \bar{\mathbf{x}}_\mathrm{h}. \tag{7.1}$$

In the present work, the dimension of $\mathcal{E}$ is not a fixed parameter but a parameter, h, that is iteratively increased by the planning algorithm, as required by the task. Correspondingly, the number of columns of $\mathbf{E}^\top$ is iteratively increased.

- The arm configurations are obtained selecting two strategies.
  - LINEAR: A sampling region for the arm configurations is defined around the segment s_a that connects the initial and the final arm configurations, $\mathbf{x}_\mathrm{a}^\mathrm{i}$ and $\mathbf{x}_\mathrm{a}^\mathrm{f}$. This region is defined as the union of hypercubes, $\mathcal{B}_\mathrm{a}(\mathbf{x}_\mathrm{a}^d)$, of side $2\rho_\mathrm{b}$ centered at evenly spaced points $\mathbf{x}_\mathrm{a}^\mathbf{d} \in s_\mathrm{a}$ separated a distance $d \leq \rho_\mathrm{a}$.
  - SPHERE: A sampling region is defined as the union of hypercubes in the Cartesian space, particularly, in the subset where the end-effector satisfies the wrist orientation constraint (Chapter 6), that is a cube of side $2\rho_x$ for the position and a segment of length $2\rho_\alpha$ for the orientation centered at the

end-effector pose corresponding to the configuration $\mathrm{x}_\mathrm{a}^\mathrm{d} \in s_\mathrm{a}$. Therefore, it relies on a known, and preferable complete, inverse kinematic procedure for the arm to determine its configuration sub-space.

- To obtain a collision-free hand-arm configuration, an arm configuration is sampled from the corresponding hypercube and a hand configuration sampled from $\mathcal{B}_\mathrm{h}$ is associated to it, until a non-collision hand-arm configuration is found. This is done trying up to n_a arm configurations and for each of them up to n_h hand configurations, using each time an increasing number of PMDs.

Algorithms 7.1 and 7.2 detail, respectively, the sampling procedures for the arm and the hand. They are called from the main algorithm (Section 7.3) that set the values n_h and n_a as fixed input parameters, and ρ_a as an input parameter that takes increasing values. The following functions are used in Algorithms 7.1 and 7.2:

RAND$(v, [a, b])$: Returns a vector of dimension v whose components have random values in the range $[a, b]$.

SELFCOLLISION(x): Takes as a parameter either a hand-arm configuration or an arm configuration. The function returns true if x makes the corresponding system to be in self-collision, or false otherwise.

COLLISION(x): Returns true if the input configuration x makes the hand-arm system to be in collision with the environment, or false otherwise.

DIM$(\mathcal{S})$: Returns the dimension of the space $\mathcal{S}$.

MAP$(\boldsymbol{\epsilon})$: Returns the configuration $\mathrm{x}_\mathrm{h} \in \mathcal{X}_\mathrm{h}$ corresponding to $\boldsymbol{\epsilon} \in \mathcal{E}$, as computed by Eq. (7.1).

GETPOSITION(x_a): Returns the position of the end-effector with coordinates (x, y, z), using forward kinematics.

GETORIENTATION$(\mathrm{x}_\mathrm{a}, {\mathrm{x}_\mathrm{a}}^f)$: Returns the rotation angle α of the wrist frame about the segment connecting the wrist positions at the configurations x_a and ${\mathrm{x}_\mathrm{a}}^f$.

SETWRISTCONSTRAINT$(x, y, z, \alpha, {\mathrm{x}_\mathrm{a}}^f)$: Returns the wrist frame represented in the homogenous matrix $\mathbf{T}$ that satisfies the orientation constraints.

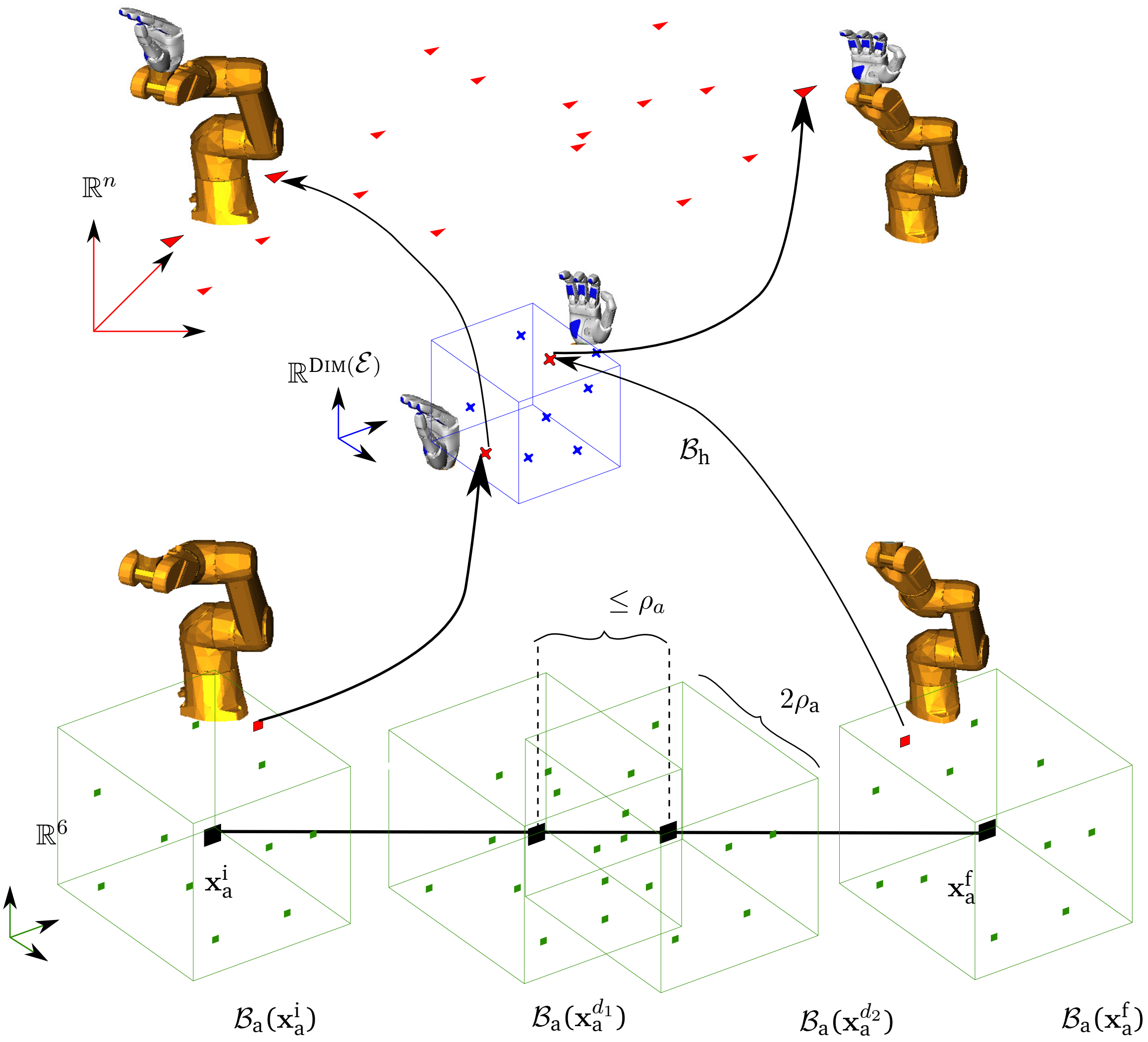

$\mathbb{R}^n$
$\mathbb{R}^{\text{DIM}(\mathcal{E})}$
$\mathcal{B}_h$
$\leq \rho_a$
$2\rho_a$
$\mathbb{R}^6$
$\mathbf{x}_a^i$
$\mathbf{x}_a^f$
$\mathcal{B}_a(\mathbf{x}_a^i)$
$\mathcal{B}_a(\mathbf{x}_a^{d_1})$
$\mathcal{B}_a(\mathbf{x}_a^{d_2})$
$\mathcal{B}_a(\mathbf{x}_a^f)$

Figure 7.2: Samples of the hand-arm system as a composition of arm and hand configurations.

INVKIN($\mathbf{T}$): Returns the arm configuration $\mathrm{x_a}$ where the wrist satisfies the orientation constraint. In our case, the arm has a closed-form solution, which is complete and very fast. If there is no solution to the inverse kinematic problem, the function returns the NOIKSOLUTION exception.

NULL: In the case that no collision-free sample could be obtained using the sampling procedures, the algorithms return the NULL exception.

Algorithm 7.1: Obtention of an arm configuration

```
SAMPLEARM($\mathbf{x_a}^{\mathrm{i}}, \mathbf{x_a}^{f}, d, \rho_{\mathrm{b}}, \rho_x, \rho_\alpha, \mathtt{mode}, n$)
input  : The initial, final and valid arm configuration, the distance $d$
         within the range $[0, 1]$, the half-sizes $\rho_{\mathrm{b}}$, $\rho_x$ and $\rho_\alpha$ of the
         sampling hypercubes for position and orientation, respectively,
         the strategy mode to be used, and the maximum number $n$ of
         trials.
output: An arm configuration $\mathbf{x_a}$ free of self-collisions, if found, or
         NULL otherwise.
$i \leftarrow 0$
while $i < n$ do
    switch mode do
        case LINEAR
            $\mathbf{x_a} \leftarrow \mathbf{x}_\mathrm{a}^\mathrm{i} + d(\mathbf{x}_\mathrm{a}^\mathrm{f} - \mathbf{x}_\mathrm{a}^\mathrm{i}) + \mathrm{RAND}(\mathrm{DIM}(\mathcal{X}_\mathrm{a}), [-\rho_\mathrm{b}, \rho_\mathrm{b}])$
        case SPHERE
            $\mathbf{x_a}^d \leftarrow \mathbf{x}_\mathrm{a}^\mathrm{i} + d(\mathbf{x}_\mathrm{a}^\mathrm{f} - \mathbf{x}_\mathrm{a}^\mathrm{i})$
            $(x^d, y^d, z^d) \leftarrow \mathrm{GETPOSITION}(\mathbf{x_a}^d)$
            $(x, y, z) \leftarrow (x^d, y^d, z^d) + \mathrm{RAND}(3, [-\rho_x, \rho_x])$
            $\alpha^d \leftarrow \mathrm{GETORIENTATION}(\mathbf{x_a}^d)$
            $\alpha \leftarrow \alpha^d + \mathrm{RAND}(1, [-\rho_\alpha, \rho_\alpha])$
            $\mathbf{T} \leftarrow \mathrm{SETWRISTCONSTRAINT}(x, y, z, \alpha, \mathbf{x_a}^\mathrm{f})$
            try $\mathbf{x_a} \leftarrow \mathrm{INVKIN}(\mathbf{T})$
            catch NOIKSOLUTION
                return NULL
    if FALSE=SELFCOLLISION($\mathbf{x_a}$) then
        return $\mathbf{x_a}$
    $i \leftarrow i + 1$
return NULL
```

7.2 Sample interconnection

As explained in Chapter 6, the hand is not able to change the configuration along any direction in $\mathcal{X}$, because the thumb-base joint has not velocity control. Therefore, the motion of the hand is performed by interleaving the motion of the thumb-base joint with the coordinated motion of all of the others joints of the hand, in a Manhattan-like manner. Then, the motion of the whole hand-arm system between two neighboring configurations is performed by dividing the arm motion in two parts and coordinating them with the two interleaved hand movements [99], one with the arm and th hand

Algorithm 7.2: Obtention of a hand-arm configuration

```
SampleHand($\mathrm{x}_a, n$)
input : The configuration $\mathrm{x}_a$ of the arm, the dimension $e$
        of $\mathcal{E}$, and the maximum number $n$ of trials.
output: A hand-arm configuration x free of collisions if
        found, or NULL otherwise.
$e \leftarrow 1$
while $e < h$ do
    $i \leftarrow 0$
    while $i < n$ do
        $\epsilon \leftarrow$Rand$(e, [0,1])$
        $\mathrm{x}_h \leftarrow$Map$(\epsilon)$
        $\mathrm{x} \leftarrow (\mathrm{x}_a, \mathrm{x}_h)$
        if False=SelfCollision(x) then
            if False=Collision(x) then
                return x
        $i \leftarrow i + 1$
    $e \leftarrow e + 1$
return Null
```

joints but the thumb base joint, and the other with the arm and the thumb base joint only. Therefore, the interconnection must be carried out as follows.

Let the thumb range associated to a given collision-free configuration x be the set of values that the thumb-base joint can sweep without producing any collision, provided that no other joint is changed. Then, two configurations are considered as neighboring configurations if the distance between them is below a given threshold and their thumb ranges intersect.

Then, in order to build a roadmap, neighboring samples are interconnected. The main features of the interconnection procedure are the following:

- The maximum number of neighboring samples is limited to the closest K samples, being K a given value.

- All the samples generated within the hypercube centered at x_a^i, $\mathcal{B}_a(\mathrm{x}_a^i)$, are forced to have x^i as a neighboring configuration, provided that their thumb ranges intersect and irrespective of whether x_a^i belongs to the closest K neighbors or not. The same is done for the final configuration x_a^f.

Algorithm 7.3: Connect a sample to the roadmap

```
ConnectSample($\mathcal{G}, \mathbf{x}$)
input  : The roadmap $\mathcal{G}$ and the configuration $\mathbf{x}$
output: The updated roadmap $\mathcal{G}$
$\mathcal{N} \leftarrow$FindNeighbors($\mathbf{x}$)
forall the $\mathbf{x}_\mathrm{n} \in \mathcal{N}$ do
    try
        $\mathcal{L} \leftarrow$M-LocalPlan($\mathbf{x}_\mathrm{n}, \mathbf{x}$)
        AddEdge($\mathcal{L}, \mathcal{G}$)
    catch NoEdge
        AddNode($\mathbf{x}, \mathcal{G}$)
return $\mathcal{G}$
```

- The local planner searches for a collision-free path between two neighboring configurations by uncoupling the motion of the thumb-base joint from the coordinated motion of all of the others joints of the hand, in a Manhattan-like manner, and dividing the motion of the arm in two parts and coordinating them with the two uncoupled phases of the hand [99].

Algorithm 7.3 shows the procedure that performs the connection of a sample to the roadmap. The following functions are used in this algorithm:

FINDNEIGHBORS($\mathbf{x}$): Finds the K-nearest neighbors of $\mathbf{x}$ from all the nodes of the roadmap $\mathcal{G}$. The neighboring threshold is set equal to the distance between $\mathbf{x}^\mathrm{i}$ and $\mathbf{x}^\mathrm{f}$.

M-LOCALPLAN($\mathbf{x}_i, \mathbf{x}_j$): Returns the collision-free path $\mathcal{L}$ connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfying that the thumb-base motions are interleaved with those of the other hand joints using with Manhattan-like maneuvers. If it can not connect the two nodes using a collision-free path it returns the NOEDGE exception.

ADDNODE($\mathbf{x}, \mathcal{G}$): Adds node $\mathbf{x}$ to graph $\mathcal{G}$. It automatically creates a new component and updates the graph.

ADDEDGE($\mathcal{L}, \mathcal{G}$): Adds edge $\mathcal{L} = [\mathbf{x}_i, \mathbf{x}_j]$ to graph $\mathcal{G}$. It automatically adds the non-existing node, merges the components they belong to if required, and updates the connected components of the graph.

7.3 The approach path planner

The main algorithm is a probabilistic roadmap planner that samples and interconnects the configurations as detailed in the two previous sections. It is an easy-to-tune adaptive algorithm whose principal features are:

- The dimension of the hand search space $\mathcal{E}$ is iteratively increased when no collision-free hand-arm configurations is found for a given arm configuration in $\mathcal{X}_{\mathrm{a}}$.
- The volume of the arm search space is iteratively increased each time the attempt to connect the initial and the final configurations fails, i.e. if no solution is found by sampling all the hypercubes $\mathcal{B}_{\mathrm{a}}$ (Figure 7.2), the size of hypercubes is increased, and a new iteration of the algorithm is launched.
- The main algorithm keeps track of the connected components that contain $\mathbf{x}^{\mathrm{i}}$ and $\mathbf{x}^{\mathrm{f}}$ in order to explore only a subset of the hypercubes $\mathcal{B}_{\mathrm{a}}(p_i)$ that define the sampling region for the arm configurations. This is done as follows. Let:
  - d^{i} be the maximum distance from $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$ to the center of a hypercube that has generated a sample that belongs to the same connected component than $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$,
  - d^{f} be the minimum distance from $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$ to the center of a hypercube that has generated a sample that belongs to the same connected component than $\mathbf{x}_{\mathrm{a}}^{\mathrm{f}}$.

  Then, only those hypercubes centered at points located at a distance $d \in [d^{\mathrm{i}}, d^{\mathrm{f}}]$ from $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$ are likely to generate samples that aid to interconnect the connected component of $\mathbf{x}^{\mathrm{i}}$ with that of $\mathbf{x}^{\mathrm{f}}$, as illustrated in Figure 7.3 (take into account that the distance from $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$ to $\mathbf{x}_{\mathrm{a}}^{\mathrm{f}}$ is considered unitary and that the hypercubes $\mathcal{B}_{\mathrm{a}}$ are swept following the Van der Corput sequence as explained in Section 7.1).
- There are no critical parameters to be tuned.

Algorithm 7.4 formally details the planning procedure that returns a path $\mathcal{P}$ connecting $\mathbf{x}^{\mathrm{i}}$ and $\mathbf{x}^{\mathrm{f}}$. The following functions are used:

ADJUSTSIZE$(k, \boldsymbol{\rho})$**:** Scales the vector $\boldsymbol{\rho}$ of parameters proportional to a given constant k.

VANDERCORPUT$(\mathbf{x}_i, \mathbf{x}_j, \rho, D)$**:** Computes a set of points evenly spaced along the segment defined by $\mathbf{x}_i$ and $\mathbf{x}_j$ normalized to the interval D, such that the number

Algorithm 7.4: Algorithm that plans the motion of the hand-arm system towards a final desired configuration.

```
ApproachPath($\mathbf{x}^\mathrm{f}, \mathbf{x}^\mathrm{i}, \rho_\mathrm{b}, \rho_x, \rho_\alpha, n_\mathrm{a}, n_\mathrm{h}, \mathtt{mode}, N$)
input : The final and initial configurations, $\mathbf{x}^\mathrm{f}$ and $\mathbf{x}^\mathrm{i}$, the initial half-sizes
        $\rho_\mathrm{b}$, $\rho_x$ and $\rho_\alpha$ of the sampling hypercubes, the maximum numbers
        $n_\mathrm{a}$ and $n_\mathrm{h}$ of trials to sample arm configurations per hypercube and
        to sample hand configurations per arm configuration, respectively,
        the strategy mode for the arm path planning, and the maximum
        number $N$ of trials for the roadmap generation.
output: The sequence of edges $\mathcal{P} \in \mathcal{G}$ forming the path that connects $\mathbf{x}^\mathrm{i}$
        and $\mathbf{x}^\mathrm{f}$
$\mathcal{P} \leftarrow \varnothing$
$\mathcal{G} \leftarrow \varnothing$
AddNode($\mathbf{x}^\mathrm{i}, \mathcal{G}$)
AddNode($\mathbf{x}^\mathrm{f}, \mathcal{G}$)
$c \leftarrow 2$
$k \leftarrow 0$
repeat
    $k \leftarrow k + 1$
    $[\rho_\mathrm{b}, \rho_x, \rho_\alpha] \leftarrow$AdjustSize($k, [\rho_\mathrm{b}, \rho_x, \rho_\alpha]$)
    $\mathcal{D} \leftarrow$VanDerCorput($\mathbf{x}^\mathrm{i}_\mathrm{a}, \mathbf{x}^\mathrm{f}_\mathrm{a}, \rho_\mathrm{b}, [0, 1]$)
    foreach $d \in \mathcal{D}$ do
        try
            $\mathbf{x}^d_\mathrm{a} \leftarrow$ SampleArm($\mathbf{x}^\mathrm{i}_\mathrm{a}, \mathbf{x}^\mathrm{f}_\mathrm{a}, d, \rho_\mathrm{b}, \rho_x, \rho_\alpha, \mathtt{mode}, n_\mathrm{a}$)
            $\mathbf{x}^d \leftarrow$SampleHand($\mathbf{x}^d_\mathrm{a}, n_\mathrm{h}$)
            $\mathcal{G} \leftarrow$ConnectSample($\mathbf{x}^d, \mathcal{G}$)
            $\mathcal{P} \leftarrow$FindPath($\mathcal{G}, \mathbf{x}^\mathrm{i}, \mathbf{x}^\mathrm{f}$)
            return $\mathcal{P}$
        catch NotInSameComponent
            $c \leftarrow c + 1$
            UpdateSequence($\mathcal{D}$)
            next $d$
        catch Null
            $c \leftarrow c + 1$
            next $d$
until $c < N$
return $\mathcal{P}$
```

of points in the set is a power of two and the distance between two points in the configuration space is below the given threshold ρ. In our case, the interval

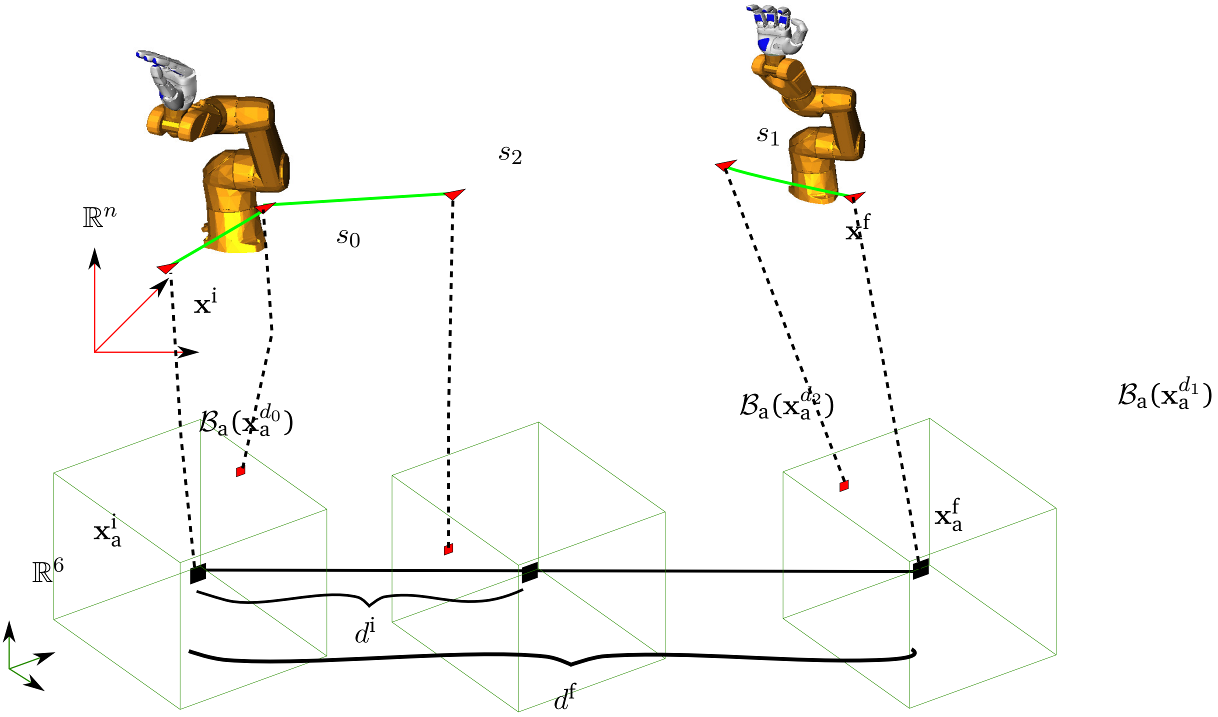

s_2
s_1
$\mathbb{R}^n$
s_0
$\mathbf{x}^{\mathrm{f}}$
$\mathbf{x}^{\mathrm{i}}$
$\mathcal{B}_{\mathrm{a}}(\mathbf{x}_{\mathrm{a}}^{d_2})$
$\mathcal{B}_{\mathrm{a}}(\mathbf{x}_{\mathrm{a}}^{d_1})$
$\mathcal{B}_{\mathrm{a}}(\mathbf{x}_{\mathrm{a}}^{d_0})$
$\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$
$\mathbf{x}_{\mathrm{a}}^{\mathrm{f}}$
$\mathbb{R}^6$
d^{i}
d^{f}

Figure 7.3: Roadmap under construction composed of two connected components after having sampled three configurations. Further exploration of the sampling region for the arm configurations is constrained to the hypercubes $\mathcal{B}_{\mathrm{a}}$ located at a distance $d \in [d^{\mathrm{i}}, d^{\mathrm{f}}] = [0.5, 1.0]$ from $\mathbf{x}_{\mathrm{a}}^{\mathrm{i}}$.

is initially set to $[0, 1]$. The elements in the set are finally re-ordered such that they follow the Van der Corput sequence [52], for which the points would be $d = 0, 1, 0.5, 0.25, 0.75, 0.125, \ldots$.

UPDATESEQUENCE($\mathcal{D}$): Updates the distances d^{i} and d^{f}, hence the interval $D = [d^{\mathrm{i}}, d^{\mathrm{f}}]$ used to defined the set $\mathcal{D}$. It automatically discards the points $d \in \mathcal{D}$ that fall outside the interval.

FINDPATH($\mathcal{G}, \mathbf{x}^{\mathrm{i}}, \mathbf{x}^{\mathrm{f}}$): First, verifies whether the nodes $\mathbf{x}^{\mathrm{i}}$ and $\mathbf{x}^{\mathrm{f}}$ belong to same connected component or not. If they belong to the same component, it returns a path $\mathcal{P} \in \mathcal{G}$ connecting nodes $\mathbf{x}^{\mathrm{i}}$ and $\mathbf{x}^{\mathrm{f}}$ using the A^* algorithm and a smoothing procedure. If they do not, it returns the NOTINSAMECOMPONENT exception.

7.4 Test cases

All the cases presented here were studied using the robot simulation toolkit for motion planning and teleoperation guiding, which has served to generate and validate the paths before executing them on the physical devices. For the simulator development, three guidelines were considered [74]: ability to run on different platforms, code accessability and software modularity. The first two allow the use of cross-platform and open-source tools. Regarding the software modularity, the project was conceived to be library-based, thus, different libraries have been developed such as a Geometric library for the treatment of the bodies and their kinematic relation, a Sampling library with different sampling strategies, a Motion Planning library, a Device library for the communication with different devices, and, finally, the GUI library that implements the user interface and library management.

For the validation, the hand is mounted on the industrial arm Säubli TX90, which has 6 revolute joints. The validation of the proposed approach has been carried out both in a virtual environment with simulated elements, as well as in a real scene with the actual hand-arm system. In order to evaluate the use of the PMDs, the planner will be casted using the LINEAR strategy, and the results are compared with the case where no PMDs are used (Subsection 7.4.1), and secondly, using the SPHERE strategy using the PMDs, and the results are compared with the case where no strategy is used for the sampling of arm configurations, that is, the samples are obtained over the full configuration space (Subsection 7.4.2).

7.4.1 Evaluation of the use of PMDs

As a benchmark, the task of grasping a can on a table was selected. The result of the proposed planner using the LINEAR strategy is compared with the case where no PMDs are used, i.e. samples of the hand configuration are obtained from the whole hand configuration space.

The quantitative results of using PMDs in motion planning are summarized and compared in Table 7.2. These results were obtained using a desktop computer equipped with a 3.00GHz Intel Core2 CPU, running Windows operating system and initial parameters of the main algorithm shown in Table 7.1. It is noticeable the decrease in the time required to find a solution when using PMDs (more than 50 times faster), due basically

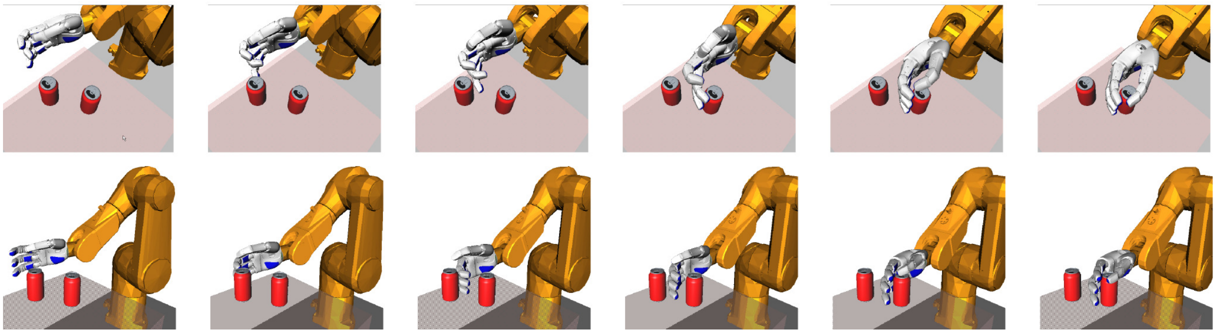


Figure 7.4: Qualitative comparison between the approach that considers the full hand configuration space (top) and the proposed approach that reduces the hand workspace using the PMDs (bottom). The use of PMDs resulted in a path composed of a smooth sequence of human-like postures.

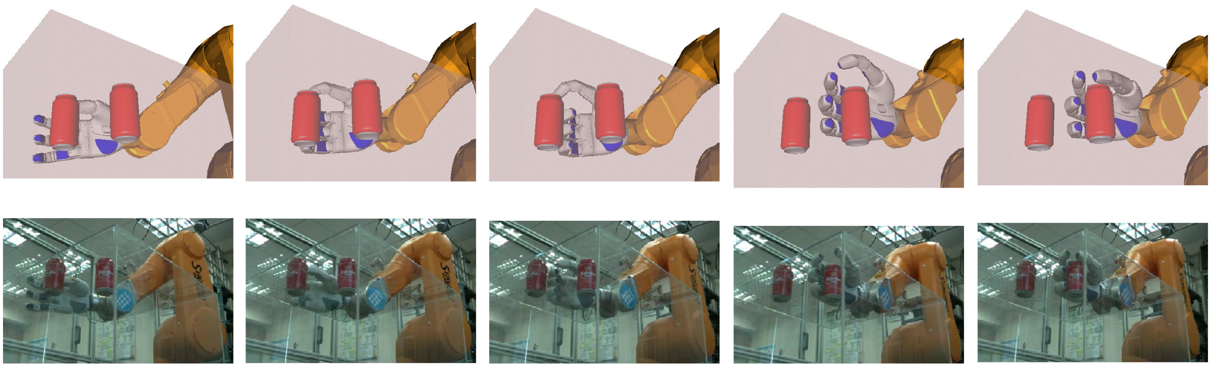


Figure 7.5: Simulation of a solution path and real execution in the actual hand-arm system.

to the fact that using PMDs collision-free samples are more easily found. Also, the number of samples required resulted very small (less than 1% of the samples required when no PMDs were used). The qualitative results are even more interesting (see Figure 7.4). Even though a smoothing procedure is always applied, the solutions found when sampling the whole hand configuration space contain awkward hand postures, ressembling movements of humans with coordination problems. On the contrary, the proposed method provides a sequence of well-coordinated human-like postures, which requires less smoothing and prunning post-processing.

As it was previously mentioned, a solution path was successfully implemented on the actual hand-arm system, the screenshots are exposed in Figure 7.5, where both

Parameters	ρ_a	K	N	n_a	n_h
Value	0.001	10	1000 (10,000)	10 (20)	10 (20)

Table 7.1: Parameters used in the approach path planner. The values shown for the adaptive parameters are the initial one; the values in parenthesis are used when no PMDs are considered.

Comparison factors	Proposed	Classical
Time to find a solution [s]	16.83	915.28
Smoothing time [s]	5.08	15.20
Final neighboring threshold (ρ_b)	0.0013	0.0631
Maximum number of samples required (c)	698.82	7274.78
Proportion of samples generated by using:		
thumb-base + 2 PMDs	22.4%	–
thumb-base + 3 PMDs	26.5%	–
thumb-base + 4 PMDs	16.5%	–
thumb-base + 5 PMDs	34.7%	–
Total nodes in the PRM	22.55	270.44
Total nodes in the solution path	7.27	3.22

Table 7.2: Comparison in averaged values between the proposed approach in which the hand workspace has been reduced using PMDs and the classical approach in which the workspace is the full configuration space.

the virtual and the real path are shown in their equivalent points on the path. The implementation on the real hand-arm system makes the usefulness of the proposed algorithms visible, along with the particular specific hardware constraints (the non-customizability of the thumb-base joint controller of the SA hand) have been correctly tackled with an appropriate local-planner.

Performance study

Assuming a given grasp or pre-grasp configuration, the proposed approach looks for the final approaching motion, where the collisions are more likely to occur with the hand rather than with the arm (i.e. collision-free solution paths will require finger motions and only slight arm deviations from the straight motion). With this in mind, the planner has been evaluated on several problems, four of them shown in Figure 7.6, with different degrees of difficulty. In comparison with the task used for the evaluation, the tasks used for the performance shown Figure 7.6 have: (1) a narrower passage, (2) a goal configuration closer to the obstacles, (3) a rectilinear path to the goal more obstructed by the presence of the T-shaped object and of the shelf itself, (4) a more cluttered environment with a longer narrow passage (this task is similar to that used in [2]). The solution paths required motions of the finger joints, maintaining the robot configurations close to the rectilinear path, and resulted in smooth sequences of human-like configurations (Figure 7.7). The algorithm was run in a computer with a I5 processor with 4 cores and 4 Gb of RAM, under Windows 7 64-bit. The testing procedure was parallelized using the MPI library [39] in order to use all cores. The quantitative results are shown in Table 7.3. Note that the fourth task required the generation of

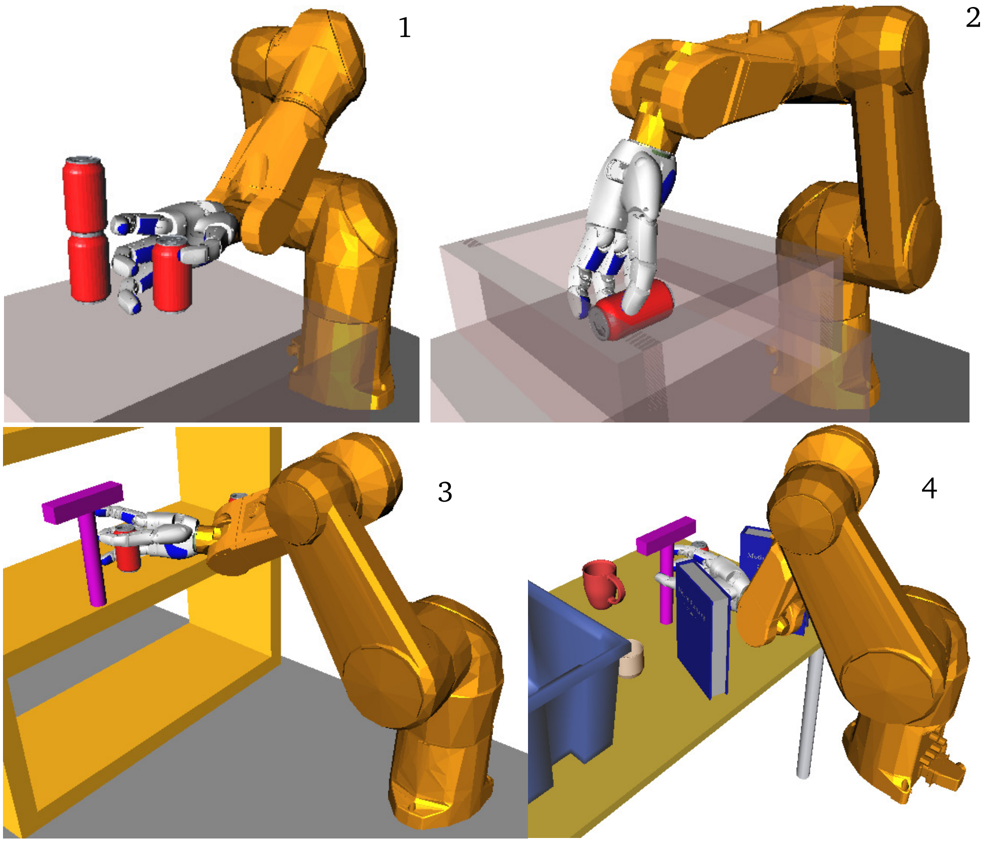


Figure 7.6: Goal configurations of the hand-arm system for some of the tasks used to test the planner: 1) Cans on a desk; 2) Can in a box; 3) Cans in a shelf; 4) T-shape object in a complex scene.

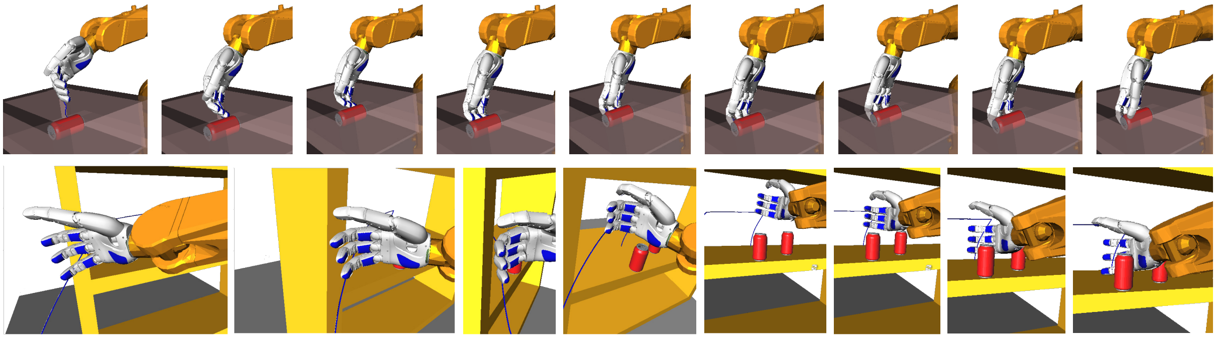


Figure 7.7: Some screen shots of the solution paths of tasks (2) and (3) illustrated in Figure 7.6.

Problem	1	2	3	4
Time to find a solution [s]	21.06	7.75	71.64	24.09
Smoothing time [s]	0.124	0.022	0.821	0.142
Number of samples	1188.7	437.7	2093.8	2205.6
Nodes in the PRM	53.6	24.6	131.7	81.9
Nodes in solution path	3.9	3.4	4.2	4.9

Table 7.3: Comparison of the performance of the planner used to solve the problems illustrated in Figure 7.6. These values are the means from 1000 runs.

more samples than the third task, because the environment is more cluttered and many samples resulted in collision, but it could be solved with a PRM composed of less nodes because the narrow passage was more aligned with the direction connecting $\mathbf{x}_{\text{ini}}$ and $\mathbf{x}_{\text{goal}}$. Therefore, the time to find a solution was larger in the third task because the validation of the PRM edges is time-consuming.

Discussion

The value of ρ_{b} determines how far the arm path can be from the rectilinear segment in $\mathcal{X}_{\text{a}}$ that connects $\mathrm{x}_{\text{a}}^{\text{i}}$ and $\mathrm{x}_{\text{a}}^{\text{f}}$. During the final approaching motion to grasp an object, the potential collisions are likely to occur with the hand, not with the arm. The value of ρ_{b} also determines the number of samples considered for each iteration of the general loop, i.e. the number of hypercubes $\mathcal{B}_{\text{a}}$ considered, although the neighboring threshold is an independent value and configurations sampled from non-contiguous hypercubes can be connected in the roadmap. The value of ρ_{b} is iteratively increased, and the initial chosen value is not a critical issue. It has to be neither too small (since then its increase could be too slow and too many samples could be required), nor to large (since then the search space could be too large and also too many samples could be required). Values between 0.001 and 0.05 gave good results for different tasks (the parameter is non-dimensional since each direction of the configuration space is normalized in the range $[0, 1]$).

The proposed approach searches the hand postures using as few PMDs as possible, which results in smoother motions all along the solution path. Moreover, the use of PMDs results in a better computational efficiency because the percentage of collision-free samples is much higher than when sampling directly the finger joints [97].

The values n_{h} and n_{s} allow several trials in the difficult parts of the path, increasing the posisblity to find a collision-free hand-arm configuration. These values are by no means critical, since the successive iterations of the main loop also permit the resampling of the difficult areas.

The distance threshold used to consider two configurations as neighboring samples is set equal to the distance between the initial and the final configuration, hence, it is not a user-defined parameter but calculated from the input of the problem.

7.4.2 Evaluation of the use of wrist orientation constraints

As an example to illustrate the SPHERE strategy, the SA hand is required to move, among obstacles, from an unconstrained configuration to a preshape configuration to grasp a can. Figure 7.8 (top) shows the snapshots of the solution path found when the task has been programmed with an orientation constraint and using the PMDs as well. Snapshots (1) and (5) show the initial and the final configurations, respectively. Snapshots (2),(3), and (4) are intermediate configurations.

The same task has been programmed without using the orientation constraint with

			Sampling	Total
Constraints	Success	# Nodes	Time (s)	Time (s)
With	100%	90 ± 71	8.9 ± 6.7	85 ± 68
Without	62%	37 ± 9	169 ± 85	198 ± 80

Table 7.4: Computation time of the planner when considering and omitting the orientation constraint.

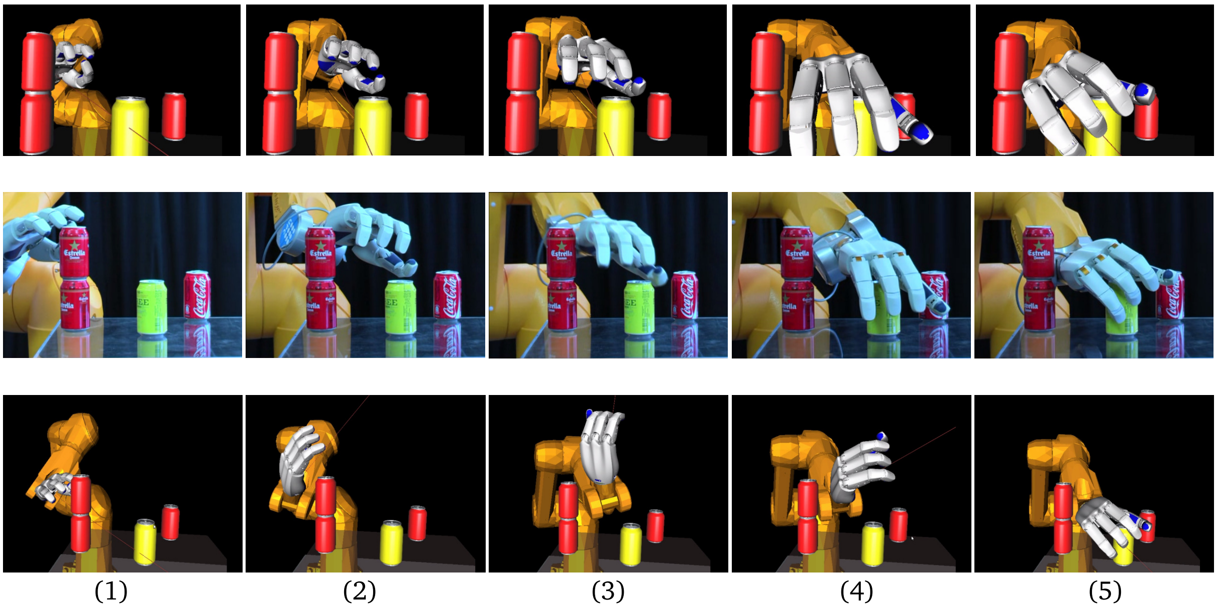

(1) (2) (3) (4) (5)

Figure 7.8: Snapshots of the results: (Top) task simulation using the orientation constraint; (Middle) task execution using the orientation constraint; (Bottom) task simulation without using the orientation constraint.

the same initial and final configurations. The result is shown in Figure 7.8 (bottom). Table 7.4 shows the results of the comparison in terms of computational efficiency measured as computational time (running on a PC@3GHz) used by the sampling procedures. The sampling time is much shorter when considering constraints because in this case samples are more often collision-free and reachable. Using virtual constraints further benefits in both senses as shown in Table 7.4 and in the snapshots of Figure 7.8. The task has been executed in the real robotic system, as shown in Figure 7.8 (middle), where it is shown how the arm movements are more realistic, in the sense that it moves directly and smoothly towards the object.

7.5 Summary

This chapter presented a motion planner for a hand-arm robotic system. The proposal pursues efficiency and human-likeliness in the hand postures while finding the approach path that connects the system from any initial configuration to any desired configuration, including those grasps obtained in the Part II. Planning is done with a probabilistic roadmap planner, and the dimensionality reduction in the hand search space poposed in the previous chapter results in lower computational time. The dimension of the hand search space and the volume of the arm search space are iteratively increased as much as it is required by the difficulty of the task. The arm configurations are sampled using two strategies. With the first one, the sampling is performed on a box swept around the rectilinear segment connecting the initial and final arm configurations. With the second strategy, that uses the orientation constraints proposed in the previous chapter, the dimension of the search space is reduced to 4, allowing a wider workspace exploration while keeping the path biased toward the goal. The proposed method has no critical parameters to be tuned such as thresholds for transitions configurations.

The approach has been implemented for the SA hand, which requires a particular local planner to decouple the motion of the thumb-base joint from the other hand joints, but is a general procedure for any other mechanical hand. The validity of the approach has been demonstrated both in simulation and in real experiments.

Part IV

Closing Remarks

8
Conclusions

As in any research, the present work opens more questions than it settles, suggesting new avenues for future investigation. This chapter summarizes the main contributions of the thesis, and highlights several points deserving further attention.

8.1 Summary of contributions

The main contribution of the thesis is a general procedure for planning grasps subject to specific contact constraints. The fact that such constraints allow to define particular grasps for specific tasks is the main motivation behind our work, but the enforcement of such constraints also provides a means of ensuring selective grasps suitable to arbitrary purposes, such as grasps in which the fingers should avoid the touching of fragile areas, or object regions that might be harmful to the robot.

The problem has been approached by subdividing it into the grasp synthesis and approach path planning subproblems, for which the following solution methods have been proposed:

1. **A complete method to compute kinematically-feasible grasps** (Chapter 3). The method is based on a previous formulation for general multi-body systems articulated through lower pair joints [80], extended to include additional constraints for modelling general patch-patch contact constraints. The formulation is suitable to apply a linear relaxation technique to compute a solution, which allows to find a

kinematically-feasible grasp whenever one exists, or to prove grasp non-existence otherwise. As opposed to other methods in the literature, the proposed method does not need to be fed with an initial estimation of the final grasp and can cope with general region-to-region contact constraints.

2. **A grasp optimization method that circumvents the local minima problem** (Chapter 4). The optimization process starts with a kinematically-feasible grasp— such as one computed by the previous method— and then focuses the exploration on a relevant subset of feasible grasps. This subset is traced exhaustively by using recently-developed techniques of higher-dimensional continuation [42]. A detailed atlas of the subset is obtained as a result, on which the highest-quality grasp according to any desired criterion, or a combination of criteria, can be readily identified.

3. **A method to account for compliant joints and contacts** (Chapter 5). The approach is inspired by the notion of soft synergies [6, 82]. It effectively introduces an elastic model of the hand whereby the physical hand is attracted towards a reference hand configuration through a set of virtual springs representing the compliance of the musculoskeletal system, while at the same time being repelled by the compliant surface of the object. This model allows to tackle all kinematic and restrainablity constraints of the grasp simultaneously, and the computed solution provides a set of hand configurations that can be used to execute the grasp using a position controller. The method is advantageous in comparison to methods of a similar spirit, which either neglect some of the mentioned constraints, limit the approach to planar grasps, or use a reduced number of fingers.

4. **A technique to reduce the dimension of the hand configuration space in a meaningful way** (Chapter 6). The reduction comes from finger coordination and wrist orientation constraints. As for the former constraints, they are inspired by the concept of hand postural synergies, with a difference on how the data points to be processed are generated. These points are typically obtained from static object prehensions, so that the results of a Principal Component Analysis provide hand postural synergies. In our case, the points are obtained by moving the hand in the free space instead, obtaining principal motion directions in the hand configuration space that allow the coordination of the joint movements. As for the

latter constraints, two approaches were explored. The first approach is a linear path reduction similar to the one usually performed in previous works [3, 61, 67, 118], but sweeping a box along the path and sampling in it in our case, which allows the arm to deviate slightly from the adopted path. The second approach introduces a way to constrain the wrist orientation, which biases the path towards the object while keeping the palm facing to a given point in the object. In both cases, the reduction will allow an efficient search with a tradeoff between the mobility of the system and the probability of finding a collision-free path in a complex scene.

5. **An adaptive planner to compute approach paths to the object** (Chapter 7). The planner provides methods to generate and interconnect samples in a reduced configuration space induced by the previously-mentioned finger-coordination and wrist-orientation constraints. The main feature of such planner is its ability to adapt to the complexity of the scene. If the environment is simple, only a few principal motion directions are used. As the environment gets more complex, and finding a path is harder, additional principal motion directions are considered, allowing to increase the overall mobility of the system, and hence the probability of finding a collision-free path.

Appendix A summarizes the publications through which these methods have been disseminated so far.

8.2 Future research directions

The following points deserving further attention have been identified:

1. **Automatic generation of the contact constraints**. *A priori* knowledge of the contact constraints is assumed throughout the work. As exposed in Section 1.2, there exist algorithms that might be helpful to determine them, but further work needs to be done on automating the process of deciding which object regions should be placed in contact with which specific hand regions. While some heuristic methods have been proposed for the case in which such regions reduce to isolated points [32, 122], algorithms able to cope with general free-form regions are still

to be developed. Furthermore, determining a suitable set of contact constraints from a higher-level specification of the manipulation task is an open issue as well.

2. **Generation of postural synergies of the hand in contact with the object**. In Chapter 4, hand postural synergies were introduced to reduce the dimension of the hand configuration space, specifically introducing constraints in $\mathbf{x}_\mathrm{h}$. But postural synergies are generally obtained using the joint values, so that the amount of dimension reduction is limited by the number of joints. In order to obtain further suitable reductions, an interesting approach would be to study how the contacts on the hand may vary with respect to the hand configuration. For instance, when we have the hand almost closed, we typically use the furthest part of the fingertip right under the nail. Oppositely, when the hand is widely open, we tend to use the middle part of the fingerprint. Thus, a generalized dimension reduction procedure could be envisaged based on introducing constraints in both $\mathbf{x}_\mathrm{h}$ and $\mathbf{x}_\mathrm{c}$.

3. **Optimization of the grasp stability from a control standpoint**. Whereas the optimization framework proposed in Chapter 4 can in principle tackle any kind of grasp quality metric, the stability of a grasp requires a more detailed study. To achieve stability, the positive definiteness of the so-called grasp impedance matrix must be ensured, which involves a non-linear programming problem for a known configuration [108], thus making the problem more complex.

4. **Optimization on all connected components of the manifold of kinematically-feasible grasps**. The presented optimization framework operates in the connected component of the manifold of relevant grasps that contains an initial kinematically-feasible grasp. In most robotic hands this is not an issue since, due to joint range limitations, such manifold only contains one connected component. However, this might not be the case in general. Ways to obtain one starting point in each connected component of the manifold of relevant kinematically-feasible grasps would extend this work to more general settings.

5. **Study of virtual constraints for the approach path to the object**. In Chapter 6, a virtual constraint was proposed to reduce the wrist configuration space. This constraint enforces the hand to face a center point on the object, which seems appropriate for a wide range of cases. However, other relational positioning constraints might be used in principle, such as enforcing the hand to face the

axis of a cylinder for grabbing long tubes, or to face a plane for particular pick-and-place operations.

6. **Consideration of time-varying environments**. The presented grasp planning method considers that the environment is static with objects and obstacles not changing over time. Even thought this is a typical scenario in every-day activities, it would be desirable to have a grasp planning method, with the ability to consider time-varying environments, including moving objects in them. In this case, switching to Rapidly-exploring Random Tree techniques [51] seems appropriate, since they are faster than the Probabilistic Roadmap technique used here, which is an asset for real-time applications.

7. **Optimization of the approach path to the object**. The path determined by the proposed method accounts for collision avoidance constraints, but it would be advisable to include the treatment of objective functions so as to optimize the approach path according to a desired criterion.

8. **Consideration of uncertainty in the models and their locations**. The present work assumes that the geometric models and locations of the hand-arm system, the object, and the environment are accurately known. Ways to introduce uncertainties in the values defining these models and locations are necessary to make the grasp planning process more robust.

9. **Unification of the grasp synthesis and the approach path planning into a one-step method**. A recent work presents an approach to unify these two subproblems [118]. However, as discussed in Chapter 2, this approach is not targeted to find grasps that allow manipulation tasks requiring the object to be contacted at specific locations. Even though the modules developed in Parts II and III are able to solve the grasp planning problem, a higher-level planner needs to be implemented to iteratively call such modules whenever a solution is not found in a first attempt.

A
List of Publications

The results of the present work have been disseminated through the following publications:

Journal papers

- **Rosales, C.**, L. Ros, J. M. Porta, and R. Suárez (**2011**). Synthesizing grasp configurations with specified contact regions. *The International Journal of Robotics Research 30*(4), 431–443 [93].

- Rosell, J., R. Suárez, **C. Rosales**, and A. Pérez (**2011**). Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. *Autonomous Robots 31*(1), 87–102 [100].

(under review)

- **Rosales, C.**, J. M. Porta and L. Ros (**2012**). Grasp Optimization under Task-dependent Contact Constraints (*submitted to the IEEE Transactions on Robotics*).

Conference papers

- **Rosales, C.**, R. Suarez, M. Gabiccini, and A. Bicchi (**2012**). On the synthesis of feasible and prehensile robotic grasps. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pp. 550–556 [94].

- **Rosales, C.**, J. Porta, and L. Ros (**2011**). Global optimization of robotic grasps. In *Proceedings of Robotics: Science and Systems VII* [91].

- Rosell, J., R. Suárez, A. Pérez, and **C. Rosales** (**2011**). Including virtual constraints in motion planning for anthropomorphic hands. In *Proceedings of the 2011 IEEE International Symposium on Assembly and Manufacturing*, pp. 1–6 [96].

- Sun, S.-C., **C. Rosales**, and R. Suárez (**2010**). Study of coordinated motions of the human hand for robotic applications. In *Proceedings of the 2010 IEEE International Conference on Information and Automation*, pp. 776–781 [115].

- Suárez, R., J. Rosell, A. Perez, and **C. Rosales** (**2009**). Efficient search of obstacle-free paths for anthropomorphic hands. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1773–1778 [114].

- Rosell, J., R. Suárez, **C. Rosales**, J. A. García, and A. Pérez (**2009**). Motion planning for high DOF anthropomorphic hands. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 4025–4030 [98].

- **Rosales, C.**, J. M. Porta, R. Suárez, and L. Ros (**2008**). Finding all valid hand configurations for a given precision grasp. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pp. 1634–1640 [92].

Technical reports

- Rosell, J., R. Suárez, **C. Rosales**, and A. Pérez (**2009**). A practical motion-planning approach for the Schunk Anthropomorphic Hand. *Technical Report IOC-DT-P-2009, IOC-UPC.*

- Rosell, J., R. Suárez, **C. Rosales**, J. A. García and A. Pérez (**2008**). Efficient motion planning for high DOF hands using principal motion directions. *Technical Report IOC-DT-P-2008-13, IOC-UPC*

- **Rosales, C.**, J. M. Porta, R. Suárez, and L. Ros (**2008**). Solving hand inverse kinematics for precision grasps. *Technical Report IOC-DT-P-2007-09, IOC-UPC.*

References

[1] Barber, C., D. Dobkin, and H. Huhdanpaa (1996). The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software 22*(4), 469–483.

[2] Berenson, D., R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner (2007). Grasp planning in complex scenes. In *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robots*, pp. 42–48.

[3] Berenson, D. and S. S. Srinivasa (2008). Grasp synthesis in cluttered environments for dexterous hands. In *Proceedings of the 2008 IEEE-RAS International Conference on Humanoid Robots*, pp. 189–196.

[4] Bicchi, A. (1994). On the problem of decomposing grasp and manipulation forces in multiple whole-limb manipulation. *Robotics and Autonomous Systems 13*(2), 127–147.

[5] Bicchi, A. (1995). On the closure properties of robotic grasping. *The International Journal of Robotics Research 14*(4), 319–334.

[6] Bicchi, A., M. Gabiccini, and M. Santello (2011). Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society B: Biological Sciences 366*(1581), 3153–3161.

[7] Bicchi, A. and D. Prattichizzo (2000). Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms. *IEEE Transactions on Robotics and Automation 16*(4), 336–345.

[8] Bicchi, A. and P. van der Smagt (2010). THE Hand Embodied project (2010-2014). http://www.thehandembodied.eu/.

[9] Birglen, L., T. Laliberté, and C. M. Gosselin (2008). *Underactuated Robotic Hands (Springer Tracts in Advanced Robotics)* (1st ed.). Springer.

[10] Borst, C., M. Fischer, and G. Hirzinger (1999). A fast and robust grasp planner for arbitrary 3D objects. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 1890–1896.

[11] Borst, C., M. Fischer, and G. Hirzinger (2002). Calculating hand configurations for precision and pinch grasps. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 1553–1559.

[12] Brown, E., N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger (2010). Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences 107*(44), 18809–18814.

[13] Buss, M., H. Hashimoto, and J. B. Moore (1996). Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation 12*(3), 406–418.

[14] Butterfass, J., M. Fischer, M. Grebenstein, S. Haidacher, and G. Hirzinger (2004). Design and experiences with DLR hand II. In *Proceedings of the World Automation Congress*, pp. 105–110.

[15] Caffaz, A. and G. Cannata (1998). The design and development of the DIST-Hand dextrous gripper. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 2075–2080.

[16] Carloni, R. (2006). *Robotic Manipulation: Planning and Control for Dexterous Grasp*. Ph. D. thesis, Università di Bologna, Bologna, Italy.

[17] Castellet, A. and F. Thomas (1998). An algorithm for the solution of inverse kinematics problems based on an interval method. In M. Husty and J. Lenarcic (Eds.), *Advances in Robot Kinematics*, pp. 393–403. Kluwer Academic Publishers.

[18] Cheng, F.-T. and D. E. Orin (1991). Optimal force distribution in multiple-chain robotic systems. *IEEE Transactions on Systems, Man and Cybernetics 21*(1), 13–24.

[19] Ciocarlie, M. and P. Allen (2011). A constrained optimization framework for compliant underactuated grasping. *Mechanical Sciences 2*(1), 17–26.

[20] Ciocarlie, M. T. and P. K. Allen (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research 28*(7), 851–867.

[21] Claret, J. A. and R. Suárez (2011). Efficient and practical determination of grasping configurations for anthropomorphic hands. In *Proceedings of the 18th IFAC World Congress*.

[22] Cornellà, J. and R. Suárez (2006). A new framework for planning three-finger grasps of 2D irregular objects. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5688–5694.

[23] Cox, D., J. Little, and D. O'Shea (1997). *An Introduction to Computational Algebraic Geometry and Commutative Algebra* (2nd ed.). Springer.

[24] Cutkosky, M. R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation 5*(3), 269–279.

[25] Cutkosky, M. R. and I. Kao (1989). Computing and controlling compliance of a robotic hand. *IEEE Transactions on Robotics and Automation 5*(2), 151–165.

[26] Denavit, J. and R. Hartenberg (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Transactions of the ASME Journal of Applied Mechanics 23*, 215–221.

[27] Didrit, O., M. Petitot, and E. Walter (1998). Guaranteed solution of direct kinematic problems for general configurations of parallel manipulators. *IEEE Transactions on Robotics and Automation 14*(2), 259–266.

[28] Dizioğlu, B. and K. Lakshiminarayana (1984). Mechanics of form closure. *Acta Mechanica 52*(1), 107–118.

[29] Dollar, A. M. and R. D. Howe (2007). Simple, robust autonomous grasping in unstructured environments. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 4693–4700.

[30] Farin, G. (2001). *Curves and Surfaces for CAGD: A Practical Guide* (5th ed.). Morgan Kaufmann.

[31] Feix, T., J. Romero, CarlHenrik, D. Kragic, and O. Co. (2009). Human grasping database. http://grasp.xief.net/.

[32] Fernandez, C., O. Reinoso, A. Vicente, and R. Aracil (2005). Kinematic redundancy in robot grasp synthesis. an efficient tree-based representation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1203–1209.

[33] Ferrari, C. and J. Canny (1992). Planning optimal grasps. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2290–2295. IEEE Comput. Soc. Press.

[34] Gabiccini, M. and A. Bicchi (2010). On the role of hand synergies in the optimal choice of grasping forces. In *Proceedings of Robotics: Science and Systems VI*, Zaragoza, Spain.

[35] Gazeau, J. P., S. Zehloul, M. Arsicault, and J. P. Lallemand (2001). The LMS hand: force and position controls in the aim of the fine manipulation of objects. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 2642–2648.

[36] Giurintano, D. J., A. M. Hollister, W. L. Buford, D. E. Thompson, and L. M. Myers (1995). A virtual five-link model of the thumb. *Medical Engineering & Phys 17*(4), 297–303.

[37] Goldfeder, C., M. Ciocarlie, H. Dang, and P. Allen (2009). The columbia grasp database. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 1710–1716.

[38] Gorce, P. and N. Rezzoug (2005). Grasping posture learning with noisy sensing information for a large scale of multifingered robotic systems: Research articles. *Journal of Robotic Systems 22*(12), 711–724.

[39] Gropp, W., E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface (Scientific and Engineering Computation)* (1st ed.). The MIT Press.

[40] Henderson, M. E. (2002). Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos 12*(3), 451–476.

[41] Henderson, M. E. (2005). Multiparameter parallel search branch switching. *International Journal of Bifurcation and Chaos in Applied Science and Engineering 15*(3), 967–974.

[42] Henderson, M. E. (2007). *Numerical continuation methods for dynamical systems: path following and boundary value problems*, Chapter Higher-Dimensional Continuation. Springer.

[43] Iwata, H. and S. Sugano (2009). Design of human symbiotic robot TWENDY-ONE. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 580–586.

[44] Jia, Y.-B. Y., F. Guo, and J. Tian (2011). On two-finger grasping of deformable planar objects. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 5261–5266.

[45] Jolliffe, I. (2002). *Principal Component Analysis*. Springer Series in Statistics.

[46] Juan, S. H. and J. M. Mirats Tur (2008). A method to generate stable, collision free configurations for tensegrity based robots. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3769–3774.

[47] Kamakura, N., M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura (1980). Patterns of static prehension in normal hands. *The American Journal of Occupational Therapy 34*(7), 437–445.

[48] Kawasaki, H., T. Komatsu, and K. Uchiyama (2002). Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. *IEEE/ASME Transactions on Mechatronics 7*(3), 296–303.

[49] Kerr, J. and B. Roth (1986). Analysis of multifingered hands. *The International Journal of Robotics Research 4*(4), 3–17.

[50] Kragic, D., A. T. Miller, and P. K. Allen (2001). Real-time tracking meets online grasp planning. In *Proceedingsf the 2001 IEEE International Conference on Robotics and Automation*, pp. 2460–2465.

[51] Kuffner, J. J. and S. M. LaValle (2000). RRT-connect: An efficient approach to single-query path planning. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 995–1001.

[52] Kuipers, L. and H. Niederreiter (2005). *Uniform distribution of sequences*. Dover Publications.

[53] Kumar, V. and K. J. Waldron (1989). Suboptimal algorithms for force distribution in multifingered grippers. *IEEE Transactions on Robotics and Automation 5*(4), 491–498.

[54] Kwon, W. and B. H. Lee (1996). Optimal force distribution of multiple cooperating robots using nonlinear programming dual method. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, pp. 2408–2413 vol.3.

[55] Lau, D., D. Oetomo, and S. K. Halgamuge (2011). Wrench-Closure workspace generation for cable driven parallel manipulators using a hybrid Analytical-Numerical approach. *Journal of Mechanical Design 133*(7), 071004+.

[56] Li, J.-W., H. Liu, and H.-G. Cai (2003). On computing three-finger force-closure grasps of 2-D and 3-D objects. *IEEE Transactions on Robotics and Automation, 19*(1), 155–161.

[57] Liu, G., J. Xu, X. Wang, and Z. Li (2004). On quality functions for grasp synthesis, fixture planning, and coordinated manipulation. *IEEE Transactions on Automation Science and Engineering 1*(2), 146–162.

[58] Liu, Y.-H. (1998). Computing n-finger force-closure grasps on polygonal objects. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 2734–2739.

[59] Lotti, F., P. Tiezzi, G. Vassura, L. Biagiotti, and C. Melchiorri (2004). UBH 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 4736–4741.

[60] Lovchik, C. S. and M. A. Diftler (1999). The Robonaut hand: a dexterous robot hand for space. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Volume 2, pp. 907–912.

[61] Lozano-Perez, T., J. L. Jones, E. Mazer, and P. A. O'Donnell (1989). Task-level planning of pick-and-place robot motions. *Computer 22*(3), 21–29.

[62] Markenscoff, X., L. Ni, and C. H. Papadimitriou (1990). The geometry of grasping. *The International Journal of Robotics Research 9*(1), 61–74.

[63] Merlet, J.-P. (2004). Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *The International Journal of Robotics Research 23*(3), 221–236.

[64] Miller, A. T. and P. K. Allen (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine 11*(4), 110–122.

[65] Mirtich, B. and J. Canny (2002). Easily computable optimum grasps in 2-D and 3-D. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 739–747.

[66] Mishra, B., J. T. Schwartz, and M. Sharir (1987). On the existence and synthesis of multifinger positive grips. *Algorithmica 2*(1), 541–558.

[67] Morales, A., T. Asfour, P. Azad, S. Knoop, and R. Dillmann (2006). Integrated grasp planning and visual object localization for a humanoid robot with Five-Fingered hands. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5663–5668.

[68] Murray, R. M., Z. Li, and S. S. Sastry (1994). *A Mathematical Introduction to Robotic Manipulation* (1st ed.). CRC Press.

[69] Napier, J. (1993). *Hands* (Revised ed.). Princeton University Press.

[70] Nguyen, V.-D. (1988). Constructing force- closure grasps. *The International Journal of Robotics Research 7*(3), 3–16.

[71] Nocedal, J. and S. J. Wright (2006). *Numerical Optimization* (2nd ed.). Springer Series in Operational Research. Springer.

[72] Ott, C., M. A. Roa, and G. Hirzinger (2011). Posture and balance control for biped robots based on contact force optimization. In *Proceedings of the 2011 IEEE-RAS International Conference on Humanoid Robots*, pp. 26–33.

[73] Park, Y. and G. Starr (1990). Grasp synthesis of polygonal objects. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pp. 1574–1580.

[74] Pérez, A. and J. Rosell (2010). A roadmap to robot motion planning software development. *Computer Applications in Engineering Education 18*(4), 651–660.

[75] Platt, R., A. H. Fagg, and R. A. Grupen (2010). Null-space grasp control: theory and experiments. *IEEE Transactions on Robotics 26*(2), 282–295.

[76] Pollard, N. S. (1990). The grasping problem: Toward Task-Level programming for an articulated hand. Technical Report 1214, MIT Artificial Intelligence Laboratory.

[77] Pollard, N. S. (2004). Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research 23*(6), 595–613.

[78] Ponce, J., S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet (1997). On computing Four-Finger equilibrium and Force-Closure grasps of polyhedral objects. *The International Journal of Robotics Research 16*(1), 11–35.

[79] Porta, J. M., L. Ros, T. Creemers, and F. Thomas (2007). Box approximations of planar linkage configuration spaces. *Journal of Mechanical Design 129*(4), 397–405.

[80] Porta, J. M., L. Ros, and F. Thomas (2009). A linear relaxation technique for the position analysis of multiloop linkages. *IEEE Transactions on Robotics 25*(2), 225–239.

[81] Pratichizzo, D., M. Malvezzi, and A. Bicchi (2010, June). On motion and force controllability of grasping hands with postural synergies. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain.

[82] Prattichizzo, D., M. Malvezzi, M. Gabiccini, and A. Bicchi (2012). On the manipulability ellipsoids of underactuated robotic hands with compliance. *Robotics and Autonomous Systems 60*(3), 337–346.

[83] Prattichizzo, D. and J. C. Trinkle (2008). Grasping. In B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, Chapter 29, pp. 671–700. Springer Berlin Heidelberg.

[84] Pérez, A. and J. Rosell (2009). A roadmap to robot motion planning software development. *Computer Applications in Engineering Education*.

[85] Rao, R. S., A. Asaithambi, and S.K. Agrawal (1998). Inverse kinematic solution of robot manipulators using interval analysis. *Transactions of the ASME Journal of Mechanical Design 120*, 147–150.

[86] Reuleaux, A. F. (1876). *The Kinematics of Machinery: Outlines of a Theory of Machines*. Macmillan (Reprinted by Dover, 1963).

[87] Roa, M. and R. Suarez (2007). Geometrical approach for grasp synthesis on discretized 3D objects. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3283–3288.

[88] Roa, M. and R. Suárez (2009). Computation of independent contact regions for grasping 3-D objects. *IEEE Transactions on Robotics 25*(4), 839–850.

[89] Roa, M. A., R. Suárez, and J. Cornellà (2008). Medidas de calidad para la prensión de objetos. *Revista iberoamericana de Automatica e Informatica Industrial 5*(1), 66–82.

[90] Rodríguez, A., L. Basañez, and E. Celaya (2008). A relational positioning methodology for robot task specification and execution. *IEEE Transactions on Robotics 24*(3), 600–611.

[91] Rosales, C., J. Porta, and L. Ros (2011). Global optimization of robotic grasps. In *Proceedings of Robotics: Science and Systems VII*.

[92] Rosales, C., J. M. Porta, R. Suárez, and L. Ros (2008). Finding all valid hand configurations for a given precision grasp. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pp. 1634–1640.

[93] Rosales, C., L. Ros, J. M. Porta, and R. Suárez (2011). Synthesizing grasp configurations with specified contact regions. *The International Journal of Robotics Research 30*(4), 431–443.

[94] Rosales, C., R. Suárez, M. Gabiccini, and A. Bicchi (2012). On the synthesis of feasible and prehensile robotic grasps. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pp. 550–556.

[95] Rosell, J., X. Sierra, L. Palomo, and R. Suárez (2005). Finding grasping configurations of a dexterous hand and an industrial robot. In *Proceedings of the 2005 IEEE International Conference onRobotics and Automation*, pp. 1178–1183.

[96] Rosell, J., R. Suárez, A. Pérez, and C. Rosales (2011). Including virtual constraints in motion planning for anthropomorphic hands. In *Proceedings of the 2011 IEEE International Symposium on Assembly and Manufacturing*.

[97] Rosell, J., R. Suárez, C. Rosales, J. A. García, and A. Pérez (2009). Motion planning for high DOF anthropomorphic hands. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 4025–4030.

[98] Rosell, J., R. Suárez, C. Rosales, J. A. García, and A. Pérez (2009). Motion planning for high DOF anthropomorphic hands. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 4025–4030.

[99] Rosell, J., R. Suárez, C. Rosales, and A. Pérez (2009). A practical motion-planning approach for the Schunk Anthropomorphic Hand. In *Technical Report IOC-UPC, submitted to an international conference*.

[100] Rosell, J., R. Suárez, C. Rosales, and A. Pérez (2011). Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. *Autonomous Robots 31*(1), 87–102.

[101] Safonova, A., J. K. Hodgins, and N. S. Pollard (2004). Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions Graph. 23*(3), 514–521.

[102] Salisbury, K. and B. Roth (1983). Kinematics and force analysis of articulated mechanical hands. *Journal of Mechanisms, Transmissions and Actuation in Design 105*, 35–41.

[103] Salisbury, K. J. and J. J. Craig (1982). Articulated hands: Force control and kinematic issues. *The International Journal of Robotics Research 1*(1), 4–17.

[104] Santello, M., M. Flanders, and J. F. Soechting (1998). Postural hand synergies for tool use. *Journal of Neuroscience 18*(23), 10105–10115.

[105] Schulz, S., C. Pylatiuk, and G. Bretthauer (2001). A new ultralight anthropomorphic hand. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 2437–2441.

[106] Schunk GmbH & Co. KG (2012). Schunk anthropomorphic hand. http://www.schunk.com/.

[107] Shadow Robot Company (2003). Design of a dextrous hand for advanced CLAWAR applications. In *Proceedings of the Climbing and Walking Robots and the Supporting Technologies for Mobile Machines*, pp. 691–698.

[108] Shimoga, K. B. (1996). Robot grasp synthesis algorithms: A survey. *The International Journal of Robotic Research 15*(3), 230–266.

[109] Sommese, A. J. and C. W. Wampler (2005). *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific.

[110] Somov, P. (1900). Üeber gebiete von schraubengeschwindigkeiten eines starren korpers bei verschiedener zahl von stutzflachen. *Zeitricht Mathematik Physik 45*, 245–306.

[111] Song, S. K., J. B. Park, and Y. H. Choi (2012). Dual-Fingered stable grasping control for an optimal force angle. *IEEE Transactions on Robotics 28*(1), 256–262.

[112] Suárez, R. and P. Grosch (2005, April). Mechanical hand MA-I as experimental system for grasping and manipulation. In *VideoProceedings of the 2005 IEEE International Conference on Robotics and Automation*.

[113] Suárez, R., M. Roa, and J. Cornella (2006). Grasp quality measures. Technical Report IOC-DT-P-2006-10, Institut d'Organizació i Control de Sistemes Industrial, UPC.

[114] Suárez, R., J. Rosell, A. Perez, and C. Rosales (2009). Efficient search of obstacle-free paths for anthropomorphic hands. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1773–1778.

[115] Sun, S.-C., C. Rosales, and R. Suárez (2010). Study of coordinated motions of the human hand for robotic applications. In *Proceedings of the 2010 IEEE International Conference on Information and Automation*, pp. 776–781.

[116] Trinkle, J. C. (1992). A quantitative test for form closure grasps. In *Proceedings of the 1992 lEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1670–1677.

[117] Tsoli, A. and O. C. Jenkins (2007). 2D subspaces for user-driven robot grasping. In *Proceedings of the RSS 2007 Workshop on Robot Manipulation: Sensing and Adapting to the Real World*.

[118] Vahrenkamp, N., T. Asfour, and R. Dillmann (2012). Simultaneous grasp and motion planning: Humanoid robot ARMAR-III. *IEEE Robotics & Automation Magazine 19*(2), 43–57.

[119] Valero-Cuevas, F. J., M. E. Johanson, and J. D. Towles (2003). Towards a realistic biomechanical model of the thumb: the choice of kinematic description may be more critical than the solution method or the variability/uncertainty of musculoskeletal parameters. *Journal Biomechanics 36*(7), 1019–1030.

[120] Waldron, K. (1986). Force and motion management in legged locomotion. *IEEE Journal of Robotics and Automation 2*(4), 214–220.

[121] Wampler, C. and A. Morgan (1991). Solving the 6R inverse position problem using a generic-case solution methodology. *Mechanism and Machine Theory 26*(1), 91–106.

[122] Woelfl, K. and F. Pfeiffer (1994). Grasp strategies for a dextrous robotic hand. In *Proceedings of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 366–373.

[123] Yamaguchi, Y. (1997). Bezier normal vector surface and its applications. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications*, pp. 26–35.

[124] Yoshikawa, T. and K. Nagai (1991). Manipulating and grasping forces in manipulation by multifingered robot hands. *IEEE Transactions on Robotics and Automation 7*(1), 67–77.

[125] Zuo, B.-R. and W.-H. Qian (2000). A general dynamic force distribution algorithm for multifingered grasping. *IEEE Transactions on Systems, Man, and Cybernetics 30*(1), 185–192.