UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Object manipulation based on tactile information

Andrés Felipe Montaño Sarria

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Doctoral programme

AUTOMATIC CONTROL, ROBOTICS AND COMPUTER VISION

Ph.D. Thesis

Object Manipulation Based on Tactile Information

ANDRÉS FELIPE MONTAÑO SARRIA

Thesis advisor

Raúl Suárez Feijóo

January 2021

This thesis is dedicated to Annamária

Acknowledgements

The work contained in this thesis is the result of several years of research and would not have been possible without the help of many people, people who end up being part of your life, as teachers, mentors, friends, becoming family.

I would like to thank my thesis advisor Raúl Suárez for his support, dedication, and patience in the development of this thesis. I thank him for all the opportunities he offered me, his friendship and advice, in one way or another sharing with him ended up forming me not only academically but also personally.

I also would like to thank all the members of the Robotics Division of the Institute of Industrial and Control Engineering (IOC). Especially to Leo for his friendship, support, and affection, for being the guide, the voice of conscience, not letting me stray from the good path of free software. For the talks during meals, for everything I have learned at his side. I would also like to remember all the colleagues who were at some point during the journey of this path: Carlos Aldana, Carlos Rodríguez, Abiud, Nestor, Josep, Ali, Muhayy Ud Din, Diab, Isiah, Henry, Diana, Niliana, Marcos, José, thanks for everything guys. And of course, I would like to thank professors Luis Basañez and Jan Rosell, always attentive and cordial, both personally and academically.

On the other hand, I would like to thank my friends and family. To my moms: Martha, Eliza, and Eva; to my parents: Walther, Humberto, and Jorge; and to my sisters: Carmén, Nathalie, María, and Gabriela, for always believing in me and wishing me the best.

Finally, to my beloved wife Anna, always Anna, for listening to my concerns, for giving me the strength to finish this work and celebrate my achievements as if they were hers. Without her support, company, and all her love I would not have been able to undertake this adventure, this life experience that we went through together without knowing how far it would take us.

Andrés

This work has been partially supported by the Spanish Government through the projects: DPI2010-15446 (MUMA): Multi-hand Systems for Complex Robotized Manipulation Tasks; DPI2013-40882-P (DEMCO): Dexterous, mobile and cooperative grasping and manipulation; and DPI2016-80077-R (AUDECO): Dexterous Robots as Coworkers with Human Operators, and also through the scholarship BES-2014-068233.

Agradecimientos

El trabajo contenido en esta tesis es el resultado de varios años de investigación y no hubiese sido posible sin la ayuda de muchas personas, personas que terminan siendo parte de tu vida, convirtiéndose en maestros, mentores, amigos, convirtiéndose en familia.

Me gustaría agradecer a mi director de tesis Raúl Suárez por su apoyo, dedicación y paciencia en el desarrollo de esta tesis. Le agradezco por todas las oportunidades que me ofreció, su amistad y consejo, de un modo u otro compartir con él terminó formándome no solo en lo académico si no también en lo personal.

También me gustaría agradecer a todos los miembros de la División de Robótica del Instituto de Organización y Control de Sistemas Industriales (IOC). En especial a Leo por su amistad, su apoyo y cariño, por ser el guía, la voz de la conciencia, no dejándome apartar del buen camino del software libre. Por las charlas durante las comidas, por todo lo que he aprendido a su lado. También me gustaría recordar a todos los compañeros que estuvieron en algún momento durante el recorrido de este camino: Carlos Aldana, Carlos Rodríguez, Abiud, Nestor, Josep, Ali, Muhayy Ud Din, Isiah, Diab, Henry, Diana, Niliana, Marcos, José, gracias por todo muchahos. Y por supuesto agradecer a los profesores Luis Basañez y Jan Rosell, siempre atentos y cordiales, tanto en lo personal como lo académico.

Por otro lado, me gustaría agradecer a mis amigos y familia. A mis mamás: Martha, Eliza y Eva; a mis papás: Walther, Humberto y Jorge; y a mis hermanas: Carmén, Nathalie, María y Gabriela, por siempre creer en mi y desearme lo mejor.

Y, finalmente, a mi amada esposa Anna, siempre Anna, por escuchar mis preocupaciones, por darme fuerzas para terminar este trabajo y celebrar mis logros como si fuesen suyos. Sin su apoyo, compañía y todo su amor no hubiese podido emprender esta aventura, esta experiencia de vida que recorrimos juntos sin saber hasta donde nos llevaría.

Pipe

Este trabajo ha recibido soporte por parte del Gobierno de España a través de los proyectos: DPI2010-15446 (MUMA): Multi-hand Systems for Complex Robotized Manipulation Tasks; DPI2013-40882-P (DEMCO): Dexterous, mobile and cooperative grasping and manipulation; y DPI2016-80077-R (AUDECO): Dexterous Robots as Coworkers with Human Operators, así como a través de la beca de estudios BES-2014-068233.

OBJECT MANIPULATION BASED ON TACTILE INFORMATION

Andrés Felipe Montaño Sarria

Abstract

In-hand dexterous manipulation of an object is the ability to change the configuration (position and/or orientation) of the object held in the hand. This is an ability that has allowed humans to use tools and interact with the environment effectively. In the past decades, robotics researchers have worked to provide dexterous manipulation skills to the robots by designing robotic hands that mimic the human hand and by developing applications that allow performing autonomous manipulation or teleoperation in harsh environments. Despite the progress made, managing the uncertainties that exist in the real world is one of the problems that still need to be worked on. Many existing manipulation methods for controlling robotic hands require a priori information about the object and high-fidelity sensors that are typically limited only to laboratory setups.

The main objective of this thesis is to develop strategies for the dexterous manipulation of unknown objects, using the tactile information generated during the grasp of the object and the manipulation process itself. In manipulation applications based on tactile information, the robotic hand has access only to tactile and proprioceptive data, in addition, no a priori information is known about the manipulated object. This reflects real-world applications, where there is uncertainty in the models of the objects that are commonly manipulated in daily activities, as well as in the sensorial measurements.

In this thesis, novel manipulation strategies based on heuristic and gradient optimization methods are proposed. Three quality indexes are selected to measure the goodness of the grasp during the manipulation, related to the configuration of the hand, the quality of the grasp, and the configuration of the object. Starting from a given initial grasp, the manipulation strategies are able to improve one quality index or a combination of them. The manipulation strategies are validated with real experimentation using robotic hands equipped with tactile sensors, allowing the execution of practical applications, such as object recognition, force optimization, and telemanipulation.

Notation and Acronyms

Notation

$\mathbb{R}^n$	n-dimensional Euclidean space
f_i	i-th finger
n_i	Number of DOF of the i-th finger
l_{ij}	Length of the j-th link of the i-th finger
q_{i_j}	Joint value of the j-th joint of the i-th finger
q_{i_v}	Joint value of the virtual joint of the i-th finger
$\mathbf{q}_i$	Finger configuration
$\mathbf{Q}$	Hand configuration
$\mathbf{v}$	Vector
$\hat{\mathbf{u}}$	Unit vector
v_x, v_y, v_z	Components of the vector $\mathbf{v}$
$\mathcal{W}$	Absolute reference frame
$\mathcal{O}$	Object reference frame
Σ_{i_j}	Reference frame located at j-th joint of the i-th finger
$O_{\Sigma_{i_j}}$	Origin of the reference frame Σ_{i_j}
Σ_{i_s}	Reference frame located at the sensor pad
$\mathbf{P}$	General point
$\mathbf{C}_i$	Contact point made by the i-th finger
γ	Object orientation
F_i	Force applied by a finger at the i-th contact point
$\boldsymbol{\tau}$	Torque
μ	Friction coefficient
s_α, c_α , t_α	$\sin(\alpha)$, $\cos(\alpha)$, $\tan(\alpha)$
Π_i	Working plane associated to finger i
$\lvert \cdot \rvert$	Euclidean norm
$\mathcal{F}(\cdot)$	Forward kinematics
$\mathcal{I}(\cdot)$	Inverse kinematics
J	Jacobian matrix

Acronyms

API	Application Programming Interface
CAD	Computer-Aided Design
CAN	Controller Area Network
DOF	Degrees Of Freedom
FC	Force Closure
FK	Forward Kinematics
GUI	Graphical User Interface
IK	Inverse Kinematics
IOC	Institute of Industrial and Control Engineering
PID	Proportional-Integral-Derivative
ROS	Robot Operating System
URDF	Unified Robot Description Format
XML	Extensible Markup Language

Latin abbreviations

i.e	-id est-	that is
e.g.	-exempli gratia-	for instance

Contents

List of Figures

List of Tables

"He had them as spellbound as a room full of Ewoks listening to C-3PO."

Cory Doctorow (1971 -)

1

Introduction

The word *dexterous* comes from the Proto-Indo-European (PIE) root *desk-* which means "right, on the right hand, opposite left", formed in English from Latin *dexter* that means "skillful" plus *-ous* that is a word-forming element making adjectives from nouns, meaning "having, full of, having to do with, doing, inclined to". The word *manipulation* is a combination of the PIE roots *man-* which means "hand" and *pele-* which means "to fill", this combination give rise to the Latin term *manipulus* which means *"a handful"*. Later in 1826, it was given the sense of "skillful handling of objects". The work presented in this thesis deals with the problem of *dexterous manipulation* of unknown objects using robotic hands and tactile sensors. This chapter presents the treated problem, defines the thesis objectives, and presents a brief outline of the contents.

1.1 Motivation and objectives

Dexterous manipulation has taken diverse definitions in robotics literature, for instance, Shimoga (1996) defines dexterity as "the ability of a grasp to achieve one or more useful secondary objectives while satisfying the kinematic relationship (between joint and Cartesian spaces) as the primary objective"; Han and Trinkle (1998) define it as "the manipulation [action] that achieves the goal configuration for the object and the [grasp] contacts"; Dafle et al. (2014) refer to dexterity as "the manipulation of an object in the hand, with the hand"; more recently Prattichizzo et al. (2020) define the dexterous manipulation as "the skillful execution of object reorienting and repositioning maneuvers, especially when performed within the grasp of an articulated mechanical hand". There are other definitions but all of them refer explicitly or implicitly to the manipulation of the object by properly locating/changing the positions of the grasp contact points, that is, by properly managing the finger configurations, which in turn give rise to the expression "in-hand manipulation" to explicitly refer to the object manipulation using only finger movements (Ho et al. 2012; Funabashi et al. 2015; Liarokapis and Dollar 2017; Shi et al. 2017).

To properly formulate the dexterous manipulation problem, an object-centered point of view must be adopted (Okamura et al. 2000). The aim is to move the grasped object (by a robotic hand) from a pose A to a pose B with respect to the palm. In order to achieve this, suitable finger movements are computed considering some assumptions: the object is a rigid body in contact with a rigid link of the robot hand, the object is grasped by the distal phalanges of the fingers, and precise models of the hand and the object are known (Murray et al. 1994). In order to deal with the manipulation problem, one option is to model it using the manipulation space, which is the n-dimensional space defined by the values of all the finger joints. In this space, a point represents a configuration of the hand and a curve represents finger movements (i.e. a sequence of hand configurations). Thus, following a proper curve in this space it is possible to achieve a successful object manipulation. However, when the model of the object is not known, computing a manipulation curve in advance may not be possible, i.e. the manipulation constraints cannot be computed a priori and therefore, planning a sequence of finger movements is not possible. Under these conditions, manipulation must be handled as a reactive procedure that determines on-line the proper hand movements. This is the approach followed in this thesis.

Dexterous manipulation is a relevant topic in robotics research, even more when a precise model of the object is not known. Usually, in a realistic scenario, the geometric model of the manipulated object is only partially known or even unknown. Several approaches to the development of new manipulation strategies are inspired in the human capability to manipulate quite different objects with the hands, and this has led to the development of a wide variety of anthropomorphic robotic hands (Bicchi and Kumar 2000). The inclusion of tactile sensors into robotic hands improves their manipulation capabilities, because these sensors provide contact information during the manipulation process allowing the execution of more complex tasks with better results, both in industrial and in everyday environments (Kappassov et al. 2015; Nadon et al. 2018; Indri et al. 2019). A tactile sensor in robotics, like human tactile receptors, is able to detect the contact and measure the applied forces. It can be used to obtain information about the shape of the object, its pose, the location of the contact points and the contact force applied to the object by the robotic fingers. Slippage detection and estimation of the friction coefficient between the fingers and the object are also some of the intended common applications of tactile sensors (Sadigh and Ahmadi 2009). Summarizing, tactile sensors help to recognize the manipulated object and to reduce the uncertainty in their geometric model.

The object manipulation process usually pursues three goals (Montaño and Suárez 2015), either independently or in a combined way:

- From the hand point of view, the optimization of the hand configuration, i.e., searching for a particular hand configuration satisfying some specific constraints that can be arbitrarily defined.
- From the grasp point of view (relation hand-object), the optimization of the grasp quality, i.e., searching for a grasp that can resist external force perturbations on the object.
- From the object point of view, the optimization of the object configuration, i.e., searching for an appropriate object position and orientation that satisfy the requirements of a given task.

The general dexterous manipulation problem leads to a variety of particular problems that still require a satisfactory solution. The main objective of this work is to develop methods for the dexterous in-hand manipulation of unknown objects using robotic hands with tactile sensors. The methods developed herein apply to the real-world scenario where the robotic hand is commanded using only proprioceptive and tactile information with no knowledge of the object being grasped. It must be remarked that the expression “unknown object” means that the model of the object is not used at all in the manipulation procedure. The objectives of this thesis are:

- Development of manipulation strategies to optimize: the hand configuration, the grasp quality, and the object configuration. Strategies are validated using robotic hands with tactile sensors to obtain information about the contact with the manipulated objects.
- Selection of grasp quality indexes considering their applicability to the dexterous manipulation problem of unknown objects.
- Generalization of the manipulation strategies in order to apply them to hands with different kinematic structures and number of fingers.
- Application of the developed manipulation strategies to real dexterous manipulation problems.

1.2 Thesis layout

The structure of the remainder of this thesis is presented below, including a brief summary of the content introduced in each chapter.

Chapter 2 reviews the existing literature that addresses dexterous manipulation for robotic hands, besides, it introduces the foundations of this thesis, presenting the basic concepts and definitions involved in the manipulation problem.

Chapter 3 presents the principal features of the robotic hands and tactile sensors used in the experimental validation of the proposed manipulation strategies. For each of the hardware components, the libraries and software modules that were designed for the validation of the proposed manipulation strategies are presented. The framework used for the hardware-software integration, which allows the implementation of the manipulation strategies, is also introduced.

Chapter 4 presents the proposed dexterous manipulation strategies. After an introduction section, the chapter includes another four sections. The first section presents the quality indexes to be optimized during the manipulation process. The second section presents manipulation strategies based on heuristic methods. The third section introduces manipulation strategies based on gradient methods. The fourth section presents an approach to manipulate unknown object using three fingers.

Chapter 5 presents applications that use the information generated during the manipulation to achieve an extra goal, such as, the object identification, the force optimization, and the object telemanipulation with and without tactile feedback.

Chapter 6 summarizes the contributions of the thesis, lists the derived publications and presents directions for future research work.

“Whether we are based on carbon or on silicon makes no fundamental difference; we should each be treated with appropriate respect.”

Arthur C. Clarke (1917 - 2008)

2

Related work

This chapter presents an overview of previous work related to the dexterous manipulation problem dealt with in this Thesis. It also provides a general review of robotic hands, tactile sensors, and grasp quality measures, all useful elements for the Thesis development.

2.1 Robotic hands

The two most distinguished features of humans among animals are the hands and the mind. Since the time of the ancient philosophers it has been debated whether the humans had dexterous hands and then they became intelligent, or the other way around (Aristotle and Ogle 1908). Later anthropological studies showed that the dexterity of the human hand has been a major factor in the changes of the *homo sapiens* brain size and organization (Sherwood et al. 2008).

Human hand dexterity is an admirable feature that has been a source of inspiration in robotics, in hardware design as well as in behavior emulation. Several approaches on manipulation are inspired in the human capability to manipulate quite different objects leading to the development of a wide variety of robotic hands (Bicchi and Kumar 2000). The mechanical design of robotic hands can be categorized according to their number of degrees of freedom (DOF) and actuators as: fully actuated mechanisms, when the number of actuator is equal to the number of DOF, redundantly actuated mechanisms, the number of actuators is larger than DOF, and underactuated mechanisms, when the number of DOF is larger than the number of actuators (Birglen et al. 2008). The two categories followed by the designers are the fully actuated and underactuated mechanisms, in fact, redundantly actuated mechanisms are neither useful nor desired, but they are present in some designs that use tendons. In these designs two actuators (tendons) are required to extend and flex a single articulation, this mechanism has been replaced by an actuator working jointly with a passive mechanism used to recover the position of the articulation by moving it in the opposite direction to the active actuator.

One of the earliest designs of robotic hands was done by Mason and Salisbury (1985), who established that nine is the minimum number of DOF to achieve dexterity using a hand with rigid hard fingers. This is because, in order to completely restrain an object at least three hard-finger contacts are necessary, and without rolling nor sliding movements, the fingers must keep the contact locations and follow the path generated for the corresponding contact point on the object, while the object moves in a three-dimensional space. Therefore, three DOF per finger are strictly necessary. Following this principle, the Salisbury's hand (Pellerin 1991) was designed with nine joints divided into three fingers. Other contemporary models that also followed the same design were built at the University of Karlsruhe (Wöhlke 1990), the Technical University of Darmstadt (Weigl and Seitz 1994), and the Delft University (Jongkind 1993). Hands that follow this design criterion and, in general, kinematically optimized hands (Tischler et al. 1998), are not anthropomorphic.

Other hand designs add redundant DOF in order to achieve more flexibility of use, for example, the Okada hand has two four-joint fingers and one three-joint thumb (Okada 1979); The hand of the Technical University of München (Menzel et al. 1994) has three fingers, each one with four joints, however the most distal joints are mechanically coupled, leaving the hand with nine DOF. Other hand designs have more than three fingers, looking for two goals: four- and five-fingered hands are more similar to the human hand, and with more fingers involved in the grasp it is possible to apply richer manipulation strategies. The list of anthropomorphic robotic hands is long but some representative examples are: the Belgrade-USC Hand (Bekey et al. 1990), the University of Bologna Robotic Hand (Bonivento et al. 1991), the DLR-HIT Hand II (Butterfass et al. 2004), the MA-I Hand (Grosch and Suárez 2004), the Robonaut 2 Hand (Bridgwater et al. 2012), the Pisa/IIT SoftHand (Catalano et al. 2014), the Shadow Hand (Shadow Robot Company 2015), the Allegro Hand (Wonik Robotics 2018), the HRI hand (Park and Kim 2020), among several others. Figure 2.1 shows examples of robotic hands, some of them with three fingers and others with four or five fingers having a more anthropomorphic design. Hands like the Barrett, SDH, and Robonaut also include tactile sensors to increase the capabilities and manipulation options of the hand.

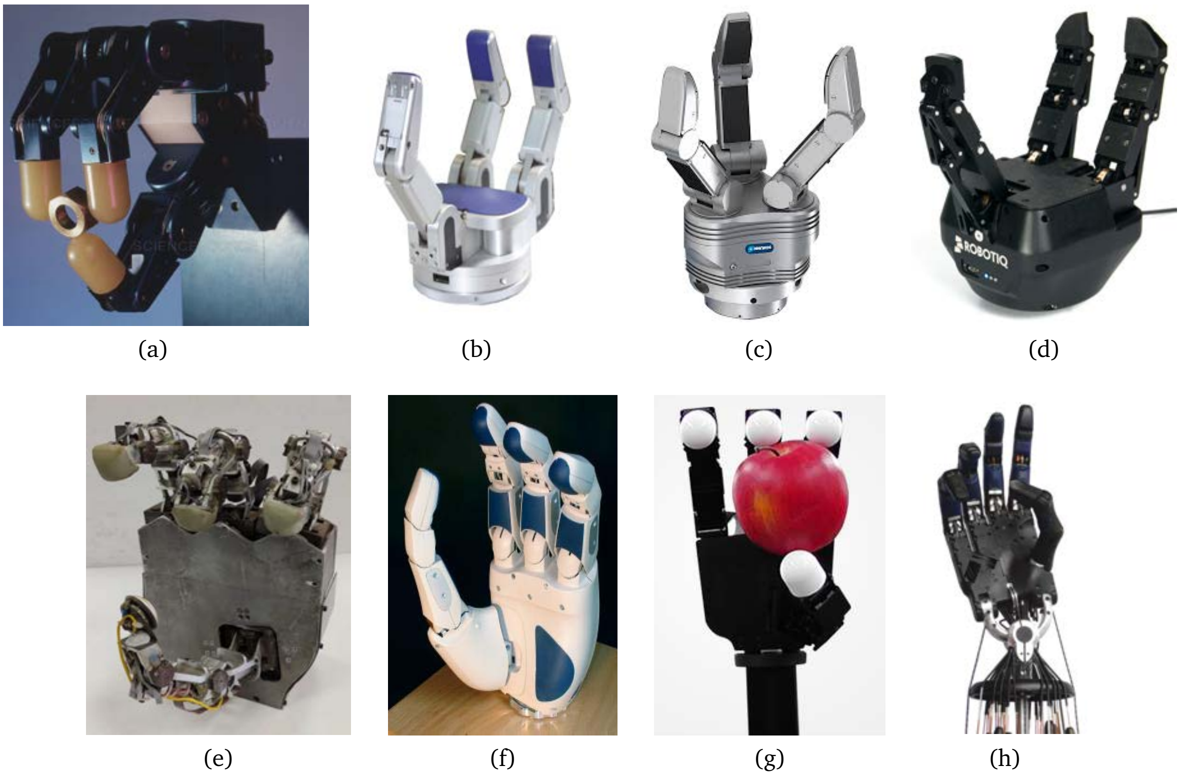

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2.1: Examples of robotic hands. (a) Salisbury's hand. (b) Barrett hand. (c) Schunk SDH2 hand. (d) Robotiq three-finger gripper. (e) MA-I Hand. (f) DLR-HIT Hand. (g) Allegro Hand. (h) Shadow Hand.

2.2 Tactile sensing

The design of robotic hands is a complex process that involves diverse knowledge areas as mechanics, digital control, and digital signal processing (Righetti et al. 2014). In particular, tactile sensing systems based on different techniques have been developed during the last decades in order to equip robots with tactile feedback (Natale and Cannata 2017; García et al. 2009; Zou et al. 2017). In general, the touch sensors improve the manipulation capabilities of the robotic hands while providing contact information during the manipulation process, allowing to perform complete tasks both in industrial environments and in daily life. Two recent reviews on the state of the art regarding tactile sensors for dexterous hands were presented by Kappassov et al. (2015) and Nadon et al. (2018). A complete overview of different types of robot applications and the types of tactile information that they employ is presented by Li et al. (2020).

A tactile sensor in robotics, like human tactile receptors, is able to detect the contact region between the sensor (generally located at the fingertip) and the manipulated object, and also to measure the exerted force on the object (Dahiya and Valle 2013). The information generated by the tactile sensors can be used to stabilize and make safer the grasp (Sadigh and Ahmadi 2009; Laaksonen et al. 2012), to classify the manipulated object according to its material stiffness (Delgado et al. 2017a), to identify the shape of the grasped object (Pozo-Espín 2012; Montaño and Suárez 2013b; Chebotar et al. 2014), its pose (Li et al. 2013), the location of the contact points (Liu et al. 2015), the contact force applied by the robotic fingers (Bimbo et al. 2016), and the friction coefficient between the object and the fingertip (Chen et al. 2018). Besides, they provide robustness in front of variations in object properties (Takahashi et al. 2008; Laaksonen et al. 2012), perturbations (Zhang and Chen 2000), and sensing errors (Hsiao et al. 2010; Jentoft et al. 2014). Slippage detection, including grasp instabilities, and estimation of the friction coefficient between the fingers and the object are also some of the intended common applications of tactile sensors (Sadigh and Ahmadi 2009; Li et al. 2014; Shirafuji and Hosoda 2014; Su et al. 2015; Agriomallos et al. 2018).

In object manipulation, tactile sensors can aid to reduce the uncertainty, allowing, for instance, an improvement of the grasp stability and safety (Bekiroglu et al. 2011; Dang et al. 2011; Boutselis et al. 2014). When the object model is partially or completely unknown tactile sensors are used to reduce uncertainty and adjust the object geometric model, which can be used to recognize the object (using a data base) (Kaboli et al. 2015; Delgado et al. 2017b; Agulló 2017; Funabashi et al. 2018). The reconstruction of the shape of an unknown object can be performed using tactile sensors without requiring object immobilization, instead, the robot manipulates the object without grasping it. The robot can infer the shape, mass and the center of mass of the object based on the motion of the contact points measured by tactile sensors (Moll and Erdmann 2001). Another use of the tactile information is to generate a contact point cloud and then, use statistical point cloud features to provide a robust descriptions of the grasped object (Gorges et al. 2011). Besides, a compact 3-D representation of unknown objects can be obtained using a probabilistic spatial approach based on Kalman filters to build a probabilistic model of the contact point cloud (Meier et al. 2011). The tactile information can be treated as a sequence of images in order to extract information about the contact conditions between an object and the hand (Ho et al. 2012), and therefore, image processing techniques can be used in order to process the tactile sensor information. In some object recognition approaches the tactile information is treated as low-resolution images. Then, different techniques are applied in order to perform the object identification, for instance, the bag-of-words approach which, by unsupervised clustering of training data, learns a vocabulary from tactile observations that is used to generate a histogram codebook (Schneider et al. 2009). Using a fusion sensor approach, a multi-sensory object representation is built by fusion of tactile and kinesthetic features. The recognition approach is based on extracting key features of tactile and kinesthetic data from multiple palpations using a clustering algorithm (Navarro et al. 2012). On the other hand, machine learning techniques has been also applied to recognize the manipulated object (Velasco et al. 2020), or to improve the object manipulation using tactile information, specifically, in order to estimate the grasp stability (Bekiroglu et al. 2011; Dang et al. 2011).

Tactile sensors are also used in manipulation strategies based on the tactile feedback without caring about the object model, that is, the manipulation is performed even when the object model is completely unknown (Montaño and Suárez 2015; Montaño and Suárez 2018c). Tactile feedback is critical in applications that consider interactions of the manipulated object with the environment (Kappassov et al. 2020), for example, in object relocation within the hand, where the initial grasp is modified by pushing the object against other fixed objects in the workspace (Chavan-Dafle et al. 2020). The tactile information obtained during the manipulation can also be used jointly with visual feedback to improve the control performance (Jara et al. 2014). The grasp stability of deformable objects can be improved by adjusting the forces applied by the fingers when there are changes in the center of mass of the grasped object (Kaboli et al. 2016). Contact forces were also used for assembly planning and execution (Suárez et al. 1995). In some cases, the design of the gripper was particularly influenced by a tactile application, designing specific fingers (Bicchi et al. 1999; Ward-Cherrier et al. 2017; Palli and Pirozzi 2019), for instance, to roll on an unknown object in order to do dexterous manipulation and identify the object surface or to manipulate wires for manufacturing applications.

2.3 Grasp quality measures

A grasp quality measure is an index that quantifies the goodness of a grasp. There are many quality indexes to evaluate the grasp quality, which are mostly related with the position of the contact points on the object or with the hand configuration (Roa and Suárez 2014; León et al. 2014; Roa et al. 2008). Other quality indexes combine different measures from the two previous groups to obtain global quality measures.

The quality measures related with the position of the contact points are grouped into measures based on algebraic properties of the grasp matrix (Li and Sastry 1988; Byoung-Ho et al. 2001), measures based on geometric relations (Park and Starr 1992; Mirtich and Canny 1994), and measures considering limitations on the finger forces (Ferrari and Canny 1992; Chinellato et al. 2003; Chinellato et al. 2005). However, all these measures require some knowledge of the object model, mainly the center of mass and its shape.

On the other hand, the quality measures associated with hand configuration relate features as the distance to singular configurations of the hand (Klein and Blaho 1987), the volume of the manipulability ellipsoid (Yoshikawa 1985), the positions of the finger joints (Salisbury and Craig 1982), or the task compatibility (Borst et al. 2004; Haschke et al. 2005). In this case, the quality measures do not depend on the features of the object. These quality indexes have been considered in order to improve the design of robotic hands (Rubert et al. 2014).

In this thesis, the manipulation is carried out without having a priori information about the object, so the quality indexes belonging to the group of measures related to the position of the contact points on the object are not appropriate. Then, the used quality indexes are related to the configuration of the hand, the configuration of the grasp without considering the shape of the object, and the task. These quality measures are introduced in Chapter 4, Section 4.2, jointly with the proposed manipulation strategies.

2.4 Contact models

Contact models can be considered as kinematic pairs. The type of pair is determined by the contact surfaces of the finger and the object. The contact can occur at points, along lines, or on planes. The possible combinations of these three elements generate nine possible types of pairings. Besides, these contacts can occur with or with friction (Salisbury and Roth 1983). Among the variety of contact models, the more common ones used in grasping are the *frictionless point contact*, the *frictional point contact* and the *soft-finger contact*, shown in Figure 2.2.

As the name implies, a *frictionless point contact* considers that the contact takes place only at a point, $\mathbf{C}_i$, and that there is no friction between the fingertip and the object. In this case, the finger can apply a force, $\mathbf{F}_i$, only in the direction normal, $\hat{\mathbf{n}}_i$, to the object surface, as shown

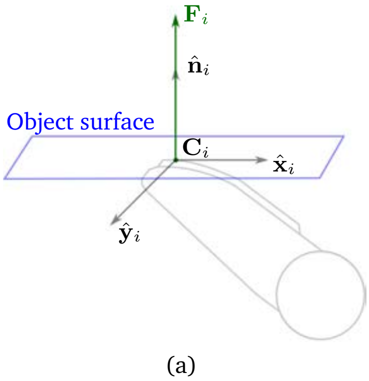


Figure 2.2: Contact models commonly used in grasping and the forces that can be applied on the object: (a) Frictionless point contact. (b) Frictional point contact. (c) Soft-finger contact.

in Figure 2.2a. This model does not represent the real contact situation that appears in real grasps (Cutkosky 1989).

In a *frictional point contact*, the contact between the fingertip and the object occurs at a frictional point, $\mathbf{C}_i$, so the exerted force, $\mathbf{F}_i$, have a component normal to the contact surface and also can have a component tangential to the object surface. Several models have been proposed to describe the frictional behavior (Howe et al. 1988). The most common is Coulomb's friction model, which is based on the idea that friction opposes to the motion and that its magnitude is independent of the velocity and contact area. This is an empirical model that states that slippage is avoided when $F_t \leq \mu F_n$, where, F_n is the magnitude of the normal component, F_t is the magnitude of the tangential component and μ is the friction coefficient, which depends on the materials that are in contact. The possible friction forces are geometrically constrained to be inside a friction cone centered at the contact point, as shown in Figure 2.2b.

In a *soft-finger contact*, the contact between the fingertip and the object occurs on a contact region, and there is friction between the parts. This type of contact allows the application of the same forces described by the frictional contact model, plus a torque around the direction normal to the contact region, as shown in Figure 2.2c, thus, the model is valid only for 3D objects (Kao and Cutkosky 1992; Ciocarlie et al. 2005). Despite the soft-finger contact model is more accurate than the other contact models, the *frictional point contact* model is the most used in the field of robotic grasping because it presents a lower complexity and time-consuming restrictions in the simulation of the contact behavior (León et al. 2014) and the application results are good enough.

One of the most desired properties of multi-finger grasps is the resistance to external disturbances, ensuring the object immobilization. A grasp that guaranteed this property satisfies one out of two conditions: form closure or force closure. A form closure grasp is such that the location of the contact points on the object ensures a total kinematic constraint of the object. A force closure (FC) grasp is such that the forces applied by the fingers on the object can resist any external wrench applied on the object up to a certain magnitude (Bicchi 1995). Both force closure and form closure allow the object immobilization, but form closure is a purely geometrical condition while force closure relies on the applied forces to guarantee the immobility of the object. A benefit of including friction in the analysis is a reduction in the number of contact points needed for the closure. For instance, in the general case, a three-dimensional object with six DOF requires seven contacts for form closure, but only three (non-collinear) contacts are needed for force closure, if they are frictional contact points, and two contacts, if soft-finger contacts are used (Prattichizzo and Trinkle 2008).

2.5 Dexterous manipulation

Dexterous manipulation could be decomposed into four basic manipulation modes, illustrated in Figure 2.3: *Coordinated manipulation*, *rolling motion*, *sliding motion* and *finger relocation*. These basic manipulation modes can be combined to get more complex manipulation actions (Li et al. 1989).

Coordinated manipulation refers to the coordinated control of the fingers to move an object from an initial configuration to a desired one, without changing the position of the fingertips on the object surface, as shown in Figure 2.3a. The movements of the fingers must guarantee that the contacts are fixed during the manipulation process, without allowing rolling and sliding (Li, Canny, and Sastry 1989). Li, Hsu, and Sastry (1989) introduced a torque control algorithm to perform in-hand relocation of an object using coordinated manipulation while guaranteeing the grasp stability. Several works have been presented related to the coordinated manipulation control, some of them integrating exteroceptive feedback, as for instance, tactile feedback to control the contact forces (Liu et al. 2004; Shaw-Cortez et al. 2018), or visual feedback for object tracking (Andrychowicz et al. 2020). Another approach in based on a bio-inspired controller that guarantees the grasp stability and realizes dexterous manipulation of tools to reach a desired grasp having minimal sensor information, but a good knowledge of the kinematic model of the hand (Garate et al. 2018).

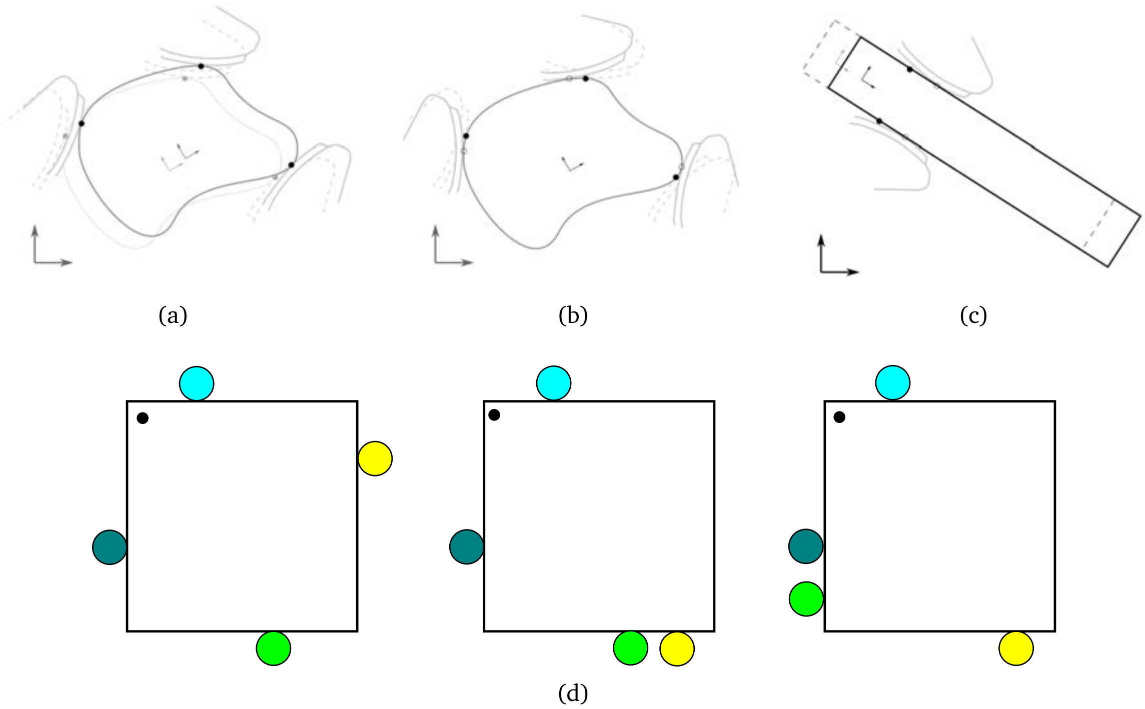


Figure 2.3: Example of manipulation modes. (a) Example of *coordinated manipulation*. The fingers change the object configuration keeping the contact points (in black). (b) Example of *rolling motion*. The fingers roll on the object surface from the initial contact point locations (in black) generating new contact point locations (in white). (c) Example of *sliding motion*. The fingers relax the contact forces allowing the object to slide due to the force of gravity. (d) Snapshots of an example of *finger gaiting*, where a four-finger hand relocates the finger positions on the object surface.

As the name implies, in the manipulation using *rolling motion*, the object is manipulated by explicitly exploiting rolling movements of the fingers on the object, as shown in Figure 2.3b. Kerr and Roth (1986) introduced the kinematics of rolling contacts for grasps using multi-fingered hands. The analysis of manipulation in the presence of rolling was introduced by Cai and Roth (1987) and Montana (1988). An analysis of the motion controllability of objects manipulated using rolling movements was presented by Li and Canny (1990). The geometry of the surfaces of the fingertip and the manipulated object is used to determine the differential equations that describe the rolling motion, using such equations, control strategies have been designed to manipulate an object using only rolling motions (Cole et al. 1989).

In the frictional point contact model the tangential and normal components of the contact force are related by the Coulombs friction law, and a local *sliding motion* occurs when the tangential component is larger than the normal component multiplied by the friction coefficient (Murray et al. 1994). Figure 2.3c shows an example of sliding motion, where the fingers relax the contact force applied on the object and this moves due to the gravity force. Contact sliding is a way to increase the mobility of the grasped object and the dexterity of the hand. However, when a heavy or delicate object is manipulated sliding motion is usually unwanted (Prattichizzo et al. 2020). One of the pioneers works on sliding motion was presented by Cole et al. (1992), who introduced a dynamic model of a multi-fingered hand manipulating an object with the fingertips, allowing that one of the fingers slides on the object. Later, Howe and Cutkosky (1996) introduced a comprehensive analysis of multi-fingered hand manipulation with contact sliding. More recently, a dynamic manipulation planner for n-finger hands was developed using sliding motions and taking advantage of external forces such as the gravity and object inertia (Shi et al. 2017).

In a manipulation with *finger relocation*, also known as “finger gaiting”, one or more fingers can lose the contact with the object and touch it again at another point on the object surface, changing the grasp configuration. While relocating fingers, the set of remaining contacts has to maintain the grasp. Figure 2.3d shows snapshots of an example of finger gaiting with a four-fingered hand grasping a squared object. The analysis of the dexterous manipulation considering finger relocation and assuming a planar grasp was introduced by Hong et al. (1990), providing solutions for finger gaiting for hands with three and four fingers. Finger gaiting helps in the problem of dexterous manipulation by increasing the range of movements of the object (Han and Trinkle 1998).

There exists a considerable amount of work analytically describing the motion of the object, the fingertips and the contact points during manipulation (Billard and Kragic 2019). Generally, these approaches assume knowledge of: the hand kinematics, object properties like shape, mass and center of mass, the contact locations, friction coefficients, and the surface geometry of both the object and fingertips. With this knowledge it is possible to compute, offline, finger trajectories, and determine slipping and rolling motions of the fingertips to produce a desired movement of the object during manipulation (Murray et al. 1994).

Control strategies have been proposed to deal with the manipulation of unknown objects, for instance, Arimoto et al. (2005) proposed a grasping scheme, called “Blind Grasping” that can realize dynamic object grasping without any external sensing. This blind grasping has been mostly inspired by the unique configuration of the human hand, called “Fingers-Thumb Opposability”, then, the approach was extended by considering the gravity force effect during the manipulation (Arimoto et al. 2006). A torque controller was used to optimize the grasping force applied on an object with smooth curvatures and a predefined shape, the approach can grasp objects with different shapes, but the experimental results were only performed in simulations without tactile sensors (Song et al. 2012). A position-force control scheme was used to manipulate the object following a predefined trajectory, but it was also evaluated only in simulation introducing noise in the sensor measurements to simulate a real environment (Li et al. 2012). A position-force controller with passive flexibility was proposed for versatile in-hand manipulation based on posture interpolation (Or et al. 2016a; Or et al. 2016b). Another approach uses only a position control law to change the pose of the manipulated object, but it lacks of sensory feedback which is a hard limitation (Tahara et al. 2010).

On the other hand, many approaches to dexterous manipulation are based on planning methods, and in this case the models of both the hand and the object must be known beforehand in order to compute trajectories for each manipulation mode, for instance, rolling motions of the object using the fingertips (Bicchi and Sorrentino 1995; Cherif and Gupta 1999; Doulgeri and Droukas 2013), rolling objects over the hand palm (Bai and Liu 2014), sliding motions of the fingers on the object (Cherif and Gupta 1999), or finger gaiting (Han and Trinkle 1998; Xu et al. 2007). Besides, planing methods allow to avoid the complex analysis of geometric relations and apply offline techniques like Rapidly-exploring Random Trees (RRT) and Probabilistic Road Maps (PRM) to the dexterous manipulation problem (Yashima 2004; Saut et al. 2007). Exploiting the geometric features of the objects, for instance, the axial symmetry of a light bulb, Xue et al. (2008) found contact trajectories to screw the bulb. Hertkorn et al. (2013) formulated the problem of dexterous manipulation as the search for a set of contact points on the manipulated object and the corresponding hand configurations compatible with the task to be executed, and the task is represented as a desired movement of the object and an external force to be applied or resisted. All these approaches use physics simulations to model the interaction between the hand and the object. Dexterous manipulation was also achieved by moving the fingers following a set of grasps, which are computed by applying a sampling method that provides samples of force-closure or non force-closure grasps on a discrete object (Roa and Suárez 2009).

Reinforcement learning has been also used to deal with the problem of reaching and grasping objects (Brock et al. 2009; Lampe and Riedmiller 2013; Kroemer et al. 2015), as well as changing the position of the object with respect to a global reference frame (Pastor et al. 2011; Kroemer et al. 2015). However, regarding dexterous manipulation using anthropomorphic hands, the amount of works is more limited (Andrychowicz et al. 2020). Reinforcement learning has been used to optimize trajectories of a robot arm-hand system and the desired contact forces along this trajectory, this optimization allows to solve problems like opening a door with a lever door handle and picking up a pen from the table (Kalakrishnan et al. 2011).

Dexterous manipulation can also be achieved by combining the different methods discussed above (i.e. methods based on control, planning, and learning) but, in any case, sensory feedback aids to reduce uncertainty and to increase the capabilities of the robotic system (Li et al. 2012; Funabashi et al. 2015). For instance, Ozawa et al. (2005) proposed a controller, with the only feedback of the internal joint angles and angular velocities, to achieve stable grasping, orientation control, and position control of the grasped object. As stated in Section 2.2, tactile sensing allows dealing with the lack of knowledge of some properties of manipulated objects. Contact locations obtained using tactile sensors can be used to define a “virtual object frame” determined by the centroid of the contact points, using this reference frame control (Tahara et al. 2010) or learning methods can be applied to manipulate the object (Li et al. 2014). Fast tactile feedback allows performing impressive demonstrations like, for instance, to spin a pen of known shape at a high speed (Ishihara et al. 2006).

Others manipulation strategies decompose the manipulation problem into small movements that allow the description of a complex task in terms of simpler actions (Felip et al. 2012) or use geometric reasoning to manipulate unknown objects to improve the grasp quality from the point of view of the hand, the grasp and the task (Montaño and Suárez 2018c).

"The nitrogen in our DNA, the calcium in our teeth, the iron in our blood, the carbon in our apple pies were made in the interiors of collapsing stars. We are made of starstuff."

Carl Sagan (1934 - 1996)

3 Robotic Systems Used for Experimental Validations

The approaches proposed in this thesis were validated experimentally using the Schunk Dexterous Hand (SDH2) and the Allegro hand. The software implementation was done in C++ using the Robotic Operating System (ROS) framework allowing the connection between the hardware, drivers and the developed software modules. Details of the used robotic hardware, hands and tactile sensors, as well as the developed software modules, are presented in this chapter.

3.1 Tactile Sensors

One of the main features of the robotic hands used in the experimental validation of this thesis is that they have the ability to obtain information about the contacts with the manipulated object by using tactile sensors. Both, the Schunk Dexterous hand and the Allegro Hand are equipped with tactile sensor systems developed by Weiss Robotics[1]. In this section, common functions to exploit both tactile sensor systems are presented, which were developed as part of this thesis. Specific details of each particular tactile system are discussed below when the robotic hands are introduced.

The Weiss tactile sensors are arrays of tactile sensor cells (texels). The tactile sensors have

[1] https://weiss-robotics.com/

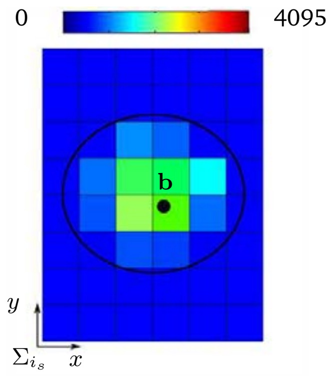


Figure 3.1: Example of a contact region on a tactile sensor.

an associated reference frame Σ_{i_s}, located in the left-bottom corner of the pad, as shown in Figure 3.1. In this work, the contacts between the fingertips and the manipulated object are modeled using the frictional point-contact model (Salisbury and Roth 1983). In the real experimentation, in general, the contact between the object and the tactile sensor takes place on a contact region including several texels, thus, the barycenter of this region, $\mathbf{b} \in \mathbb{R}^2$, is considered as the current effective contact point, and the summation of the forces sensed over all the texels is considered as the current contact force F. Figure 3.1 shows an example of a contact region highlighted with an ellipse. The measures of pressure on the texels are represented by colors. A measurement in a texel returns a value between 0, when no force is applied, and 4095 for the maximum measurable normal force, whose value depends on the type of sensor.

Tactile sensors can be connected to a PC via serial port or USB with a CDC profile[2]. The driver provided by the manufacturer of the sensors has the basic functionalities for the configuration of the sensor, the communication with it, and the reading of the pressure measurement on each texel. The pressure measurement is done using the function *getPressureOnTexel*. A C++ library, *WEISSlib*, was developed to manage, configure and read information from the tactile sensors. The library provides more complex functions that simplify the data management. Among these functions there is one, *getContactInfo*, to obtain processed information about the contact: the barycenter, the contact force, and the contact area. The pseudo-code of the *getContactInfo* function is shown in Algorithm 3.1. First, all the texels of the sensor pad are read using the function *getPressureOnTexel*(x, y), if the returned pressure value on each texel, p, is larger than a threshold, *pressureThreshold*, it is considered a valid value and the measurement is processed to compute the contact information. The contact area is the product of the number of texels with a valid measurements by the area of a single texel. The contact force is the product of the

[2] Communication Device Class (CDC) is a profile that allows the emulation of a Virtual COM-port using the ACM (Abstract Control Model) function to connect devices over USB.

Algorithm 3.1 getContactInfo

```
Ensure: contactArea, contactForce, b
  procedure GETCONTACTINFO
      sumPressure, sumX, sumY, numTexels ← 0
      contactArea, contactForce, $b_x$, $b_y$ ← 0
      for all $x$ in sensorPadColumns do
          for all $y$ in sensorPadRows do
              $p$ ← getPressureOnTexel($x, y$)
              if $p >$ pressureThreshold then
                  sumPressure ← sumPressure + $p$
                  sumX ← sumX + $x\ p$
                  sumY ← sumY + $y\ p$
                  numTexels ← numTexels + 1
              end if
          end for
      end for
      contactArea ← numTexels by areaTexel
      contactForce ← forceFactor by sumPressure by contactArea
      $b_x$ ← $\text{texel}_{\text{wide}}$ ( sumX / sumPressure )
      $b_y$ ← $\text{texel}_{\text{high}}$ ( sumY / sumPressure )
      return contactArea, contactForce, b
  end procedure
```

summation of pressures by the contact area, scaled by a calibration constant, *forceFactor*. The barycenter (b_x, b_y) of the contact region is computed as,

$$b_x = \text{texel}_{\text{wide}} \frac{\sum_{x=1}^{x_{\max}} (x\ p)}{\sum p} \tag{3.1}$$

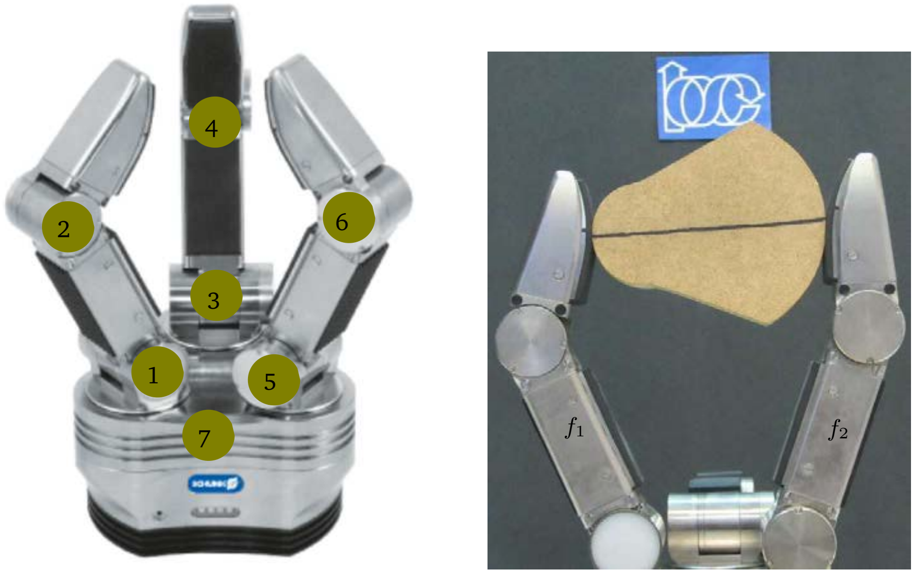
$$b_y = \text{texel}_{\text{high}} \frac{\sum_{y=1}^{y_{\max}} (y\ p)}{\sum p} \tag{3.2}$$

where $x_{\max}$ and $y_{\max}$ are the total numbers of columns and rows in the sensor pad, x is the column, y is the row, and p is the measured pressure on the texel, respectively.

3.2 Schunk Dexterous Hand

The Schunk Dexterous Hand (SDH2)[3], shown in Figure 3.2a, is a three finger hand with seven active DOF. Each finger has two DOF, located at the proximal and distal links. The hand has an additional DOF allowing the rotation of two fingers around their bases to work opposite to each other in the same plane, as shown in Figure 3.2b.

[3] https://schunk.com/nl_en/gripping-systems/series/sdh/



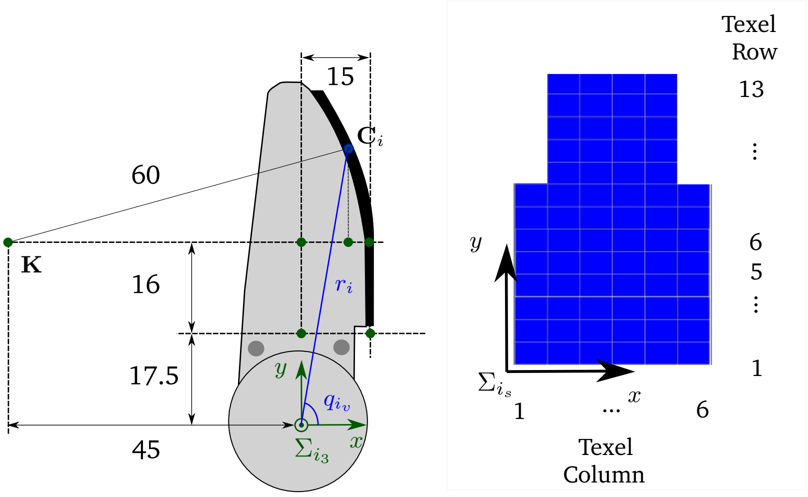
(a)

(b)

Figure 3.2: (a) Schunk Dexterous Hand (SDH2) with labels showing its seven DOF. (b) Two fingers of the SDH2 working in the same plane.

Each finger has two tactile sensor pads attached on the surfaces of the proximal and the distal links, thus, the tactile sensor system of the whole hand has six sensor pads. The pad over each proximal link has 84 texels of 3.4 mm by 3.4 mm, and the pad over each distal link (fingertip) has 68 texels, of the same size and with the following features. The pad on each fingertip has a planar part with length 16 mm and a curve part with radius 60 mm, as shown in Figure 3.3a. The planar part of the sensor pad includes the rows of texels 1 to 5 with 6-texel wide, and the curved part the rows of texels 6 to 13 with 6-texel wide from rows 6 to 8 and 4-texel wide from rows 9 to 13, making the total of 68 texels as shown in Figure 3.3. The maximum measurable normal force per texel is 3 N. It must be noted that, when the contact is produced only on one or two texels the measured force is therefore limited to up to 3 or 6 N, respectively, and these cases must be specially considered to avoid pushing the fingers trying to get larger forces. Besides, since the tactile sensors do not provide tangential components of the grasping forces, in the experiments the actual contact force could be larger than the measured one, which is not a significant problem, unless extremely fragile objects are manipulated and the normal forces are quite close to the maximal tolerated forces. In this work, only the fingertips of the two fingers working opposite to each other are used for the manipulation.

The SDH2 actuators are DC motors coupled with high-ratio gears. The hand is connected to a computer using two interfaces, one to command/read information from the motors, which can



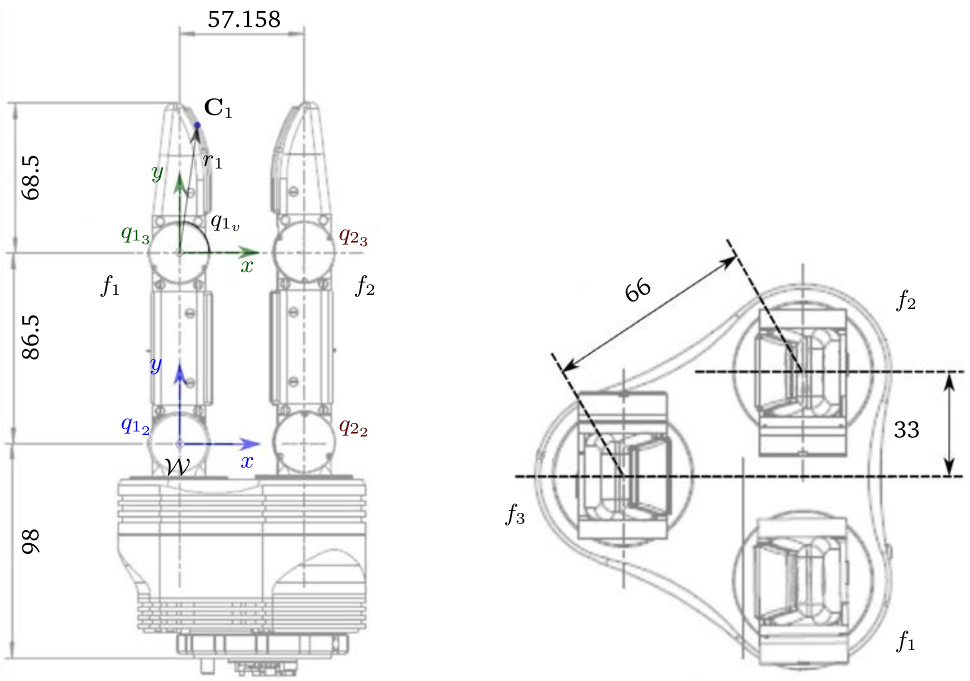
(a) (b)

Figure 3.3: (a) Lateral view of a SDH2 fingertip. (b) Front representation of a SDH2 fingertip sensor pad (all dimensions are in millimeters).

be a serial port or a controller area network (CAN) bus, and another interface to communicate with the tactile sensors, which is a serial port. The manufacturer provides a C++ driver for the basic communication and control of the hand. The tactile sensors have independent drivers and they are managed using the *WEISSlib* library introduced in the Section 3.1. A C++ library, *SDHlib*, was developed based on the manufacturer library, to manage, configure and read information from the hand. ROS modules were also developed to integrate the hand (*SDHlib*) and sensors (*WEISSlib*) libraries with other ROS modules, as for instance robotic hardware like the robot arms, or software components to test the approaches proposed in this thesis. Details of the libraries are presented later in Section 3.4, where the developed software tools are introduced.

Figure 3.4 shows the geometric model used for the kinematic analysis of the hand. A finger f_i, $i \in \{1, 2\}$, is a kinematic serial chain with n_i DOF and n_i links with length l_{ij}, $j \in \{1, ..., n_i\}$. A joint angle q_{i_j} relates the position of each link to the previous one. The configuration of the finger f_i is given by its joints angles as $\mathbf{q}_i = \{q_{i_1}, \cdots, q_{i_{n_i}}\}$. A hand configuration is given by the concatenation of the configurations of the two used fingers as $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2\}$. Each finger f_i has a reference frame Σ_{i_0} with the origin $O_{\Sigma_{i_0}}$ fixed at its base. The absolute reference frame $\mathcal{W}$ is located at the base of the finger f_1, i.e. $\mathcal{W}$ coincides with $O_{\Sigma_{1_0}}$.

Denavit-Hartenberg (DH) parameters (Craig 1986) are used to describe the kinematic model of the fingers. Table 3.1 shows the DH-parameters for finger f_i. The range of the joints q_{i_2} and q_{i_3} is from $-90°$ to $+90°$. For the coupled joint, q_{i_1}, the range is from $0°$ to $90°$, being 90° the value



(a) (b)

Figure 3.4: SDH2 kinematics model.

Table 3.1: Denavit-Hartenberg parameters for SDH2 fingers.

j	α_{i_j}	a_{i_j}	d_{i_j}	q_{i_j}
1	90	0	0	q_{i_1}
2	0	86.5	0	q_{i_2}
3	0	68	0	q_{i_3}

to work with the fingers opposite to each other.

Since the tactile sensor of each fingertip is not planar, the shape of the sensor has to be considered in order to compute the position of the contact point in the absolute reference frame $\mathcal{W}$. The processing of a sensor measurement during contact returns, as effective punctual contact point $\mathbf{b} \in \mathbb{R}^2$, the barycenter of the contact area in the sensor reference system Σ_{i_s} and, as the effective force F, the summation of all the measured forces on all the texels. Given $\mathbf{b}$, the coordinates of the contact point $\mathbf{C}_i$ in the reference system of the fingertip Σ_{i_3} are given by (Karakatsani 2011),

$$C_{i_x} = \begin{cases} 17.5 + b_x & \text{if } b_x < 16 \text{ mm} \\ 33.5 + 60 \sin\left(\frac{b_x - 16}{60}\right) & \text{if } b_x \geq 16 \text{ mm} \end{cases} \tag{3.3}$$

$$C_{i_y} = \begin{cases} 15 & \text{if } 17.5 < C_{i_x} < 33.5 \text{ mm} \\ -45 + \sqrt{60^2 - (C_{i_x} - 33.5)^2} & \text{if } 33.5 \leq C_{i_x} < 66.4 \text{ mm} \end{cases} \tag{3.4}$$

A virtual link is used to include the contact point information into the kinematics of each finger f_i. This virtual link adds an extra non-controllable DOF, q_{i_v}, to f_i (see Figure 3.4). q_{i_v} and the length of the virtual link r_i are directly determined by $\mathbf{b}$. In the reference frame Σ_{i3}, q_{i_v} and r_i are given by

$$q_{i_v} = \tan^{-1}\left(\frac{C_{i_y}}{C_{i_x}}\right) \tag{3.5}$$

$$r_i = \sqrt{{C_{i_x}}^2 + {C_{i_y}}^2} \tag{3.6}$$

The position of the contact point $\mathbf{C}_i$ referenced to the finger frame Σ_{i0}, can be computed using the virtual link values q_{i_v} and r_i, and the finger joint values q_{i_j}, $j \in \{1, 2, 3\}$, as,

$$\mathbf{C}_i = \begin{bmatrix} 86.5\ c_{q_{i_1}} s_{q_{i_2}} + r_i c_{q_{i_1}} c_{q_{i_2}} c_{(q_{i_3} - q_{i_v})} - c_{q_{i_1}} s_{q_{i_2}} s_{(q_{i_3} - q_{i_v})} \\ 86.5\ s_{q_{i_1}} s_{q_{i_2}} + r_i s_{q_{i_1}} c_{q_{i_2}} c_{(q_{i_3} - q_{i_v})} \\ 86.5\ c_{q_{i_2}} + r_i(-c_{(q_{i_3} - q_{i_v})} s_{q_{i_2}} - c_{q_{i_2}} s_{(q_{i_3} - q_{i_v})} \end{bmatrix} \tag{3.7}$$

The inverse kinematics (IK) problem is set out as follows: given the position of the contact point $\mathbf{C}_i$ with respect to the finger frame Σ_{i0} and the value of the virtual joint q_{i_v}, find the joint values q_{i_j}, $j \in \{2, 3\}$ to make the expected contact point on the fingertip (which coincides with the barycenter, $\mathbf{b}$, of the contact region) be coincident with $\mathbf{C}_i$. There are different approaches to compute the IK, for instance using numerical, analytical, or geometrical methods. In the case of a finger with 3 DOF, a geometric approach gives a close-form solution to the IK problem. Figure 3.5 shows the geometric model used to solve the IK problem of the SDH2. The values of the angles ρ, σ and γ are computed using the cosine law over the triangle described by the origins of the frames $O_{\Sigma_{i_1}}$ and $O_{\Sigma_{i_3}}$, and by the position of the contact point $\mathbf{C}_i$ as,

$$\rho = \cos^{-1}\left(\frac{-|O_{\Sigma_{i_1}}\mathbf{C}_i|^2 + L_2^2 + r_i^2}{2L_2 r_i}\right) \tag{3.8}$$

$$\sigma = \cos^{-1}\left(\frac{-r_i^2 + L_2^2 + |O_{\Sigma_{i_1}}\mathbf{C}_i|^2}{2L_2|O_{\Sigma_{i_1}}\mathbf{C}_i|}\right) \tag{3.9}$$

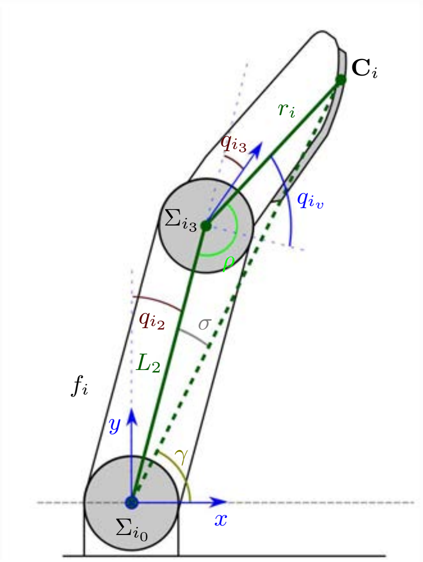


Figure 3.5: Geometric model used to solve the inverse kinematic problem of the SDH2.

$$\gamma = \tan^{-1}\left(\frac{C_{i_y}}{C_{i_x}}\right) \tag{3.10}$$

As mentioned above, the value of q_{i_1} is fixed to 90° to allow the fingers work opposite to each other. Then, the values of q_{i_2} and q_{i_3}, are given by,

$$q_{i_2} = -\sigma - \gamma + \mathrm{sgn}(C_{i_x})\pi/2 \tag{3.11}$$

$$q_{i_3} = -\rho + \pi/2 + q_{i_v} \tag{3.12}$$

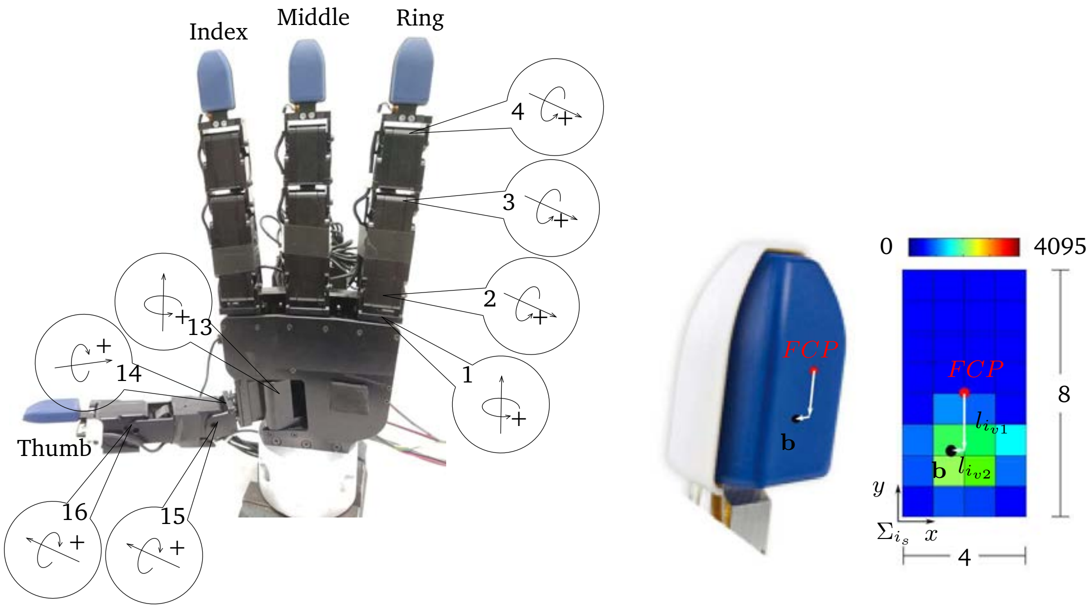
where

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

3.3 Allegro Hand

The Allegro Hand from Wonik Robotics[4] is a four-finger anthropomorphic hand with 4 DOF in each finger, as shown in Figure 3.6a. The Index (I), Middle (M), and Ring (R) fingers have the

[4] http://www.wonikrobotics.com/Allegro-Hand.htm



(a) (b)

Figure 3.6: (a) Anthropomorphic robotic hand Allegro Hand, with 16 degrees of freedom (DOF) and 4 fingertips with tactile sensors WTS-FT 0408. (b) Detail of the fingertip with tactile sensor WTS-FT 0480.

same kinematic structure, the first degree of freedom fixes the orientation of the working plane Π_i, $i \in \{I, M, R\}$, within the finger workspace, while the other three DOF (flexion/extension) are used to make the fingertip reach a point and an orientation in Π_i. In the case of the Thumb (T), the first DOF produces the abduction movement and the second DOF fixes the orientation of the working plane Π_T of the Thumb, leaving only two DOF to work on it, that is, for the Thumb, the position and the orientation of the fingertip are not independent of each other.

The fingertips of the commercial version of the Allegro Hand do not have tactile sensors, thus, for this work they were replaced in the Robotics Lab of the IOC by sensorized fingertips Weiss Tactile Sensors WTS-FT 0408[5], increasing the capabilities of the hand for dexterous manipulation. The WTS-FT are sensorized fingertips with the same functionality principle that the tactile sensors of the SDH2, but with a different shape and size of the tactile sensing matrix, which in the case of the WTS-FT has 4 by 8 texels of 3.8 mm by 3.8 mm, as shown in Figure 3.6b. The WTS-FT sensors are connected to a PC via serial port with a transmission rate of 50 Hz. A measurement of the pressure in each texel returns a value between 0 when no force is applied and 4095 for the maximum measurable normal force of 1.23 N.

Figure 3.7 shows the dimensions and the reference systems used to compute the kinematics of

[5] https://weiss-robotics.com/

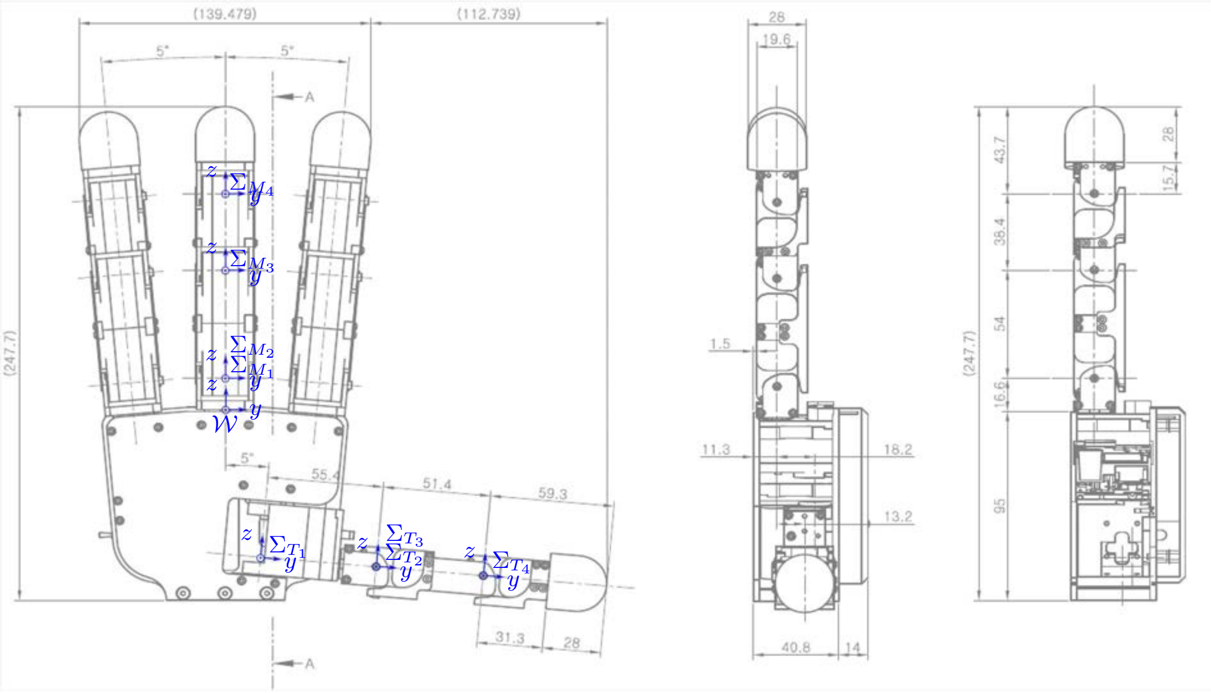


Figure 3.7: Allegro hand dimensions and reference frames used to compute the hand kinematics.

the Allegro Hand. The positions of the origins of the reference systems of the joints of the hand are given in the Table 3.2.

Similarly to the SDH2, a finger, f_i, is modeled as a kinematic serial chain with n_i DOF, $i \in \{I, M, R, T\}$. Each finger has n_i links with length l_{ij}, $j \in \{1, ..., n_i\}$, a joint angle q_{i_j} relates the position of each link to the previous one. The configuration of the finger f_i is given by its joints angles as $\mathbf{q}_i = \{q_{i_1}, \cdots, q_{i_{n_i}}\}$. A hand configuration is given by the concatenation of the configurations of the fingers as $\mathbf{Q} = \{\mathbf{q}_I, \mathbf{q}_M, \mathbf{q}_R, \mathbf{q}_T\}$ Each finger f_i has a reference frame Σ_{i_0}, with the origin, $O_{\Sigma_{i_0}}$, fixed at its base. The absolute reference frame $\mathcal{W}$ is located at the base of the finger f_M, in the intersection point with the palm of the hand, as shown in Figure 3.7.

The Kinematics and Dynamics Library (KDL)[6] is used to compute the kinematics of the hand. This library requires the robot to be described using a Unified Robot Description Format (URDF) which is based on extensible markup language (XML) and uses computer-aided design (CAD) models for the graphical representation of the hand and the sensors. In order to include the contact information in the kinematics of the hand, two virtual prismatic links $l_{i_{v1}}$ and $l_{i_{v2}}$ are used to locate the contact point on the sensor pad, assuming that the sensor surface is flat. These

[6] https://www.orocos.org/kdl

Table 3.2: Position of the reference frames of the fingers of the Allegro hand.

Index

Frame	x	y	z
Σ_{I_1}	0	45.098	14.293
Σ_{I_2}	0	45.098	14.293
Σ_{I_3}	0	49.804	68.087
Σ_{I_4}	0	53.151	106.341

Middle

Frame	x	y	z
Σ_{M_1}	0	0	16.6
Σ_{M_2}	0	0	16.6
Σ_{M_3}	0	0	70.6
Σ_{M_4}	0	0	109

Ring

Frame	x	y	z
Σ_{R_1}	0	-45.098	14.293
Σ_{R_2}	0	-45.098	14.293
Σ_{R_3}	0	-49.804	68.087
Σ_{R_4}	0	-53.151	106.341

Thumb

Frame	x	y	z
Σ_{T_1}	-18.2	16.958	-73.288
Σ_{T_2}	-13.2	72.147	-78.116
Σ_{T_3}	-13.2	72.147	-78.116
Σ_{T_4}	-13.2	123.351	-82.596

virtual links add two non-controlled DOF $q_{i_{v1}}$ and $q_{i_{v2}}$, at the end of the finger kinematic chain. Figure 3.6b shows the barycenter $\mathbf{b}$ of the contact region and the virtual links that localize the contact point with respect to the fingertip center point (FCP) on the sensor surface.

The joints of the Allegro hand have DC motors as actuators and potentiometers to measure the joint values with a resolution of 0.002°. The hand is connected to a computer by a controller area network (CAN) bus. In order to control the position of the hand, a position controller was developed to command the torque to be applied to each joint. The proposed controller includes gravity compensation, and it is implemented according to the following control law,

$$\boldsymbol{\tau}(t) = \mathbf{u}(t) + \boldsymbol{\tau}_{gc} \tag{3.13}$$

where $\boldsymbol{\tau}_{gc}$ is the torque to compensate the weight of the finger structure (gravity compensation), and $\mathbf{u}(t)$ is the torque to reduce the position error, computed as,

$$\mathbf{u}(t) = K_P\mathbf{e}(t) + K_D\frac{d\mathbf{e}(t)}{dt} + K_I\int_0^t \mathbf{e}(t)dt \tag{3.14}$$

where the gains K_P, K_D, and K_I are the proportional, derivative and integral gains, respectively, and $\mathbf{e}(t)$ is the position error, defined as the difference between the desired and the current hand configurations $\mathbf{Q}_d$ and $\mathbf{Q}_k$ respectively, i.e.

$$\mathbf{e}(t) = \mathbf{Q}_d(t) - \mathbf{Q}(t) \tag{3.15}$$

For the implementation of the controller the integral term is discretized, with a sampling time

Algorithm 3.2 updateController

```
Require: $\mathbf{Q}_k$, $\mathbf{Q}_d$, $K_P$, $K_D$, $K_I$, $\boldsymbol{\tau}_{gc}$
Ensure: $\boldsymbol{\tau}(t_{k+1})$
  procedure UPDATECONTROLLER
      Compute the position error $\mathbf{e}(t_k)$ using Eq. (3.15)
      Compute the integral component of the controller as
          $\mathbf{I} = \mathbf{I} + \mathbf{e}(t_k)dt$
      Compute the derivative component of the controller as
          $\mathbf{D} = (\mathbf{e}(t_k) - \mathbf{e}(t_{k-1}))/dt$
      Compute $\mathbf{u}(t_k)$ as
          $\mathbf{u}(t_k) = K_P\mathbf{e}(t_k) + K_D\mathbf{D} + K_I\mathbf{I}$
      Compute $\boldsymbol{\tau}(t_{k+1})$ as
          $\boldsymbol{\tau}(t_{k+1}) = \mathbf{u}(t_k) + \boldsymbol{\tau}_{gc}$
  end procedure
```

Δt, as

$$\int_0^t \mathbf{e}(t)dt \cong \sum_{k=1}^{T} \mathbf{e}(t_k)\Delta t \tag{3.16}$$

and the derivative term is approximated as

$$\frac{d\mathbf{e}(t)}{dt} \cong \frac{\mathbf{e}(t_k) - \mathbf{e}(t_{k-1})}{\Delta t} \tag{3.17}$$

Algorithm 3.2 shows the pseudo-code of the function *updateController* which computes the torque applied to the joints in each control cycle.

3.4 Developed Software Tools

Robot Operating System (ROS) (Quigley et al. 2015) is an open-source flexible framework that aims to simplify the development of general-purpose collaborative software for robotics. It provides services such as hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, and package management. In the ROS framework, the processing takes place in nodes that can be running in the same or in different CPU-cores or computers. Nodes, coordinated by a master node, send data streams to each other and they can be dynamically configured by services. Data transferring between nodes can be performed using service or topic kind messages. Services work under a request/reply interaction, while topics under a publish/subscribe interaction.

Figure 3.8 shows the developed software modules to interact with the hardware used for the validation of the manipulation approaches proposed in this thesis, the modules are interconnected using the ROS framework. As can be seen in the figure, the designed software

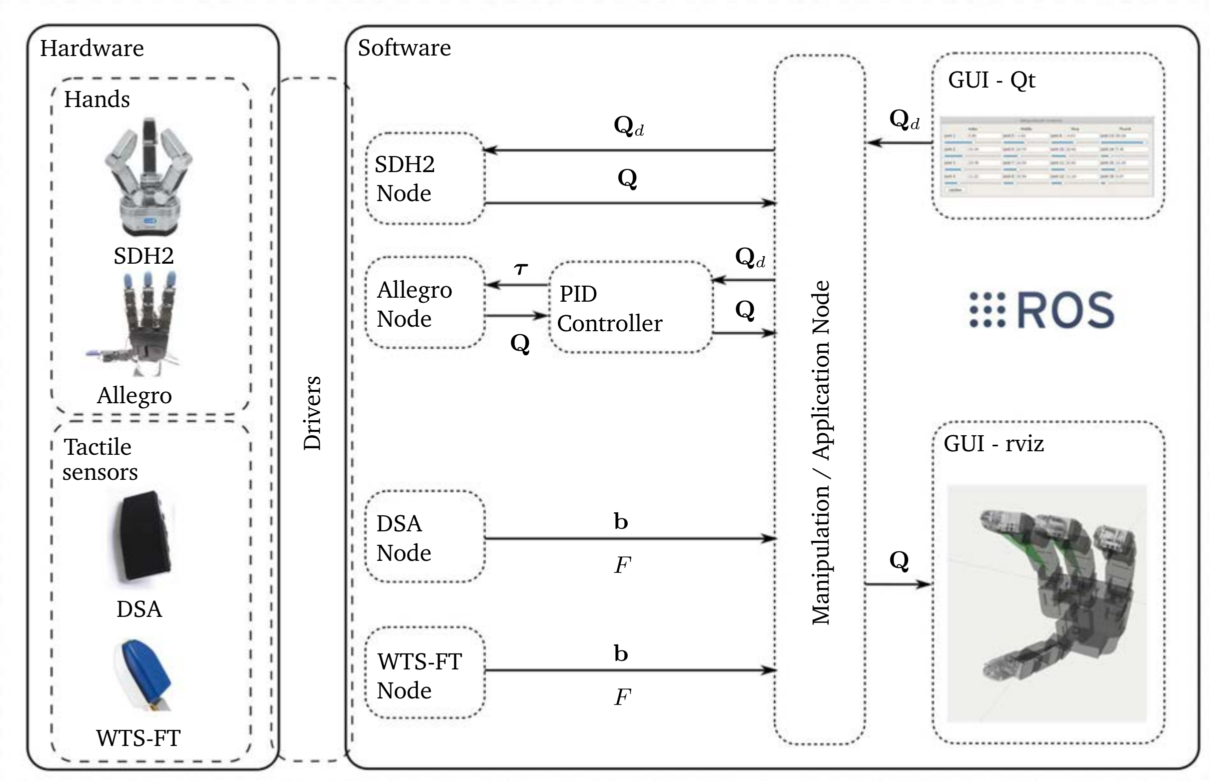


Figure 3.8: Hardware used in the experimentation and developed software modules.

modules are:

- two modules to command an read the current state of the robotics hands (SDH2 and Allegro).
- a module to control the Allegro Hand using a PID controller.
- two modules to get the measurements of the tactile sensor system (DSA and WTS-FT).
- a module to run the manipulation algorithms proposed in this work.
- a module GUI to command the movements of each joint of the hands (GUI-Qt).
- a module GUI to visualize the hands and the applied and measured forces (GUI-rviz).

The SHD2 and Allegro Nodes are based on the *SDHlib* and *AHandlib* libraries respectively, these nodes offer the basic functionalities to command movements and read the current state of the hands, as well as functions to compute the kinematics of the hand. In the case of the SDH2 hand, the hand is commanded by a set of desired joint values (desired hand configuration) $\mathbf{Q}_d$, while the Allegro hand receives the torques $\boldsymbol{\tau}$ to be applied by the joint motors, in both cases feedback is given by the encoders of the joints that measure the hand configuration $\mathbf{Q}$.

The aim of the PID controller Node in the Allegro Hand is to compute proper torques $\boldsymbol{\tau}$ to command the movements of the hand. The Node has as input the desired hand configuration $\mathbf{Q}_d$ from the Manipulation/Application Node and as feedback the current hand configuration $\mathbf{Q}$ from the Allegro Node.

As with the robotic hands, there are two nodes to manage the DSA 9210 and the WTS-FT tactile sensors; the nodes allow to compute the contact information, represented by the barycenter of the contact region $\mathbf{b}$ and the contact force F applied on that region, both nodes are based on the *weisslib* library, introduced in Subsection 3.1.

The approaches proposed in this thesis are coded in the Manipulation/Application Node, which is able to connect with SDH2 and DSA or Allegro and WTS-FT nodes, respectively. Besides, the Manipulation/Application Node also generates the hand state for the virtual representation of the hand in the GUI-rviz Node and receives the desired configuration of the hand from the GUI-Qt Node. rviz[7] is a 3D visualization tool for ROS-based applications.

The GUI-Qt[8] Node has independent inputs to command each joint of the hands. The number of inputs depends on the selected hand. As output, the node publishes the desired hand configuration $\mathbf{Q}_d$.

The GUI-rviz Node allows visualizing a rviz scene with the model of the hand, the forces involved in the manipulation (forces measured by the tactile sensors and forces computed using the applied torques on the hand joints), working planes for each finger, and the contact points, among other elements. The scene is described using an URDF file which allows the inclusion of CAD models for the graphical representation of the hand and the sensors.

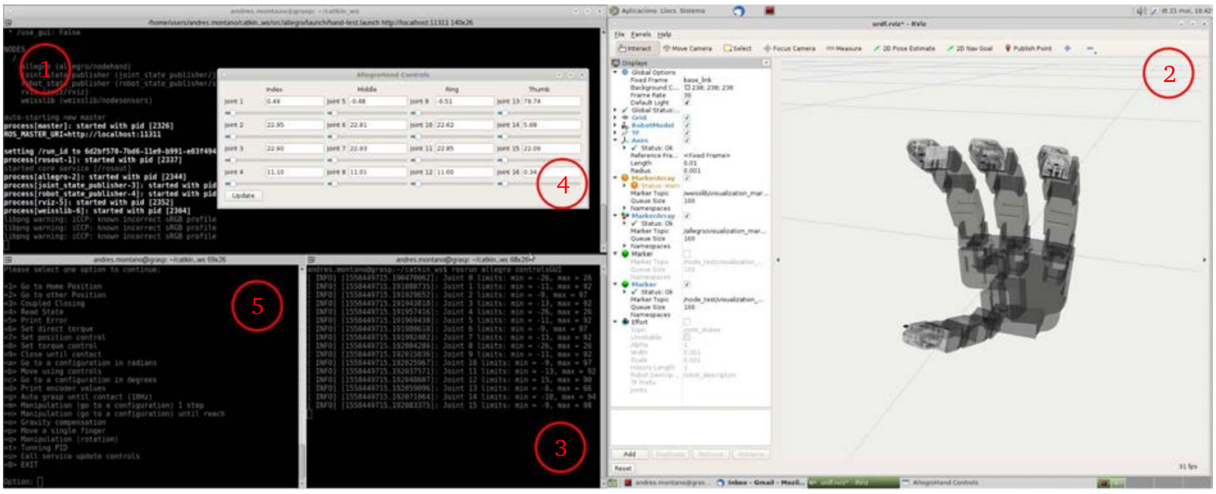
The measured forces are represented by rviz markers (arrows in this particular case) using the contact information, location and contact force, returned by the sensors. The computed contact forces are obtained as

$$\mathbf{F} = J(q)^{T^*}\boldsymbol{\tau} \tag{3.18}$$

where $J(q)^{T^*}$ is the inverse of the transposed Jacobian matrix, which depends on the joint values

[7] http://wiki.ros.org/rviz

[8] https://www.qt.io/



1
2
3
4
5
andres.montano@grasp: ~/catkin_ws
/home/users/andres.montano/catkin_ws/src/allegro/launch/hand-test.launch http://localhost:11311 140x26
andres.montano@grasp: ~/catkin_ws 69x26
andres.montano@grasp: ~/catkin_ws 68x26
AllegroHand Controls
Update
Aplicaciones Llocs Sistema
urdf.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Add
Duplicate
Remove
Rename
Reset
31 fps
Inbox - Gmail - Mozil...
AllegroHand Controls

```
 * /use_gui: False

NODES
  /
    allegro (allegro/nodehand)
    joint_state_publisher (joint_state_publisher/j
    robot_state_publisher (robot_state_publisher/s
    rviz (rviz/rviz)
    weisslib (weisslib/nodesensors)

auto-starting new master
process[master]: started with pid [2326]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 6d2bf570-7bd6-11e9-b991-e03f494
process[rosout-1]: started with pid [2337]
started core service [/rosout]
process[allegro-2]: started with pid [2344]
process[joint_state_publisher-3]: started with pid
process[robot_state_publisher-4]: started with pid
process[rviz-5]: started with pid [2352]
process[weisslib-6]: started with pid [2364]
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
```

	Index		Middle		Ring		Thumb
joint 1	0.49	joint 5	-0.48	joint 9	-0.51	joint 13	79.74
joint 2	22.95	joint 6	22.81	joint 10	22.62	joint 14	5.68
joint 3	22.90	joint 7	22.93	joint 11	22.85	joint 15	22.09
joint 4	11.10	joint 8	11.01	joint 12	11.00	joint 16	0.34

```
Please select one option to continue:

<1> Go to Home Position
<2> Go to other Position
<3> Coupled Closing
<4> Read State
<5> Print Error
<6> Set direct torque
<7> Set position control
<8> Set torque control
<9> Close until contact
<a> Go to a configuration in radians
<b> Move using controls
<c> Go to a configuration in degrees
<d> Print encoder values
<g> Auto grasp until contact (10Hz)
<m> Manipulation (go to a configuration) 1 step
<n> Manipulation (go to a configuration) until reach
<o> Gravity compensation
<p> Move a single finger
<q> Manipulation (rotation)
<t> Tunning PID
<u> Call service update controls
<0> EXIT

Option:
```

```
andres.montano@grasp:~/catkin_ws$ rosrun allegro controlsGUI
[ INFO] [1558449715.190478062]: Joint 0 limits: min = -26, max = 26
[ INFO] [1558449715.191888735]: Joint 1 limits: min = -11, max = 92
[ INFO] [1558449715.191929652]: Joint 2 limits: min = -9, max = 97
[ INFO] [1558449715.191943818]: Joint 3 limits: min = -13, max = 92
[ INFO] [1558449715.191957416]: Joint 4 limits: min = -26, max = 26
[ INFO] [1558449715.191969438]: Joint 5 limits: min = -11, max = 92
[ INFO] [1558449715.191986618]: Joint 6 limits: min = -9, max = 97
[ INFO] [1558449715.191992402]: Joint 7 limits: min = -13, max = 92
[ INFO] [1558449715.192004286]: Joint 8 limits: min = -26, max = 26
[ INFO] [1558449715.192015036]: Joint 9 limits: min = -11, max = 92
[ INFO] [1558449715.192025967]: Joint 10 limits: min = -9, max = 97
[ INFO] [1558449715.192037571]: Joint 11 limits: min = -13, max = 92
[ INFO] [1558449715.192048687]: Joint 12 limits: min = 15, max = 90
[ INFO] [1558449715.192059096]: Joint 13 limits: min = -6, max = 88
[ INFO] [1558449715.192071064]: Joint 14 limits: min = -10, max = 94
[ INFO] [1558449715.192083375]: Joint 15 limits: min = -9, max = 98
```

```
Global Options
  Fixed Frame          base_link
  Background C...      238; 238; 238
  Frame Rate           30
  Default Light        ☑
Global Status:...
Grid                   ☑
RobotModel             ☑
TF                     ☑
Axes                   ☑
  Status: Ok
  Reference Fra...     <Fixed Frame>
  Length               0.01
  Radius               0.001
MarkerArray            ☑
  Status: Warn
  Marker Topic         /weisslib/visualization_mar...
  Queue Size           100
  Namespaces
MarkerArray            ☑
  Status: Ok
  Marker Topic         /allegro/visualization_mar...
  Queue Size           100
  Namespaces
Marker                 □
  Marker Topic         /node_test/visualization_...
  Queue Size           100
  Namespaces
Marker                 ☑
  Status: Ok
  Marker Topic         /node_test/visualization_...
  Queue Size           100
  Namespaces
Effort                 □
  Topic                joint_states
  Unreliable           □
  Alpha                1
  Width                0.001
  Scale                0.001
  History Length       1
  Robot Descrip...     robot_description
  TF Prefix
  Joints
```

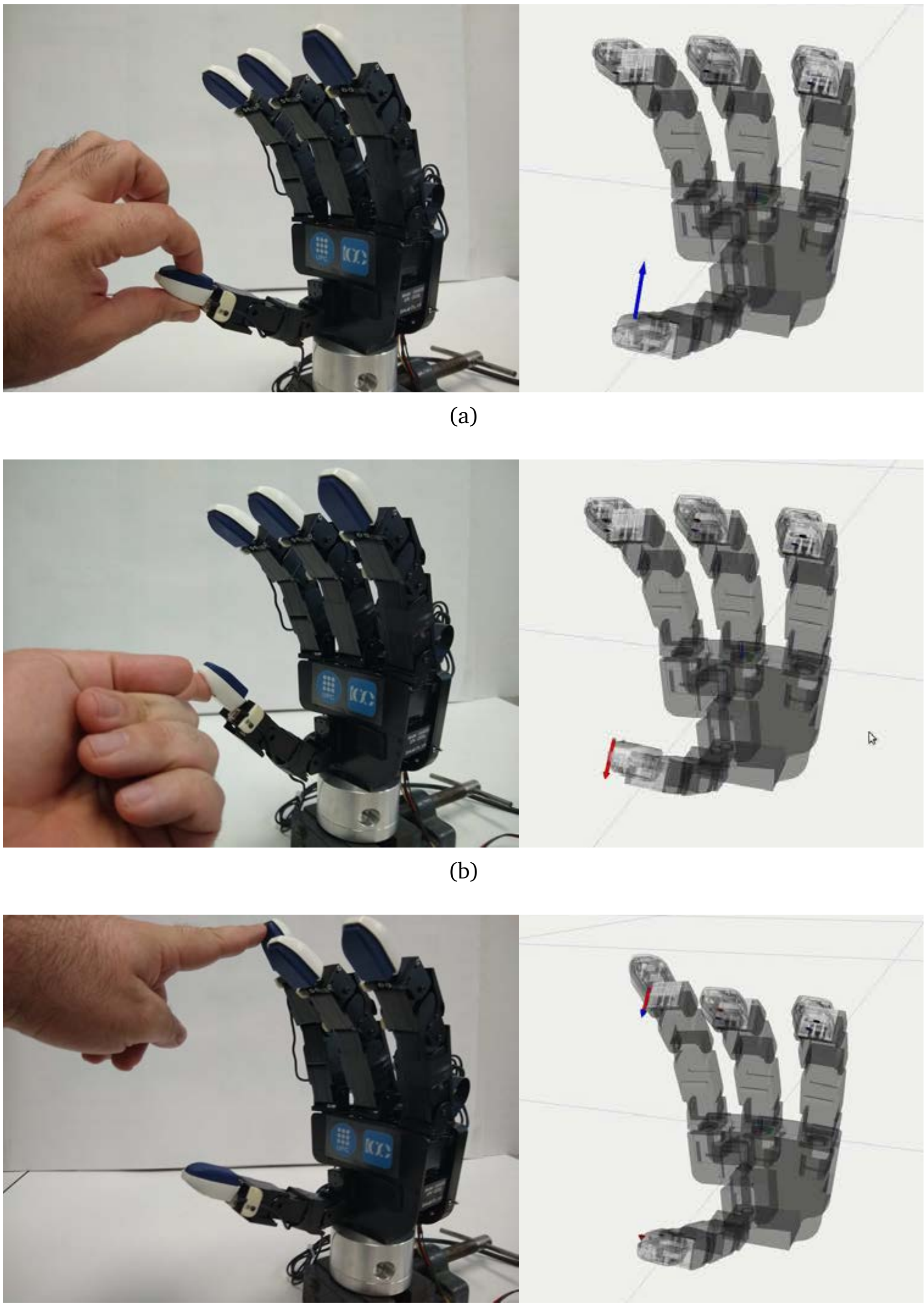
Figure 3.9: Screenshot of the visualization tool during an example using the Allegro Hand with tactile sensors.

of the hand, and $\boldsymbol{\tau}$ is the vector of the applied torques at each joint.

A screenshot of the visualization tool during an example is being run using the Allegro hand with the WTS-FT tactile sensors is shown in Figure 3.9. In the figure, there are five windows identified with a circled number in red. The first window corresponds to a system console that executes the nodes involved in the communication with the hand *AHandlib* and the sensors *weisslib*. The second window shows the GUI-rviz Node, where the models of the hand and the tactile sensors and the measured and the computed forces are displayed. The third window corresponds to a system console that launches the GUI-Qt Node to control the individual movement of the joints of the Allegro hand. The fourth window shows the graphical interface of the GUI-Qt Node. The fifth window corresponds to the Manipulation/Application Node, which has an user interface to execute a manipulation strategy and displays relevant information, as the measured forces and computed forces exerted by the fingers.

The following examples illustrate the main functionalities of the developed system. Figure 3.10 shows three examples in which the operator interacts with the robot hand and touch the sensors to generate forces without holding any object, the forces are displayed on the GUI. In the three examples the hand was placed in a predetermined configuration and held static in it. The forces registered by the tactile sensors are represented by blue arrows, while those calculated using the torques applied to the joints are represented by red arrows. In the first example, pressure is exerted directly on one of the tactile sensors and the force registered by the sensor is displayed on the graphic interface. In the second example the finger is pushed from the back of the fingertip trying to change the finger configuration. As a result of this action, the hand controller increases the torque at the finger joints to maintain the default setting, increasing the torques produces a force at the fingertips that is displayed in red on the graphical interface but the tactile sensor does not measure anything, so there is not any force represented in blue. In the third example, the finger is pushed from the fingertip trying to change its configuration. In this case, the force registered by the sensor and the calculated force due to the torques applied to the joints are simultaneously displayed.

Figure 3.11 shows four examples grasping different objects. In the first example, a spherical object is held using two fingers; in the second example, the hand holds a bottle with flat faces using three fingers; in the third example, a jar is held at the upper and lower ends using three fingers again; and, in the fourth example, a cylinder-shaped bottle is held using the four fingers of the hand. In each example it is shown: an image with the grasp executed by the hand and two screen captures of the graphical interface showing, in the first one, the forces computed using the torques applied to the joints, and, in the second one, the forces registered on the tactile sensors. Note that in the last two examples there is a noticeable difference between the forces produced by the torques at the joints and those measured by the sensors. This is due to the fact that tangential forces are being applied to the surface of the sensor pad and the sensors cannot measure them. This effect shows the usefulness of the developed system that quickly allows to graphically visualize this difference, and therefore what is actually happening with the grasping forces.

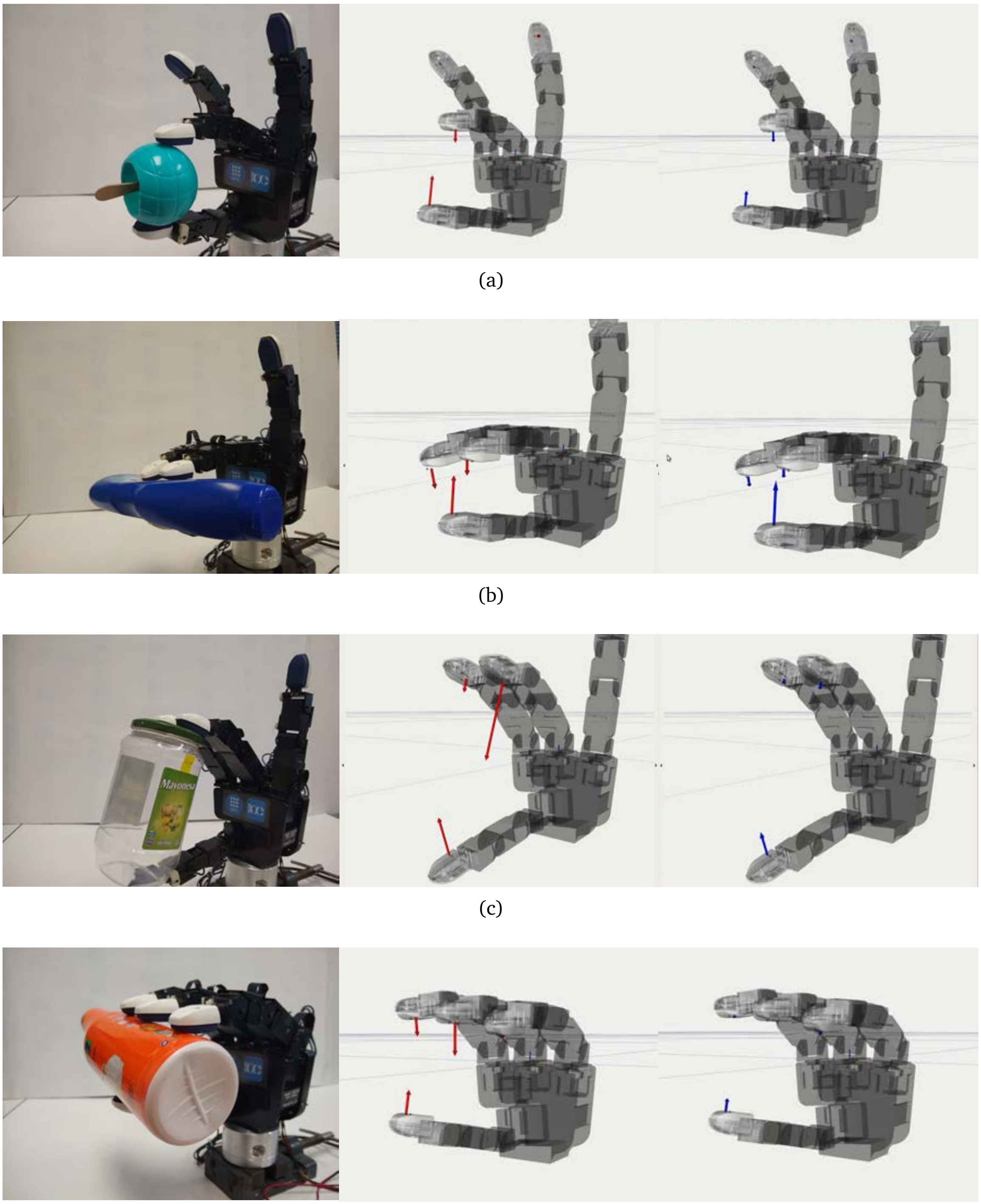


(a)

(b)

(c)

Figure 3.10: Examples of the interaction with fingers and tactile sensors producing forces that are visualized on the GUI of the software tool. (a) Force measured by the tactile sensor due to the pressure exerted on the sensor. (b) Computed force using the torques commanded by the controller to hold the hand in a static configuration while an external force is applied to the finger but not on the sensor pad. (c) Measured and computed forces when a force is pushing the finger on the sensor pad.



(a)

(b)

(c)

(d)

Figure 3.11: Grasping examples of objects with different shapes: (a) Spherical object grasped using two fingers. (b) Bottle with flat faces grasped using three fingers. (c) Jar grasped by the top and bottom faces using three fingers. (d) Cylindrical bottle grasped using four fingers.

*"If at first an idea does not sound absurd,
then there is no hope for it."*
Albert Einstein (1879 - 1955)

4

Dexterous Manipulation of Unknown Objects

A set of manipulation strategies to manipulate unknown objects is described in this chapter. The manipulation strategies use as inputs only tactile and kinematic information obtained during the manipulation process itself, thus, the manipulation is performed as a reactive process.

4.1 Introduction

In general, the manipulation process usually pursues three main goals, one from the hand point of view, one from the grasp (relation hand-object) point of view and one from the object point of view. These goals are, respectively:

- The optimization of the hand configuration, i.e. the search for a "comfortable" hand configuration while holding the object;
- The optimization of the grasp quality, i.e. the search for a "secure" grasp such that the hand can resist external perturbations applied on the object;
- The optimization of the object configuration, i.e. the search for an "appropriate" object position and orientation to accomplish the requirements of a given task.

The manipulation strategies introduced in this chapter allows the manipulation of unknown objects pursuing these three goals. The expression "unknown object" is used to indicate that the model of the object is not needed during the manipulation process (note that "model of the object" includes shape, texture, stiffness, center of mass, friction coefficient and any other related physical property of the object).

Manipulation of unknown objects can be performed as a reactive procedure using the tactile information obtained at the contacts at the fingertips, and kinematic information of the hand. It must be remarked that tactile information is used only to obtain knowledge of two relevant things: the position of the contact point on each fingertip and the module of the corresponding contact force.

The main goal of the approaches proposed in this thesis is the dexterous in-hand manipulation (i.e. no need of wrist or arm movements) of an unknown object starting from a given initial grasp, keeping the contact between each finger and the object during the manipulation (i.e. finger gaiting is not applied) and the grasping forces within a desired range, while preventing the object from falling. The manipulation process determines the movements (sequences of hand configurations) to manipulate the object improving a quality index. The initial grasp could be non-optimal due to several reasons (e.g. accessibility, position uncertainty or incomplete information about the object), but, in any case, the planning and the execution of the initial grasp are outside the scope of this work.

The computation of the finger movements is done following a specific manipulation strategy for each of the goals mentioned above, and a specific index to be minimized is defined to measure the quality of the manipulation actions.

The following assumptions are considered:

- The robotic hand used for the manipulation has tactile sensors to obtain information about the contacts with the manipulated object, and no other external feedback source is available, as, for instance, visual information.
- The manipulated objects are rigid bodies and their shape is unknown. The proposed approaches could work also for soft objects, there is not any specific constraint for it, but a limit for the acceptable softness is not determined in this work.
- The friction coefficient is not known neither identified during the manipulation. It is assumed to be above a minimum security value which can be roughly determined considering the object material and the rubber surface of the fingertips. In the experimentation we compute the movements using the minimum value of the possible

friction coefficient between the material of the fingertips and the used objects, i.e. a conservative value below the real friction coefficient.

- The finger joints have a position control to make them trying to reach the commanded positions, which is the most frequent case in a commercial hand with a closed controller. No force control is required at the level of the joint controllers. The proposed approaches use the tactile measurements to generate commanded positions taking into account the current contact forces, thus, it is actually acting as an implicit upper level force control loop.

As it was discussed in Section 3.3, the contact between a fingertip and the object produces contact regions on the sensor pad. The contact between each fingertip and the object is modeled using the punctual contact model (Mason 2001), thus, in order to adjust the contact model, the barycenter of the actual contact region (either a single one or a set of disjoints sub-regions) is considered as the current contact point. Besides, the summation of the forces sensed at each texel in the actual contact region is considered as the current contact force applied by the finger at the equivalent punctual contact.

Many actions in every-day and industrial tasks can be performed using only two fingers, for instance, rotate one object for vision inspection or change the grasp posture and the object position for more comfortable manipulation (Li et al. 2013). Increasing the number of fingers increases the redundancy of the grasp allowing other types of movements such as finger gaiting, re-grasping or finger pivoting/tracking (Ma and Dollar 2011), but also increases the complexity of the grasp analysis.

In the particular case of dexterous manipulation using two fingers, the fingers perform a grasp comparable with a human grasp when the thumb and index fingers are used to manipulate an object, with the fingertips movements lying on a plane (MacKenzie and Iberall 1994). This type of grasp limits the movement of the object to a plane but it allows different practical actions in every-day and industrial tasks, like, for instance, matching the orientation of two pieces to be assembled or inspecting an object (Toh et al. 2012). Figure 4.1 shows the geometric model of a two-finger grasp.

The indexes to evaluate the quality of the grasp are introduced below, then, the manipulation strategies using two fingers are presented considering heuristics and gradient based methods, finally an extension for three fingers is discussed in the last section of the chapter.

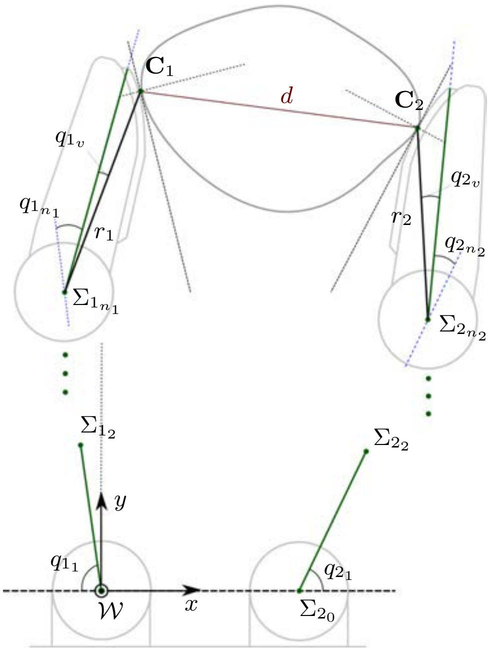


Figure 4.1: Geometric model for a two-finger grasp.

4.2 Quality Indexes

The following subsections introduce the indexes used to quantify the quality related to the hand configuration, the grasp and the object orientation. The indexes, that are expressed as functions of the hand joint values, must be optimized during the manipulation process.

4.2.1 Related to the Hand Configuration

The optimization of the hand configuration implies that the fingers must try to reach specific positions while preventing the fall of the object. These positions are generally defined by the middle-range of the joints because, in the middle-range positions the joints are far away from their mechanical limits, thus, there is a potential wider range of movements, but it could be also arbitrarily defined by the user according to the particular features of the robotic hand.

Let $q_{i_{j_0}}$ be the predefined desired specific position of the j-th joint of the finger i, then $\mathbf{Q}_0 = \{q_{{i_j}_0}, i \in \{1,2\}, j \in \{1, n_i\}\}$ is the desired specific configuration of the hand. Then, the goodness of the hand configuration is indicated by a quality index $\mathcal{Q}_c$ computed according to

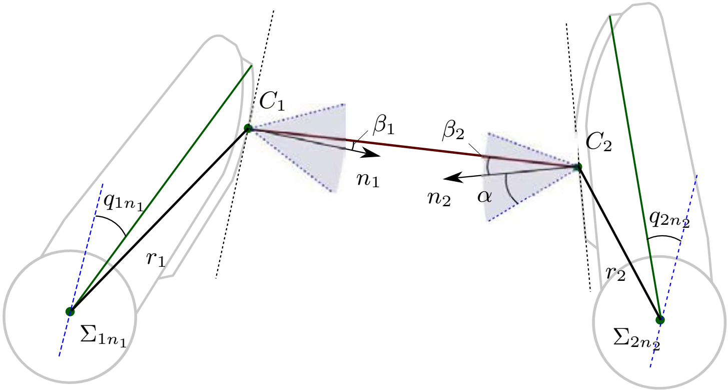


Figure 4.2: Geometric model used to compute the friction constraints.

the current joint values q_{i_j} as

$$\mathcal{Q}_c = \sum_{i=1}^{2}\sum_{j=1}^{n_i}\left(\frac{q_{i_j} - q_{i_{j_0}}}{q_{i_{j_{max}}} - q_{i_{j_{min}}}}\right)^2 \tag{4.1}$$

where $q_{i_{j_{max}}}$ $q_{i_{j_{min}}}$ and q_{i_j} are the maximum limit, minimum limit and the actual position of the j-th joint of f_i, respectively. The hand configuration is improved by minimizing $\mathcal{Q}_c$, which favors the hand configurations with the joints as close as possible to the desired specific positions (Liégeois 1977).

4.2.2 Related to the Grasp

When the goal of the manipulation is the optimization of the grasp quality, the hand has to manipulate the object optimizing the current FC grasp. We use a quality index $\mathcal{Q}_g$ that relates the angle between the normal forces at the contact points and the segment between the contact points. $\mathcal{Q}_g$ favors the finger forces closer to the surface normal. If the segment is close to the boundary of the friction cone, i.e. far from the normal directions, the object could easily slip in presence of perturbations (Liu et al. 2004). $\mathcal{Q}_g$ is expressed as,

$$\mathcal{Q}_g = \frac{\beta_1 + \beta_2}{2} \tag{4.2}$$

where β_1 and β_2 are the angles between the normal direction at each contact point and the segment between contact points (see Figure 4.2).

The optimization of the grasp quality implies that the fingers must manipulate the object increasing the security margin of the FC grasp given by the angles β_i. i.e. the segment connecting both contact points must lie far from the boundary of the friction cones.

4.2.3 Related to the Object Orientation

The optimization of the object orientation γ_k implies that the fingers must rotate the object towards a desired goal orientation γ_d. The orientation of the object in the initial grasp is considered as $\gamma_0 = 0$, and the desired orientation of the object γ_d is relative to it. Then, the manipulation strategy must reduce the difference between γ_d and the current object orientation γ_k. The quality index could be just the orientation error $|\gamma_d - \gamma_k|$, but in order to constrain it to the range $[0, 1]$ it is normalized dividing by $\gamma_i - \gamma_d$, γ_i being the current orientation at the time γ_d is given, i.e.

$$\mathcal{Q}_r = \left| \frac{\gamma_d - \gamma_k}{\gamma_d - \gamma_i} \right| \tag{4.3}$$

The orientation of the object γ_k can be computed using basic geometry and the information obtained from the tactile sensors and the finger kinematics, no other external feedback is considered (like, for instance, a vision system) although it could exist at a higher level (for instance to determine γ_d, but this is outside of the scope of this work). For fingertips with circular shape, the current object orientation γ_k is given by (Ozawa et al. 2004)

$$\gamma_k = \frac{2R + d_k}{d_k}(\theta_0 - \theta) + \frac{R}{d_k}\sum_{j=1}^{n_1}(q_{1_{j_{\gamma_0}}} - q_{1_{j_k}}) - \sum_{j=1}^{n_2}(q_{2_{j_{\gamma_0}}} - q_{2_{j_k}}) \tag{4.4}$$

being

θ: the average of the two angles between an arbitrary reference axis attached to the object and the directions normal to each fingertip at the corresponding contact point,

θ_0: the value of θ at the initial grasp (i.e. for γ_0),

$q_{i_{j_k}}$: the current value of the ij-th joint, i.e. joint $j = 1, ..., n_i$ of finger $i = 1, 2$,

$q_{i_{j_{\gamma_0}}}$: the value of the ij-th joint at the initial grasp (i.e. for γ_0),

d_k: the distance between the contact points, and

R: the radius of the fingertip.

The first term in Eq. (4.4) has a factor that depends on the variation of θ, then, since θ does not change significantly during the manipulation (i.e. $\theta \approx \theta_0$) the first term can be neglected. Thus, γ_k can be approximated by

$$\gamma_k \approx \frac{R}{d_k}\sum_{j=1}^{n_1}(q_{1_{j_{\gamma_0}}} - q_{1_{j_k}}) - \sum_{j=1}^{n_2}(q_{2_{j_{\gamma_0}}} - q_{2_{j_k}}) \tag{4.5}$$

Since the finger movements are small and γ_k is recomputed in each iteration, this approximation is accurate enough for the optimization of $\mathcal{Q}_r$.

4.3 Manipulation Strategies - Heuristic Methods

One simple approach to decide how to move the fingers to improve the quality indexes is to use intuitive movements, following the human behavior when manipulating an object with the fingertips. The following subsections introduce a general algorithm and manipulation strategies based on heuristics for the optimization of the quality indexes defined in Section 4.2.

4.3.1 General Manipulation Algorithm

The general algorithm introduced below could be used with any of the manipulation strategies that will be presented in the following subsections. Indexes k and $k+1$ are used to denote the current and next iteration, respectively. The manipulation is a reactive process composed of the five following steps:

1) Obtaining of the current state of the grasp.
2) Adjustment of the distance between contact points.
3) Computation of the new hand configuration.
4) Verification of termination conditions.
5) Movement of the fingers.

The first step is the computation of the variables that determine the current state of the grasp:

- The absolute position of the current contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, which are obtained using the tactile information provided by the tactile sensors to determine the contact points on the fingertips and the hand kinematics to determine the absolute positions of these contact points.

- The magnitude of the grasping force F_k, which is obtained as the average of the contact forces F_{1_k} and F_{2_k} measured on each fingertip. Although, in the case of two fingers, F_{1_k} and F_{2_k} should have the same magnitude and opposite direction, the use of the average of both measured contact forces minimizes potential measurement errors, thus,

$$F_k = \frac{F_{1_k} + F_{2_k}}{2} \tag{4.6}$$

- The Euclidean distance d_k between the contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, which is given by

$$d_k = |\overline{\mathbf{C}_{1_k}\mathbf{C}_{2_k}}| = \sqrt{(C_{1_{kx}} - C_{2_{kx}})^2 + (C_{1_{ky}} - C_{2_{ky}})^2} \tag{4.7}$$

The second step is the adjustment of distance between contact points to try to keep the force applied on the object around a desired value F_d. Since the shape of the object is unknown, any movement of the fingers may alter the current contact force F_k, allowing potential damage of the object or the hand if it increases or allowing a potential fall of the object if it decreases. The error in the contact force e_{f_k} is defined as

$$e_{f_k} = F_k - F_d \tag{4.8}$$

In order to reduce e_{f_k}, the distance d_k is adjusted in each iteration by adding an adjusting factor Δd,

$$d_{k+1} = d_k + \Delta d \tag{4.9}$$

with

$$\Delta d = \begin{cases} \mathcal{F}_1(e_{f_k}) & \text{if } e_{f_k} \leq 0 \\ \mathcal{F}_2(e_{f_k}) & \text{if } e_{f_k} > 0 \end{cases} \tag{4.10}$$

where $\mathcal{F}_1(e_{f_k})$ and $\mathcal{F}_2(e_{f_k})$ are user defined functions.

The third step is the computation of target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$, according to a desired manipulation strategy. Moving the fingers such that the contact points are moved towards the target contact points must improve the quality index associated with the desired manipulation strategy. The procedures to compute the target contact points are detailed in the following subsections for manipulation strategies with different goals.

The fourth step is the verification of certain termination conditions. The iterative manipulation procedure is applied until any of the following four termination conditions is activated, two of them associated with the quality index and the other two with the motion constraints:

- The quality index reaches the optimal value.
- The current value of the quality index is not improved after a predetermined number of iterations. Note that the index may not be improved monotonically, but it may become worst during a few iterations or oscillate alternating small improvements and worsening.
- The expected grasp at the computed target contact points does not satisfy the friction constraints.
- The computed target contact points do not belong to the workspace of the fingers. This condition is activated when the computed target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ are not reachable by the fingers, i.e. $\mathbf{Q}_{k+1} = \{\mathbf{q}_{1_{k+1}}, \mathbf{q}_{2_{k+1}}\}$ does not lie within the hand workspace.

The fifth and last step is the execution of the movements of the fingers. When none of the termination conditions is activated, the hand is moved towards $\mathbf{Q}_{k+1}$ to make the fingers try to reach the desired target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$. After the finger movements a new manipulation iteration begins.

The heuristic procedure proposed to manipulate an object using tactile feedback is summarized in Algorithm 4.1.

The objective of the manipulation strategies introduced below is the generation of the target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ that allow the optimization of the hand configuration, the grasp quality and the object orientation, respectively.

4.3.2 Optimizing the Hand Configuration

In this manipulation strategy, one finger is moved independently according to the desired goal and the other finger adapts its movement to follow the first finger and avoid the object fall. The

Algorithm 4.1 Manipulation with tactile feedback - Heuristic methods

```
Require: $F_d$, the Manipulation Strategy (MS)
  procedure REACTIVEMANIPULATION
      stop←false
      while stop≠true do
          \\ Computation of current state of the grasp
          Compute $\mathbf{C}_i$ using Forward Kinematics
          Compute $F_k$ using Eq. (4.6)
          \\ Adjustment of the distance between contact points
          Compute $d_{k+1}$ using Eq. (4.9)
          \\ Computation of target contact points using a manipulation strategy (MS)
          Compute $\mathbf{C}_{i_{k+1}}$ according to MS
          \\ Computation of new hand configuration
          Compute $\boldsymbol{Q}_{k+1}$ from $C_{i_{k+1}}$ using Inverse Kinematics
          Compute the quality index $\mathcal{Q}_{MS}$
          \\ Verification of termination conditions
          if a termination condition is not satisfied then
              \\ Finger movement
              Mover $f_i$ to $\mathbf{Q}_{k+1}$
          else
              stop←true
          end if
      end while
  end procedure
```

fingers alternate their roles in order to balance the hand movements, in this way both fingers advance to the goal configuration.

The configuration of the independent finger is computed varying each joint by a small value Δq, chosen small enough to ensure small changes in the position of the contact point. Assuming that, in the k-th iteration, the independent finger is f_i, the next joint values $q_{i_{j_{k+1}}}$ are given by,

$$q_{i_{j_{k+1}}} = q_{i_{j_k}} + \operatorname{sgn}(q_{i_{j_0}} - q_{i_{j_k}})\Delta q \tag{4.11}$$

where the function $\operatorname{sgn}(x)$, introduced to move the joints toward the desired configuration, is given by,

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{4.12}$$

The configuration of the follower finger is computed under the hypothesis that it is moved over a virtual circular path whose diameter is given by the distance d_{k+1} resulting from Eq. (4.9), as shown in Figure 4.3. Given the coordinates of the independent finger (e.g. $\mathbf{C}_{1_{k+1}}$ for the first

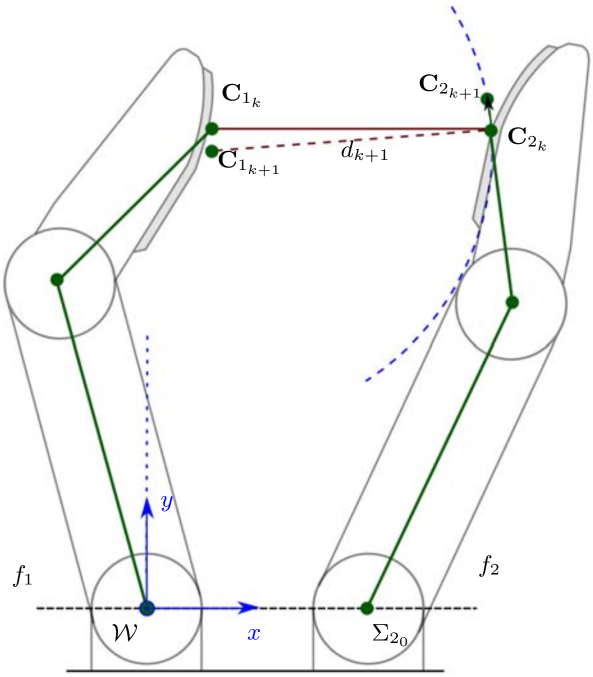


Figure 4.3: Two-finger model for the optimization of the hand configuration.

iteration), the coordinates $\mathbf{C}_{2_{k+1}}$ of the contact point of the follower finger are compute as,

$$C_{2_{y_{k+1}}} = C_{1_{y_{k+1}}} + \left(\tan(\rho) \sqrt{\frac{{d_{k+1}}^2}{\tan(\rho)^2 + 1}} \right) \tag{4.13}$$

$$C_{2_{x_{k+1}}} = C_{1_{x_{k+1}}} + \sqrt{{d_{k+1}}^2 - (C_{2_{y_{k+1}}} - C_{1_{y_{k+1}}})^2} \tag{4.14}$$

where the angle ρ is computed as,

$$\rho = \tan^{-1} \left(\frac{C_{2_{y_k}} - C_{1_{y_{k+1}}}}{C_{2_{x_k}} - C_{1_{x_{k+1}}}} \right) \tag{4.15}$$

This means that the contact point $\mathbf{C}_{2_{k+1}}$ is computed on a circumference of diameter d_{k+1} centered at the contact point $\mathbf{C}_{1_{k+1}}$. The same procedure is applied when f_2 is the independent finger and f_1 the follower.

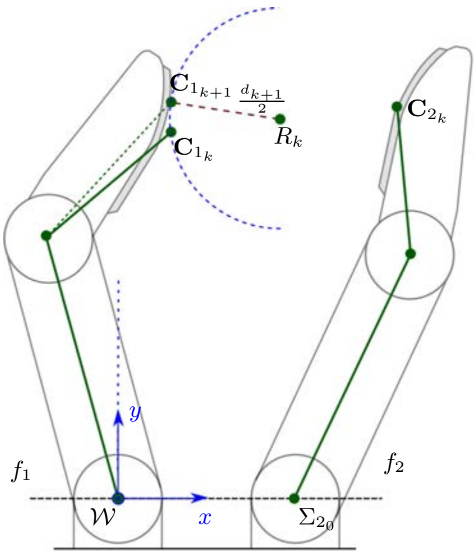


Figure 4.4: Two-finger model used for the optimization of the grasp quality.

4.3.3 Optimizing the Grasp Quality

In this manipulation strategy, we consider the hypothesis that the shape of the object is locally circular, and therefore, it is possible to compute a proper movement over the object surface to improve the grasp quality. The right direction of the movement depends on the computed angle β_i for each finger f_i. The fingers have again different roles (independent and follower) and iteratively change them. The new configuration for the independent finger is computed moving the contact point over the hypothetical circular shape of the object, as a displacement over an arc of circumference $(d_{k+1}/2)\Delta\phi$ as shown in Figure 4.4.

The angle variation $\Delta\phi$ can be positive or negative to perform displacements of the contact point in both directions with respect to the fingertip. The sign of $\Delta\phi$ have to be chosen equal to the sign of β_i. The next coordinates for the independent finger (e.g. $\mathbf{C}_{1_{k+1}}$, assuming that the independent finger is f_1) are computed as,

$$C_{1_{x_{k+1}}} = R_{x_k} - \frac{d_{k+1}}{2}\cos(\gamma_k + \Delta\phi) \tag{4.16}$$

$$C_{1_{y_{k+1}}} = R_{y_k} - \frac{d_{k+1}}{2}\sin(\gamma_k + \Delta\phi) \tag{4.17}$$

where the point R_k is the middle point of the segment between the contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$

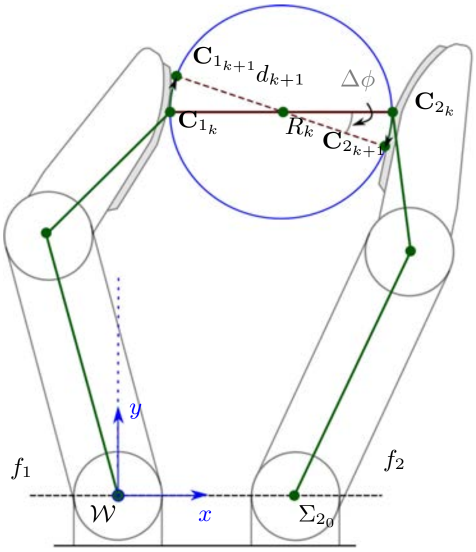


Figure 4.5: Two fingers model used for the optimization of the object orientation.

given by,

$$R_{x_k} = \frac{C_{2_{x_k}} - C_{1_{x_k}}}{2} + C_{1_{x_k}} \tag{4.18}$$

$$R_{y_k} = \frac{C_{2_{y_k}} - C_{1_{y_k}}}{2} + C_{1_{y_k}} \tag{4.19}$$

Besides, in equations (4.16) and (4.17), $\Delta\phi$ is chosen small enough to assure small movements of the object on each manipulation step. The next coordinates of the follower finger ($\mathbf{C}_{2_{k+1}}$) are computed using the same procedure as in the manipulation strategy for the optimization of the hand configuration, described in Subsection 4.3.2.

4.3.4 Optimizing the Object Orientation

In this manipulation strategy, we consider the hypothesis that the fingers are moved over a circular path whose diameter is given by the distance between the two current contact points, d_{k+1}, as shown in Figure 4.5.

In this case the fingers do not have different roles. The target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ are computed as,

$$C_{1_{x_{k+1}}} = R_{x_k} - (d_{k+1}/2)\cos(\gamma_k + \Delta\phi) \tag{4.20}$$

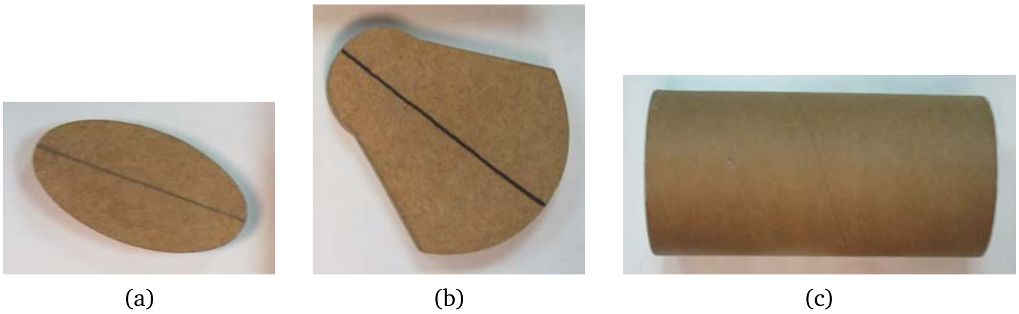


Figure 4.6: Three manipulated objects used in the experimentation with the heuristic methods. (a) Object 1: elliptical shape. (b) Object 2: two-curvature shape. (c) Object 3: cylinder.

$$C_{1_{y_{k+1}}} = R_{y_k} - (d_{k+1}/2)\sin(\gamma_k + \Delta\phi) \tag{4.21}$$

$$C_{2_{x_{k+1}}} = R_{x_k} + (d_{k+1}/2)\cos(\gamma_k + \Delta\phi) \tag{4.22}$$

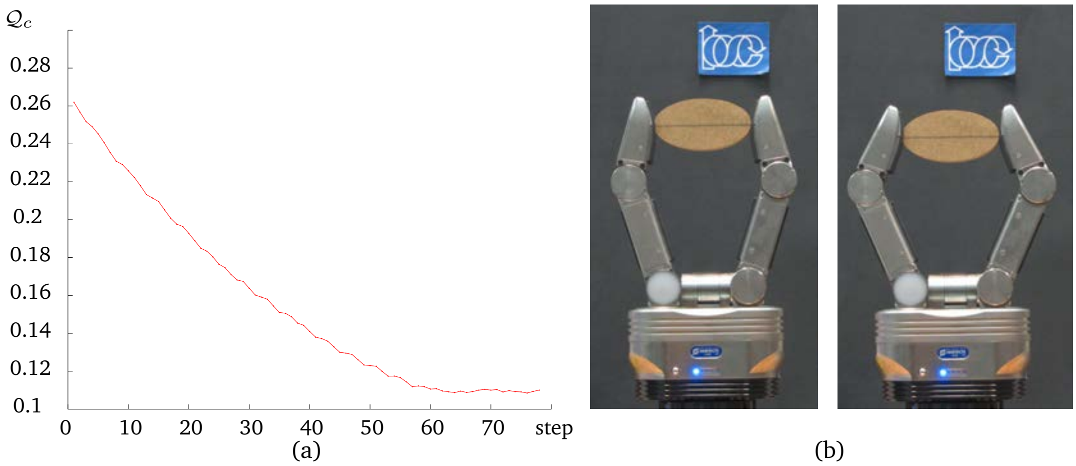
$$C_{2_{y_{k+1}}} = R_{y_k} + (d_{k+1}/2)\sin(\gamma_k + \Delta\phi) \tag{4.23}$$

where the point R_k, the center of the circular path, is computed using Eq. (4.18) and Eq. (4.19), and $\Delta\phi$ is chosen positive to turn the object clockwise or negative to turn the object counterclockwise. $\Delta\phi$ is chosen small enough to assure small movements of the object on each manipulation step.

4.3.5 Experimental Validation

The described manipulation strategies have been fully implemented using C++ for the manipulation of unknown objects with the SDH2 hand introduced in Chapter 3. Figure 4.6 shows three objects used in the real experimentation. Each object is held between the two coupled fingers of the SDH2, then, the fingers are closed until the contact forces reach a desired value $F_d = 2$ N. The initial grasp is executed without a precise knowledge of the object position and therefore, the initial contact points are unknown and the initial grasp configuration changes at each execution of the experiment.

The material of the sensor pads is rubber and the material of the objects is wood or cardboard, thus, we consider a worst case friction coefficient $\mu = 0.4$, which is lower than the friction coefficient between rubber and wood $\mu = 0.7$, and rubber and cardboard $\mu = 0.5$ (Kutz 2005).



(a) (b)

Figure 4.7: Example of optimization of the hand configuration. (a) $\mathcal{Q}_c$ during the manipulation of Object 1. (b) Snapshots of the initial and final grasps.

One simple approach to define the functions $\mathcal{F}_1$ and $\mathcal{F}_2$, that adjust the distance between contact points d_k in Eq. (4.10), is to choose them as a constant value λ with the sign opposite to the sign of e_{f_k}, i.e. $\mathcal{F}_1 = \lambda$ and $\mathcal{F}_2 = -\lambda$, being λ an empirically determined value. In the experimentation presented below λ is set to 1 mm.

The joints of the SDH2 have an operation range from -90° to 90°, therefore the middle-range position is 0°, however when the joints are in the middle-range position the hand is in a singular configuration, thus, in this experimentation, each joint range has been constrained to be between -90° and 0° for the proximal joints and, between 0° and 90° for the distal joints. The joint variation in each manipulation step was set to $\Delta\theta = 0.5°$.

In the example shown in Figure 4.7, Object 1 was manipulated with the hand configuration being optimized. Figure 4.7a shows the evolution of $\mathcal{Q}_c$ during the manipulation. Note that $\mathcal{Q}_c$ decreases in each manipulation step, which is an indicator of an improvement in the hand configuration. In this case, the manipulation had a duration of 10.31 s, and it ended because $\mathcal{Q}_c$ did not improve in the last 15 manipulation iterations. Figure 4.7b shows snapshots of the initial and final grasp configurations.

In the example shown in Figure 4.8, Object 2 was manipulated with the grasp configuration being optimized. Figure 4.8a shows the evolution of $\mathcal{Q}_g$ during the manipulation. The parameter to vary the position of the contact points was set to $\Delta\phi = 0.5°$. The manipulation ended when the module of β_i was below a threshold of 0.1, and this happened at 10.74 s. Figure 4.8b shows snapshots of the initial and final grasp configurations.

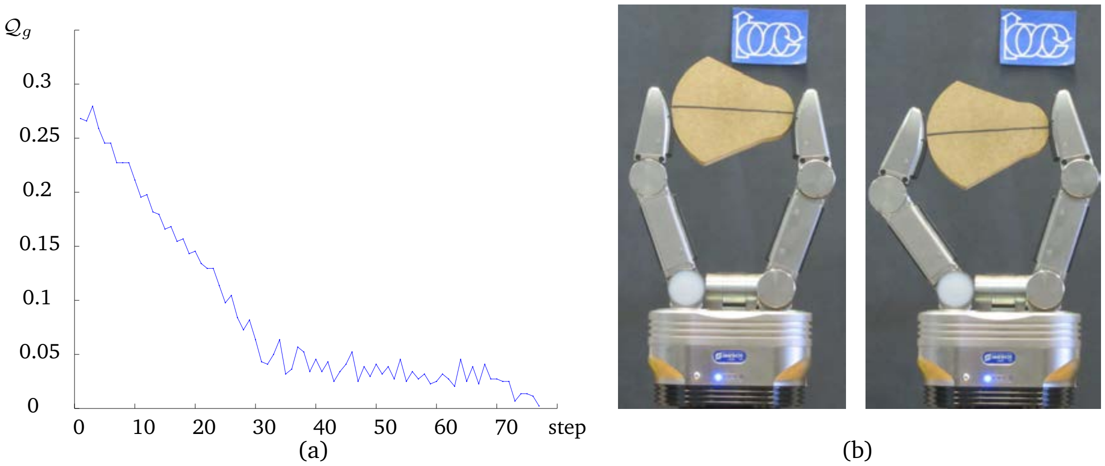


Figure 4.8: Example of optimization of the grasp configuration. (a) $\mathcal{Q}_g$ during the manipulation of Object 2. (b) Snapshots of the initial and final grasps.

In the example shown in Figure 4.9, Object 3 was rotated as much as possible in both senses. Figure 4.9a and 4.9b show the change in the object orientation with respect to the initial grasp, when the object was rotated clockwise and counterclockwise, respectively. The manipulation took 5.32 s in the counterclockwise case and 5.53 s in the clockwise. The manipulation ended when a finger reaches its workspace limit in both senses. Figure 4.9c shows snapshots of the initial grasp configuration, and the final grasp configurations when the object was rotated as much as possible counterclockwise and clockwise.

In the example shown in Figure 4.10, the goal is the improvement of the hand and grasp qualities, using for it the corresponding individual strategies, one after the other. First, the object was manipulated until the quality index $\mathcal{Q}_c$ related to the hand configuration was not improved anymore. Then, the motion strategy to improve the index $\mathcal{Q}_g$ related to the grasp quality was applied, allowing a variation of $\mathcal{Q}_c$ within a given threshold (i.e. it can worsen a limited amount). Figure 4.10 shows the obtained quality indexes $\mathcal{Q}_c$ and $\mathcal{Q}_g$ when the Object 3 was manipulated. Even when both quality indexes are shown during the whole manipulation process, until iteration 93 only $\mathcal{Q}_c$ was considered as an optimization index, using the corresponding motion strategy. At this point, since $\mathcal{Q}_c$ has not improved after 15 consecutive iterations, the optimization strategy was changed to improve $\mathcal{Q}_g$, but now checking that $\mathcal{Q}_c$ remains below the given threshold of 0.1 over the minimum reached value. The manipulation ended when $\mathcal{Q}_c$ reached the threshold value, with $\mathcal{Q}_g$ being less than 0.1, this happened after 28.81 s.

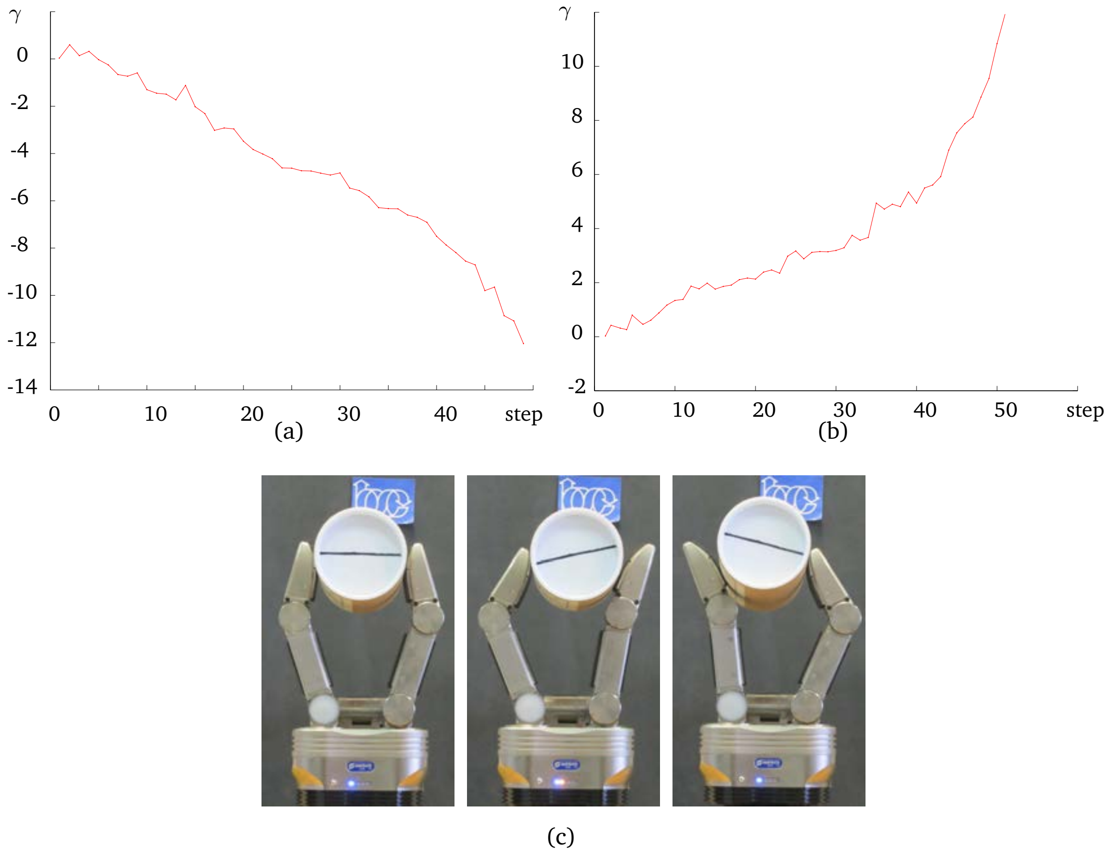


Figure 4.9: Example of optimization of the object orientation. (a) Orientation during the manipulation of Object 3 when it was rotated clockwise. (b) Orientation during the manipulation of Object 3 when it was rotated counterclockwise. (c) Snapshots of the initial grasp, and the final grasps when the object was rotated counterclockwise and clockwise.

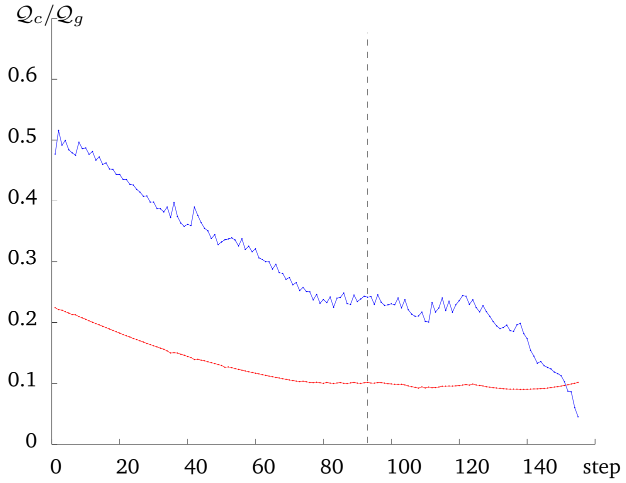


Figure 4.10: Example of the improvement of the hand and grasp qualities. $\mathcal{Q}_c$ (in Red) and $\mathcal{Q}_g$ (in Blue) resulting from applying the strategies to improve $\mathcal{Q}_c$ until iteration 93, when it has not improved anymore during 15 iterations, then the strategy to improve $\mathcal{Q}_g$ was applied while $\mathcal{Q}_c$ remains within a given threshold.

4.4 Manipulation Strategies - Gradient-based Methods

Considering that the quality indexes are expressed as functions of the hand joint values, a procedure to optimize the grasp is the determination of the gradient of these functions and make the hand movements to follow it. As a result, with relatively simple geometrical reasoning and assumptions, an unknown object can be manipulated keeping the grasping forces in a desired range and preventing the object from falling despite uncertainty.

In the same way as with the heuristics methods in Section 4.3, the following subsections introduce a general algorithm and the gradient-based strategies for the optimization of the quality indexes defined in Section 4.2.

4.4.1 General Manipulation Algorithm

In the case of gradient-based methods, the general algorithm has some differences from that already presented for the heuristic methods, in this case the manipulation process is composed of the following five steps:

1) Obtaining of the current state of the grasp.
2) Computation of auxiliary points using a manipulation strategy.
3) Computation of the target hand configuration based on the auxiliary points.
4) Verification of termination conditions.
5) Movement of the fingers.

The first step is the computation of the relevant variables of the current state of the grasp, which are the same variables as in the heuristic methods and therefore, they are computed as it was described in Section 4.3, i.e. the following variables are computed:

- The position of the current contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$ (obtained using tactile and kinematic information).
- The magnitude of the grasping force F_k (obtained with Eq. (4.6)).

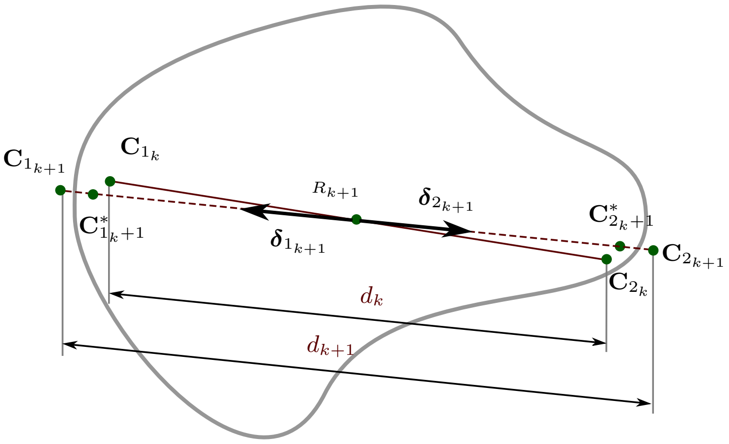


Figure 4.11: Example of the computation of $\mathbf{C}_{i_{k+1}}$ using $\mathbf{C}^*_{i_{k+1}}$, adjusting the distance d_k to d_{k+1} when the contact force F_k is larger than F_d.

- The Euclidean distance d_k between the contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$ (obtained with Eq. (4.7)).

The second step is the computation of two auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$, according to the desired manipulation strategy. Moving the fingers such that the contact points are moved towards the auxiliary points must improve the quality index associated with the desired manipulation strategy. The strategy to compute the auxiliary points for the optimization of each quality index is detailed in the following subsections.

The third step is the adjustment of the auxiliary points to obtain new target contact points, $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$, that try to keep the force applied to the object within a desired range. The force error e_{f_k} is reduced, in the same way as in the heuristic methods, adjusting the distance d_k in each iteration by adding an adjusting factor Δd, i.e.

$$d_{k+1} = d_k + \Delta d \tag{4.24}$$

with

$$\Delta d = \begin{cases} \mathcal{F}_1(e_{f_k}) & \text{if } e_{f_k} \leq 0 \\ \mathcal{F}_2(e_{f_k}) & \text{if } e_{f_k} > 0 \end{cases} \tag{4.25}$$

The positions of the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are adjusted along the line that they define in order to obtain the actual target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ at a distance d_{k+1} from each

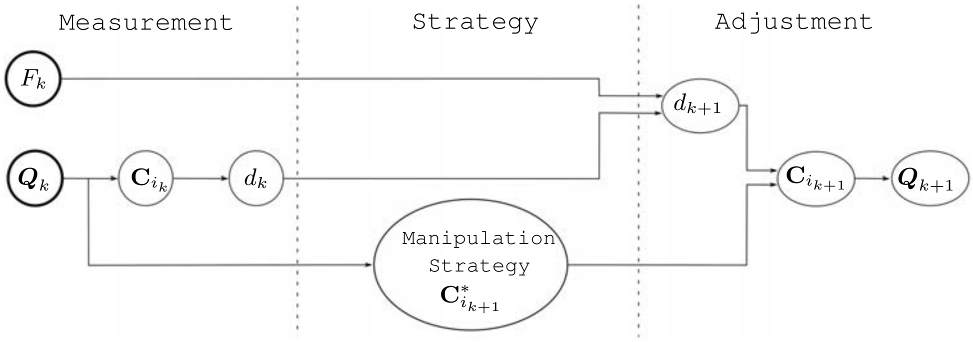


Figure 4.12: Relation between the measured variables, the role of the manipulation strategy, and the final adjustment to obtain the new hand configuration.

other. This is done as,

$$\mathbf{C}_{i_{k+1}} = R_{k+1} + \frac{d_{k+1}}{2}\boldsymbol{\delta}_{i_{k+1}}, i \in \{1, 2\} \tag{4.26}$$

where R_{k+1} is the middle point of the segment between $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ and $\boldsymbol{\delta}_{i_{k+1}}$ is the unitary vector from R_{k+1} to $\mathbf{C}^*_{i_{k+1}}$, as shown in Figure 4.11.

Finally, using the inverse kinematics of the fingers, from the points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ it is possible to obtain the corresponding hand configuration $\mathbf{Q}_{k+1} = \{\mathbf{q}_{1_{k+1}}, \mathbf{q}_{2_{k+1}}\}$. Figure 4.12 illustrates the relationship between the measured variables, the role played by the manipulation strategy in the computation of the auxiliary points $\mathbf{C}^*_{i_{k+1}}$, and the variables involved in the final adjustment to obtain the new hand configuration (with independence of the used manipulation strategy).

The fourth step is the verification of termination conditions. As in the Heuristic Methods, the iterative manipulation strategy is applied until any of the following four termination conditions is activated:

- The quality index reaches the optimal value.
- The current value of the quality index is not improved after a predetermined number of iterations.
- The expected grasp at the computed target contact points does not satisfy the friction constraints.
- The computed contact points do not belong to the workspace of the fingers.

The fifth and last step is the execution of the movements of the fingers. When none of the termination conditions is activated, the hand is moved towards $\mathbf{Q}_{k+1}$ to make the fingers try

Algorithm 4.2 Reactive manipulation algorithm

```
Require: $F_d$, tha Manipulation Strategy (MS)
  procedure REACTIVEMANIPULATION
    stop←false
    while stop≠true do
      Computation of current state of the grasp
      Compute $\mathbf{C}_i$ using Forward Kinematics
      Compute $F_k$ using Eq. (4.6)
      Computation of new contact points for a manipulation strategy (MS)
      Compute $\mathbf{C}^*_{i_{k+1}}$ according to MS
      Adjustment of the virtual contact points
      Compute $d_{k+1}$ using Eq. (4.9)
      Compute $\mathbf{C}_{i_{k+1}}$ using Eq. (4.26)
      Computation of new hand configuration
      Compute $\boldsymbol{Q}_{k+1}$ from $C_{i_{k+1}}$ using Inverse Kinematics
      Compute the quality index $\mathcal{Q}_{MS}$
      Verification of termination conditions
      if a termination condition is not satisfied then
        Finger movement
        Mover $f_i$ to $\mathbf{Q}_{k+1}$
      else
        stop←true
      end if
    end while
  end procedure
```

to reach the desired target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$. After the finger movements a new manipulation iteration begins.

Algorithm 4.2 shows the proposed reactive manipulation procedure using two fingers, which is general and valid for any of the gradient-based manipulation strategies.

The objective of the manipulation strategies introduced below is to generate the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ to allow the optimization of the hand configuration, the grasp quality and the object orientation, respectively. As it was shown in the scheme presented in Figure 4.12, the manipulation strategy generates $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ which will be used later to compute the next contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ to properly manipulate the object.

4.4.2 Optimizing the Hand Configuration

In this strategy, the goal configuration of the hand is known with independence of the object shape, thus, it is trivial to move the hand towards it, the key point is to do it allowing an adequate adjustment of the distance d_k between the contact points in each iteration to prevent

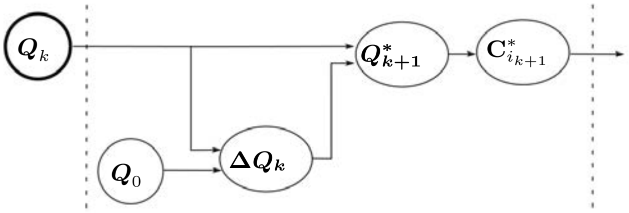


Figure 4.13: Variables involved in the optimization of the hand configuration.

the object from falling. Then, the hand configuration is updated in each iteration as

$$\mathbf{Q}^*_{k+1} = \mathbf{Q}_k + \mathbf{\Delta Q} \tag{4.27}$$

where

$$\mathbf{\Delta Q_k} = \eta(\mathbf{Q}_0 - \mathbf{Q}_k) \tag{4.28}$$

is a small enough vector pointing from the current configuration $\mathbf{Q}_k = \{\mathbf{q}_{1_k}, \mathbf{q}_{2_k}\}$ to $\mathbf{Q}_0$, i.e. η must be chosen to properly fix the advance of the hand configuration in each iteration. As a practical approach, when the angles are measured in degrees, $\mathbf{\Delta Q} \leq 1$ was found to work well, and this is achieved with

$$\eta = \frac{\tanh(||\mathbf{Q}_0 - \mathbf{Q}_k||)}{||\mathbf{Q}_0 - \mathbf{Q}_k||} \tag{4.29}$$

where $\tanh$ is used to bound η when the current configuration of the hand $\mathbf{Q}_k$ is far from $\mathbf{Q}_0$. From Eq. (4.28) and Eq. (4.29) results

$$\mathbf{\Delta Q_k} = \frac{\tanh(||\mathbf{Q}_0 - \mathbf{Q}_k||)}{||\mathbf{Q}_0 - \mathbf{Q}_k||}(\mathbf{Q}_0 - \mathbf{Q}_k) \tag{4.30}$$

Finally, from $\mathbf{Q}^*_{k+1}$ obtained in Eq. (4.27), it is straightforward to obtain the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ using the direct kinematics of the hand.

Figure 4.13 summarizes the relation between the variables involved in the computation of $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ for the optimization of the hand configuration (according to the general diagram shown in Figure 4.12).

4.4.3 Optimizing the Grasp Quality

In this strategy, using basic geometry and the information obtained from the tactile sensors and the finger kinematics, the angles β_i are computed as functions of the current contact points $\mathbf{C}_i$,

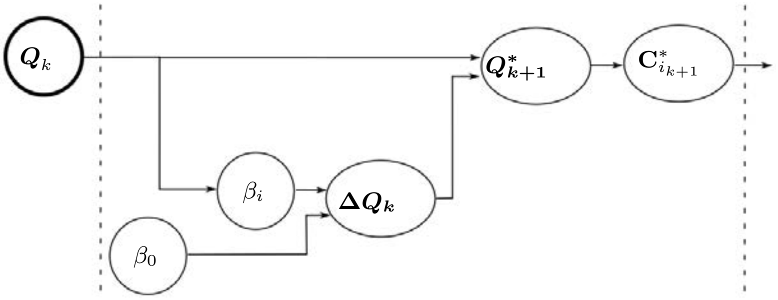


Figure 4.14: Variables involved in the optimization of the grasp quality.

the position of the origin of reference frame $\Sigma_{i_{n_i}}$, and the length r_i and the joint angle q_{i_v} of the virtual links at the fingertips as (all the variables are computed for the iteration k, thus, to improve legibility, index k has been removed in these equations),

$$\beta_1 = \cos^{-1}\left(\frac{-|\overline{\Sigma_{1_{n_i}}\mathbf{C}_2}|^2 + r_1^2 + |\overline{\mathbf{C}_1\mathbf{C}_2}|^2}{2r_1|\overline{\mathbf{C}_1\mathbf{C}_2}|}\right) + q_{1_v} - \pi \tag{4.31}$$

$$\beta_2 = \cos^{-1}\left(\frac{-|\overline{\Sigma_{2_{n_i}}\mathbf{C}_1}|^2 + r_2^2 + |\overline{\mathbf{C}_1\mathbf{C}_2}|^2}{2r_2|\overline{\mathbf{C}_1\mathbf{C}_2}|}\right) + q_{2_v} - \pi \tag{4.32}$$

Note that β_i depends on the current hand configuration $\mathbf{Q}_k$. Therefore, the gradient of $\beta_i\left(\mathbf{Q}_k\right)$ at the current hand configuration, $\boldsymbol{\nabla\beta}_i\left(\mathbf{Q}_k\right)$, is used to compute the next auxiliary configuration of the hand $\mathbf{Q}^*_{k+1}$ as

$$\mathbf{Q}^*_{k+1} = \mathbf{Q}_k + \mathbf{\Delta Q} \tag{4.33}$$

where $\mathbf{\Delta Q}$ is now given by

$$\mathbf{\Delta Q} = \frac{1}{2}\tanh(\beta_1)\frac{\boldsymbol{\nabla\beta}_1}{||\boldsymbol{\nabla\beta}_1||} + \frac{1}{2}\tanh(\beta_2)\frac{\boldsymbol{\nabla\beta}_2}{||\boldsymbol{\nabla\beta}_2||} \tag{4.34}$$

Finally, as in the previous strategy, from $\mathbf{Q}^*_{k+1}$ obtained in Eq. (4.33), it is straightforward to obtain the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ using the direct kinematics of the hand.

Figure 4.14 summarizes the relation between the variables involved in the computation of $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ for the optimization of the grasp quality (according to the general diagram shown in Figure 4.12).

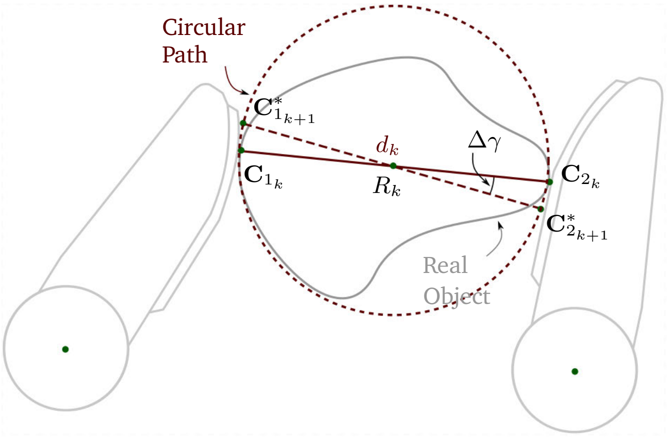


Figure 4.15: Movements used for the optimization of the object orientation. $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed over a circular path with diameter d_k centered at R_k.

4.4.4 Optimizing the Object Orientation

In this strategy, the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed considering that the fingers are moved to produce the displacement of the current contact points on the sensor pads along a circular path given as (see Figure 4.15),

$$C^*_{1_{k+1_x}} = R_{k_x} - \frac{d_k}{2}\cos(\gamma_{k+1}) \tag{4.35}$$

$$C^*_{1_{k+1_y}} = R_{k_y} - \frac{d_k}{2}\sin(\gamma_{k+1}) \tag{4.36}$$

$$C^*_{2_{k+1_x}} = R_{k_x} + \frac{d_k}{2}\cos(\gamma_{k+1}) \tag{4.37}$$

$$C^*_{2_{k+1_y}} = R_{k_y} + \frac{d_k}{2}\sin(\gamma_{k+1}) \tag{4.38}$$

i.e. the new auxiliary positions are points on a circumference with diameter d_k centered at the middle point, R_k, between the points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, and

$$\gamma_{k+1} = \gamma_k + \tanh(\gamma_k)\Delta\gamma \tag{4.39}$$

with $\Delta\gamma$ chosen empirically and small enough to assure small movements of the object in each manipulation step.

Note that in this case it was not necessary to compute $\mathbf{Q}_{k+1}$ as an intermediate step to determine the auxiliary points $\mathbf{C}^*_{i_{k+1}}$. Instead, now $\mathbf{Q}_{k+1}$ can be deduced from $\mathbf{C}^*_{i_{k+1}}$ applying inverse kinematics. This is relevant since the direction of $\mathbf{\Delta Q} = \mathbf{Q}^*_{k+1} - \mathbf{Q}_k$ is necessary to combine

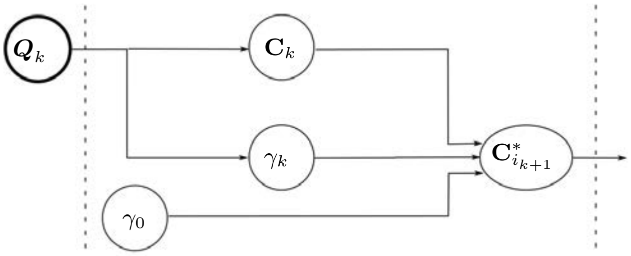


Figure 4.16: Variables involved in the optimization of the object orientation.

different manipulation strategies, as will be shown in Subsection 4.4.6.

Figure 4.16 summarizes the relation between the variables involved in the computation of $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ for the optimization of the object orientation (according to the general diagram shown in Figure 4.12).

4.4.5 Experimental Validation

The approach described above, including the general algorithm and the manipulation strategies, has been validated using the SDH2 described in Chapter 3. Some examples of experimental results are presented below to illustrate the performance of the approach.

In the following examples the fingers are blindly closed around an unknown object until the measured grasping force reaches a predefined desired value $F_d = 5$ N. This force value was chosen considering the range of the tactile sensors, the forces the hand can apply and that the manipulated objects were hard rigid bodies. The objects used for the experiments were selected looking for different object shapes (with small and large curvatures) and different object boundaries (smooth and irregular), so that the performance of the proposed approach can be illustrated under different conditions. The initial position of the object varies in each execution and is not known with precision, so the initial grasp configuration and the initial contact points are unknown a priori by the system. The friction coefficient considered in the calculations was $\mu = 0.4$ (friction cone angle of only $\alpha = 21.8°$), which is below the expected real physical value considering the object materials and that the fingertips have rubber surface. The functions $\mathcal{F}_1$ and $\mathcal{F}_2$ involved in the adjustment of the distance between contact points d_k in Eq. (4.10), are defined depending on the module of the force error e_{f_k}. In the following experiments we use $\mathcal{F}_1 = 2\lambda(e_{f_k} + e^2_{f_k})$ and $\mathcal{F}_2 = \lambda e_{f_k}$, with λ being a predefined constant. The reason for the difference between $\mathcal{F}_1$ and $\mathcal{F}_2$ is that a potential fall of the object ($F_k \to 0$) is considered more

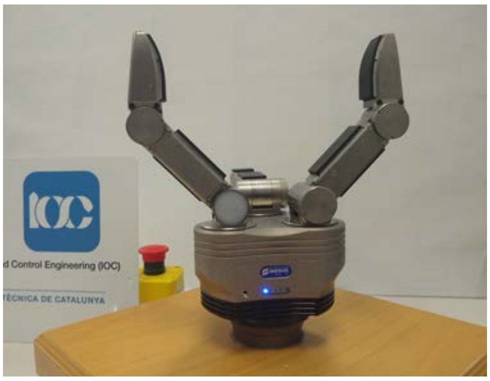
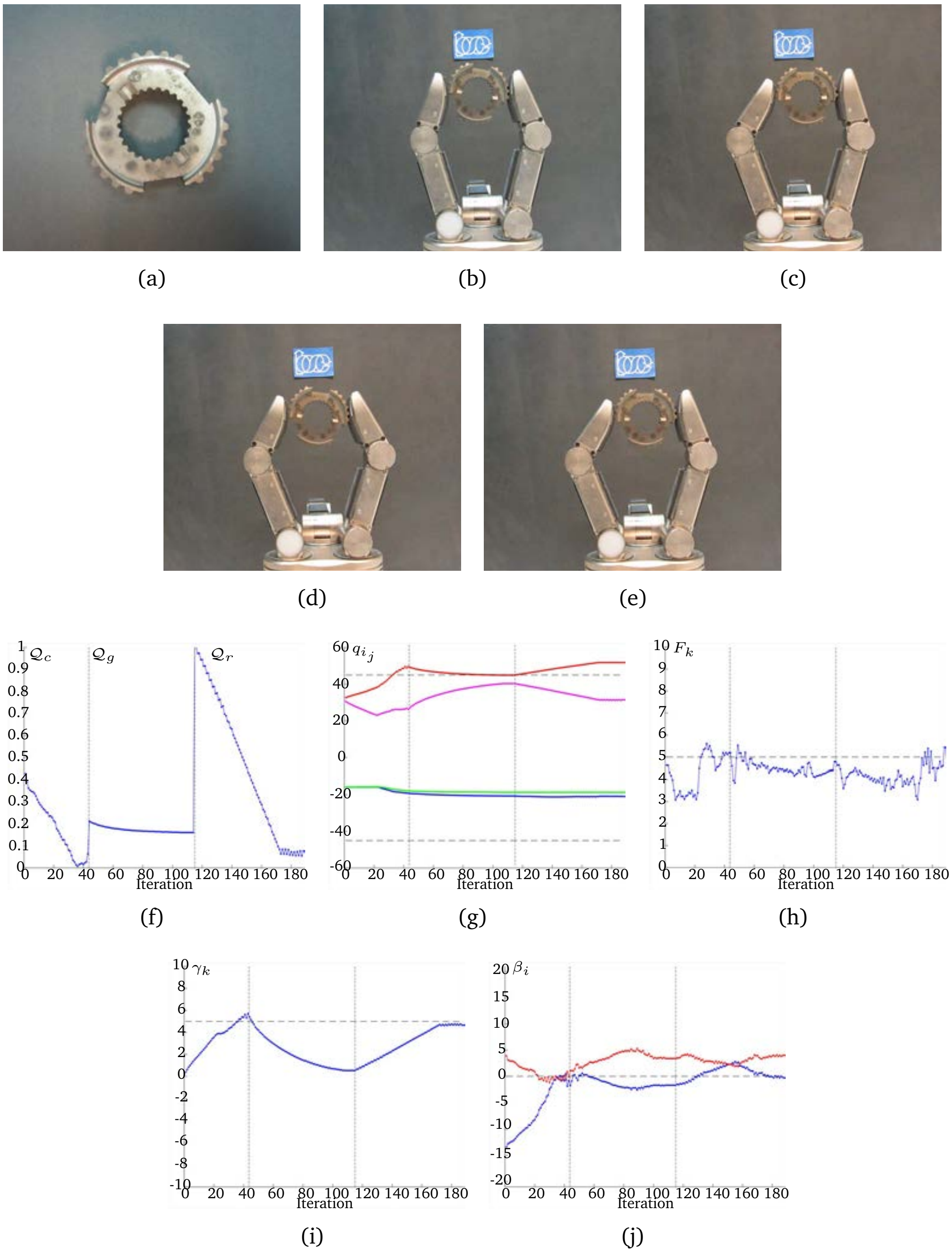


Figure 4.17: SDH2 in the desired hand configuration.

critical that a potential application of large grasping forces ($F_k \gg F_d$), and therefore, $\mathcal{F}_1$ has larger gain, specially for large $|e_{f_k}|$. The constant λ to adjust the distance between the contact points according to Eq. (4.47) was set to $\lambda = 0.25$ mm. Videos of the experimental executions can be found in http://goo.gl/ivFd0q.

In the examples shown in figures 4.18 to 4.21, four different objects are manipulated improving the three quality indexes sequentially, first the manipulation optimizes $\mathcal{Q}_c$, then, $\mathcal{Q}_g$ and finally $\mathcal{Q}_r$. For the improvement of $\mathcal{Q}_c$, $\mathbf{Q}_0 = \{-45, 45, -45, 45\}$ is considered as the desired hand configuration, shown in Figure 4.17. When $\mathcal{Q}_g$ is improved, the angles β_i are minimized according to the expected behavior of the manipulation strategy. Finally, for the improvement of $\mathcal{Q}_r$, the desired goal is an object rotation of 5° clockwise. In the sub-figures showing charting results, a vertical dotted line is depicted to highlight the iterations in which the optimization index changes. Particular details of each experiment are given in the caption of each figure.

In the example shown in Figure 4.22 the object was successively rotated clockwise and counterclockwise with desired orientations γ_d set to 5°, -5°, 10°, -10°, and 15°. The change of set-point was manually done once the system has activated a termination condition for the current set-point. In the first four cases the termination condition was the arrival of $\mathcal{Q}_r$ to the expected value according to the system internal measurements, i.e. $\gamma_k \approx \gamma_d$ (see Figure 4.22c), and in the last case the manipulation ended because the expected next value of the angle β_1 exceeded the friction cone limit before arriving to $\gamma_d = 15°$ (see the evolution of β_1 in Figure 4.22f), meaning that there was a risk of sliding and the object could flip away from the hand. The real orientations of the object when the terminal conditions were activated, measured by an external vision system, are given in Figure 4.22c between parenthesis below the corresponding values obtained from internal measurements. $\Delta\gamma$ was set to 0.25°.



(a) (b) (c)

(d) (e)

(f) (g) (h)

(i) (j)

Figure 4.18: (a) Manipulated object. (b) Initial grasp. (c) Hand configuration after optimizing $\mathcal{Q}_c$. (d) Hand configuration after optimizing $\mathcal{Q}_g$. (e) Hand configuration after optimizing $\mathcal{Q}_r$. (f)) Evolution of the quality indexes. The manipulations improving $\mathcal{Q}_c$, $\mathcal{Q}_g$ and $\mathcal{Q}_r$ ended after 3.864 s and 43 iterations, 4.486 s and 70 iterations and 9.083 s and 73 iterations, respectively. (g) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, q_{2_2} in magenta. (h) Average force F_k in Newtons, the dashed line indicates F_d. (i) Evolution of the object orientation in degrees. (j) Angles β_i in degrees, β_1 in blue and β_2 in red (the dashed line indicates the optimal value of β_i). Note that the non-smooth and toothed surface of the manipulated object produces more than one contact region on each fingertip without generating any manipulation problem.

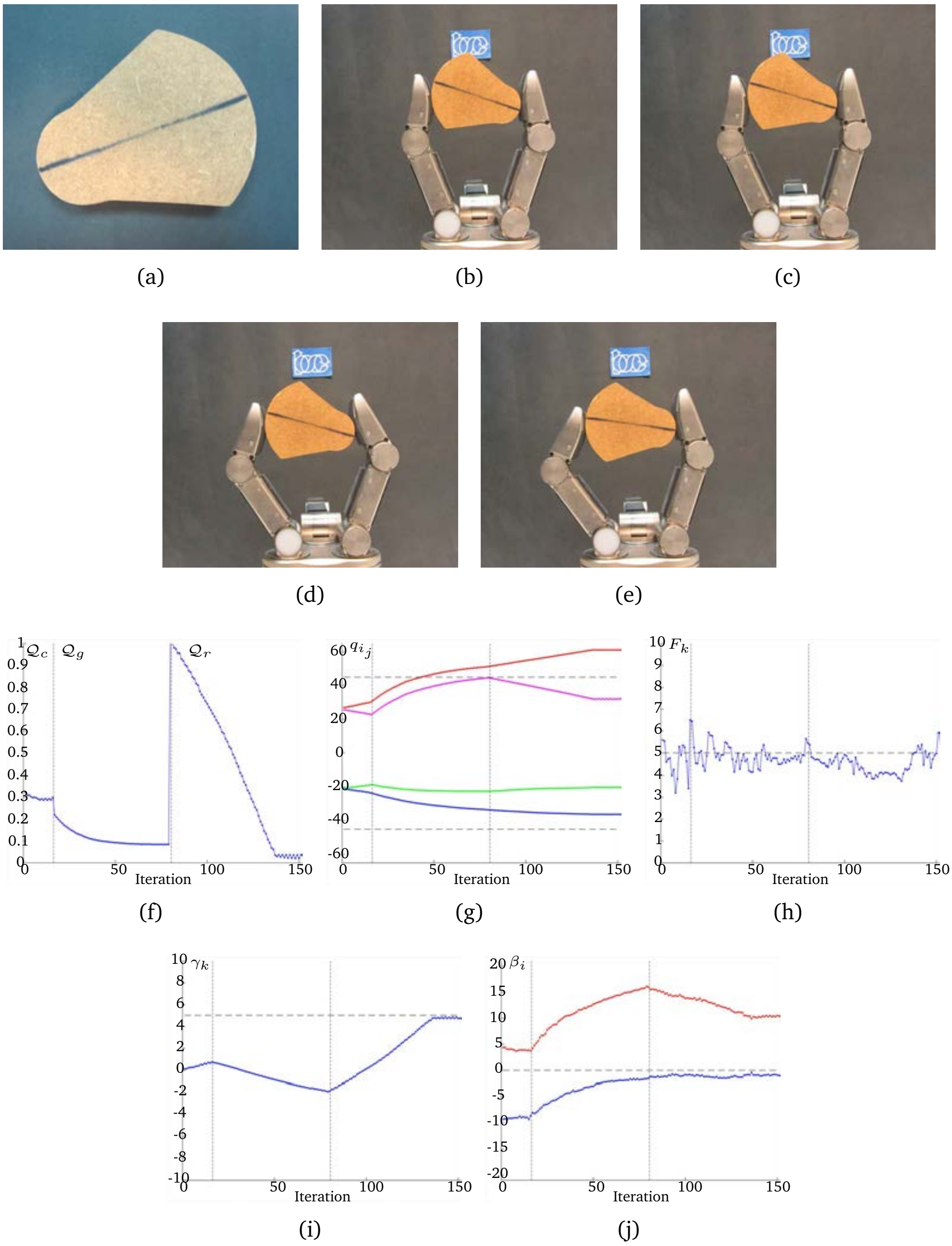


Figure 4.19: (a) Manipulated object. (b) Initial grasp. (c) Hand configuration after optimizing $\mathcal{Q}_c$. (d) Hand configuration after optimizing $\mathcal{Q}_g$. (e) Hand configuration after optimizing $\mathcal{Q}_r$. (f) Evolution of the quality indexes. The manipulations improving $\mathcal{Q}_c$, $\mathcal{Q}_g$ and $\mathcal{Q}_r$ ended after 1.171 s and 16 iterations, 4.687 s and 62 iterations and 9.709 s and 71 iterations, respectively. (g) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, q_{2_2} in magenta. (h) Average force F_k in Newtons, the dashed line indicates F_d. (i) Evolution of the object orientation in degrees. (j) Angles β_i in degrees, β_1 in blue and β_2 in red (the dashed line indicates the optimal value of β_i).

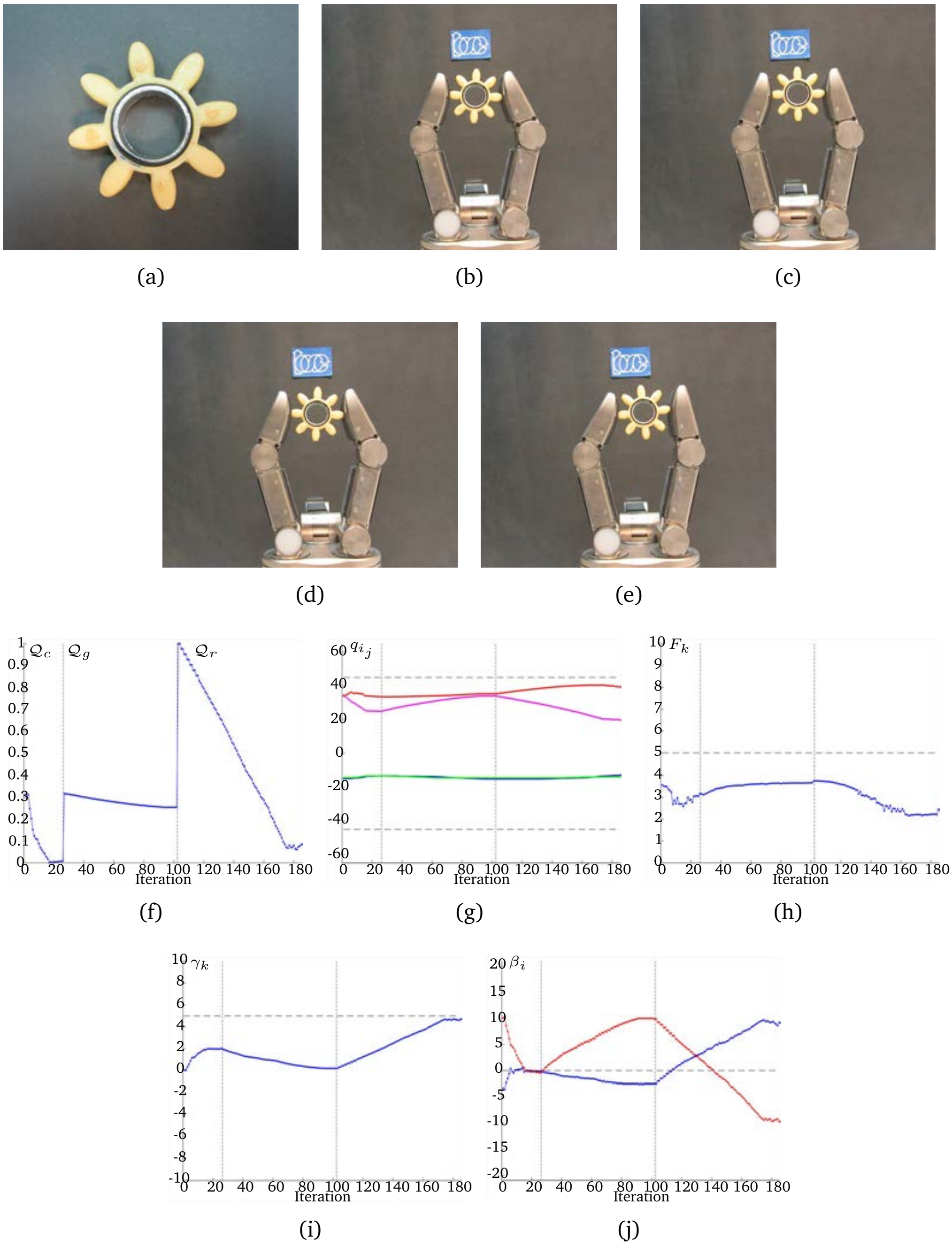


Figure 4.20: (a) Manipulated object. (b) Initial grasp. (c) Hand configuration after optimizing $\mathcal{Q}_c$. (d) Hand configuration after optimizing $\mathcal{Q}_g$. (e) Hand configuration after optimizing $\mathcal{Q}_r$. (f) Evolution of the quality indexes. The manipulations improving $\mathcal{Q}_c$, $\mathcal{Q}_g$ and $\mathcal{Q}_r$ ended after 2.439 s and 26 iterations, 4.627 s and 75 iterations and 8.779 s and 82 iterations, respectively. (g) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, q_{2_2} in magenta. (h) Average force F_k in Newtons, the dashed line indicates F_d. (i) Evolution of the object orientation in degrees. (j) Angles β_i in degrees, β_1 in blue and β_2 in red (the dashed line indicates the optimal value of β_i). Note that, due to the shape of the manipulated object, the contact is produced on a limited region of the sensor and therefore the measured force F_k cannot reach the desired force F_d.

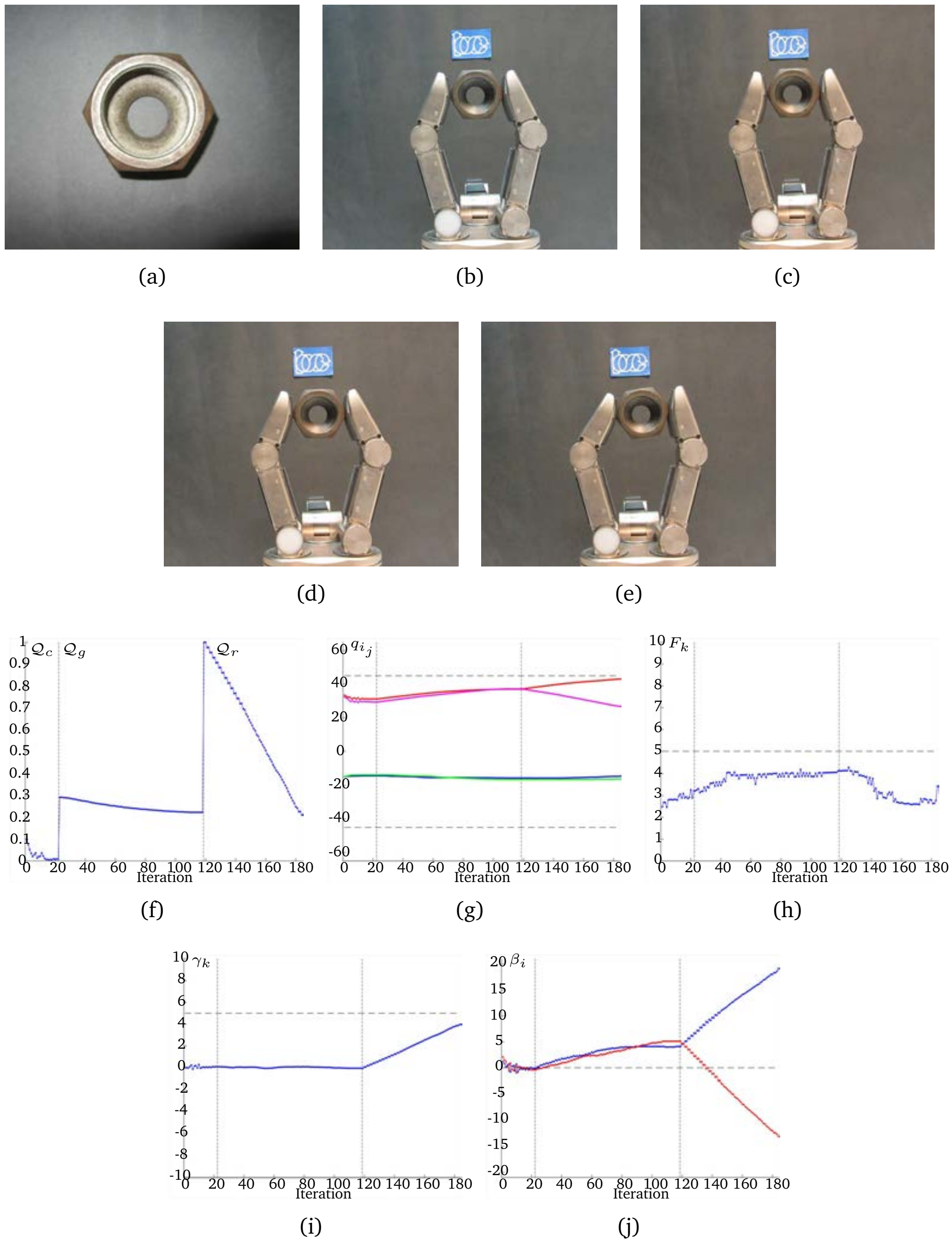
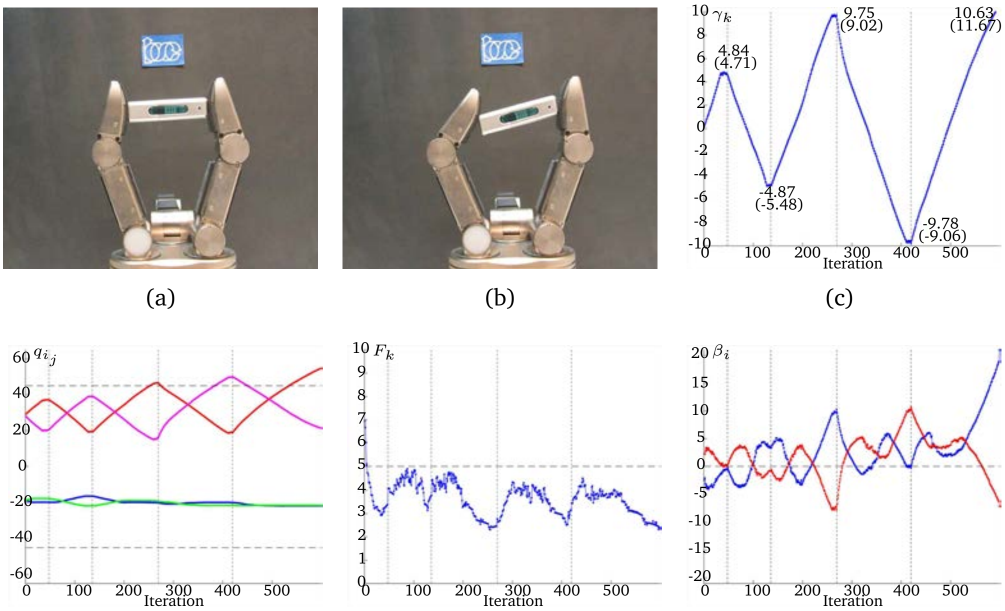


Figure 4.21: (a) Manipulated object. (b) Initial grasp. (c) Hand configuration after optimizing $\mathcal{Q}_c$. (d) Hand configuration after optimizing $\mathcal{Q}_g$. (e) Hand configuration after optimizing $\mathcal{Q}_r$. (f) Evolution of the quality indexes. The manipulations improving $\mathcal{Q}_c$, $\mathcal{Q}_g$ and $\mathcal{Q}_r$ ended after 2.122 s and 22 iterations, 5.558 s and 95 iterations and 5.347 s and 65 iterations, respectively. (g) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, q_{2_2} in magenta. (h) Average force F_k in Newtons, the dashed line indicates F_d. (i) Evolution of the object orientation in degrees. (j) Angles β_i in degrees, β_1 in blue and β_2 in red (the dashed line indicates the optimal value of β_i). Note that as in Example 3 the contact region is quite small due the object shape and therefore the force F_k cannot reach the desired force. The manipulation ended without reaching the desired object orientation because the friction constraints were not satisfied and the object could slip out of the hand.



(a) (b) (c)

(d) (e) (f)

Figure 4.22: (a) Initial grasp. (b) Final grasp. (c) Evolution of the object orientation γ_k with sequential setpoints 5, -5, 10, -10 and 15 degrees. (d) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, q_{2_2} in magenta. (e) Average force F_k in Newtons, the dashed line indicates F_d. (f) Angles β_i in degrees, β_1 in blue and β_2 in red, the dashed line indicates the optimal value of β_i.

4.4.6 Combining Manipulation Strategies

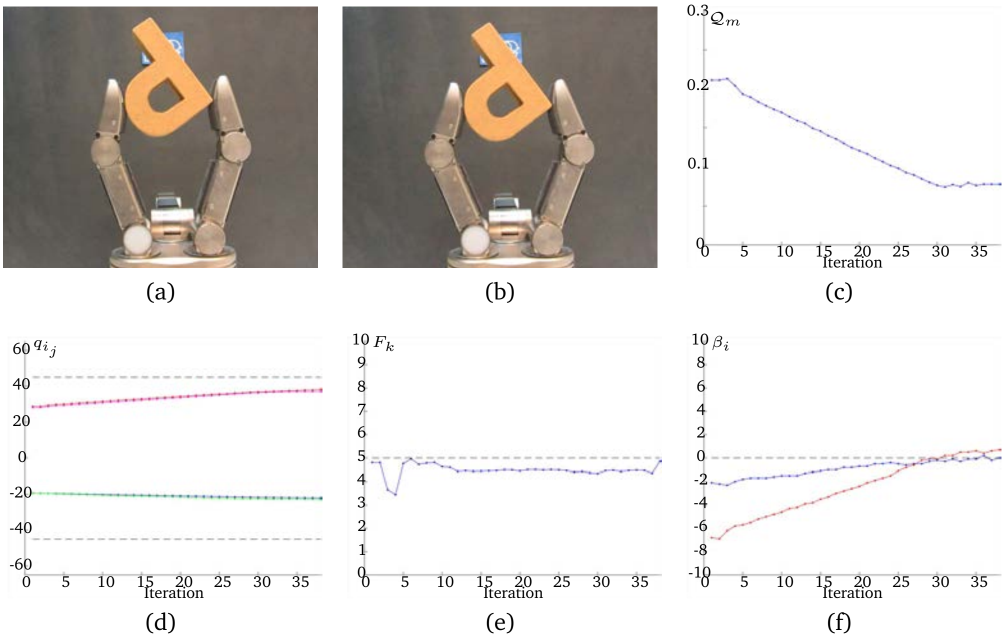
The approach allows the combination of two or more manipulation strategies, for this purpose a combined quality index $\mathcal{Q}_m$ is computed as a lineal combination of the quality indexes associated to the combined manipulation strategies, i.e.

$$\mathcal{Q}_m = \sum_j \omega_j \mathcal{Q}_j \tag{4.40}$$

where $\omega_j > 0$ are arbitrarily defined weighting coefficients.

When two or more manipulation strategies are combined, the target configuration of the hand $\mathbf{Q}_{k+1}$ is computed as the current hand configuration plus a lineal combination of the incremental movements $\mathbf{\Delta Q}_j$ obtained individually by each manipulation strategy j, i.e.

$$\mathbf{Q}^*_{k+1} = \mathbf{Q}_k + \sum_j \omega_j \mathbf{\Delta Q}_j \tag{4.41}$$



(a) (b) (c)

(d) (e) (f)

Figure 4.23: (a) Initial grasp. (b) Final grasp. (c) Evolution of the index $\mathcal{Q}_m = 0.5\mathcal{Q}_c + 0.5\mathcal{Q}_g$. (d) Evolution of the joints values in degrees, q_{1_1} in blue, q_{1_2} in red, q_{2_1} in green, $q_{2}2$ in magenta. (e) Average force F_k in Newtons, the dashed line indicates F_d. (f) Angles β_i in degrees, β_1 in blue and β_2 in red (the dashed line indicates the optimal value of β_i).

with $\omega_j > 0$ satisfying $\sum_j \omega_j = 1$ to avoid unexpected large movements. The coefficients ω_j can be arbitrarily adjusted to give different weights to each individual strategy. It must be remarked that the final movement determined to optimize the combined index does not imply the individual optimization of all the involved individual indexes.

Then, from $\mathbf{Q}^*_{k+1}$ it is straightforward to obtain the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ using the direct kinematics of the hand. Once the auxiliary points are computed, the same procedure as in the other manipulation strategies is applied, i.e. the computation of the target hand configuration based on the auxiliary points, the verification of the termination conditions, and finally the movement of the fingers.

In the example shown in Figure 4.23 two manipulation strategies were combined, optimizing the hand configuration and the grasp quality simultaneously. The strategies were combined using $\omega_1 = \omega_2 = 0.5$ in Eq. (4.40) and Eq. (4.41), i.e. $\mathcal{Q}_m = 0.5\mathcal{Q}_c + 0.5\mathcal{Q}_g$. In this example β_i tends to zero according to the optimization of the grasp quality while the joints tend to their desired specific positions. The manipulation ended after 2.85 s and 38 iterations because $\mathcal{Q}_m$ did not improve the current optimal value after 10 iterations. Note that the optimization of $\mathcal{Q}_m$ does not imply the optimization of $\mathcal{Q}_c$ and $\mathcal{Q}_g$.

4.5 Three Finger Extension using Virtual Contact Points

A natural extension of the manipulation strategies introduced in the previous sections is to increase the number of fingers involved in the manipulation process. For this, a robotic hand with a higher number of DOF per finger is considered for the manipulation. Inspired by the typical movements that a human being does to rotated an object, three fingers of an anthropomorphic hand are used to rotate an unknown object forward and backward. This new manipulation strategy introduces a simple geometric procedure based on the commanded positions of the fingertips, which are used to define a set of virtual contact points without caring about the positions of the real contact points between the fingertips and the object.

Considering that the finger joints of the hand work under position control, the commanded hand configurations must be such that the commanded positions of the fingertips lie “inside” the object in order to apply a force on the object surface. It must be noted that if the fingertips are positioned exactly on the surface of the object, they will not produce grasping forces on it. From now on, we will refer to the commanded fingertip positions located “inside” the object as “virtual contact points”, since they are not physically reachable. Furthermore, the magnitude of the force applied by each fingertip on the object surface depends on the distance between the virtual contact point and the real contact point actually reached on the object surface. Thus, each virtual contact point is adjusted as a function of the force error, i.e. the difference between the desired and the current contact force sensed on each fingertip. Determining the finger movements using only the virtual contact points allows the object manipulation without knowing its real shape or any other physical property.

In the experimentation, three fingers of a robotic hand are used to grasp and manipulate the object, using tripod grasp (Feix et al. 2016), i.e. the thumb works opposite to the other two fingers (abduction movement) in the same way as humans do it. In this work, we will consider that the Thumb works as supporting finger, while the Index and Middle fingers lead the object movements. Besides, the user continuously provides manipulation commands at a high level, that is, in each iteration the system receives a user command indicating the sense of the rotation movement and the system autonomously determines the finger movements. There is no external measurement of the object orientation but adding, for instance, a vision system, the proposed methodology could be used to positioning the object in an absolute orientation, if such orientation is actually reachable.

4.5.1 Proposed Manipulation Strategy

The manipulation is performed as an iterative process such that, in each iteration, the finger movements are computed according to the sense of rotation, s_k, indicated by the user.

As introduced in Chapter 3 Section 3.3, the fingers of the Allegro Hand f_i, $i \in \{I, M, T\}$ with I, M and T corresponding respectively to the Index, Middle and Thumb, is modeled as a kinematic serial chain with n_i DOF and n_i links. Each finger link has an associated reference frame Σ_{i_j}, $j \in \{1, ..., n_i\}$, which defines its position in the absolute reference frame $\mathcal{W}$. The position of each link j with respect to the previous one is determined by the joint angle q_{i_j}. The finger configuration $\mathbf{q}_i$ is given by the concatenation of all the joint angles as $\mathbf{q}_i = \{q_{i_1}, \cdots, q_{i_{n_i}}\}$. The hand configuration is given by the concatenation of the configurations of the fingers used for the manipulation, i.e. $\mathbf{Q} = \{\mathbf{q}_I, \mathbf{q}_M, \mathbf{q}_T\}$.

The flexion/extension joints of each finger i move the finger within a working plane Π_i, in this work Π_i is defined by three points corresponding to the positions of the reference frames Σ_{i_j} of the three phalanges of the finger. The variables involved in the manipulation are computed using the projections of the relevant points on the working plane of each finger. In a tripod grasp, the finger working planes must be oriented as parallel as possible to each other, as shown in Figure 4.24. In this way, the fingers can perform cooperative movements and the object can be rotated around an axis orthogonal to the working planes of the fingers, as it is usually done by human beings. Nevertheless, the proposed procedure can be generalized to rotate objects around any arbitrary axis, there is no restriction that prevents this, but it is evident that the kinematics of the hand may allow very small rotations around some particular axis.

Given the current virtual contact points $\mathbf{P}_{i_k}$, the computation of points $\mathbf{P}_{i_{k+1}}$ for the leading fingers (Index and Middle) is done as follows. Two auxiliary points $\mathbf{P}^*_{i_{k+1}}, i = \{I, M\}$ are defined as the points resulting from a displacement ± ζ of P_{i_k} along the line perpendicular to the segment between $\mathbf{P}_{i_k}$ and $\mathbf{P}_{T_k}$, as shown in Figure 4.25. The intention is to make the axis of rotation passing through $\mathbf{P}_{T_k}$. The sign of the displacement ζ depends on the desired sense of rotation for the current iteration. Thus,

$$\mathbf{P}^*_{i_{k+1}} = \mathbf{P}_{i_k} \pm \zeta \hat{\boldsymbol{p}} \tag{4.42}$$

with $\hat{\boldsymbol{p}} \in \mathbb{R}^3$ and $\hat{\boldsymbol{p}} \cdot (\mathbf{P}_{i_k} - \mathbf{P}_{T_k}) = 0$.

Since the shape of the object is unknown, any movement of the fingers may alter the contact

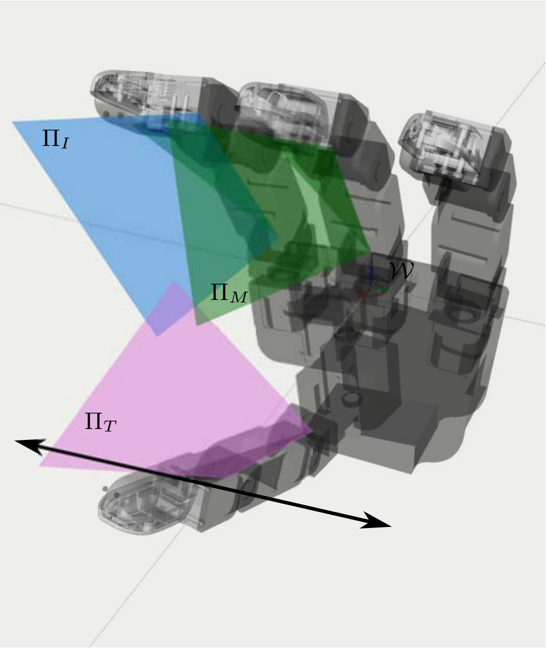


Figure 4.24: Allegro hand with the finger working planes Π_i for Index, Middle and Thumb fingers and the axis for the object rotation.

force F_{i_k}. The module of F_{i_k} must remain within a threshold around a desired value F_{i_d} because if it increases a lot the object or the hand may be damaged and if it decreases the grasp may fail and the object may fall down. In order control the value of the grasping forces, a force error $e_{f_{i_k}}$ is defined as the difference between the desired force F_{i_d} and the current force measured by the sensors F_{i_k}, i.e.

$$e_{f_{i_k}} = F_{i_k} - F_{i_d} \tag{4.43}$$

Now, consider the distance d_i defined as the Euclidean distance between each virtual contact point $\mathbf{P}_i$, $i = \{I, M\}$ and the rotation point $\mathbf{P}_T$,

$$d_{i_k} = |\mathbf{P}_{i_k} - \mathbf{P}_{T_k}| \tag{4.44}$$

An adjustment of d_{i_k} allows to change the grasping force applied on the object, then, d_{i_k} is modified in each iteration depending on the force error $e_{f_{i_k}}$ by properly determining the final positions of $\mathbf{P}_{i_{k+1}}, i = \{I, M\}$ and $\mathbf{P}_{T_{k+1}}$. $\mathbf{P}_{i_{k+1}}$ is determined as,

$$\mathbf{P}_{i_{k+1}} = \mathbf{P}^*_{i_{k+1}} + \Delta d_{i_k} \hat{\boldsymbol{p}}^*_i \tag{4.45}$$

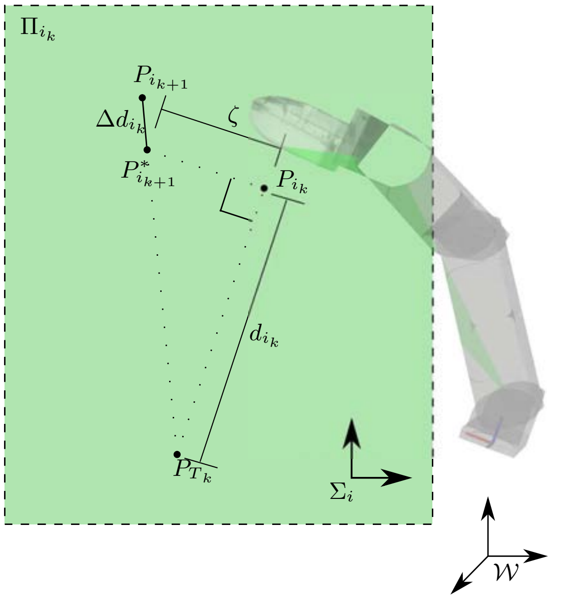


Figure 4.25: Example of the computation of $P_{i_{k+1}}$, $i = \{I, M\}$, when the contact force F_{i_k} is larger than F_{i_d} (i.e., $e_{f_{i_k}} \leq 0$). After obtaining $P^*_{i_{k+1}}$ with a displacement ζ from the current position P_{i_k}, the target virtual contact point $P_{i_{k+1}}$ is obtained applying the adjustment Δd_{i_k} to displace $P^*_{i_{k+1}}$ away from $P_{T_{k+1}}$. All the points are projections onto Π_{i_k}.

with

$$\hat{\boldsymbol{p}}_i^* = \frac{\mathbf{P}^*_{i_{k+1}} - \mathbf{P}_{T_k}}{|\mathbf{P}^*_{i_{k+1}} - \mathbf{P}_{T_k}|} \tag{4.46}$$

and

$$\Delta d_{i_k} = \begin{cases} \lambda(|e_{i_k}| + e_{i_k}^2) & \text{if } e_{i_k} \leq 0 \\ -\lambda e_{i_k} & \text{if } e_{i_k} > 0 \end{cases} \tag{4.47}$$

being λ a predefined constant, empirically obtained. In the same way than in the manipulation using two fingers, different gains are defined depending on the sign of $e_{f_{i_k}}$, because a potential fall of the object ($F_{i_k} \to 0$) is considered more critical that a potential application of large grasping forces ($F_{i_k} \gg F_{i_d}$).

In the case of the Thumb, since it is only used as supporting point for the object rotation, the computation of $\mathbf{P}_{T_{k+1}}$ is done with the only aim of adjusting the contact force without computing any intermediate virtual point. $\mathbf{P}_{T_{k+1}}$ is computed considering an adjustment with respect to the Index and Middle fingers as,

$$\mathbf{P}_{T_{k+1}} = \mathbf{P}_{T_k} + \Delta d_{T_k} \hat{\boldsymbol{p}}_T \tag{4.48}$$

Algorithm 4.3 Manipulation algorithm

```
Require: $F_d$
  procedure MANIPULATE
    $k \leftarrow 0$
    repeat
      Read the direction of rotation $s_k$
      Compute finger working planes $\Pi_{i_k}$
      Project $\mathbf{P}_{i_k}$ onto $\Pi_{i_k}$
      for $i = \{I, M\}$ do
        Compute $\mathbf{P}_{i_{k+1}}$ according to $s_k$
        Compute $\Delta d_{i_k}$
        Adjust $\mathbf{P}_{i_{k+1}}$ to obtain $\mathbf{P}_{i_{k+1}}$
      end for
      Compute $\Delta d_{T_k}$
      Adjust $\mathbf{P}_{T_k}$ to obtain $\mathbf{P}_{T_{k+1}}$
      Compute $\mathbf{Q}_{k+1}$ from $\mathbf{P}_{i_{k+1}}$ using IK
      if $\mathbf{Q}_{k+1}$ belong to the hand workspace then
        Mover hand to $\mathbf{Q}_{k+1}$
        $k \leftarrow k + 1$
      end if
    until stop by user
  end procedure
```

with

$$\Delta d_{T_k} = -\frac{\Delta d_{I_k} + \Delta d_{M_k}}{2} \tag{4.49}$$

and

$$\hat{\boldsymbol{p}}_T = \frac{\hat{\boldsymbol{p}}_I^* + \hat{\boldsymbol{p}}_M^*}{|\hat{\boldsymbol{p}}_I^* + \hat{\boldsymbol{p}}_M^*|} \tag{4.50}$$

Finally, the new hand configuration $\mathbf{Q}_{k+1}$ is computed using inverse kinematics of $\mathbf{P}_{i_{k+1}}$, $i = \{I, M, T\}$. The movements of the fingers are executed only if each $\mathbf{P}_{i_{k+1}}$ belongs to the workspace of the corresponding finger, i.e. the target $\mathbf{Q}_{k+1}$ lies within the hand workspace. Algorithm 4.3 summarizes the main steps for the computation of the hand configuration that allows the desired object manipulation.

4.5.2 Experimental Validation

The experimental validation of the proposed approach was done using the Allegro Hand introduced in Chapter 3 Section 3.3. The fingers Index, Medium and the Thumb were closed around an unknown object, until approximately reaching a desired contact force $F_d = 5$ N. In a first set of experiments, the initial grasp was obtained by moving the fingers using the graphical application introduced in Chapter 3 Section 3.4 to control individually each hand joint. This application also allows the visualization of the measured force on each sensor at the fingertips. The objects used for the experimentation, shown in Figure 4.26, were chosen looking

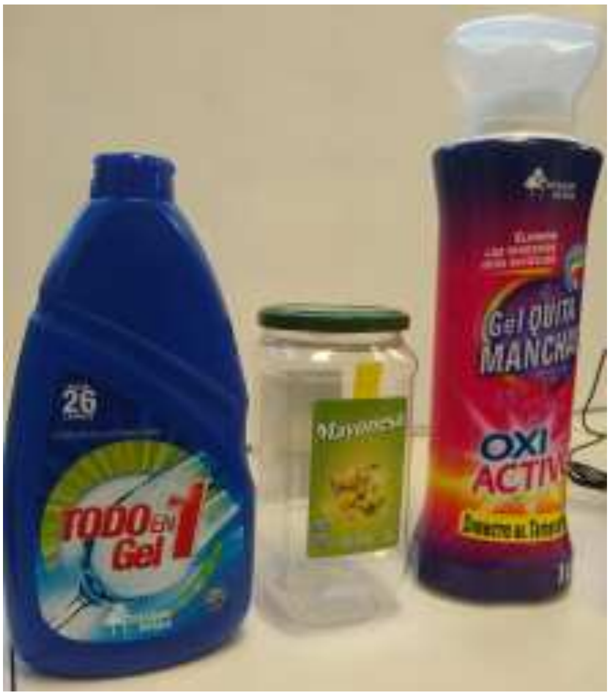
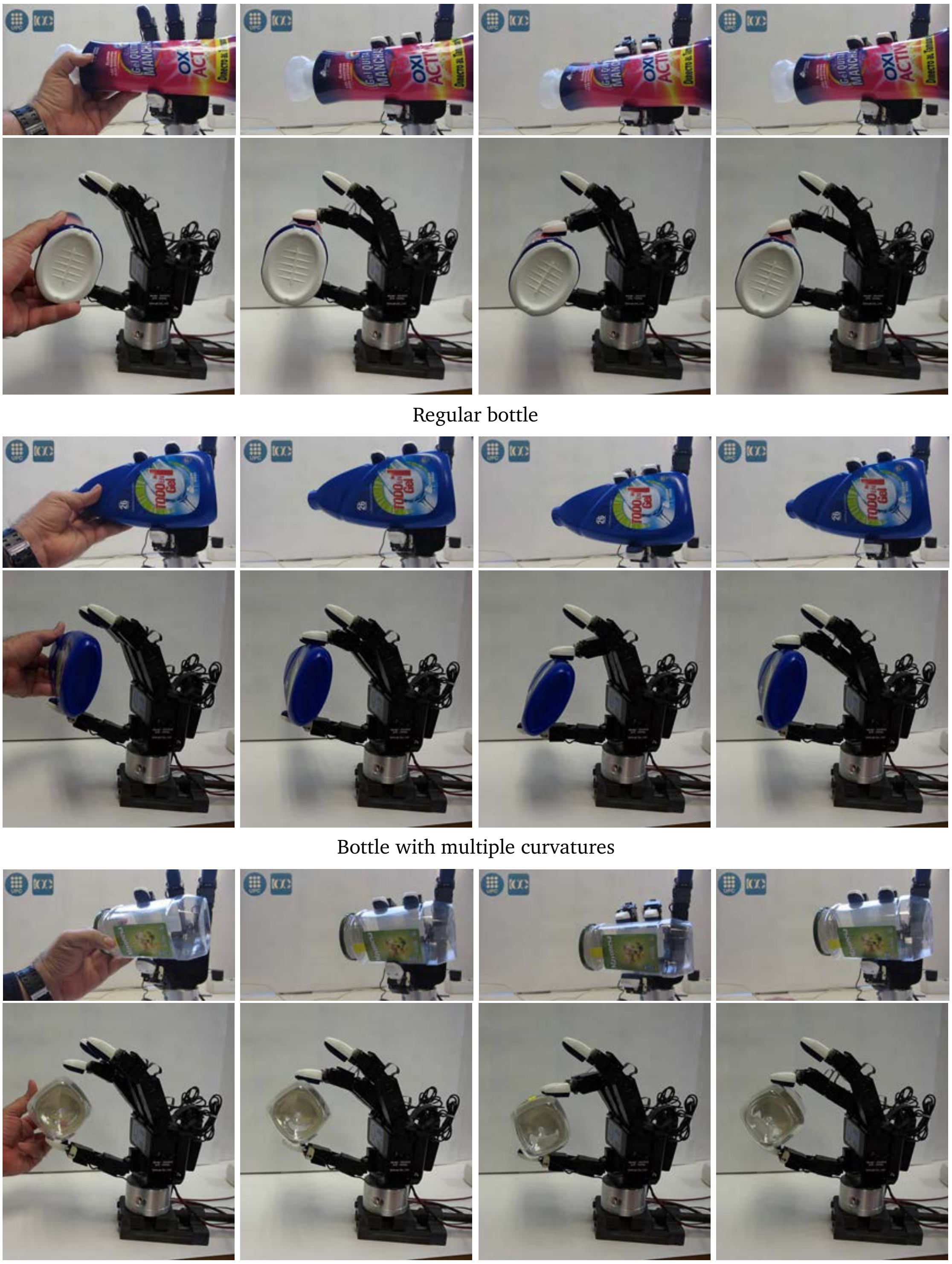


Figure 4.26: Set of everyday objects used for the first set of experiments: Bottle with multiple curvatures (left), jar with flat faces (center) and regular bottle (right).

for different shapes and stiffness, so that the proposed approach performance can be illustrated under different conditions. The constant λ to compute Δd_i and the distance ζ to compute the auxiliary points $\mathbf{P}^*_{i_{k+1}}$ were all set to 1 mm. The manipulation experiment for each object includes the following steps: a) obtaining the initial grasp; b) rotation of the object clockwise until reaching the limit of the hand workspace; c) rotation of the object counterclockwise until reaching the limit of the hand workspace; and, finally, d) release the object.

Figure 4.27 shows snapshots of the manipulation of the three objects. From left to right, the first picture shows the user putting the object in the workspace of the hand; the second picture shows the hand performing the initial grasp; the third picture shows the configuration of the hand when the limit of the hand workspace was reached after rotating the object clockwise; and the last picture shows the configuration of the hand when the limit of the hand workspace was reached after rotating the object counterclockwise.



Regular bottle

Bottle with multiple curvatures

Jar with flat faces

Figure 4.27: Snapshots of the manipulation of three objects with different shapes. Objects were rotated clockwise and counterclockwise until reaching the limits of the hand workspace.

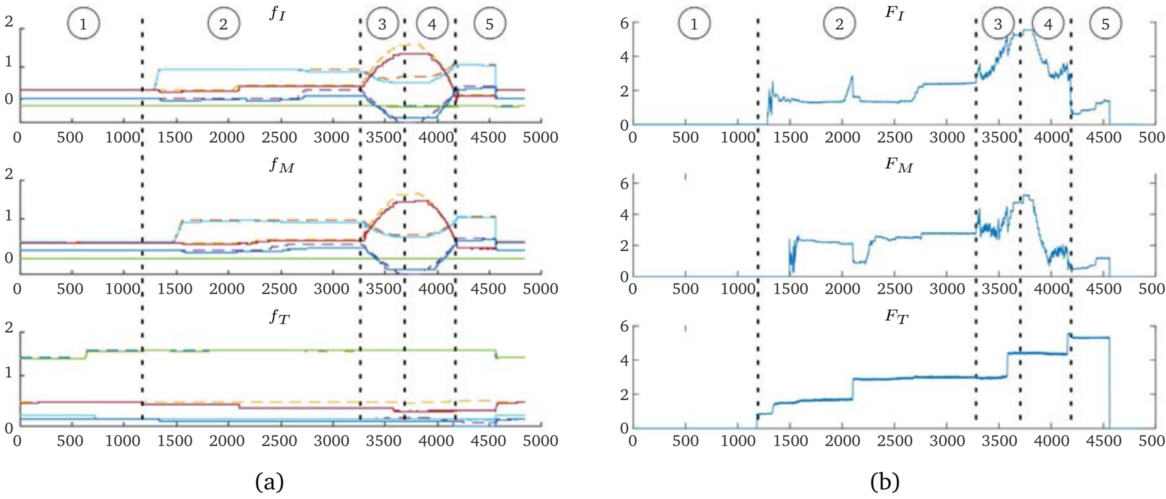
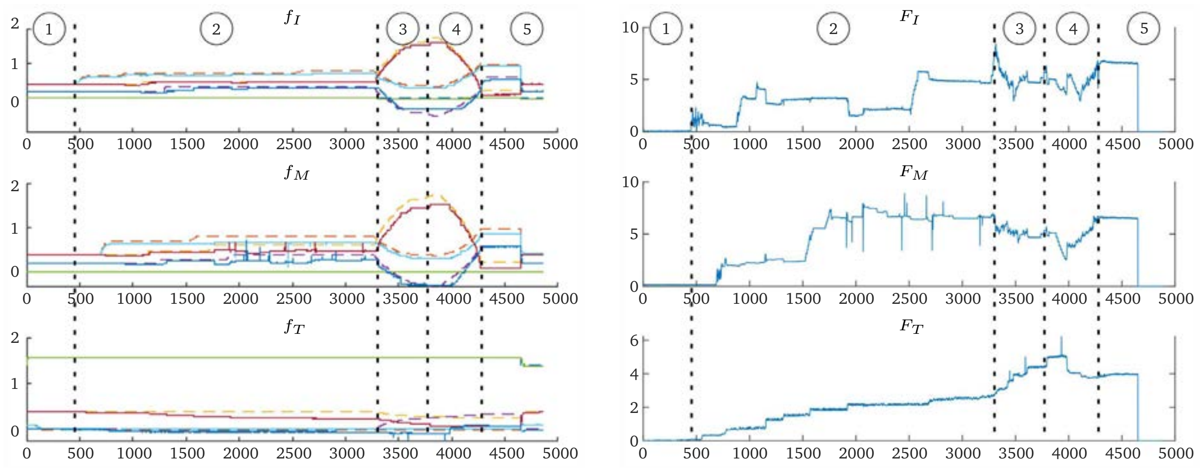


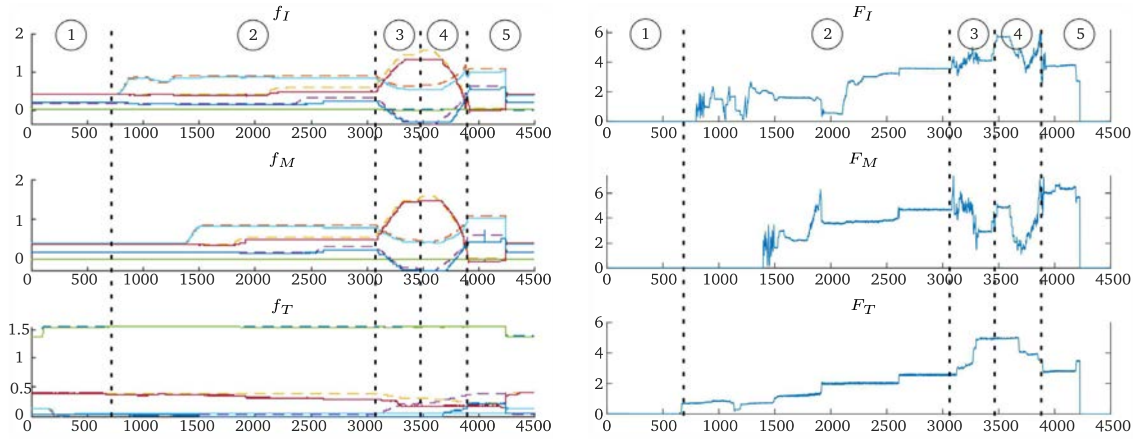
Figure 4.28: Experimental results of the manipulation of the regular bottle. (a) Evolution of joint values (in Radians): the commanded joints values in dashed line and the reached joint values in continuous line. (b) Evolution of the measured forces (in Newtons) at the fingertips.

Figures 4.28 to 4.30 show the evolution of the commanded and reached values of the finger joints, and the evolution of the measured forces at the fingertips when the regular bottle (first row in Figure 4.27), the bottle with multiple curvatures (second row in Figure 4.27) and jar with flat faces (third row in Figure 4.27) were manipulated, respectively. The commanded joint values correspond to the virtual contact points P_{i_k}, $i = \{I, M, T\}$ and the reached joint values are those obtained due to the real contact with the object surface. In these figures, five regions are remarked using vertical dashed lines and a number inside a circle: region 1 shows the values at the initial hand configuration before grasping the object; region 2 shows the evolution of the values while the initial grasp was being performed; region 3 shows the evolution of the values while the object was rotated clockwise; region 4 shows the evolution of the values while the object was rotated counterclockwise; and, finally, region 5 shows the values when the object was released and the hand returned to the initial configuration. In each region, the contact forces have the following behaviors. In region 1, the contact forces were zero at all the fingertips, since there were no contact between them and the object. In region 2, when the initial grasp was performed, the contact forces at each fingertip did not appear at the same time because the movements of the fingers were performed sequentially using the graphical interface to command the finger movements individually. In region 3 and region 4, during the manipulation, the measured forces remained close to the desired value. Finally, in region 5, the measured forces were constant until the object was released.



(a) (b)

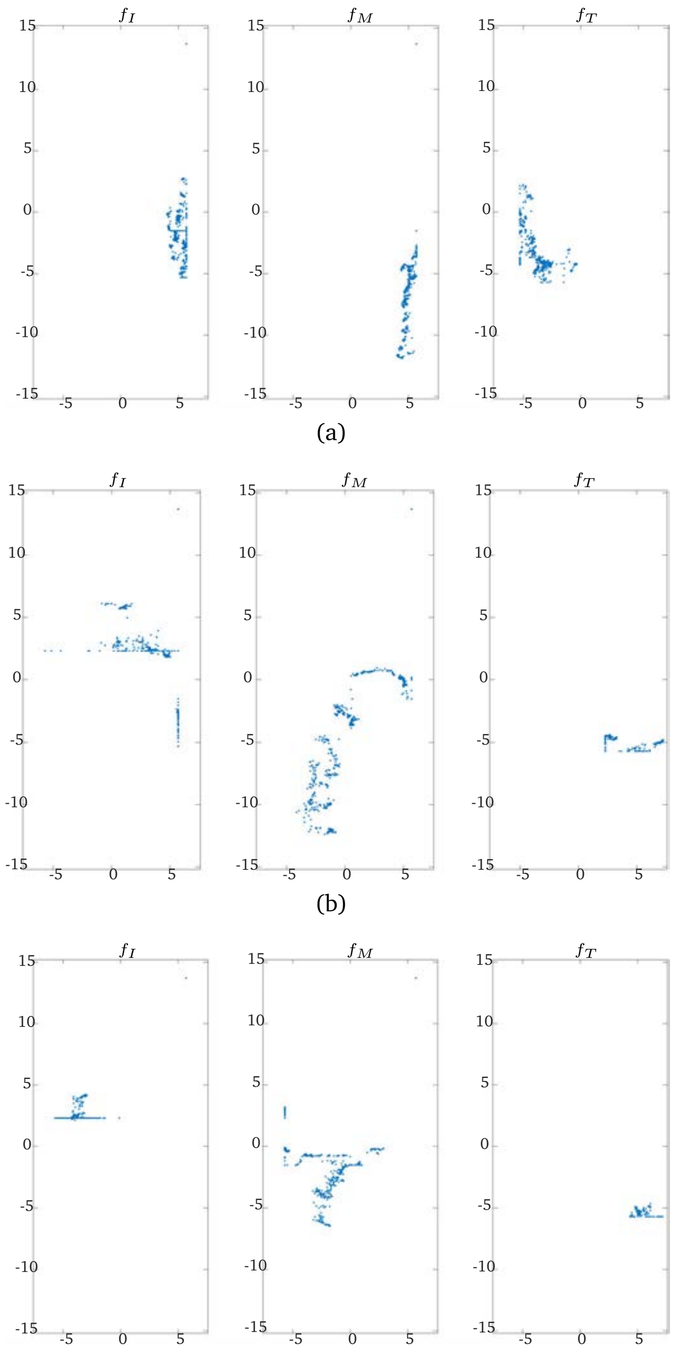
Figure 4.29: Experimental results of the manipulation of the bottle with multiple curvatures. (a) Evolution of joint values (in Radians): the commanded joints values in dashed line and the reached joint values in continuous line. (b) Evolution of the measured forces (in Newtons) at the fingertips.



(a) (b)

Figure 4.30: Experimental results of the manipulation of a jar with flat faces. (a) Evolution of joint values (in Radians): the commanded joints values in dashed line and the reached joint values in continuous line. (b) Evolution of the measured forces (in Newtons) at the fingertips.

Figure 4.31 shows the resulting contact points on the sensor surface for the three manipulation examples. In the first example, the resulting contact points for the three fingers are distributed in a similar way because the relatively small and constant curvature of the object surface produces rolling over all the sensor surfaces. In the last two examples, the contact points on the Thumb are concentrated in a smaller region because the object surface has a larger curvature at the contact regions.



(a)

(b)

(c)

Figure 4.31: Contact point positions on the tactile sensor pads (in millimeters) when manipulating: (a) The regular bottle. (b) The bottle with multiple curvatures. (c) A jar with flat faces.

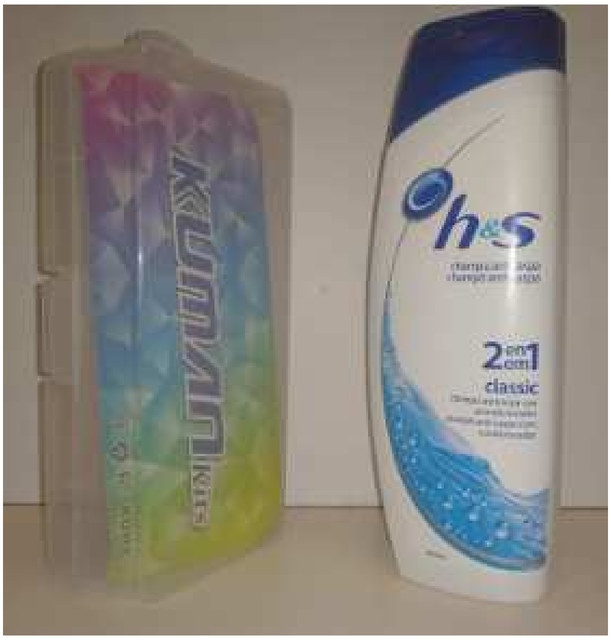
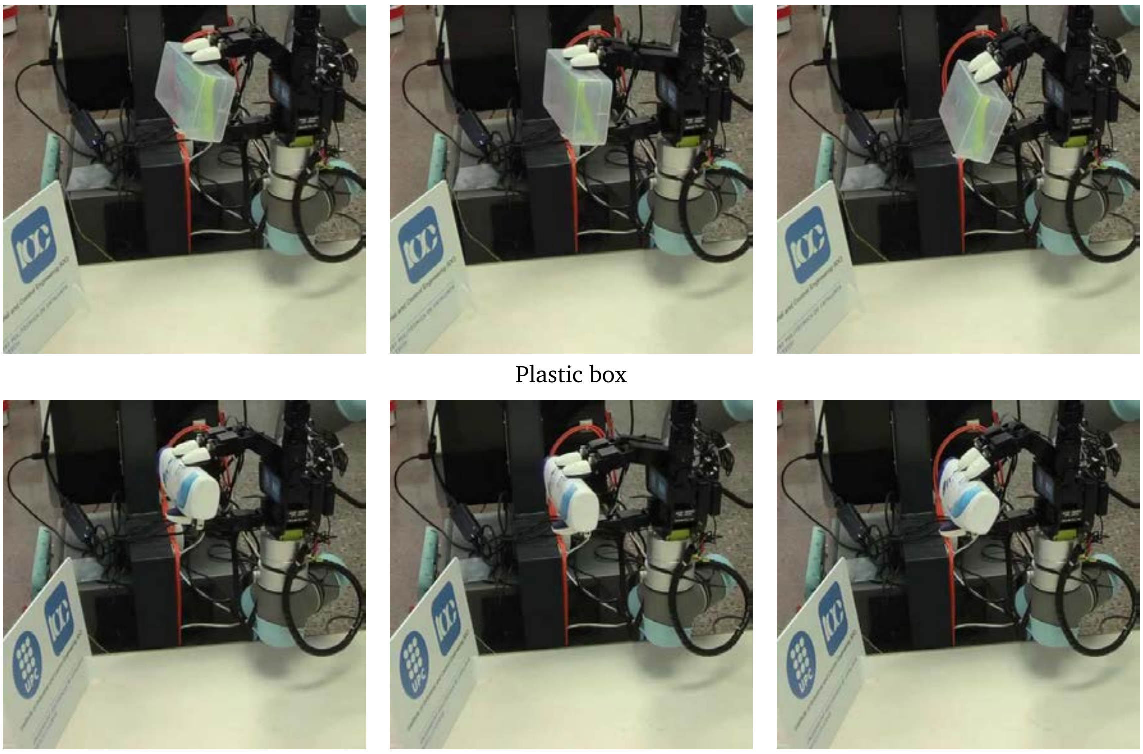


Figure 4.32: Set of objects used for the second set of experiments: plastic box (left) and shampoo bottle (right).

In a second set of experiments, the used hand is part of a dual-arm mobile manipulator and the initial grasps of the objects (shown in Figure 4.32) were done by the robot itself. In must be noted that one of the objects (the plastic box) is almost completely rigid. The arm moves the hand to a position such that it envelopes the object and then, the fingers are closed until grasping the object with contact forces close to the desired value. We have to remark that, as it was stated before, the problem of obtaining optimized initial grasps is outside the scope of this work. Once the object is grasped, it is lifted and then, rotated counterclockwise and clockwise until reaching the limits of the hand workspace. The adjustable parameters were set to the same values as in the first set of experiments.

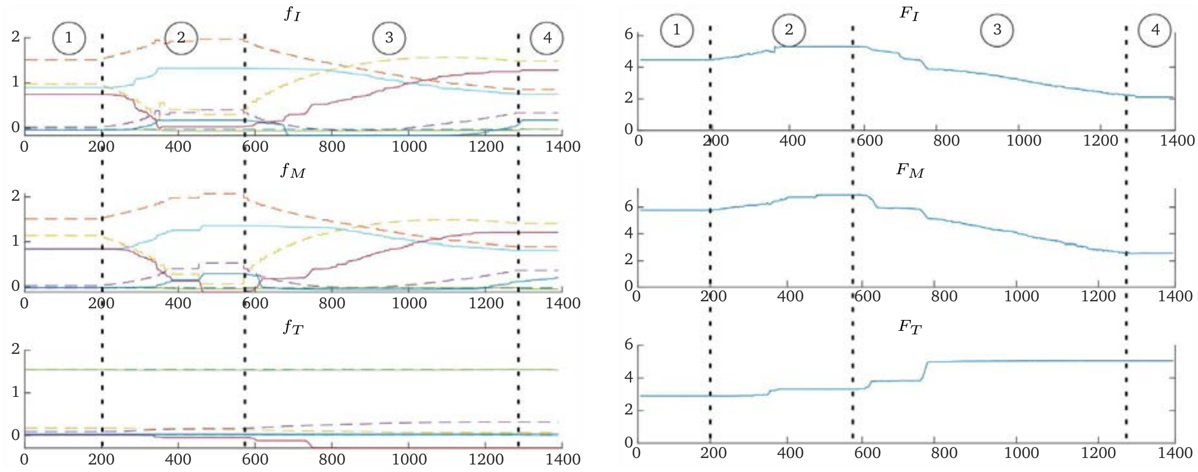
Figure 4.33 shows snapshots of the manipulation of the two objects. Figures 4.34 and 4.35 show the evolution of the commanded and reached values of the finger joints and the evolution of the measured forces at the fingertips for each manipulation example. In these figures four regions are remarked using vertical dashed lines and a number inside a circle: region 1 and region 4 show the values before and after the manipulation process; region 2 shows the values during the counterclockwise rotation and region 3 shows the values during the clockwise rotation of the objects. Videos showing the system performance for each case in both sets of experiments can be found in https://bit.ly/2lLvbDY.



Plastic box

Shampoo bottle

Figure 4.33: Snapshots of the manipulation of a plastic box and a shampoo bottle. Objects were rotated counterclockwise and clockwise until reaching the limits of the hand workspace.



(a)

(b)

Figure 4.34: Experimental results of the manipulation of the plastic box. (a) Evolution of the joint values (in Radians) of the three fingers: the commanded joints values in dashed line and the reached joint values in continuous line. (b) Evolution of the measured forces (in Newtons) at the fingertips.

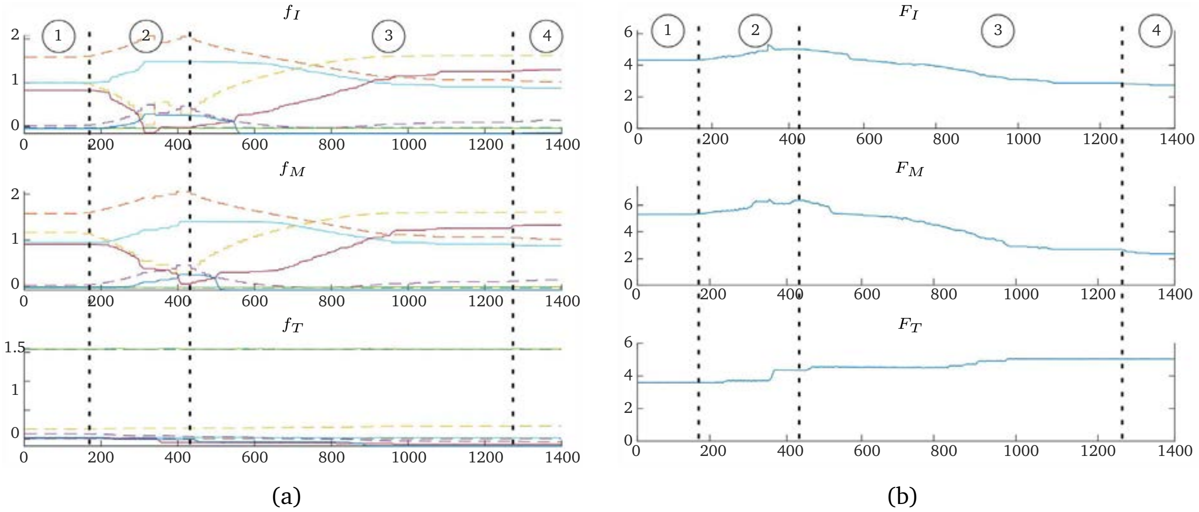


Figure 4.35: Experimental results of the manipulation of the shampoo bottle. (a) Evolution of the joint values (in Radians) of the three fingers: the commanded joints values in dashed line and the reached joint values in continuous line. (b) Evolution of the measured forces (in Newtons) at the fingertips.

"Success depends upon previous preparation, and without such it there is sure to be failure."

Confucius (551 BC - 479 BC)

5 Applications

The data generated during the manipulation can be used in several applications besides the manipulation process itself, such as, for instance, the identification of the manipulated object, the optimization of the forces applied on the manipulated object, and the teleoperation. The developments done for each of these applications and experimental results are presented in each of the following sections.

5.1 Object Recognition

Human beings use tactile feedback for retrieving object information, such as texture, temperature and shape, as well as to assist grasping and manipulation actions with the detection of important events, like slippage or object deformation. By monitoring these events, the human hand can update the grasping force according to the object weight, stiffness or friction. It has been shown that people have difficulties to perform manipulation tasks when they are deprived of tactile feedback (Johansson and Westling 1984). Besides, several characteristics of the grasped objects can also be recognized by touching. The shape of the object, the irregularity of the contact surface and the temperature of the object are some of them (Drimus et al. 2011). The object shape reconstruction using tactile information is useful in applications where it is not possible to apply artificial vision. Even providing artificial vision, the tactile information is a good complement to reduce uncertainty in the object model (Prats et al. 2010).

The tactile and kinematic data collected during the manipulation process are used here to

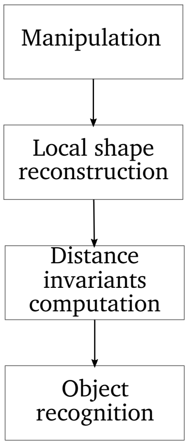


Figure 5.1: Flow chart for object identification.

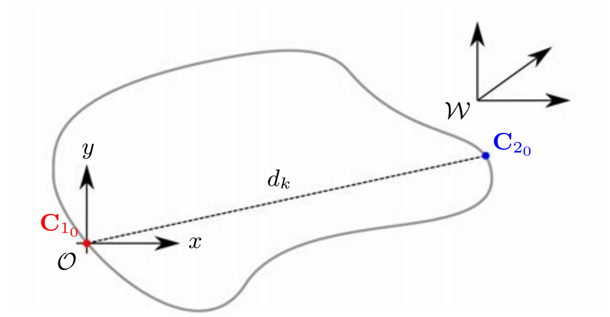


Figure 5.2: Contact points $\mathbf{C}^{\mathcal{O}}_{1_k}$ and $\mathbf{C}^{\mathcal{O}}_{1_k}$ expressed in the object reference frame $\mathcal{O}$.

identify points on the surface of the object in contact with the hand in order to allow the reconstruction of the object shape. The object identification is performed using basic distance invariants, which are measured on the reconstructed shape of the object. For this same purpose, there are other more complex and sophisticated procedures but keeping coherence with the approaches proposed, a simple, but effective, geometrical reasoning is proposed here for the object recognition. Figure 5.1 shows the flow chart of the object identification procedure.

5.1.1 Local Shape Reconstruction

The local shape reconstruction uses the data collected during the object manipulation. The contact points $\mathbf{C}_{i_k}$ are mapped from the reference frame $\mathcal{W}$ to a reference frame $\mathcal{O}$ associated to the object, being the origin of $\mathcal{O}$ at the first contact point $\mathbf{C}^{\mathcal{W}}_{1_0}$, where the superscript $\mathcal{W}$ refers to the absolute reference frame, and the orientation of $\mathcal{O}$ coincident with that of $\mathcal{W}$, both at the initial blind grasp, i.e. for $k = 0$ as show in Figure 5.2.

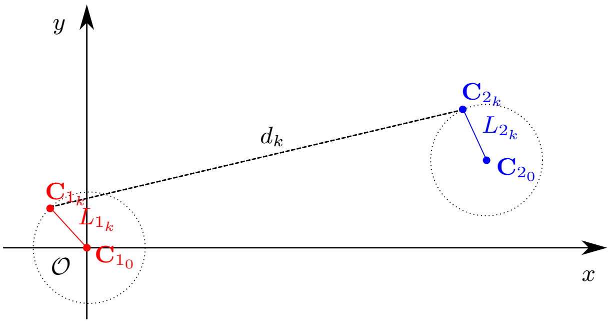


Figure 5.3: Circular regions with radius L_{1_k} and L_{2_k}, where the points belonging to the object surface are located.

The mapping of the contact points is performed using: the distance between contact points d_k, the rolled distances on the fingertip surfaces L_{i_k} (computed from consecutive measurements of the contact points on the tactile sensor), and the object orientation γ_k relative to the object orientation in the initial grasp. The mapped contact points $\mathbf{C}_{i_k}^{\mathcal{O}}, i = \{1, 2\}$, where the superscript $\mathcal{O}$ refers to the reference frame of the object, belong to two regions defined by circular paths with radius L_{1_k} and L_{2_k}, centered on the contact points $\mathbf{C}_{1_0}^{\mathcal{O}}$ and $\mathbf{C}_{2_0}^{\mathcal{O}}$ respectively, as shown in Figure 5.3.

In order to improve legibility, the subscript k is removed in the following expressions. The circular paths, referenced to the object reference frame $\mathcal{O}$, are described by:

$$C_{1_x}{}^2 + C_{1_y}{}^2 = L_1^2 \tag{5.1}$$

$$(C_{2_x} - d\cos\gamma)^2 + (C_{2_y} - d\sin\gamma)^2 = L_2^2 \tag{5.2}$$

Besides, the contact points $\mathbf{C}_1$ and $\mathbf{C}_2$ must satisfy

$$d = \sqrt{(C_{2_x} - C_{1_x})^2 + (C_{2_y} - C_{1_y})^2} \tag{5.3}$$

$$(C_{2_y} - C_{1_y}) = (C_{2_x} - C_{1_x})\tan\gamma \tag{5.4}$$

Eq. (5.1), Eq. (5.2), Eq. (5.3) and Eq. (5.4) are solved for C_{1_x}, C_{2_x}, C_{1_y} and C_{2_y} resulting in:

$$C_{1_x} = C_{2_x} - \sqrt{\frac{d^2}{1 + (\tan\gamma)^2}} \tag{5.5}$$

$$C_{1_y} = C_{2_y} - \tan\gamma\sqrt{\frac{d^2}{1+(\tan\gamma)^2}} \tag{5.6}$$

$$C_{2_x} = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \tag{5.7}$$

$$C_{2_y} = \sqrt{L_2^2 - (C_{2_x} - d\cos\gamma)^2} + d\sin\gamma \tag{5.8}$$

where

$$a = (-2d\sin\gamma + 2\beta)^2 + (-2d\cos\gamma + 2\rho)^2 \tag{5.9}$$

$$\begin{aligned} b =& -2d\cos\gamma(-2d\sin\gamma+2\beta)^2 - 2\psi(-2d\cos\gamma+2\rho) \\ &+ 2d\sin\gamma(-2d\sin\gamma+2\beta)(-2d\cos\gamma+2\rho) \end{aligned} \tag{5.10}$$

$$\begin{aligned} c =& (d\cos\gamma)^2(-2d\sin\gamma+2\beta)^2 + \psi^2 - 2\psi d\sin\gamma(-2d\sin\gamma+2\beta) \\ &+ (d\sin\gamma)^2(-2d\sin\gamma+2\beta)^2 - L_2^2(-2d\sin\gamma+2\beta)^2 \end{aligned} \tag{5.11}$$

$$\rho = \sqrt{\frac{d^2}{1+(\tan\gamma)^2}} \tag{5.12}$$

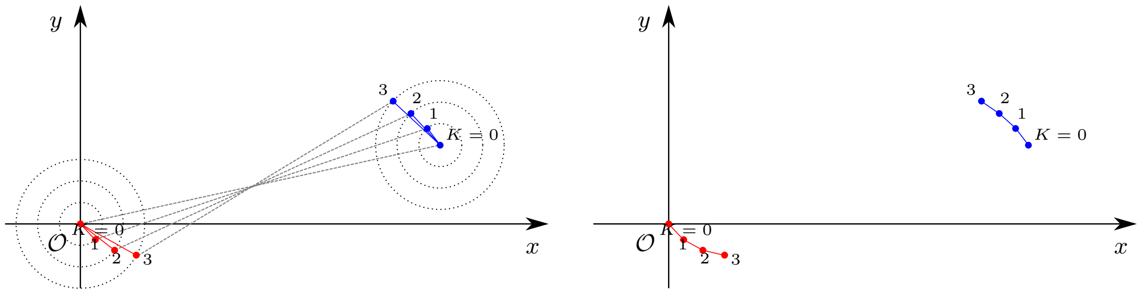
$$\beta = \tan\gamma\sqrt{\frac{d^2}{1+(\tan\gamma)^2}} \tag{5.13}$$

$$\psi = L_2^2 - L_1^2 - (d\cos\gamma)^2 - (d\sin\gamma)^2 + \rho^2 + \beta^2 \tag{5.14}$$

Computing iteratively the points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$ for each iteration k, the object shape is reconstructed as it is shown in Figure 5.4.

5.1.2 Object Recognition

The distance invariants are defined as the distances d_{ij}, with $i \neq j$ and $i, j \in \{1, ..., 4\}$, between each pair of the extreme points P_i belonging to the reconstructed object shape as shown in Figure 5.5. The distance invariants are computed after the reconstructed object shape has been obtained.



(a) (b)

Figure 5.4: (a) Representation of a reconstructed object shape (numbered points) applying iteratively the identification of boundary points. (b) Reconstructed object shape.

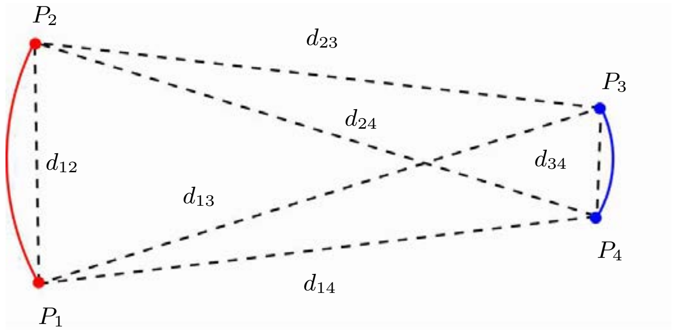


Figure 5.5: Example of reconstructed shape for finger f_1 (red) and finger f_2 (blue), and their distance invariants. Points P_1, P_2, P_3 and P_4 are the extreme points of the reconstructed shape.

The reconstructed object shape and distance invariants are considered as a pattern (signature) of the manipulated object, which is used for object recognition. Using previously stored models from a database of objects, the matching of the object signature and the points of the models is searched as follows. First, the extreme points P_1 and P_2 of the portion of the reconstructed shape corresponding to finger f_1 are matched against the object model being evaluated. If a good match is found, then, extreme points P_3 and P_4 of the portion of the reconstructed shape corresponding to finger f_2 are matched against the points in the object model. If the signature matches well with the object model, i.e. the four points P_i have a match with points on the model under consideration, then, a local comparison of all the points in the reconstructed object shape is performed against the points in the corresponding part of the model, considering a tolerance margin in the points positions.

An example of the matching of the object pattern (signature) and the database objects models is shown in Figure 5.6. In the first two cases, the computed signature does not match with the model of a circumference (Figure 5.6b), nor with that of a ellipse (Figure 5.6c). Figures 5.6d and 5.6e show a positive matching of the signature in a false case and with the correct object,

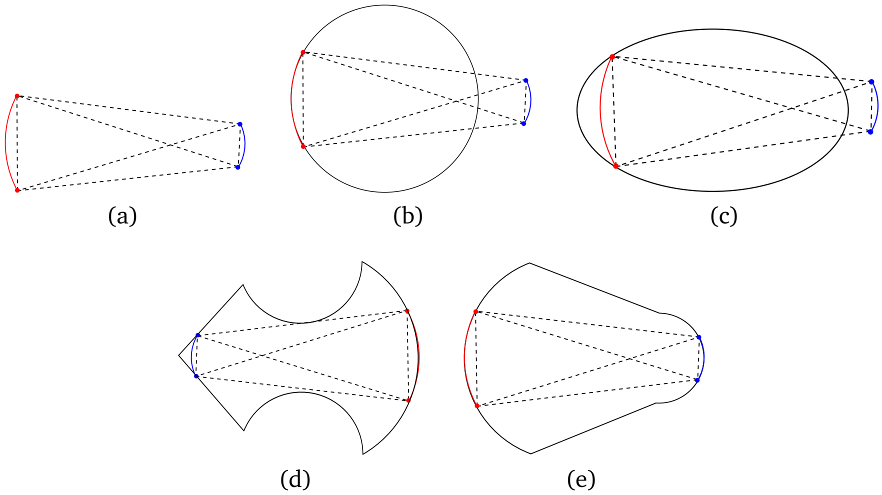


Figure 5.6: Example of the matching of the reconstructed object signature with different object models: (a) object signature. (b) Matching with a circular model. (c) Matching with an elliptical model. (d) Matching with an object model producing a false positive. (e) Positive matching with an object model.

respectively. These two positive cases are then evaluated using all the points in the recontructed object shape to select the correct model.

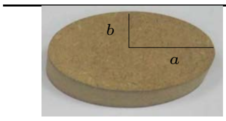
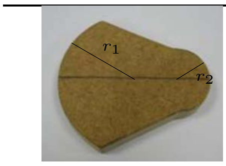
5.1.3 Experimental Validation

The proposed approach has been implemented using C++ for the manipulation process with the SDH2 hand and Matlab for the data analysis and the object identification. In order to validate the proposed approach, different experiments were carried out. Table 5.1 shows the relevant properties of three different objects, which were manipulated using the two coupled fingers of the SDH2 hand.

Each object was manipulated using the manipulation strategy that optimizes the object orientation (introduced in Chapter 4), rotating the object with two fingers. Each object was rotated three times, first counterclockwise and then clockwise, until the contact forces tend to exceed the friction cone limits or the fingers reach the limit of their workspace.

Figure 5.7 shows the real object contour and the reconstructed object shape. The noise in the reconstructed shape due to the sensor noise and the computational approximations was not significant in the matching procedure. Figure 5.8 shows snapshots of a real execution of the manipulation and recognition process where the hand holds the object with cylindrical shape.

Table 5.1: Objects used for the validation of the object recognition approach.

Object	Shape	Dimensions
	Cylindrical	$r = 40$
	Elliptical	$a = 37, b = 20$
	Two curvatures	$r_1 = 45, r_2 = 20$

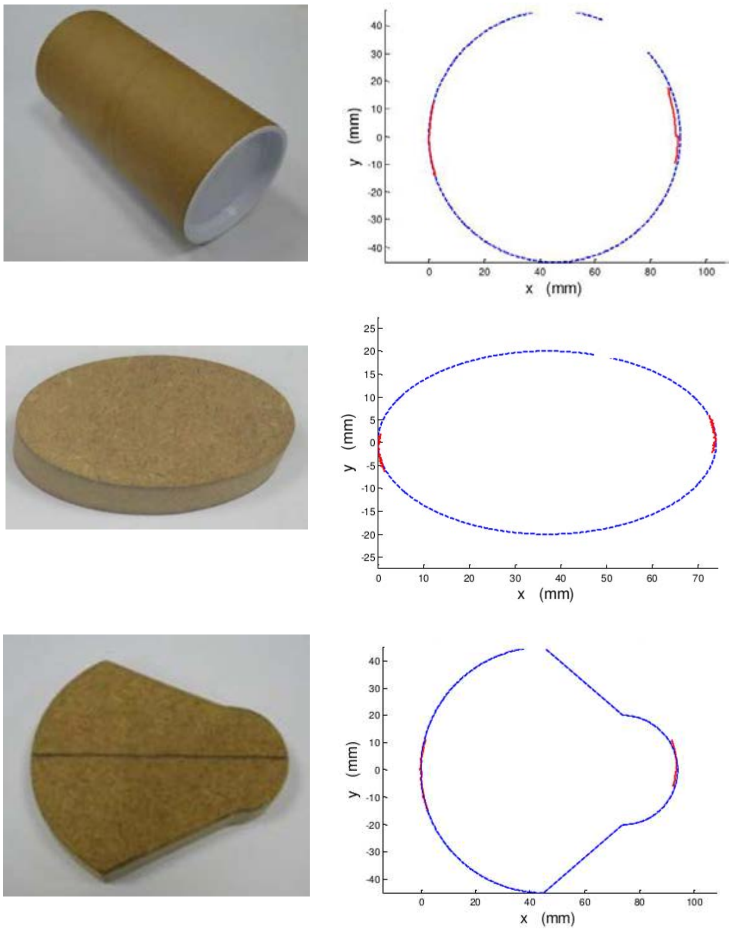


Figure 5.7: Results of the object shape reconstruction for three objects. The right images show the object model (in blue) and the reconstructed shape (in red).

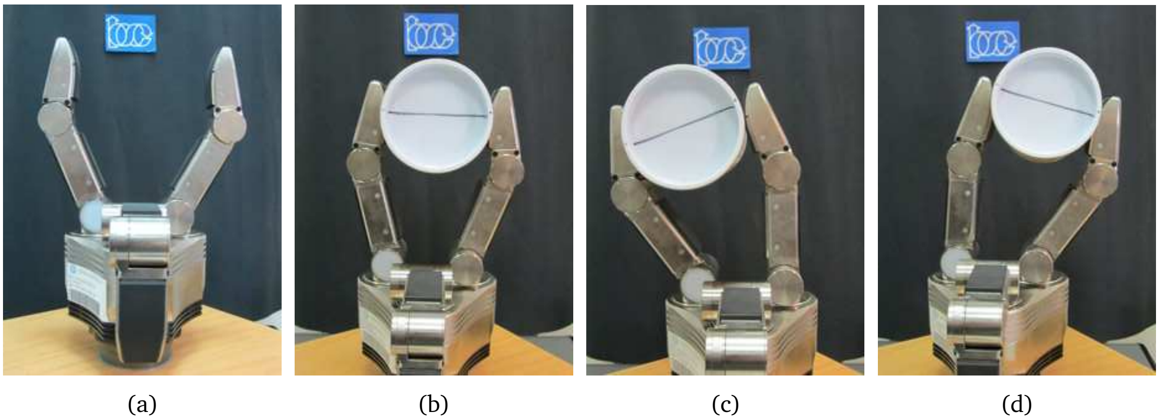


Figure 5.8: Snapshots of a real execution of the manipulation of a cylinder. (a) The SDH2 hand in the initial position. (b) Cylinder grasped before that the manipulation process starts. (c) Configuration where the fingers reach their workspace limit in counterclockwise motion. (d) Idem in clockwise motion.

5.2 Force Optimization

Another application related with the shape identification during the manipulation of unknown objects is the optimization of the applied forces on the manipulated object. Combining the recognition of the local curvature of the object surface at the contacted regions with a proper manipulation strategy it is possible to optimize the contact forces during the manipulation process. The approach described in this section iteratively determines a sequence of hand configurations that allows the secure object manipulation and, at the same time, obtain information on the object shape in order to improve the contact forces during the manipulation.

5.2.1 Manipulation Strategy

Algorithm 5.1 shows the main steps of the manipulation procedure. The desired contact force F_d and the desired object orientation γ_d relative to the initial object orientation γ_0 are the algorithm inputs. The manipulation begins once the object has been properly grasped with a FC grasp. The manipulation procedure iteratively computes the target hand configurations to rotate the object towards the desired orientation.

In order to improve the contact forces, the manipulation is divided into two stages, in the first

Algorithm 5.1 Force optimization based on manipulation

```
Require: $F_d$, $\gamma_d$
  procedure ForceOptimization
    $k \leftarrow 0$, stop $\leftarrow$ false
    while stop$\neq$true do
      Compute $\mathbf{C}_i$ using Forward Kinematics
      Compute $d_k$ using Eq. (5.38)
      Compute $F_k$ using Eq. (4.6)
      Compute $\gamma_k$ using Eq. (5.15)
      Compute $\varepsilon_{o_k}$ using Eq. (5.16)
      if $k \leq m$ then
        Compute $\mathbf{C}^*_{i_{k+1}}$ using Eq. (5.17) to (5.20)
      else
        Compute the local curvatures
        Compute $\mathbf{C}^*_{i_{k+1}}$ using Eq. (5.22) to (5.25)
      end if
      Compute $d_{k+1}$ using Eq. (4.9)
      Compute $\mathbf{C}_{i_{k+1}}$
      Compute $\boldsymbol{Q}_{k+1}$ from $C_{i_{k+1}}$ using Inverse Kinematics
      if a termination condition is not satisfied then
        Move $f_i$ to $\mathbf{Q}_{k+1}$
        $k \leftarrow k + 1$
      else
        stop$\leftarrow$true
      end if
    end while
  end procedure
```

one, the fingertip positions are computed using a path on a single virtual circumference (as it was done in the approach described in Chapter 4), and in the second one, each fingertip follows a specific circumference computed according to the estimated local curvature of the object at the contact point. These two specific circumferences are computed using a circular regression over the set of resulting contact points of the previous manipulation iterations, thus, the second stage can start only after m iterations, being m the number of points used to properly compute the circular regression. The local curvatures provide information about the object shape and their use in the computation of the resulting target hand configurations improves the forces applied on the manipulated object. Each manipulation iteration k involves the following steps:

- Computation of the current state of the grasp.
- Computation of two auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$, to be used in the computation of the target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$.
- Computation of the distance d_k between contact points to improve the contact force.
- Computation of the target contact points $\mathbf{C}_{i_{k+1}}$ using $\mathbf{C}^*_{i_{k+1}}$ and d_k.
- Computation of the target hand configuration $\mathbf{Q}_{k+1}$.

- Verification of the termination conditions.
- Movement of the fingers to the target configuration if none of the termination conditions is activated.

The computation of the current state of the grasp implies the computation of the position of the current contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, the magnitude of the grasping force F_k, the orientation of the object γ_k, and the orientation error ε_{o_k}. $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$ are computed using FK and the virtual link information, as described in Chapter 3, Section 3.2. F_k is the average of the contact forces F_{1_k} and F_{2_k} measured on each fingertip. The value of γ_k in the initial grasp is considered as $\gamma_0 = 0$, and it is used as reference for the object orientation. During the iterative process, γ_k is computed using the information obtained from the tactile sensors and the finger kinematics. For fingertips with circular shape, the current object orientation γ_k can be approximated by,

$$\gamma_k \approx \frac{R}{d_k}\sum_{j=1}^{n_1}(q_{1j_{\gamma_0}} - q_{1\mathrm{j}_k}) - \sum_{j=1}^{n_2}(q_{2\mathrm{j}_{\gamma_0}} - q_{2j_k}) \tag{5.15}$$

In this manipulation strategy, the goal is to reduce the orientation error ε_{o_k} between the desired orientation γ_d and the current object orientation γ_k, i.e.

$$\varepsilon_{o_k} = \gamma_d - \gamma_k \tag{5.16}$$

During the first m iterations there is no information about the object shape, therefore the two auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed considering them as the result of a displacement of the current contact points on the sensor pad along a virtual circular path with diameter d_k, centered at the middle point R_k between the points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, i.e. the auxiliary points are given by (see Figure 5.9):

$$C^*_{1_{k+1_x}} = R_{k_x} - \frac{d_k}{2}\cos(\gamma_{k+1}) \tag{5.17}$$

$$C^*_{1_{k+1_y}} = R_{k_y} - \frac{d_k}{2}\sin(\gamma_{k+1}) \tag{5.18}$$

$$C^*_{2_{k+1_x}} = R_{k_x} + \frac{d_k}{2}\cos(\gamma_{k+1}) \tag{5.19}$$

$$C^*_{2_{k+1_y}} = R_{k_y} + \frac{d_k}{2}\sin(\gamma_{k+1}) \tag{5.20}$$

with

$$\gamma_{k+1} = \gamma_k + \tanh(\varepsilon_o)\Delta\gamma \tag{5.21}$$

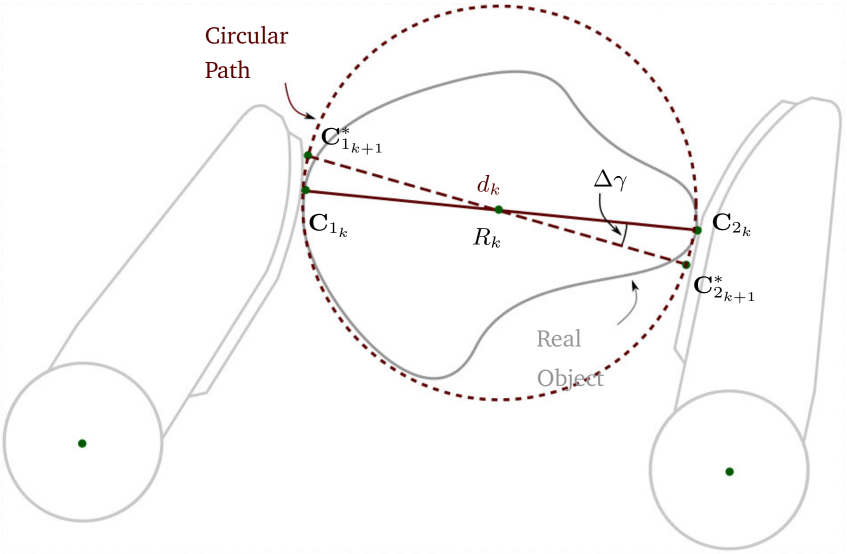


Figure 5.9: Movements used to change the object orientation. $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed over a circular path with diameter d_k centered at R_k.

where $\Delta\gamma > 0$ is chosen, empirically, small enough to assure small movements of the object in each iteration and the function $\tanh$ is used to limit the gain for large values of γ_k. Note that the point R_k is recomputed in each iteration.

Then, after m iterations (i.e. $k > m$), $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed considering them as the result of a displacement of the current contact points on the sensor pad along circular paths described by the circumferences $\mathcal{C}_{i_k}$ with center at $(c_{i_{k_x}}, c_{i_{k_y}})$ and radius $r_{c_{i_k}}$ obtained using circular regression, as described below in Subsection 5.2.2, i.e. the initial virtual circular path is replaced by paths described by circumferences computed using the data obtained during the manipulation (see Figure 5.10). Therefore, $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are given by,

$$C^*_{1_{k+1_x}} = c_{1_{x_k}} - (r_{c_{1_k}})\cos(\gamma_{k+1}) \quad (5.22)$$

$$C^*_{1_{k+1_y}} = c_{1_{y_k}} - (r_{c_{1_k}})\sin(\gamma_{k+1}) \quad (5.23)$$

$$C^*_{2_{k+1_x}} = c_{2_{x_k}} + (r_{c_{2_k}})\cos(\gamma_{k+1}) \quad (5.24)$$

$$C^*_{2_{k+1_y}} = c_{2_{y_k}} + (r_{c_{2_k}})\sin(\gamma_{k+1}) \quad (5.25)$$

From this point on, i.e. after computing the auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$, the algorithm follows the same steps as those of the general algorithm for manipulation strategies based on the gradient method presented in Chapter 4, Section 4.4.

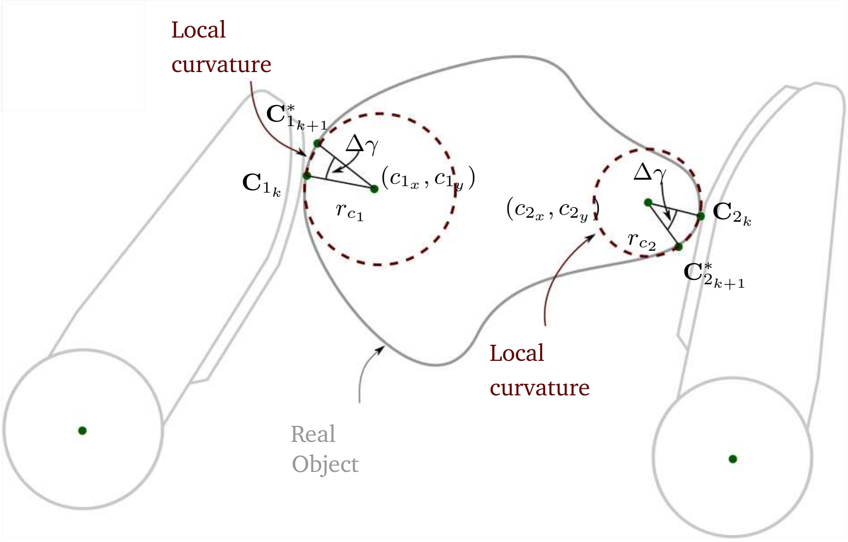


Figure 5.10: Movements used to change the object orientation improving the contact force. $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed over the circular paths generated by the circular regressions using the previous contact points.

5.2.2 Computation of the Local Curvature of the Object

The local curvature of the object is determined on-line applying a circular regression over a set of contact points already obtained during the manipulation. The contact points have to be mapped from the reference frame $\mathcal{W}$ to a reference frame associated to the object $\mathcal{O}$, being the origin of $\mathcal{O}$ at the first contact point $\mathbf{C}_1^{\mathcal{W}}$ and its orientation coincident with that of $\mathcal{W}$, both at the initial blind grasp. This mapping is done with the same procedure explained in Subsection 5.1.1 for the reconstruction of the local shape of the object.

Let $S_{\mathbf{C}_i}, i \in \{1, 2\}$ be the set of m mapped contact points $\mathbf{C}_{i_h}^{\mathcal{O}}, h \in \{k-m, ..., k-1, k\}$. The goal of the circular regression is to determine the circumference $\mathcal{C}_i$ with center point (c_{i_x}, c_{i_y}) and radius r_{c_i}, such that the sum of the squared distances from each point in $S_{\mathbf{C}_i}$ to $\mathcal{C}_i$ is minimized, i.e. the points must be as close as possible to $\mathcal{C}_i$ (Chernov 2010). The assumption of circular shapes is based on the fact that “most objects in the world are made up of circular arcs and straight segments” (Perkins 1978). Note that a straight segment can be seen as a circular arc with large enough radius. The obtained $\mathcal{C}_i$ are mapped back to the absolute reference frame $\mathcal{W}$ by locating the center point (c_{i_x}, c_{i_y}) at a distance r_{c_i} from the contact point $\mathbf{C}_{i_k}$ along the normal direction to the fingertip at $\mathbf{C}_{i_k}$, as shown in Figure 5.10. Then, $\mathcal{C}_i^{\mathcal{W}}$ is used to compute the auxiliary points $\mathbf{C}^*_{i_{k+1}}$ as described in the manipulation strategy in Eq. (5.22) to Eq. (5.25) in Subsection 5.2.1.

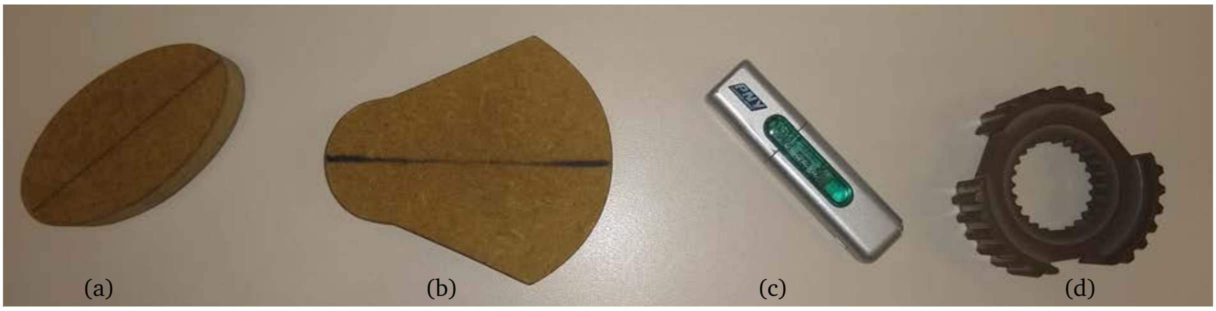


Figure 5.11: Set of objects used in the experimentation of the force optimization approach. (a) Object 1: elliptical shape. (b) Object 2: two-curvature shape. (c) Object 3: USB pendrive. (d) Object 4: industrial gear.

5.2.3 Experimental Validation

The Schunk Dexterous Hand (SDH2) was used for the experimental validation. A detailed description of the hand kinematics was presented in Chapter 3. In each of the following illustrative examples an object was brought close to the hand and the hand closed the fingers to perform a blind grasp. The desired contact force was set to $F_d = 4$ N. The friction coefficient was assumed to be $\mu = 0.4$, which is lower than the real physical value considering that the fingertips are made of rubber. The constant λ to adjust the distance between the contact points according to Eq. (4.47) was set to $\lambda = 0.25$ mm. The constant $\Delta\gamma$ to change the object orientation was set to $\Delta\gamma = 0.25°$. The number of points to compute the local curvatures was set to $m = 30$. The object was rotated first to a desired orientation $\gamma_d = 5°$ and then, to $\gamma_d = -5°$. In order to illustrate the effect of using the local curvatures in the computation of the hand configurations, after these movements the object was rotated without using the local curvatures to $\gamma_d = 5°$ and then, to $\gamma_d = 0°$. In this is way the resulting contact forces in both cases can be directly compared. Figure 5.11 shows the set of objects used in the experimentation.

Figure 5.12 shows the results of the manipulation of Object 1. Figure 5.12a to 5.12c show snapshots of the object in the initial grasp, rotated to 5° and rotated to -5°, respectively. Figure 5.12d shows the contact points $\mathbf{C}_i^{\mathcal{W}}$. Figure 5.12e shows the contact points $\mathbf{C}_i^{\mathcal{O}}$ and the last computed circumferences $\mathcal{C}_i$. Figure 5.12f shows the evolution of the object orientation γ_k, and Figure 5.12g the contact force F_k, with and without the use of the local curvatures.

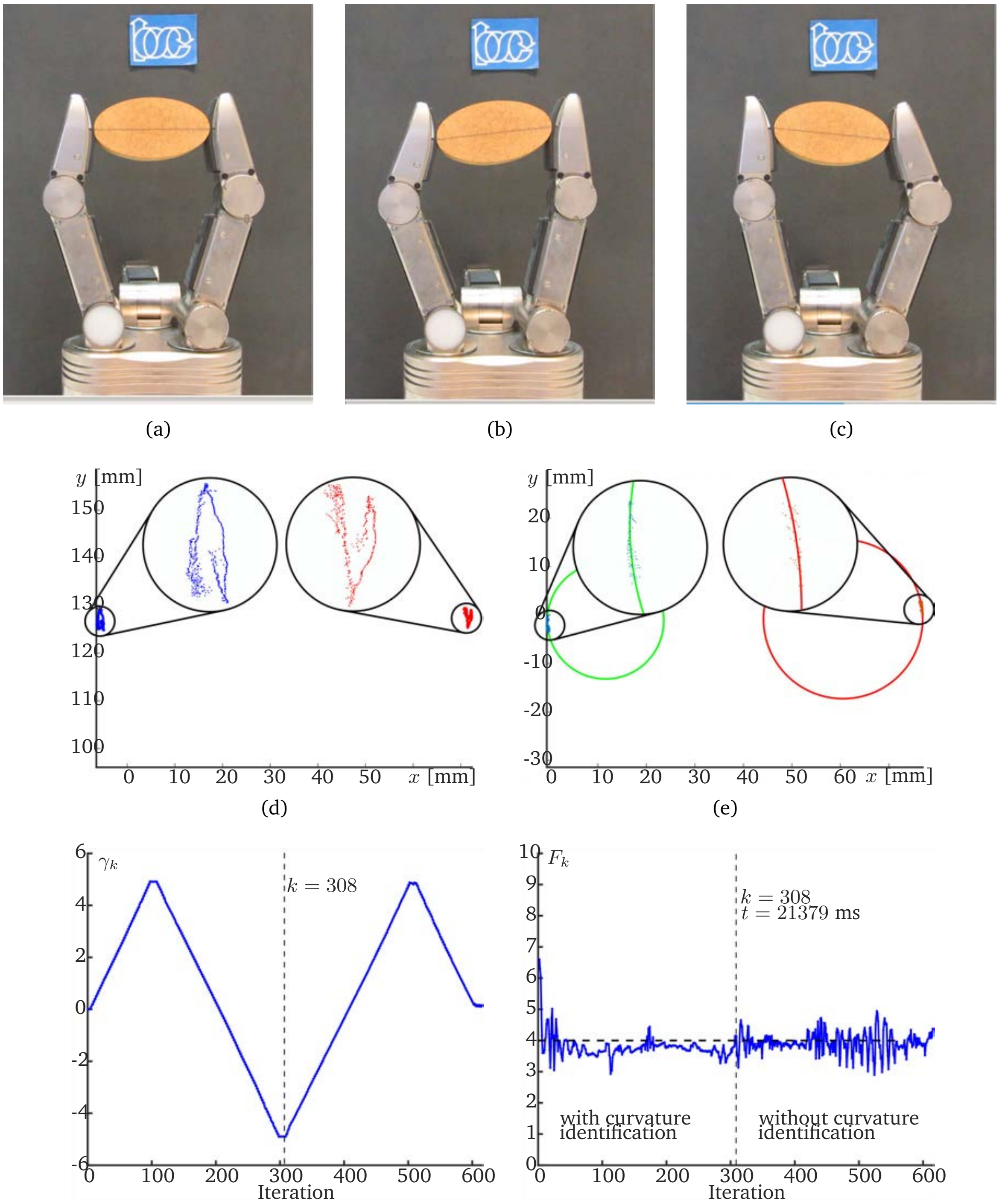


Figure 5.12: Experimental results of the manipulation of Object 1: (a) Initial grasp. (b) Object rotated to 5°. (c) Object rotated to -5°. (d) Resulting contact points $\mathbf{C}_i^{\mathcal{W}}$ with zoomed interest regions. (e) Contact points $\mathbf{C}_i$ and circumferences $\mathcal{C}i$ computed in the last circular regression, with zoomed interest regions. (f) Evolution of the orientation γ_k (in degrees). (g) Evolution of the contact force F_k (in Newtons), the horizontal dashed line indicates F_d and the vertical dashed line the iteration in which the manipulation stops using the local curvatures.

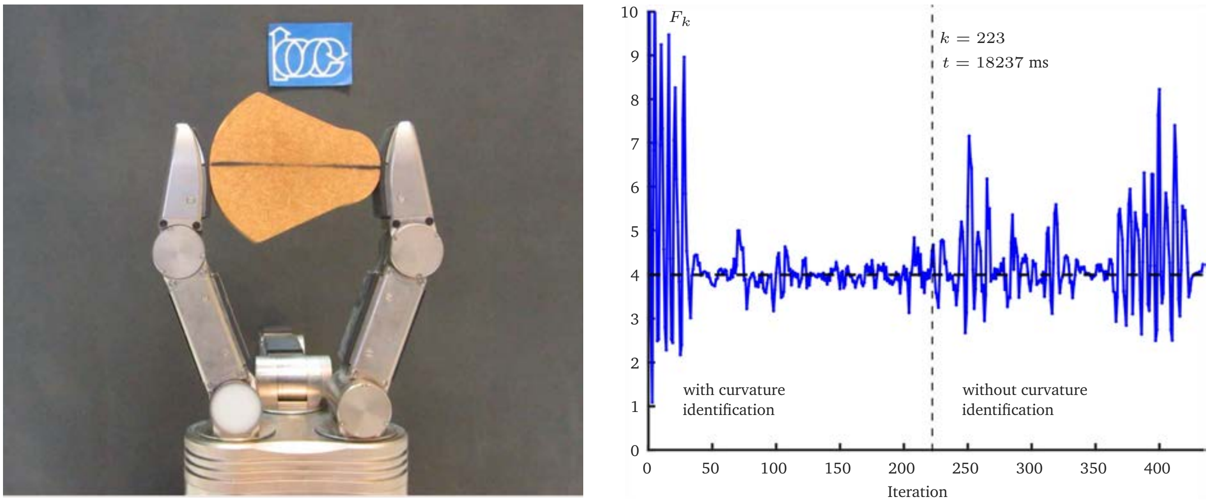


Figure 5.13: Experimental results of the manipulation of Object 2.

Figures 5.13 to 5.15 show the results of equivalent experiments using the other three objects. Each case shows: on the left, a snapshot of the initial grasp, and, on the right, a chart with the evolution of the contact force F_k (in Newtons), the horizontal dashed line indicates the desired contact force F_d and the vertical dashed line the iteration in which the manipulation stops using the local curvatures.

Table 5.2 summarizes, for each object manipulation, the average contact force $\overline{F_k}$, the corresponding force variance σ^2, the number of executed iterations, and the average time per iteration, with and without using the computed local curvatures. The improvement of the resulting contact force is indicated by its variance σ^2, when the local curvature is used in the computation of the target hand configurations the contact force has a lower variance. A video of the experimental executions can be found in http://goo.gl/Wz7UH3.

Table 5.2: Experimental results of the force optimization.

	Object 1		Object 2		Object 3		Object 4	
	with	without	with	without	with	without	with	without
$\overline{F_k}$	3.7284	3.9288	4.0072	4.2874	3.3971	3.7385	3.6060	3.9790
σ^2	0.0321	0.1056	0.0826	0.8161	0.1207	0.5450	0.0953	0.3222
Number of iterations	308	306	223	210	272	259	386	335
Time per iteration	69.41	73.48	81.78	76.06	67.96	69.17	63.67	69.97

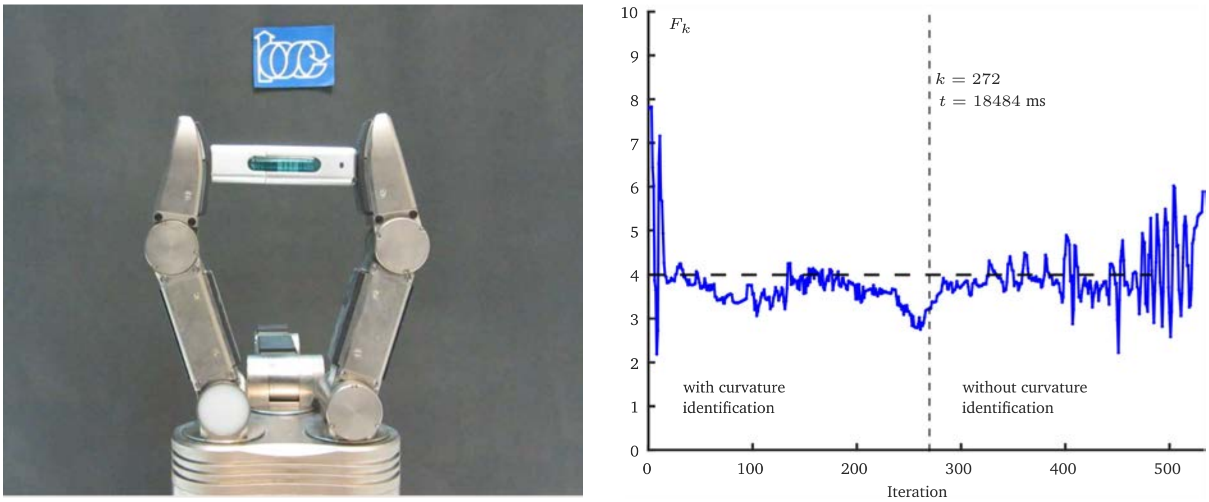


Figure 5.14: Experimental results of the manipulation of Object 3.

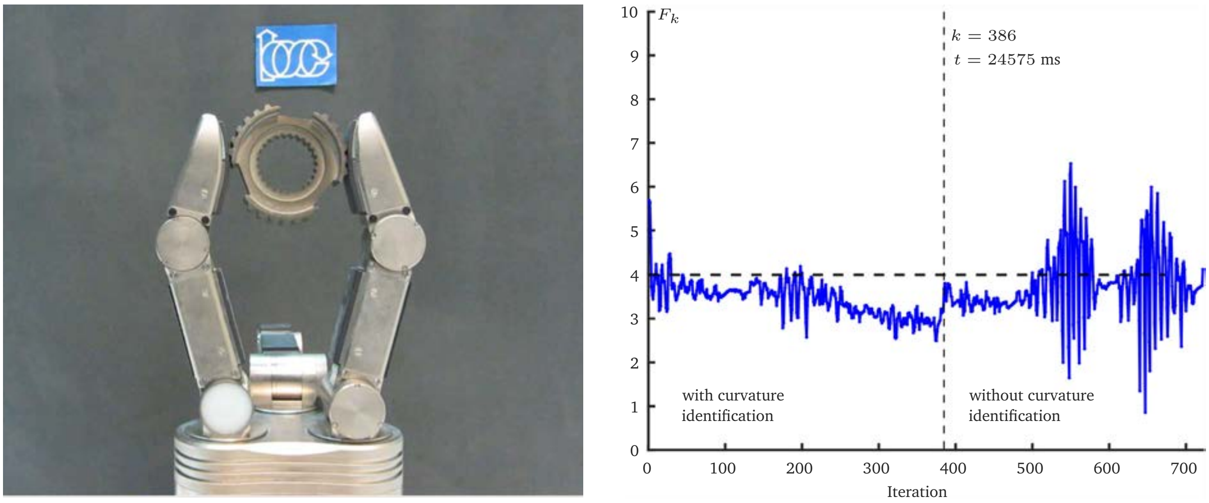


Figure 5.15: Experimental results of the manipulation of Object 4.

5.3 Telemanipulation without haptic feedback

Teleoperation of robots is a challenging subject in applications in which an operator takes decisions and the robots perform actions following the operator commands. Some scenarios where the teleoperation is relevant are: handling hazardous material, telesurgery, underwater vehicles, space robots, mobile robots, among others (Hokayem and Spong 2006; Basañez and Suárez 2009; Hvilshøj et al. 2012). Object dexterous manipulation is a problem that can be included in these fields. The problem addressed in this work is the remote dexterous manipulation of an unknown object. The proposed approach uses a shared telemanipulation schema in which the operator provides high-level commands to the robotic hand to change the orientation of the grasped object, and the robotic system uses the tactile and kinematics information to generate proper set-points for the low-level control of the finger joints (which may be a commercial close controller, as it is our case), besides, the robotic system controls locally the forces and movements in order to avoid object falls.

It must be remarked that the geometric model of the object is unknown and that, during the manipulation, the rotation limits are given by the friction constraints (supervised during the manipulation to avoid object falls) and by the kinematic constraints of the fingers (joint limits). The proposed approach is intended to perform rotations of objects in a plane, like, for instance, to match the orientation of two pieces to do an assembly or to inspect an object. This type of rotation, jointly with planar sliding, is a quite frequent manipulation action in every-day and industrial tasks (Toh et al. 2012).

5.3.1 Manipulation State Machine

The behavior of the telemanipulation process is described by the state machine shown in Figure 5.16, which includes the following states:

$S_{\textbf{init}}$: The hand is in the initial configuration, ready to perform a grasp.

$S_{\textbf{close}}$: The fingers are being closed until reaching a desired grasp force.

$S_{\textbf{open}}$: The fingers are being opened to release the grasped object.

$S_{\textbf{grasp}}$: The object is grasped and the hand is waiting for a command.

$S_{\textbf{turnC}}$: The next configuration for a clockwise rotation of the grasped object is computed.

$S_{\textbf{turnCC}}$: The next configuration for a counterclockwise rotation of the grasped object is computed.

$S_{\textbf{move}}$: The hand executes the next configuration, rotating the object.

The state transitions are determined by:

Keyboard signals K_c, K_o, $K_{\textbf{tc}}$ and $K_{\textbf{tcc}}$: These are four signals generated by the operator using a standard keyboard, each signal is simply generated by pressing a predetermined key. The signals command the four possible actions during the teleoperation: close the fingers (K_c), open the fingers (K_o), rotate clockwise the object(K_{tc}) and rotate counterclockwise the object(K_{tcc}).

Force signal S_F: It is a binary signal that is activated when $F_k > F_d$, where F_k is the grasp force in the k-th step and F_d is the desired grasping force. This condition is reached when the hand has been closed and the object is in contact with the sensor pads.

Friction signal S_G: It is a binary signal that is activated when the friction constraints allow the grasp to firmly hold the object. The binary complement of S_G is represented as $\overline{S_G}$.

The state machine for the teleoperation starts in the state S_{init}, where the hand is waiting for the command K_c to be introduced by the operator in order to close the fingers. When the command K_c is introduced, the system evolves to the state S_{close}. The system remains in the state S_{close} until the measured force on the sensor pads is greater than a desired grasp force, and the object has been actually grasped. Once the grasp is done and the proper force is detected, the system evolves to state S_{grasp}. In the state S_{grasp} the system is waiting for the commands K_{tc}, K_{tcc} or K_o, in order to do a clockwise rotation, a counterclockwise rotation or to open the fingers to release the object, respectively. The finger movements to rotate the object are computed in the states S_{turnC} and S_{turnCC} depending on the direction of rotation indicated by the operator. In these states an autonomous dexterous manipulation algorithm (introduced later in Section 5.3.2) is used to compute the next finger positions (but no movement is executed yet). If reachable finger positions are found and the friction constraints are satisfied, then, the system evolves to the state S_{move}, where the fingers are moved towards their new positions, and, once they are reached, the system comes back to S_{grasp}. If reachable finger positions are not found or they do not satisfy the friction constraints, then, the system comes back to S_{grasp} without passing

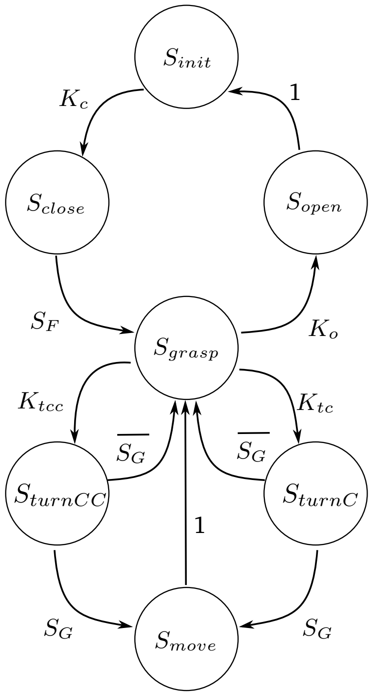


Figure 5.16: State machine of the teleoperation system.

through S_{move}, i.e. without executing any movement. Note that the time-delay that may exist in the communication channel between the operator and the hand controller does not affect the robustness of the manipulation, which is locally assured by the system. When new commands arrive while an action is being executed, i.e. many commands are sent to the hand before the end of executing a previous command, these commands are stored in a stack to be executed once the executions of the movements in progress are finished.

5.3.2 Manipulation Strategy

The manipulation strategy used in this application is the one introduced in Chapter 4, Section 4.3 for the optimization of the object orientation using heuristic methods. This strategy is briefly summarized here remarking the steps performed in each state of the manipulation state machine. The strategy determines a sequence of finger movements to perform the desired rotation of the object, while trying to keep the grasping force F_k as close as possible to the desired contact force F_d. The force F_k applied to the object is controlled by adjusting the distance d_k between the contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$.

The object orientation γ is computed using only proprioceptive information. The object orientation resulting from the initial grasp is considered as the reference orientation, i.e. $\gamma_0 = 0$. The current state of the grasp is determined by the grasping force F_k, the position of the contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$, the distance between contact points d_k and the object orientation γ_k, all of them are computed in state S_{grasp}.

The distance between contact points is adjusted by adding a factor δd, in order to control the force applied on the object. The expected positions of the contact points in the next step are computed considering the hypothesis that the fingers are moved over a circular path whose diameter is given by the distance d_{k+1}. This is actually a general approximation, since the manipulated object is unknown. The target contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$, are computed as,

$$C_{1_{x_{k+1}}} = R_{x_k} - \frac{d_{k+1}}{2}\cos(\gamma_k + \Delta\phi) \tag{5.26}$$

$$C_{1_{y_{k+1}}} = R_{y_k} - \frac{d_{k+1}}{2}\sin(\gamma_k + \Delta\phi) \tag{5.27}$$

$$C_{2_{x_{k+1}}} = R_{x_k} + \frac{d_{k+1}}{2}\cos(\gamma_k + \Delta\phi) \tag{5.28}$$

$$C_{2_{y_{k+1}}} = R_{y_k} + \frac{d_{k+1}}{2}\sin(\gamma_k + \Delta\phi) \tag{5.29}$$

where $\Delta\phi$ is chosen positive to turn the object clockwise or negative to turn the object counterclockwise. $\Delta\phi$ is chosen small enough to assure small movements of the object in each manipulation step. The point R_k is the middle point of the segment between the two contact points $\mathbf{C}_{1_k}$ and $\mathbf{C}_{2_k}$. The expected contact points $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$ are computed in states S_{turnC} and S_{turnCC} depending on the command introduced by the operator to rotate the object clockwise or counterclockwise, respectively.

In order to avoid sliding, each force applied on the object must lie inside the friction cone centered at the direction normal to the sensor surface at the contact point. When this condition is satisfied, the binary signal S_G is activated allowing a transition in the manipulation state machine to the state S_{move}, where the finger motions are executed. When S_G is not activated (i.e. $\overline{S_G}$ is activated) the state changes to S_{grasp} and the system waits for new commands.

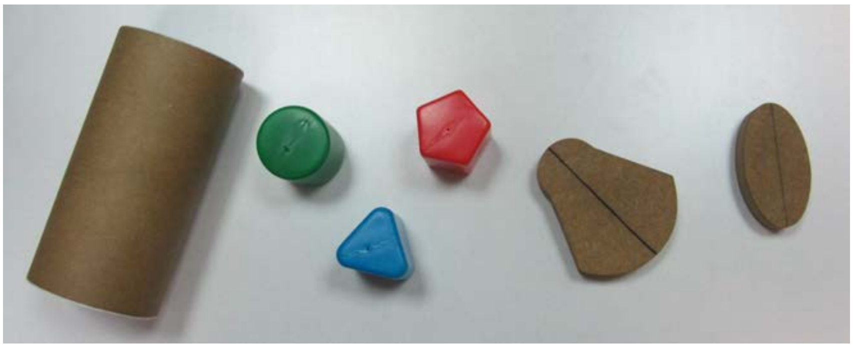


Figure 5.17: Set of telemanipulated objects.

5.3.3 Experimental Validation

The described approach has been implemented using C++ for unknown object telemanipulation with the SDH2 hand. For the experiments, the hand was fixed on a base over a table, but it could be assembled in a robotic arm as well, this is not of relevance for the proposed approach, since the teleoperation of the robotic arm is out of the scope of this approach. The absolute position of the object in the space could be controlled by the robotic arm, and therefore, only the orientation will be controlled by the fingers of the hand. In the experiments, an unknown object is grasped and its orientation is changed by rotating it with the fingers. The operator introduces commands in a very simple and intuitive way using a keyboard to indicate the desired action, i.e. there are commands to close and open the fingers, and to turn the object clockwise or counterclockwise once it has been grasped.

Figure 5.17 shows the set of objects used in the examples below. When an object is located between the two opposed fingers of the SDH2, the operator generates, clicking the proper key in the keyboard, the command to close the fingers until the measured force reaches the desired value $F_d = 2$ N. Note that the initial contact points are unknown, i.e. the initial grasp configuration changes at each execution of the experiment. After this, the operator can manipulate the object, rotating it clockwise or counterclockwise by means of simple teleoperation commands.

The material of the sensor pads is rubber and the material of the objects is wood, cardboard or plastic, thus, we consider a worst case friction coefficient $\mu = 0.4$, which is lower than the friction coefficient between rubber and wood $\mu = 0.7$, rubber and cardboard $\mu = 0.5$, and rubber and plastic $\mu = 0.6$ (Kutz 2005). The constant λ to adjust the distance d_k was set to 1 mm. The orientation variation in each manipulation step was set to $\Delta\phi = 0.25°$.

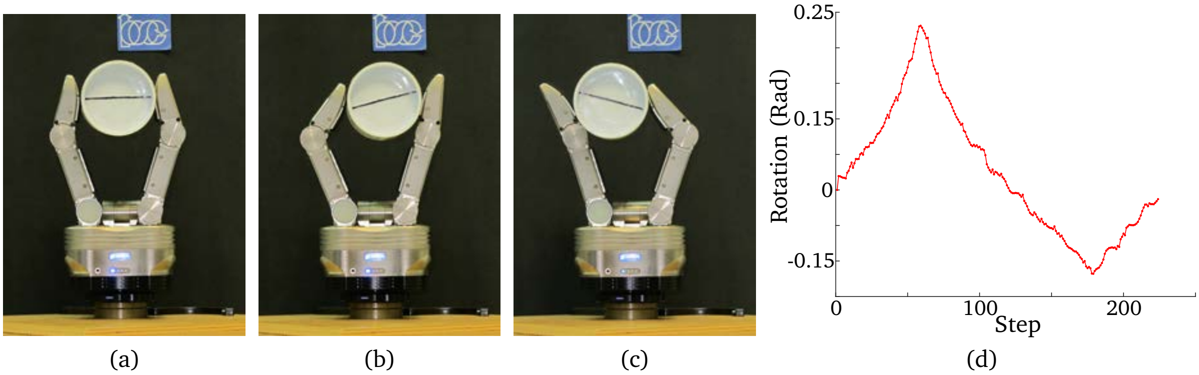


Figure 5.18: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.

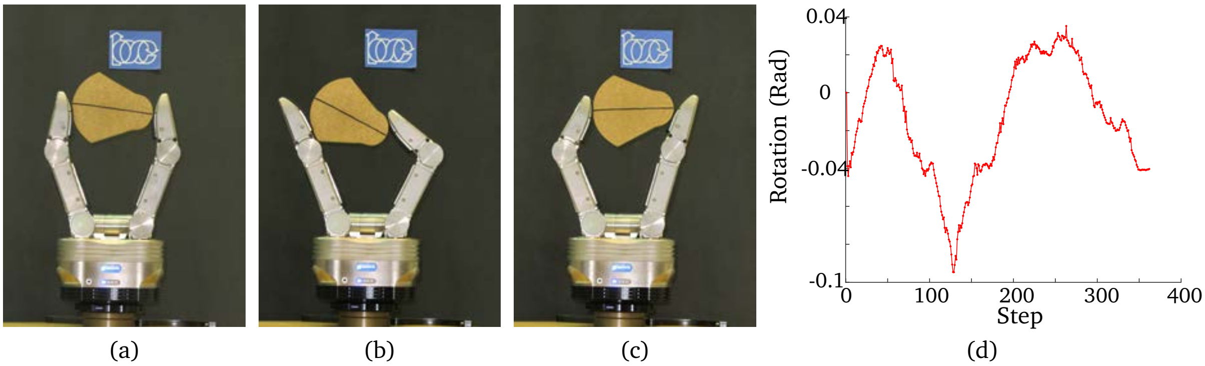
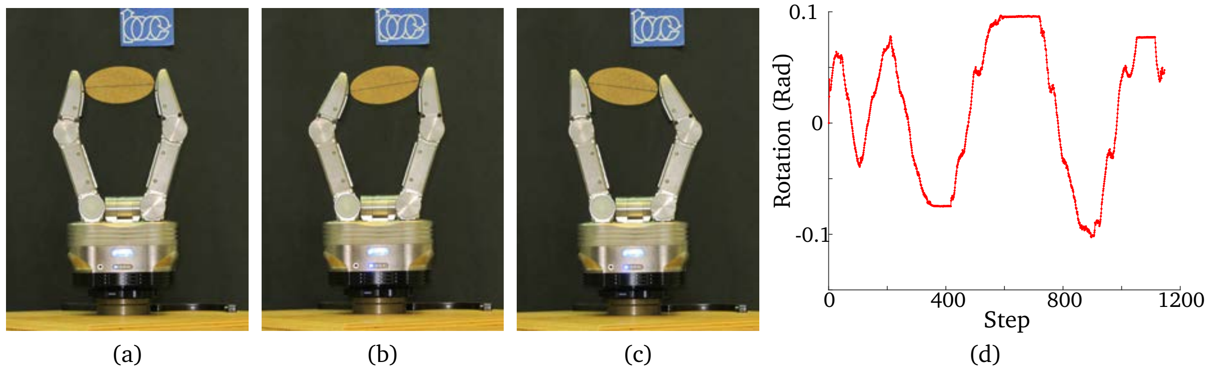


Figure 5.19: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.

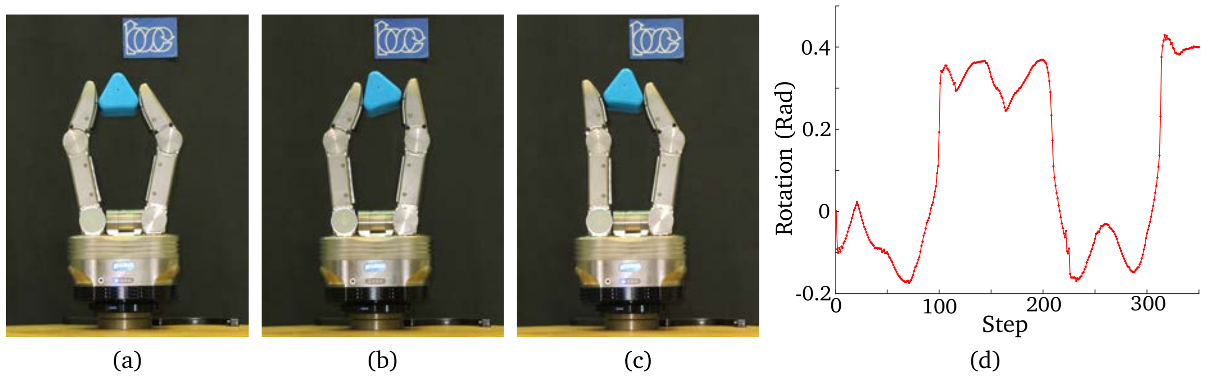
In the examples, each object was grasped and rotated as much as possible in both directions. Figures 5.18 to 5.23 show, for each example, the initial configuration of the grasped object, the configuration when the limits of rotation are reached in counterclockwise and clockwise directions, and a graphical representation of the evolution of the object orientation for each step during the telemanipulation process.

These examples have shown the evolution of the object rotation per step, Figure 5.24 shows an example detailing also the involved variables as a function of time. The commands introduced by the operator to rotate the object in each step are shown in Figure 5.24a and in Figure 5.24b they are shown as a function of time. Figure 5.24c and Figure 5.24d show the variation of the joints for each step and over time during the manipulation, respectively, and Figure 5.24e and Figure 5.24f do so for the object orientation. The time elapsed between the reception of two



(a) (b) (c) (d)

Figure 5.20: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.



(a) (b) (c) (d)

Figure 5.21: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.

commands in the remote system is shown in Figure 5.24g, it includes the system delays and the operator response time. A snapshot when the limit of rotation was reached in clockwise direction is shown in Figure 5.24h. This experiment lasted for 270 steps with a duration of 306.1 s. Note that the clockwise object rotation reached a maximum value -14.88° in the step 175 and remains there until step 186, even when the operator is still sending rotation commands in that sense. The reason for this is that the grasp is on the limit of the friction constraints and therefore, the system did not go further in order to assure the grasp robustness and avoid the object fall. Between steps 49 and 71 and between steps 234 and 253 the operator introduced commands to rotate the object alternately in both senses, and the system responds correctly.

In order to show that the delays in the network do not affect the robustness of the manipulation we include another example in which high and random time delays were included in the

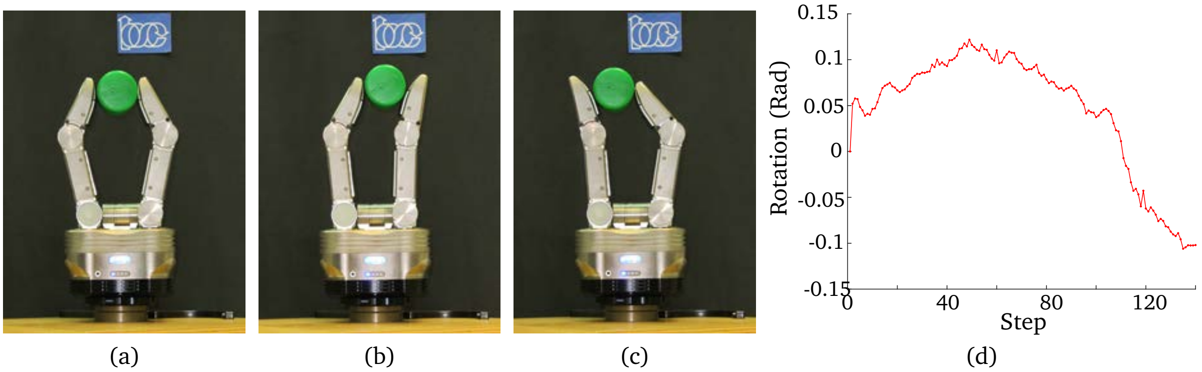


Figure 5.22: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.

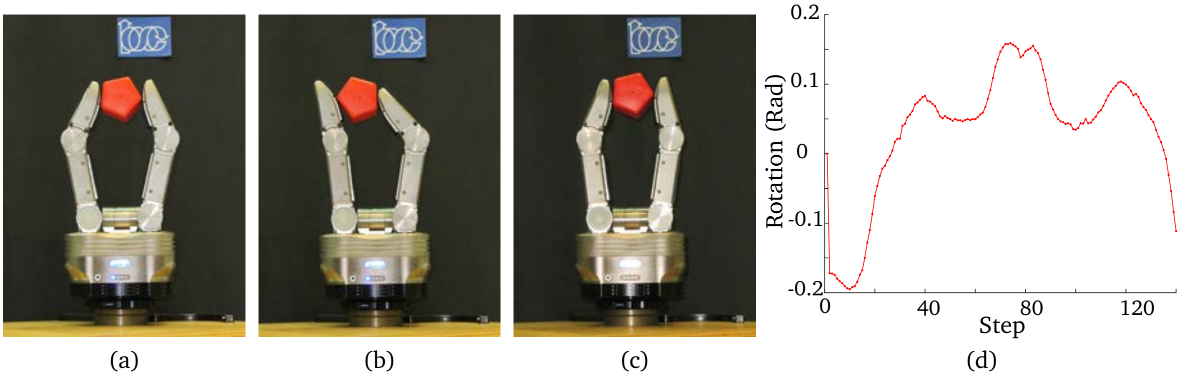
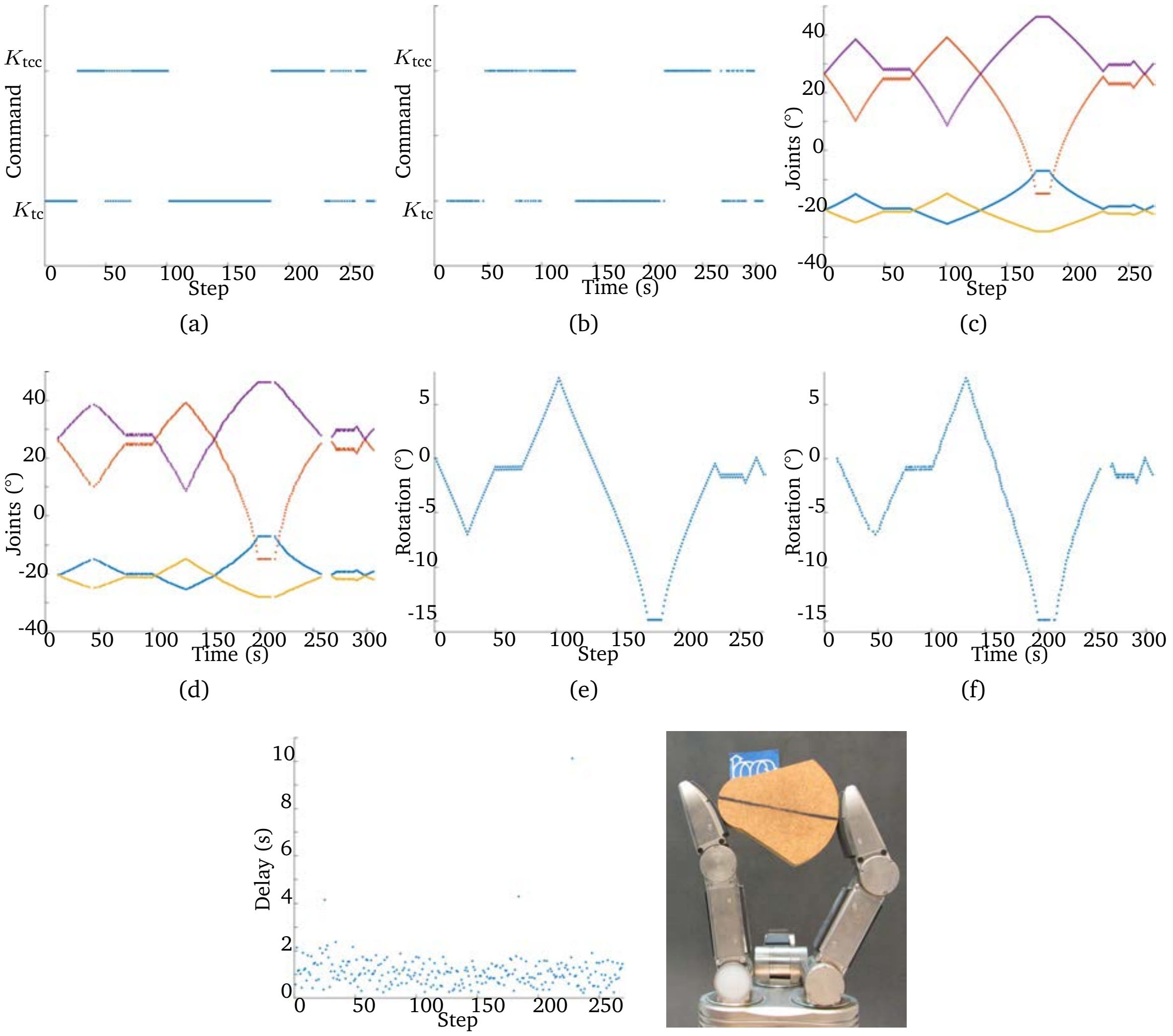


Figure 5.23: (a) Initial grasp configuration. (b) Grasp configuration when the limit of rotation is reached in counterclockwise direction. (c) Grasp configuration when the limit of rotation is reached in clockwise direction. (d) Evolution of the object orientation for each step during the telemanipulation.

communication channel between the local station with the input interface and the remote station with the robotic hand. The delays were randomly generated between 0 and 1.5 s, in both senses of the communication. In this example the aim of the telemanipulation was to rotate the object to -5° with respect to the initial orientation at the grasping time, and the feedback received by the operator in the local station is only the object orientation, no visual feedback was allowed in this case. We asked an untrained person without knowledge of the system response to execute the task. The commands introduced by the operator are shown in Figure 5.25a, per step, and in Figure 5.25b, over time. The task was accomplished after 57 steps with a duration of 114.5 s. The joint values in each step and over time are shown in Figure 5.25c and Figure 5.25d, respectively, and the orientation of the object, per step and over time, are shown in Figure 5.25e and in Figure 5.25f. Note that despite the small variations of the object orientation around -5°, the object manipulation is always robust and stable. The random delays introduced per step in both senses of the communication channel are shown in Figure 5.25g and Figure 5.25h. A video of the execution of these examples is available at https://goo.gl/3QECq1.

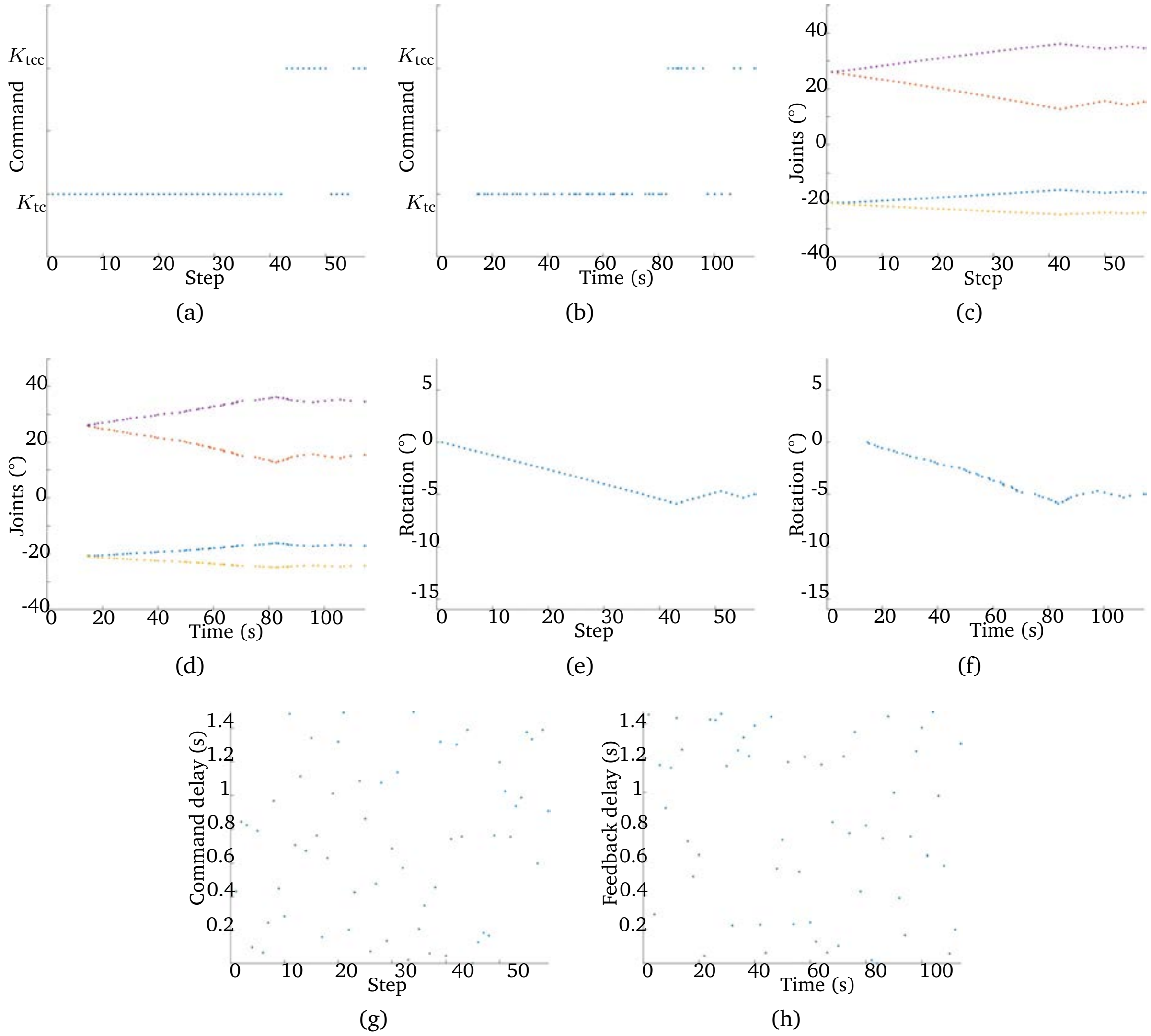


(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 5.24: (a) Introduced commands per step. (b) Introduced commands over time. (c) Evolution of the joints in each step. (d) Evolution of the joints over time. (e) Evolution of the object orientation in each step. (f) Evolution of the object orientation over time. (g) Time elapsed between the reception of two commands in the remote station. (h) Limit of rotation reached in clockwise direction (steps 175 to 186).



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.25: (a) Introduced commands per step. (b) Introduced commands over time. (c) Evolution of the joints in each step. (d) Evolution of the joints over time. (e) Evolution of the object orientation in each step. (f) Evolution of the object orientation over time. (g) Time delays introduced per step. (h) Time delays introduced over time.

5.4 Telemanipulation with Haptic Feedback

Improving teleoperation by adding haptic feedback is another proposed application related with the manipulation of unknown objects. The autonomy level of the robotic system in teleoperation has been addressed following different approaches. On the one side, the operator has the full control of the movements and actions of the robot (a fully teleoperated system), and, on the other side, the control can also be shared between the remote operator and the local robot (Mindell 1993).

Performing dexterous telemanipulation by commanding each joint of a robotic hand as a function of each joint of the human operator hand is a complex problem mainly due three factors:

- The kinematics of the robotic and the human hands are not fully coincident, which easily generates differences in the fingertip positions during the manipulation that may produce the loss of contacts on the object and therefore, the object might fall.
- Even when there exist several proposals, up to our knowledge, there is not any practical device that allows haptic feedback with enough precision at the level of finger joints, thus it is difficult for the operator to feel (in real practical situations) the precise state of the real grasping forces and therefore, how critical the grasp is at any time.
- Time-delays are ubiquitous in telemanipulation scenarios and they affect the stability of the overall closed-loop system.

In this scenario, telecommand the remote robotic hand using high-level commands to autonomously perform the object manipulation within some security margins and providing the human operator with a sense of the general state of the manipulation via haptic feedback is a useful idea. A detailed discussion of the general problems related to teleoperation as well as a description of typical applications was presented in Basañez and Suárez (2009).

Different input interfaces has been used to perform teleoperation, like, in the case of the arms, trackers (Rosell et al. 2014) or wiimotes (Ciobanu et al. 2013), and in the case of the hands, gloves (Rosell et al. 2014; Kukliński et al. 2014), multi touch interfaces (Toh et al. 2012), or video based systems (Ciobanu et al. 2013). One of the main problems in these approaches is the determination of an adequate mapping of the human pose and movements to those of the robot (Meattini et al. 2020).

5.4.1 Approach Overview

The idea is that the human operator uses a haptic device to command the movements that the robotic hand must perform in order to do the in-hand manipulation of an unknown object.

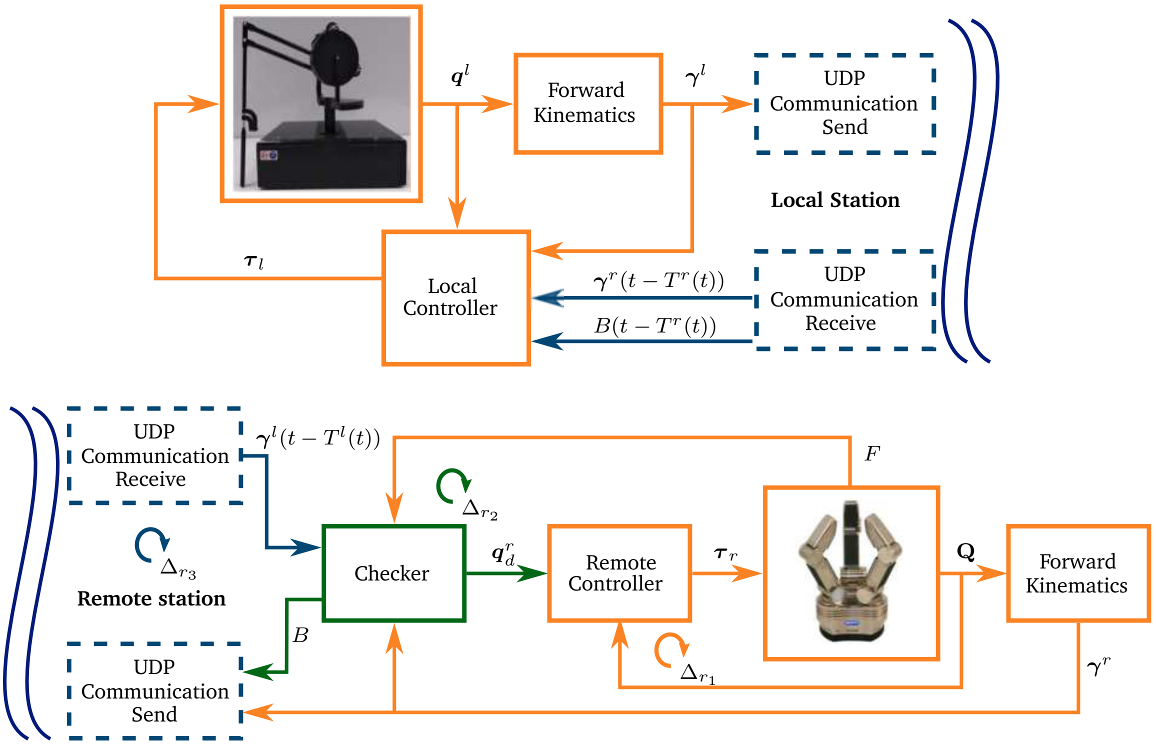


Figure 5.26: Diagram with the main elements of the proposed approach for telemanipulation with haptic feedback.

Figure 5.26 shows a diagram representing the main elements of the proposed system.

In the local station there are three main elements: a *Haptic Device* that, manipulated by the human operator, generates a vector $\boldsymbol{q}^l$ of joint values q_i^l, which is properly transformed into a vector with the desired configuration γ^l of the manipulated object by a *Forward Kinematics* module. γ^l is the information transmitted to the remote station through the communication channel. The third relevant block in the local station is the *Local Controller*, which is in charge of generating the vector $\boldsymbol{\tau}_l$ of torques τ_{l_i} that the haptic device produces as a response to the human movements. The inputs to the Local Controller are two variables generated in the local station, $\boldsymbol{q}^l$ and γ^l, plus another two variables received from the remote station, the object position $\gamma^r(t - T^r(t))$ and the special binary signal $B(t - T^r(t))$ (explained below), with T^r being the delay in the communication channel.

In the remote station there are four main elements, three of them equivalent, in some way, to those in the local station, and a special fourth element that plays a key role in the proposed approach. The first three elements are: a *Robotic Hand* whose configuration $\mathbf{Q}$ is the vector of its joint values q_i^r, a *Forward Kinematics* module that uses $\mathbf{Q}$ to compute the current configuration of the manipulated object γ^r, and a *Remote Controller*, which is in charge of generating the vector of torques $\boldsymbol{\tau}_r$ that commands the robotic hand as a function of the error between its current configuration $\mathbf{Q}$ and a set-point $\boldsymbol{q}_d^r$. This set-point $\boldsymbol{q}_d^r$ is generated by the special fourth element, called *Checker,* as a function of the error between γ^r and the commanded variable $\gamma^l(t - T^l(t))$ received from the local station with a delay T^l. Basically, the Checker computes $\boldsymbol{q}_d^r$ using a manipulation strategy that tends to displace γ^r towards $\gamma^l(t - T^l(t))$ with a reasonable small displacement that ensures the physical robustness of the grasp during the manipulation. Besides, since the object is unknown, it is not possible to predict the variation of the contact forces after each movement of the fingers and, for this reason, the potential increment or decrement of the grasping forces is also considered in the computation of $\boldsymbol{q}_d^r$, trying to keep the forces stable around a predefined value. The Checker is also in charge of detecting when a finger configuration is close to a joint limit as well as of predicting when a manipulation movement (computed according to the used manipulation strategy) may produce a grasping failure that makes the object to flip away or to fall down because a grasping force reaches the limit of the friction cone. In order to communicate these situations to the local station, the Checker generates the binary signal B that indicates whether the expected movement of the hand, computed to follow the command $\gamma^l(t - T^l(t))$ received from the local station, is valid (safe and reachable) or not.

The elements mentioned above work within control loops with different frequencies, which must be taken into account for the correct operation of the whole system. The Remote Controller controls the hand actuators with a sampling period Δ_{r_1}, but it receives the set-points from the Checker with a different period Δ_{r_2} and the Checker receives the information from the communication channel with a different sampling time Δ_{r_3}.

In the next section, this approach is particularized for a specific type of manipulation and details of the particular implementation are also provided.

5.4.2 Particular Implementation

The proposed approach has been implemented with the particular goal of remotely commanding the rotation of an unknown grasped object around a predetermined axis. As stated in previous chapters, this type of manipulation is frequently done by human beings in daily tasks, for instance, to inspect an object. The main features and assumptions of this particular implementation are:

1) The robotic hand uses only two fingers to grasp and rotate the object, as when the human being uses the thumb and index finger to hold and rotate an object.

2) The hand fingers are equipped with tactile sensors that allow the determination of the contact points on the fingertips and an estimation of the grasping forces.

3) One DOF of a haptic device in the local station is used to command the rotation of the grasped object in the remote station.

4) The hand movements executed to grasp the object are outside the scope of this work. The telemanipulation is done starting with the object already hold by the two used fingers.

5) There is no knowledge about the shape or any other physical property of the manipulated object, like the center of mass. Nevertheless, it is assumed that the friction coefficient between the object and the fingertips is above an estimated value. This is a realistic assumption because, even when the object is unknown, the material of the fingertip is known and usually it has a large friction coefficient (like rubber or a similar materials).

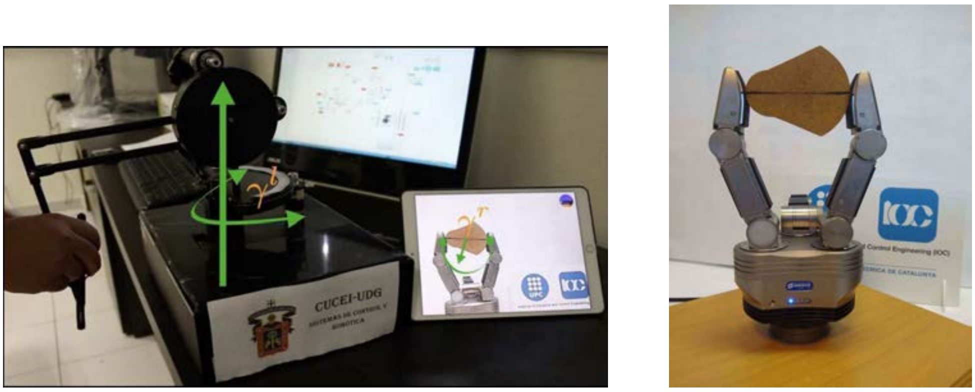
The implementation of the proposed approach for the desired type of manipulation as well as the setup used in the real experimentation is described below.

Experimental setup

For the validation of the proposed approach, transatlantic experiments have been carried on between the local station located at the Robotics lab of the CUCEI-UDG in Guadalajara, Mexico, and the remote station located at the Robotics lab of the IOC-UPC in Barcelona, Spain. The data transmission between the local and remote stations is done through UDP ports, and the main hardware is the following:

- A PHANTOM Premium 1.5 High Force® is used as the haptic device. It provides three DOF with positional sensing and force feedback. This device is commercially available from 3D SYSTEMS®. The communication at the local station between Simulink® and the haptic is done using the homemade library *PhanTorque_3Dof*, which is publicly available at http://shorturl.at/kAE17.

- The Schunk Dexterous Hand (SDH2), introduced in Chapter 3 Section 3.2, is used as robotic hand, which is suitable for both, service robotics and industrial applications. In



(a)

(b)

Figure 5.27: (a) Testbed at local station located in Guadalajara, Mexico. (b) Testbed at the remote station located in Barcelona, Spain, with the SDH2 hand using two fingers opposite to each other to grasp an object.

this work, the fingertips of the two coupled fingers are used to grasp and manipulate an unknown object, performing a tip-pinch grasp (Feix et al. 2016).

Local station

As mentioned is Subsection 5.4.1 and illustrated in Figure 5.26, the local station has three elements: a haptic device, a Forward Kinematics module and the Local Controller. One DOF q_1 of the haptic device is used to command the object rotation γ^l. For this reason the Forward Kinematics module is straightforward, and only a scaling factor is introduced to take advantage of the haptic device workspace, establishing the relation between these variables as $\gamma^l = 0.4\, q^l$.

Regarding the Local Controller, it is implemented according to the following control law (Aldana et al. 2018),

$$\tau^l = -k^l \left(\gamma^l - \theta^l\right) \tag{5.30}$$

where the gain k^l is any positive number and θ^l stands for the generalized coordinate of the controller, that is obtained by solving the following second order ordinary-differential-equation

$$\ddot{\theta}^l = -k^l(\theta^l - \gamma^l) - d^l\dot{\theta}^l - p^l(\theta^l - \gamma^r(t - T^r(t))) \tag{5.31}$$

with $d^l, p^l > 0$ being the damping and the proportional gains.

Since the SDH2 hand does not allow the user to set a desired torque, the Remote Controller is implemented as a set-point position-based scheme that works with the proprietary SDH2 hand controller. For this application, it is assumed that such controller is a simple proportional plus damping scheme with control gains $d^l \approx d^r$ and $p^l \approx p^r$. In order for the controller to be robust to time-delays, the control gains have to satisfy

$$d^l > \frac{1}{2}(^\circ T^l + {}^\circ T^r)p^l \tag{5.32}$$

where $^\circ T^l$ and $^\circ T^r$ are the bounds of the time-delays (Aldana et al. 2018).

This proposal does not require velocity measurements and, provided that sufficiently large damping is injected, i.e. Eq. (5.32) holds, this scheme guarantees that $\lim\limits_{t\to\infty} \gamma^l(t) - \gamma^r(t) = 0$ (Aldana et al. 2018; Nuño et al. 2018).

Remote station

The main element of the remote station is the Checker, which has two main duties: a) compute the set-point $\boldsymbol{q}_d^r$ for the Remote Controller to move the hand fingers according to the information received from the local station, and, b) generate the binary signal B to indicate whether the computed set-point is not reachable or unsafe, because, respectively, the fingers would reach joint limits or the grasping forces would be close to the limits of the friction cone and therefore, the object could fall down or flip away.

Duty (a) is based on the manipulation strategy for the optimization of the object orientation using gradient-based methods introduced in Chapter 4, Section 4.4, and briefly summarized here. First, the current state of the grasp is determined by computing the current absolute positions of the contact points $\mathbf{C}_{i_k}$, $i \in \{1, 2\}$, and the current grasping forces F_{i_k}. $\mathbf{C}_{i_k}$, $i \in \{1, 2\}$, are directly computed using the tactile information (to identify the contact points on the sensor pads) and the hand forward kinematics. In the case of a two-finger grasp, the contact forces F_{i_k} should have the same magnitude and opposite direction, but due to different sources of uncertainties and measurement errors this may not be exactly true. In order to minimize errors, F_{i_k}, $i \in \{1, 2\}$, are considered to have the right direction defined by the fingertip contact points and a magnitude F_k equal to the average of the measured values at each fingertip.

Then, new contact points $\mathbf{C}_{i_{k+1}}$ are computed aiming for a proper object rotation and adjustment of F_{k+1}. For this matter, two auxiliary points $\mathbf{C}^*_{1_{k+1}}$ and $\mathbf{C}^*_{2_{k+1}}$ are computed by displacing $\mathbf{C}_{i_k}$ along a circular path with diameter d_k and centered at $\boldsymbol{R}_k$ (middle point between $\mathbf{C}_{1_k}$ and $\mathbf{C}_{_k}$),

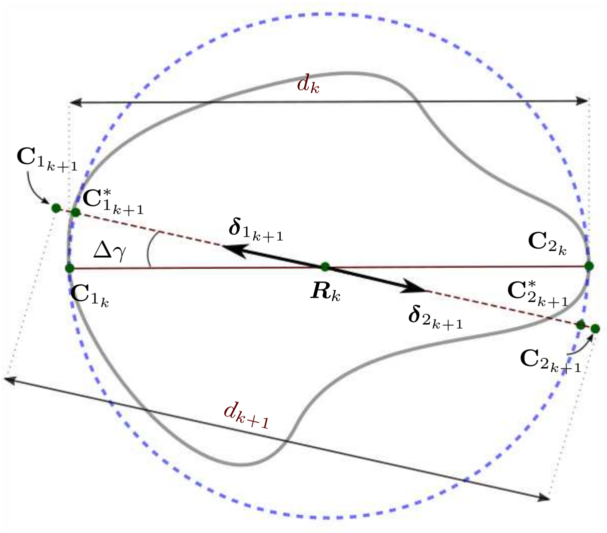


Figure 5.28: Example of computation of $\mathbf{C}_{i_{k+1}}$ using the auxiliary points $\mathbf{C}^*_{i_{k+1}}$ and the adjustment of the distance d_k.

as shown in Figure 5.28, i.e.

$$C^*_{1_{k+1_x}} = R_{k_x} - \frac{d_k}{2}\cos(\phi + s_r\Delta\gamma) \tag{5.33}$$

$$C^*_{1_{k+1_y}} = R_{k_y} - \frac{d_k}{2}\sin(\phi + s_r\Delta\gamma) \tag{5.34}$$

$$C^*_{2_{k+1_x}} = R_{k_x} + \frac{d_k}{2}\cos(\phi + s_r\Delta\gamma) \tag{5.35}$$

$$C^*_{2_{k+1_y}} = R_{k_y} + \frac{d_k}{2}\sin(\phi + s_r\Delta\gamma) \tag{5.36}$$

where d_k is the Euclidean distance between the contact points $\mathbf{C}_{1_k}{}^{\cdot}\,\Delta\gamma$ is an empirical value chosen small enough to assure small movements of the object on each manipulation step k, and s_r depends on the difference between the current object orientation $\boldsymbol{\gamma}^r$ (computed in the remote station) and the desired orientation $\boldsymbol{\gamma}^l(t - T^l(t))$ (received from the local station), as

$$s_r = -\tanh(\boldsymbol{\gamma}^r - \boldsymbol{\gamma}^l(t - T^l(t))) \tag{5.37}$$

Note that the direction of the displacement of $\mathbf{C}^*_{i_{k+1}}$ with respect to $\mathbf{C}_{i_k}$ corresponds to the desired sense of rotation of the object given by the sign of $\boldsymbol{\gamma}^r - \boldsymbol{\gamma}^l(t - T^l(t))$.

Any finger movement may alter the grasping force F_k, which must remain within a threshold around a desired value F_d, thus in each iteration F_k is adjusted by adjusting d_k as

$$d_{k+1} = d_k + \Delta d_k \tag{5.38}$$

with Δd_k depending on e_{f_k} as

$$\Delta d_k = \begin{cases} 2\lambda(\|e_{f_k}\| + e_{f_k}^2) & \text{if } e_{f_k} \leq 0 \\ -\lambda e_{f_k} & \text{if } e_{f_k} > 0 \end{cases} \tag{5.39}$$

being λ a predefined constant, empirically obtained. As discussed in Chapter 4, the reason for the different gain is the same than in other manipulation strategies, a potential fall of the object is considered more critical that a potential application of large grasping forces.

The next (expected) contact points $\mathbf{C}_{i_{k+1}}$ that intend to change the object orientation γ^r and also adjust the force F_k, are computed as,

$$\mathbf{C}_{i_{k+1}} = \boldsymbol{R}_k + \frac{d_{k+1}}{2}\boldsymbol{\delta}_{i_{k+1}} \tag{5.40}$$

where $\boldsymbol{\delta}_{i_{k+1}}$ is the unitary vector from $\boldsymbol{R}_k$ to $\mathbf{C}^*_{i_{k+1}}$ (see Fig. 5.28).

Finally, the hand configuration $\boldsymbol{q}^r_{d_{k+1}}$ is obtained from the points $\mathbf{C}_{i_{k+1}}$ using the inverse kinematics of the fingers, i.e. $\boldsymbol{q}^r_{d_{k+1}} = \text{IK}(\mathbf{C}_{1_{k+1}}, \mathbf{C}_{2_{k+1}})$.

Duty (b) of the Checker is solved as follows. The binary signal is set as $B = 0$ when $\boldsymbol{q}^r_{d_{k+1}}$ is reachable and safe to keep the grasp, and therefore, the manipulation movement can be safely done, and $B = 1$ otherwise. Determining whether $\boldsymbol{q}^r_{d_{k+1}}$ is reachable is straightforward, it simply means that $\boldsymbol{q}^r_{d_{k+1}} = \text{IK}(\mathbf{C}_{1_{k+1}}, \mathbf{C}_{2_{k+1}})$ returns a valid hand configuration. Determining whether $\boldsymbol{q}^r_{d_{k+1}}$ produces a safe grasp is done by checking whether the expected applied force $F_{i_{k+1}}$ lies inside the friction cone. As stated above, the grasping forces $F_{i_{k+1}}$ are aligned with the segment defined by $\mathbf{C}_{1_{k+1}}$ and $\mathbf{C}_{2_{k+1}}$, so their directions are known, and the direction $\hat{n}_i$ normal to each fingertip at $\mathbf{C}_{1_{k+1}}$ is also known (the shape of the fingertip is known). Then, if the angle β_i between $F_{i_{k+1}}$ and $\hat{n}_i$ is smaller that the angle of the friction cone, $\alpha = \arctan(\mu)$, being μ the friction coefficient, the grasp can be considered safe, i.e. $B = 0$ if $\beta_i < \alpha$. Note that the actual friction coefficient is not known, neither explicitly computed during the manipulation, thus, a given minimum value is assumed in order to determine B. This is a reasonable assumption, since the material of the fingertips (rubber) is known and it produces a reasonable μ for most of the unknown manipulated objects. Algorithm 5.2 summarizes the steps done by the Checker.

Algorithm 5.2 Steps done in the Checker.

```
Require: $F_{i_k}$, $F_d$, $\gamma^r$, $\gamma^l(t - T^l(t))$
Ensure: $\mathbf{Q}_{k+1}$, $B$

  procedure CHECKER
      $k \leftarrow 0$
      loop
          Check for new values of $\gamma^l(t - T^l(t))$
          Compute $\mathbf{C}_{i_k}$ using FK
          Compute $F_k$
          Compute $d_k$ using (4.7)
          Compute $\mathbf{C}^*_{i_{k+1}}$ using (5.33 to 5.36)
          Compute $d_{k+1}$ using (5.38)
          Compute $\mathbf{C}_{i_{k+1}}$ using (5.40)
          Compute $\mathbf{Q}_{k+1}$ using IK($\mathbf{C}_{1_{k+1}}$,$\mathbf{C}_{2_{k+1}}$)
          if $\mathbf{Q}_{k+1}$ is reachable and safe then
              Send $\mathbf{Q}_{k+1}$ to Remote Controller
              $B \leftarrow 0$
          else
              $B \leftarrow 1$
          end if
          Send $B$ to local station
          $k \leftarrow k + 1$
      end loop
  end procedure
```

Another element of the remote station is the Forward Kinematics module, which is in charge of computing the current orientation of the object γ^r from the current configurations of the fingers $\mathbf{Q}_{k+1}$. Since the object is unknown, the object orientation cannot be fully determined without using external sensors (like, for instance, a vision system), but, as shown by Montaño and Suárez (2018c), it can be estimated with enough precision using the following expression initially proposed for fingertips with circular shape,

$$\gamma^r \approx \frac{R}{d_k}\left(\sum_{j=1}^{n_1}(q_{1j_{\gamma_0}} - q_{1\mathrm{j}_k}) - \sum_{j=1}^{n_2}(q_{2\mathrm{j}_{\gamma_0}} - q_{2j_k})\right) \tag{5.41}$$

5.4.3 Experimental Validation

The grasped object shown in Figure 5.29 is used in the experiments described below, using the following gains and parameters. At the local station the Local Controller gains are $k^l = 15$, $d^l = 8$ and $p^l = 2$. At the remote station the desired grasping force is set to $F_d = 5$ N, the constant to adjust the distance between contact points is set to $\lambda = 0.1$, the constant $\Delta\gamma$ is set to 0.25°, and the minimum friction coefficient is assumed to be $\mu = 0.4$ (i.e. $\alpha \approx 0.38$ rad). Figure 5.29 shows snapshots of the telemanipulation experimentation. A video showing the system performance can be found in https://sir.upc.edu/projects/dexterous_telemanipulation/.

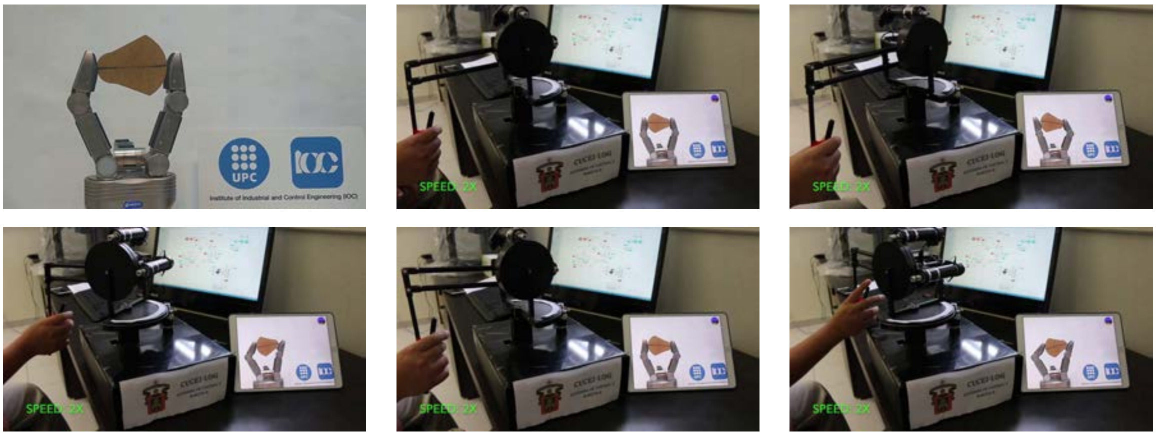


Figure 5.29: Snapshots of the telemanipulation experimentation from Guadalajara to Barcelona.

Figure 5.30 shows the current orientation of the object (γ^r) and the desired orientation (γ^l) commanded by the local operator through the first joint of the haptic device. It can be observed that the object follows the desired orientation and, when it is pushed to perform a non valid movement, it keeps the current orientation until a valid movements is demanded, this effect can be clearly appreciated in the intervals 12 s to 22 s, 38 s to 48 s and 65 s to 72 s. Figure 5.31 shows the torque applied to the haptic device and the behavior of the binary signal B. It can be observed that when the non-valid movement signal is received (i.e. $B = 1$), an increasing torque is applied to the haptic device emulating a wall for the local operator, this allows the local operator to realize that the movement of the object is reaching a limit. Figure 5.32 shows the friction cone angle α and the angles β_1 and β_2 that indicate whether the movement defined by $\mathbf{Q}_{k+1}$ is safe. Angles β_i are computed using the kinematic information of the target hand configuration $\mathbf{Q}_{k+1}$ and if one of their values exceeds the friction cone, then, the movement of the fingers is not executed and the binary signal B is set to 1 to inform of this event to the local station. When $\|\beta_1\|$ and $\|\beta_2\|$ are smaller than $\|\alpha\|$, the binary signal $B = 0$ indicates to the local station that the movement can be executed safely. Figure 5.33 shows the grasping force F_k and the desired force F_d. The variable time-delay between the local and remote stations along the experiment can be observed in Figure 5.34, being the bound values $^\circ T^l \approx {^\circ T^r} \approx 0.65$ s.

Figures 5.35 to 5.38 show the results of a second experiment. The variable time-delays are similar to those shown in Figure 5.34 so they are not shown. Note that in this case the local operator demands twice a non-valid movement, in the intervals 34s to 40s and 85s to 90s.

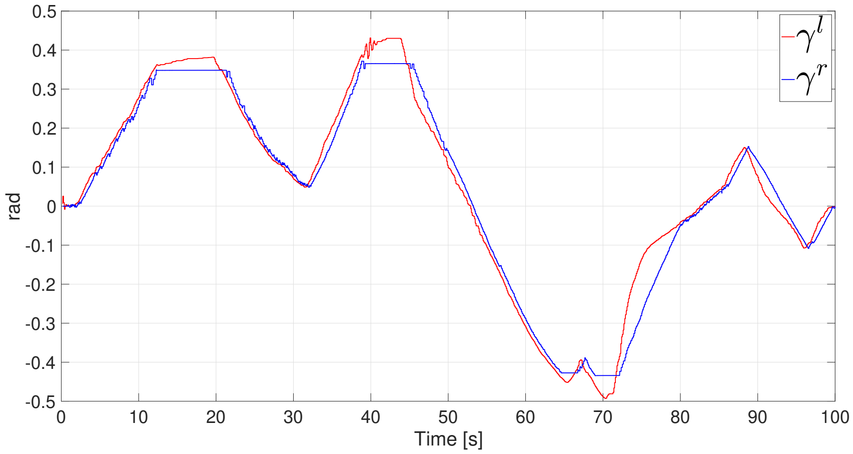


Figure 5.30: Local orientation γ^l and remote object orientation γ^r.

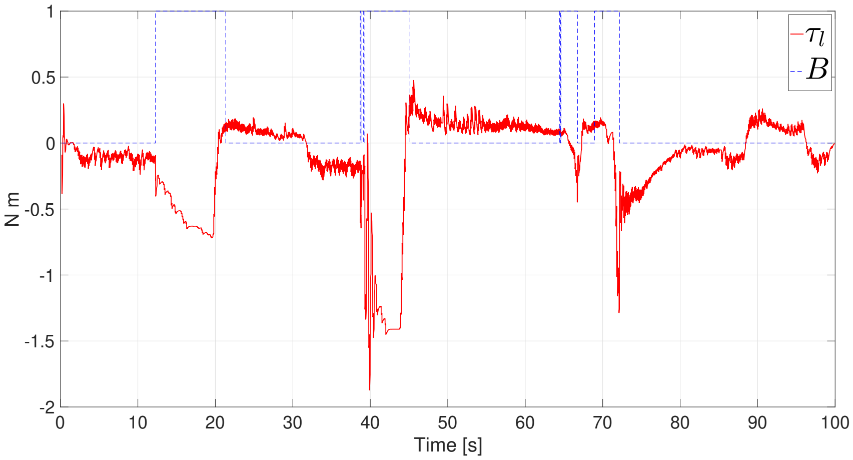


Figure 5.31: Local torque τ^l and the signal B indicating valid ($B = 0$) and non valid ($B = 1$) movements.

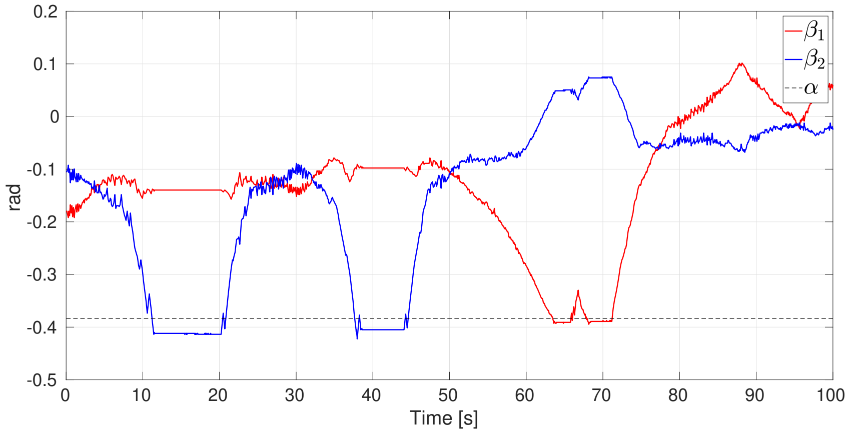


Figure 5.32: Angles β_1 and β_2, and the friction cone angle α.

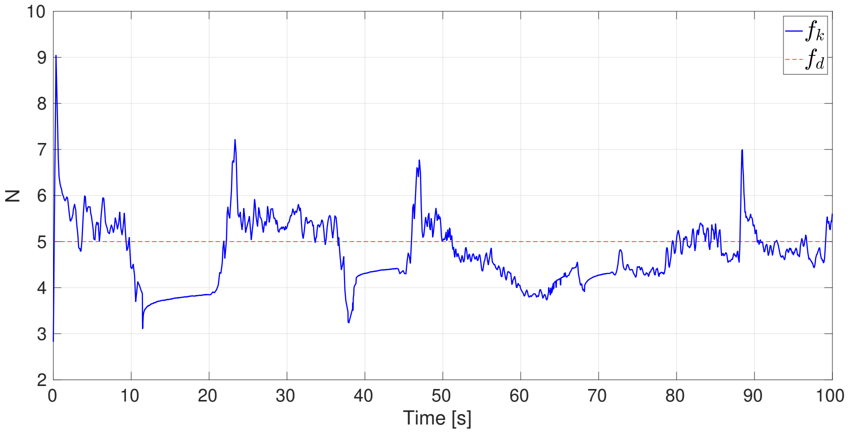


Figure 5.33: Grasping force F_k and desired force F_d.

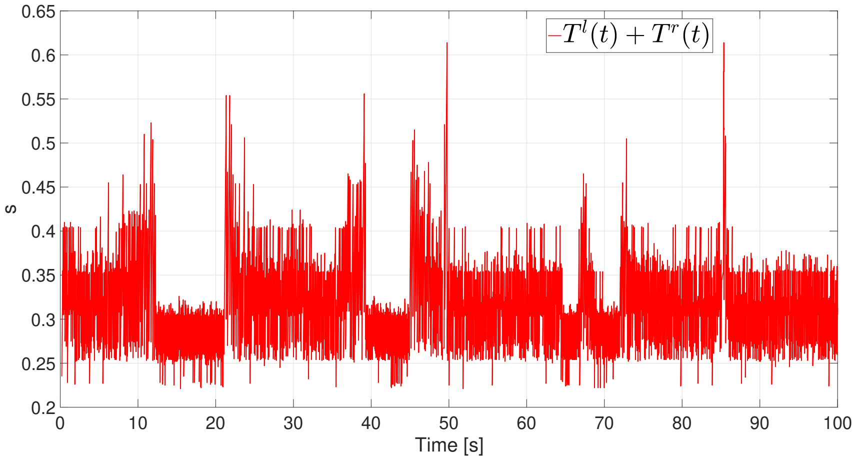


Figure 5.34: Time-delay in the communication channel $T^l + T^r$.

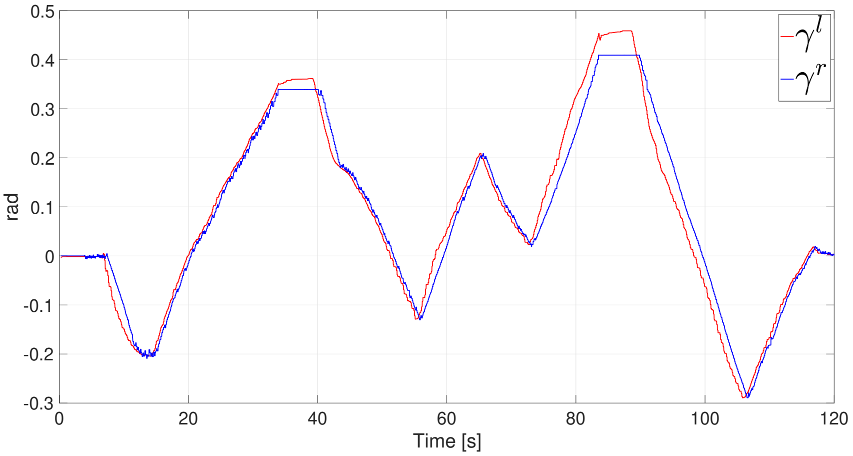


Figure 5.35: Local orientation γ^l and remote object orientation γ^r.

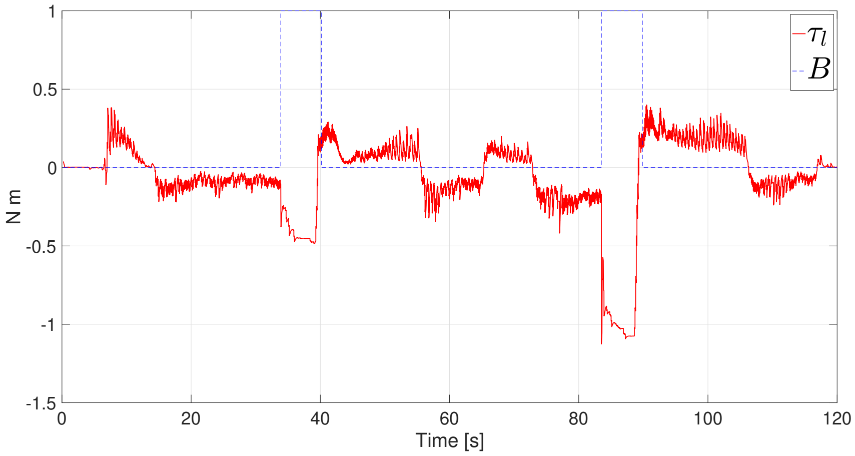


Figure 5.36: Local torque τ^l and the signal B indicating valid ($B = 0$) and non valid ($B = 1$) movements.

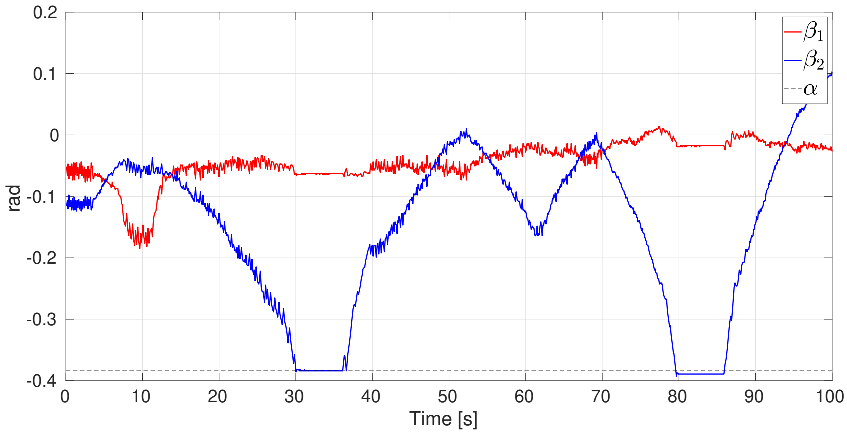


Figure 5.37: Angles β_1 and β_2, and the friction cone angle α.

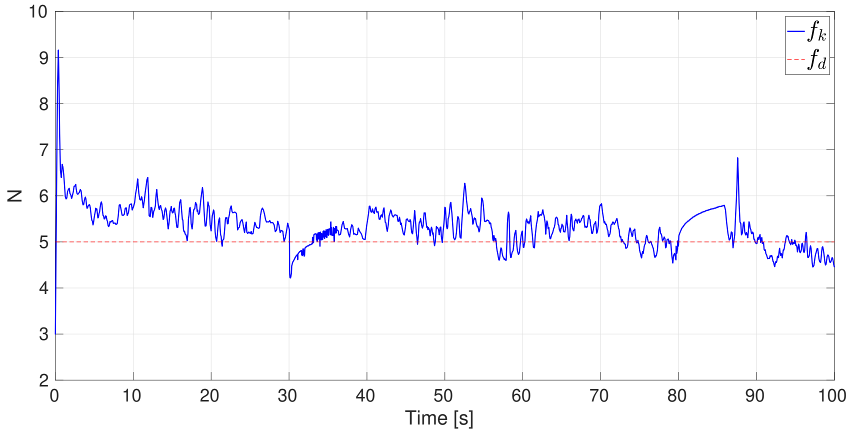


Figure 5.38: Grasping force F_k and desired force F_d.

"To surrender to ignorance and call it God has always been premature, and it remains premature today."

Isaac Asimov (1920 - 1992)

6

Conclusions and Future Research Directions

As in any research, the work presented in this thesis raises more questions than it solves. This chapter presents the conclusions, summarizes the contributions of the thesis, discusses the future research directions and finally lists the publications already presented as a result of this work.

6.1 Contributions of the thesis

This thesis is focused on the manipulation of unknown objects using tactile information as feedback. The manipulation starts from a given initial grasp and keeps the contact between each finger and the object (i.e. finger gaiting is not considered) while the grasping forces are kept within the desired range preventing the object from falling. Three quality indexes were chosen to evaluate the quality of the configuration of the hand, of the grasp, and of the object configuration related to a given task. The tactile sensors are used to detect the contact points that provide information about the position and orientation of the object. Using this information, the fingers are able to rotate the object until the contact forces reach the friction cone limits or the fingers reach the mechanical limit of movement. Furthermore, the measured contact forces are used to adjust the forces that the fingers apply on the object during the manipulation, adapting them to the object shape. In this context, the main contributions of this work are:

- The proposal of three manipulation strategies based on heuristics, imitating the human behavior, to improve three quality indexes related to the configuration of the hand, the quality of the grasp, and the configuration of the object. The manipulation is done as a reactive procedure considering only the tactile information and the kinematic data of the hand measured during the manipulation itself.

- The proposal of a relationship between the quality indexes and the finger joints, i.e. the indexes are expressed as functions of the hand joint values. Using these relationships, a simple procedure to determine on-line the hand movements to manipulate the object following the gradient of these functions is also proposed.
- The proposal of an approach that, based on the commanded positions of the fingertips, defines a set of virtual contact points between the fingertips and the object (without caring about the positions of the real contact points) that allows the object manipulation using a relatively simple geometric reasoning.

In addition to the manipulation strategies mentioned above, other original contributions of this work are the implementation of:

- A C++ library, *WEISSlib*, to configure, manage and read information from the Weiss tactile sensors.
- A C++ library, *AHandlib*, including a PID controller to command the Allegro hand.
- An application, based on the ROS framework, for the visualization of the robotic hand, the forces measured by the tactile sensors and the contact forces computed from the joint torques, while the hand interacts with the environment.

The proposed manipulation approaches were successfully tested in real experimentation using two robotic hands with different features, the SDH2 that is a three-fingered industrial gripper, and the Allegro hand that is a four-fingered anthropomorphic hand. In addition, the following practical approaches were developed applying the proposed manipulation strategies and the contact and kinematic data obtained during the manipulation process:

- An approach for the object shape reconstruction and the object recognition. The object shape is reconstructed by geometrical reasoning about the location of the contact points and the kinematic data of the hand. Then, distance invariants are computed using the reconstructed object shape. The reconstructed object shape and the distance invariants form a signature of the object, which is compared with a database of models of the manipulated objects.

- An approach to improve the grasping forces based on the recognition of the local curvature of the object surface at the contacted points. The computation of the finger movements using a model of the object closer to the real one (obtained during the manipulation) allows the reduction of the error in the contact forces during the manipulation.
- Two dexterous telemanipulation approaches, with and without haptic feedback, robust against variable time-delays in the communication channel. The approaches employ tactile and kinematic information to manipulate an unknown object with the commanded orientation given remotely by a human operator. When haptic feedback is used, the human operator receives force feedback information regarding the object manipulation and the feasibility of the commanded movements.

The manipulated objects in the experimental evaluation of the proposed manipulation strategies were selected having different shapes, with small and large curvatures and smooth and irregular boundaries. The proposed manipulation strategies can deal with any type of objects, as long as they fit between the fingers of the hand. These strategies are independent of the hardware, so they can be implemented for any robotic hand with tactile feedback that may be provided by current commercial tactile sensors. It must be remarked that in the procedures proposed to compute the hand movements only the inverse kinematic of the hand is necessary, thus, independently of the hand complexity the only requirement to apply the strategies is to know the hand kinematics.

To the best of the author knowledge, other works dealing with dexterous manipulation of unknown objects do not solve problems equivalent to the ones addressed in this thesis. Another types of manipulation tasks are performed in these works or the software implementation is not available, therefore, there is not enough information to compare them with the approaches proposed in this thesis. For instance, Shaw-Cortez et al. (2018) proposed a robust discrete-time controller to manipulate unknown objects, ensuring that the object does not slip during the manipulation. In order to define the manipulation task, they define a reference frame (task frame) depending on the hand kinematic and the main goal of the proposed controller is to minimize the position and orientation error of the task frame. In the experimentation they rotate an object without using real tactile information, instead of that, they simulate the tactile sensor signals. Another work proposed an approach based on machine learning and visual feedback. A fixed point in the object surface is defined using a marker and the manipulation task emulates handwriting by controlling the movements of the fingers by tracking the fixed reference point on the object, i.e. the reference point on the object is considered as the pen and the hand must move it following a defined path for each letter (Morgan et al. 2019). The used robotic hand is an underactuated hand without tactile feedback and visual markers on the fingertips are used to obtain the fingertip locations.

This thesis also opens new research problems that require further exploration, as described below.

6.2 Future works

As discussed in Chapter 2, in the last decade, the development of robotic hardware has had significant advances, both in mechanical hands and tactile sensors, thus, it can be considered that the development of new manipulation strategies is currently necessary to exploit the exiting hardware. Even when the approaches proposed in this thesis solve some problems related to the dexterous manipulation of unknown objects, additional work is still required in this field. The following are some open research lines that could be treated in future work:

- The improvement of the force controller used to keep the contact force around a desired value, like, for instance, in the case of the Allegro hand, exploring a hybrid controller that combines position and force.
- The inclusion of finger gaiting like a possible manipulation mode, allowing to manipulate the object with a wider range of movements.
- The extension of the proposed approaches to allow the rotation of the object around any arbitrary axis in the space.
- The combination of the proposed manipulation strategies with planning-based or learning-based approaches, to extend the manipulation capabilities of the robotic system.
- The inclusion in the object identification approaches of the raw data generated by the tactile sensors during the manipulation, considering in this way more detailed information about the manipulated object.
- The implementation of the proposed approaches considering different types of tactile sensors.

- The extension of the teleoperation approaches to use of more than two fingers of a force-controlled hand.
- The integration of the dexterous object telemanipulation with the teleoperation of the robotic arm.

6.3 Derived publications

The following published papers were derived from this thesis.

Refereed Journals

- Montaño and Suárez (2017), Robust dexterous telemanipulation following object-orientation commands. *Industrial Robot: An International Journal 44*(5), 648–657.
  The paper contributions were described in Chapter 5.
- Montaño and Suárez (2018c), Manipulation of Unknown Objects to Improve the Grasp Quality Using Tactile Information. *Sensors 18*(5), 1412.
  The paper contributions were described in Chapter 4.
- Montaño and Suárez (2019a), Dexterous Manipulation of Unknown Objects Using Virtual Contact Points. *Robotics 8*(4), 86.
  The paper contributions were described in Chapter 4.

International Refereed conferences

- Montaño and Suárez (2013b), Object shape reconstruction based on the object manipulation. In *16th International Conference on Advanced Robotics (ICAR 2013)*, Montevideo, Uruguay, pp. 1–6.
  The paper contributions were described in Chapter 5.
- Montaño and Suárez (2014), Getting comfortable hand configurations while manipulating an object. In *19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2014)*, Barcelona, Spain, pp. 1–8.
  The paper contributions were described in Chapter 4.

- Montaño and Suárez (2015), Unknown object manipulation based on tactile information. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, pp. 5642–5647.
  The paper contributions were described in Chapter 4.

- Montaño and Suárez (2016), Commanding the Object Orientation Using Dexterous Manipulation. In L. Reis, A. Moreira, P. Lima, L. Montano, and V. Muñoz-Martinez (Eds.), *Robot 2015: Second Iberian Robotics Conference. Advances in Intelligent Systems and Computing*, Volume 418, pp. 69–79. Springer.
  The paper contributions were described in Chapter 5.

- Montaño and Suárez (2018a), Improving Grasping Forces During the Manipulation of Unknown Objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, pp. 3490–3495, 2018.
  The paper contributions were described in Chapter 5.

- Montaño and Suárez (2019d), Model-free in-hand manipulation based on commanded virtual contact points. In *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2019)*, Zaragoza, Spain, pp. 586–592, 2019.
  The paper contributions were described in Chapter 4.

- Montaño, Suárez, Aldana, and Nuño (2020), Bilateral telemanipulation of unknown objects using remote dexterous in-hand manipulation strategies. In *21st International Federation of Automatic Control World Congress (IFAC-V 2020)*, Berlin, Germany.
  The paper contributions were described in Chapter 5.

National Refereed conferences

- Montaño and Suárez (2013a), Manipulación de objetos con dos dedos usando información táctil. In *XXXIV Jornadas de Automática*, Terrassa, Spain, pp. 618–625.
  The paper contributions were described in Chapter 3.

- Montaño and Suárez (2018b), Manipulación de Objetos Desconocidos Analizando Localmente su Forma para Optimizar las Fuerzas de Prensión. In *XXXIX Jornadas de Automática*, Badajoz, Spain, pp. 276–282.
  The paper contributions were described in Chapter 5.

- Montaño and Suárez (2019b), Herramienta para visualización gráfica de fuerzas de contacto y de movimientos de una mano robótica con sensores táctiles. In *XL Jornadas*

de Automática, Ferrol, Spain, pp. 749–755.
The paper contributions were described in Chapter 3.

- Montaño and Suárez (2019c), Manipulación Diestra de Objetos Desconocidos Usando Puntos de Contacto Virtuales. In *Jornadas Nacionales de Robótica (Spanish National Robotics Conference)(JNR19)*, Alicante, Spain, pp. 221–228.
  The paper contributions were described in Chapter 4.

Obtained Awards

ROBOTNIK 2013 Award for the best work in Robotics, awarded to: Andrés Montaño and Raúl Suárez for the work "Manipulación de objetos con dos dedos usando información táctil", presented in the *XXXIV Jornadas de Automática*, 2013, Terrassa, Spain.

Bibliography

Agriomallos, I., S. Doltsinis, I. Mitsioni, and Z. Doulgeri (2018). Slippage Detection Generalizing to Grasping of Unknown Objects using Machine Learning with Novel Features. *IEEE Robotics and Automation Letters 3*(2), 942–948.

Agulló, C. M. (2017). *Reconocimiento geométrico de objetos 3D y detección de deformaciones en manipulación robótica*. Ph. D. thesis, Universidad de Alicante.

Aldana, C. I., E. Cruz, E. Nuño, and L. Basañez (2018). Control in the operational space of bilateral teleoperators with time-delays and without velocity measurements. In *IFAC Proceedings Volumes*, pp. 204–209. Elsevier.

Andrychowicz, M., B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research 39*(1), 3–20.

Arimoto, S., R. Ozawa, and M. Yoshida (2005). Two-Dimensional Stable Blind Grasping under the Gravity Effect. In *IEEE International Conference on Robotics and Automation*, pp. 1196–1202.

Arimoto, S., M. Yoshida, and J.-H. Bae (2006). Stable "blind grasping" of a 3-D object under non-holonomic constraints. In *IEEE International Conference on Robotics and Automation*, pp. 2124–2130.

Aristotle and W. Ogle (1908). *De partibus animalium*. Oxford: Clarendon Press.

Bai, Y. and C. K. Liu (2014). Dexterous manipulation using both palm and fingers. In *IEEE International Conference on Robotics and Automation*, pp. 1560–1565.

Basañez, L. and R. Suárez (2009). Teleoperation. In *Springer Handbook of Automation*, pp. 449–468. Springer.

Bekey, G. A., R. Tomovic, and I. Zeljkovic (1990). *Control Architecture for the Belgrade/USC Hand*, pp. 136–149. Springer.

Bekiroglu, Y., R. Detry, and D. Kragic (2011). Learning tactile characterizations of object- and pose-specific grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1554–1560.

Bicchi, A. (1995). On the Closure Properties of Robotic Grasping. *The International Journal of Robotics Research 14*(4), 319–334.

Bicchi, A. and V. Kumar (2000). Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, pp. 348–353.

Bicchi, A., A. Marigo, and D. Prattichizzo (1999). Dexterity through rolling: manipulation of unknown objects. In *IEEE International Conference on Robotics and Automation*, pp. 1583–1588.

Bicchi, A. and R. Sorrentino (1995). Dexterous manipulation through rolling. In *IEEE International Conference on Robotics and Automation*, pp. 452–457.

Billard, A. and D. Kragic (2019). Trends and challenges in robot manipulation. *Science 364*(6446), eaat8414.

Bimbo, J., S. Luo, K. Althoefer, and H. Liu (2016). In-Hand Object Pose Estimation Using Covariance-Based Tactile To Geometry Matching. *IEEE Robotics and Automation Letters 1*(1), 570–577.

Birglen, L., T. Laliberté, and C. Gosselin (2008). *Underactuated Robotic Hands*. Springer.

Bonivento, C., E. Faldella, and G. Vassura (1991). The University of Bologna Robotic Hand Project: current state and future developments. In *IEEE International Conference on Advanced Robotics*, pp. 349–356.

Borst, C., M. Fischer, and G. Hirzinger (2004). Grasp planning: how to choose a suitable task wrench space. In *IEEE International Conference on Robotics and Automation*, pp. 319–325.

Boutselis, G. I., C. P. Bechlioulis, M. V. Liarokapis, and K. J. Kyriakopoulos (2014). An integrated approach towards robust grasping with tactile sensing. In *IEEE International Conference on Robotics and Automation*, pp. 3682–3687.

Bridgwater, L. B., C. A. Ihrke, M. A. Diftler, M. E. Abdallah, N. A. Radford, J. M. Rogers, S. Yayathi, R. S. Askew, and D. M. Linn (2012). The Robonaut 2 hand - designed to do work with tools. In *IEEE International Conference on Robotics and Automation*, pp. 3425–3430.

Brock, O., J. Trinkle, and F. Ramos (2009). Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In *Robotics: Science and Systems*, pp. 254–261. MIT Press.

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

Byoung-Ho, K., O. Sang-Rok, Y. Byung-Ju, and S. Il Hong (2001). Optimal grasping based on non-dimensionalized performance indices. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 949–956.

Cai, C. and B. Roth (1987). On the spatial motion of a rigid body with point contact. In *IEEE International Conference on Robotics and Automation*, pp. 686–695.

Catalano, M. G., G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi (2014). Adaptive synergies for the design and control of the Pisa/IIT SoftHand. *The International Journal of Robotics Research 33*(5), 768–782.

Chavan-Dafle, N., R. Holladay, and A. Rodriguez (2020). Planar in-hand manipulation via motion cones. *The International Journal of Robotics Research 39*(2-3), 163–182.

Chebotar, Y., O. Kroemer, and J. Peters (2014). Learning robot tactile sensing for object manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3368–3375.

Chen, W., H. Khamis, I. Birznieks, N. F. Lepora, and S. J. Redmond (2018). Tactile Sensors for Friction Estimation and Incipient Slip Detection—Toward Dexterous Robotic Manipulation: A Review. *IEEE Sensors Journal 18*(22), 9049–9064.

Cherif, M. and K. K. Gupta (1999). Planning quasi-static fingertip manipulations for reconfiguring objects. *IEEE Transactions on Robotics and Automation 15*(5), 837–848.

Chernov, N. (2010). *Circular and Linear Regression: Fitting Circles and Lines by Least Squares*. Monographs on Statistics and Applied Probability. Chapman & Hall CRC.

Chinellato, E., R. B. Fisher, A. Morales, and Á. P. del Pobil (2003). Ranking planar grasp configurations for a three-finger hand. In *IEEE International Conference on Robotics and Automation*, pp. 1133–1138.

Chinellato, E., A. Morales, R. B. Fisher, and Á. P. del Pobil (2005). Visual Quality Measures for Characterizing Planar Robot Grasps. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 35*(1), 30–41.

Ciobanu, V., N. Popescu, A. Petrescu, and M. Noeske (2013). Robot telemanipulation system. In *International Conference on System Theory, Control and Computing*, pp. 681–686.

Ciocarlie, M., A. T. Miller, and P. K. Allen (2005). Grasp analysis using deformable fingers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4122–4128.

Cole, A. A., J. E. Hauser, and S. Sastry (1989). Kinematics and Control of Multifingered Hands with Rolling Contact. *IEEE Transactions on Automatic Control 34*(4), 398–404.

Cole, A. A., P. Hsu, and S. Sastry (1992). Dynamic control of sliding by robot hands.pdf. *IEEE Transactions on Robotics and Automation 8*(1), 42–52.

Craig, J. J. (1986). *Introduction to robotics: mechanics & control*. Addison-Wesley Pub. Co.

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.

Dafle, N. C., A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. A. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge (2014). Extrinsic dexterity: In-hand manipulation with external forces. In *IEEE International Conference on Robotics and Automation*, pp. 1578–1585.

Dahiya, R. S. and M. Valle (2013). Touch Sensing—Why and Where? In *Robotic Tactile Sensing*, pp. 3–12. Springer.

Dang, H., J. Weisz, and P. K. Allen (2011). Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 5917–5922.

Delgado, A., C. A. Jara, and F. Torres (2017a). Adaptive tactile control for in-hand manipulation tasks of deformable objects. *International Journal of Advanced Manufacturing Technology 91*(9-12), 4127–4140.

Delgado, A., C. A. Jara, and F. Torres (2017b). In-hand recognition and manipulation of elastic objects using a servo-tactile control strategy. *Robotics and Computer-Integrated Manufacturing 48*, 102–112.

Doulgeri, Z. and L. Droukas (2013). On rolling contact motion by robotic fingers via prescribed performance control. In *IEEE International Conference on Robotics and Automation*, pp. 3976–3981.

Drimus, A., G. Kootstra, A. Bilberg, and D. Kragic (2011). Classification of rigid and deformable objects using a novel tactile sensor. In *International Conference on Advanced Robotics*, pp. 427–434.

Feix, T., J. Romero, H. B. Schmiedmayer, A. M. Dollar, and D. Kragic (2016). The GRASP Taxonomy of Human Grasp Types. *IEEE Transactions on Human-Machine Systems 46*(1), 66–77.

Felip, J., J. Bernabe, and A. Morales (2012). Contact-based blind grasping of unknown objects. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 396–401.

Ferrari, C. and J. Canny (1992). Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pp. 2290–2295.

Funabashi, S., S. Morikuni, A. Geier, A. Schmitz, S. Ogasa, T. P. Tomo, S. Somlor, and S. Sugano (2018). Object Recognition Through Active Sensing Using a Multi-Fingered Robot Hand with 3D Tactile Sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2589–2595.

Funabashi, S., A. Schmitz, T. Sato, S. Somlor, and S. Sugano (2015). Robust in-hand manipulation of variously sized and shaped objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 257–263.

Garate, V. R., M. Pozzi, D. Prattichizzo, N. Tsagarakis, and A. Ajoudani (2018). Grasp Stiffness Control in Robotic Hands Through Coordinated Optimization of Pose and Joint Stiffness. *IEEE Robotics and Automation Letters 3*(4), 3952–3959.

García, G. J., J.-A. Corrales, J. Pomares, and F. Torres (2009). Survey of visual and force/tactile control of robots for physical interaction in Spain. *Sensors 9*(12), 9689–9733.

Gorges, N., S. E. Navarro, and H. Worn (2011). Haptic object recognition using statistical point cloud features. In *International Conference on Advanced Robotics*, pp. 15–20.

Grosch, P. and R. Suárez (2004). Dexterous Robotic Hand MA-I, Sofware and Hardware Architecture. In *Intelligent Manipulation and Grasping International Conference*, pp. 91–96.

Han, L. and J. C. Trinkle (1998). Dextrous manipulation by rolling and finger gaiting. In *IEEE International Conference on Robotics and Automation*, pp. 730–735.

Haschke, R., J. J. Steil, I. Steuwer, and H. Ritter (2005). Task-Oriented Quality Measures for Dextrous Grasping. In *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 689–694.

Hertkorn, K., M. A. Roa, and C. Borst (2013). Planning in-hand object manipulation with multifingered hands considering task constraints. In *IEEE International Conference on Robotics and Automation*, pp. 617–624.

Ho, V. A., T. Nagatani, A. Noda, and S. Hirai (2012). What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation? In *IEEE International Conference on Automation Science and Engineering*, pp. 461–468.

Hokayem, P. F. and M. W. Spong (2006). Bilateral teleoperation: An historical survey. *Automatica 42*(12), 2035–2057.

Hong, J., G. Lafferriere, B. Mishra, and X. Tan (1990). Fine manipulation with multifinger hands. In *IEEE International Conference on Robotics and Automation*, pp. 1568–1573.

Howe, R. D. and M. R. Cutkosky (1996). Practical Force-Motion Models for Sliding Manipulation. *The International Journal of Robotics Research 15*(6), 557–572.

Howe, R. D., I. Kao, and M. R. Cutkosky (1988). The sliding of robot fingers under combined torsion and shear loading. In *IEEE International Conference on Robotics and Automation*, pp. 103–105.

Hsiao, K., S. Chitta, M. Ciocarlie, and E. G. Jones (2010). Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1228–1235.

Hvilshøj, M., S. Bøgh, O. S. Nielsen, O. Madsen, and O. Skov Nielsen (2012). Autonomous industrial mobile manipulation (AIMM): Past, present and future. *Industrial Robot 39*(2), 120–135.

Indri, M., L. Lachello, I. Lazzero, F. Sibona, and S. Trapani (2019). Smart Sensors Applications for a New Paradigm of a Production Line. *Sensors 19*(3), 650.

Ishihara, T., A. Namiki, M. Ishikawa, and M. Shimojo (2006). Dynamic Pen Spinning Using a High-speed Multifingered Hand with High-speed Tactile Sensor. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 258–263.

Jara, C. A., J. Pomares, F. Candelas, and F. Torres (2014). Control Framework for Dexterous Manipulation Using Dynamic Visual Servoing and Tactile Sensors' Feedback. *Sensors 14*(1), 1787–1804.

Jentoft, L. P., Q. Wan, and R. D. Howe (2014). Limits to compliance and the role of tactile sensing in grasping. In *IEEE International Conference on Robotics and Automation*, pp. 6394–6399.

Johansson, R. S. and G. Westling (1984). Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Experimental Brain Research 56*(3), 550–564.

Jongkind, W. (1993). Dextrous gripping in a hazardous environment guidelines, fault tolerance and control. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 509–514.

Kaboli, M., A. De La Rosa T, R. Walker, and G. Cheng (2015). In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 1155–1160.

Kaboli, M., K. Yao, and G. Cheng (2016). Tactile-based manipulation of deformable objects with dynamic center of mass. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 752–757.

Kalakrishnan, M., L. Righetti, P. Pastor, and S. Schaal (2011). Learning force control policies for compliant manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4639–4644.

Kao, I. and M. R. Cutkosky (1992). Quasistatic Manipulation with Compliance and Sliding. *The International Journal of Robotics Research 11*(1), 20–40.

Kappassov, Z., J.-A. Corrales, and V. Perdereau (2015). Tactile sensing in dexterous robot hands - Review. *Robotics and Autonomous Systems 74*, 195–220.

Kappassov, Z., J.-A. Corrales, and V. Perdereau (2020). Touch driven controller and tactile features for physical interactions. *Robotics and Autonomous Systems 123*, 103332.

Karakatsani, D. (2011). Object manipulation with two robotic fingers using tactile information. Master’s thesis, Universitat Politècnica de Catalunya.

Kerr, J. and B. Roth (1986). Analysis of Multifingered Hands. *The International Journal of Robotics Research 4*(4), 3–17.

Klein, C. A. and B. E. Blaho (1987). Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators. *The International Journal of Robotics Research 6*(2), 72–83.

Kroemer, O., C. Daniel, G. Neumann, H. van Hoof, and J. Peters (2015). Towards learning hierarchical skills for multi-phase manipulation tasks. In *IEEE International Conference on Robotics and Automation*, pp. 1503–1510.

Kukliński, K., K. Fischer, I. Marhenke, F. Kirstein, M. V. Aus Der Wieschen, D. Sølvason, N. Krüger, and T. R. Savarimuthu (2014). Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control. In *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 346–351.

Kutz, M. (2005). *Mechanical Engineers’ Handbook*. John Wiley & Sons, Inc.

Laaksonen, J., E. Nikandrova, and V. Kyrki (2012). Probabilistic sensor-based grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2019–2026.

Lampe, T. and M. Riedmiller (2013). Acquiring visual servoing reaching and grasping skills using neural reinforcement learning. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

León, B., A. Morales, and J. Sancho-Bru (2014). *From Robot to Human Grasping Simulation*. Cognitive Systems Monographs. Springer.

León, B., C. Rubert, J. Sancho-Bru, and A. Morales (2014). Characterization of grasp quality measures for evaluating robotic hands prehension. In *IEEE International Conference on Robotics and Automation*, pp. 3688–3693.

Li, M., Y. Bekiroglu, D. Kragic, and A. Billard (2014). Learning of grasp adaptation through experience and tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3339–3346.

Li, M., H. Yin, K. Tahara, and A. Billard (2014). Learning object-level impedance control for robust grasping and dexterous manipulation. In *IEEE International Conference on Robotics and Automation*, pp. 6784–6791.

Li, Q., R. Haschke, H. Ritter, and B. Bolder (2012). Towards unknown objects manipulation. In *IFAC Proceedings Volumes*, pp. 289–294. Elsevier.

Li, Q., O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter (2020). A Review of Tactile Information: Perception and Action Through Touch. *IEEE Transactions on Robotics 36*(6), 1–16.

Li, Q., M. Meier, R. Haschke, H. Ritter, and B. Bolder (2013). Rotary object dexterous manipulation in hand: a feedback-based method. *International Journal of Mechatronics and Automation 3*(1), 36.

Li, Z. and J. Canny (1990). Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation 6*(1), 62–72.

Li, Z., J. Canny, and S. Sastry (1989). On motion planning for dexterous manipulation. I. The problem formulation. In *IEEE International Conference on Robotics and Automation*, pp. 775–780.

Li, Z., P. Hsu, and S. Sastry (1989). Grasping and Coordinated Manipulation by a Multifingered Robot Hand. *The International Journal of Robotics Research 8*(4), 33–50.

Li, Z. and S. Sastry (1988). Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation 4*(1), 32–44.

Liarokapis, M. V. and A. M. Dollar (2017). Deriving dexterous, in-hand manipulation primitives for adaptive robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1951–1958.

Liégeois, A. (1977). Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics 7*(12), 868–871.

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.

Liu, H., K. C. Nguyen, V. Perdereau, J. Bimbo, J. Back, M. Godden, L. D. Seneviratne, and K. Althoefer (2015). Finger contact sensing and the application in dexterous hand manipulation. *Autonomous Robots 39*(1), 25–41.

Ma, R. R. and A. M. Dollar (2011). On dexterity and dexterous manipulation. In *IEEE International Conference on Advanced Robotics*, pp. 1–7.

MacKenzie, C. L. and T. Iberall (1994). The Grasping Hand. *Advances in Psychology 104*, 482.

Mason, M. T. (2001). *Mechanics of Robotic Manipulation*. The MIT Press.

Mason, M. T. and J. K. Salisbury (1985). *Robot Hands and the Mechanics of Manipulation*. The MIT Press.

Meattini, R., D. Chiaravalli, L. Biagiotti, G. Palli, and C. Melchiorri (2020). Combined Joint-Cartesian Mapping for Simultaneous Shape and Precision Teleoperation of Anthropomorphic Robotic Hands. In *IFAC Proceedings Volumes*, pp. 1–6. Elsevier.

Meier, M., M. Schopfer, R. Haschke, and H. Ritter (2011). A Probabilistic Approach to Tactile Shape Reconstruction. *IEEE Transactions on Robotics 27*(3), 630–635.

Menzel, R., K. Woelfl, and F. Pfeiffer (1994). Grasping with a dexterous robot hand. In *IFAC symposium Robot Control*, pp. 303–308. Elsevier.

Mindell, D. A. (1993). Telerobotics, automation, and human supervisory control [Review]. *IEEE Technology and Society Magazine 12*(3), 7.

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

Moll, M. and M. A. Erdmann (2001). Reconstructing shape from motion using tactile sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pp. 692–700.

Montana, D. J. (1988). The Kinematics of Contact and Grasp. *The International Journal of Robotics Research 7*(3), 17–32.

Montaño, A. and R. Suárez (2013a). Manipulación de objetos con dos dedos usando información táctil. In *XXXIV Jornadas de Automática*, Terrassa, Spain, pp. 618–625.

Montaño, A. and R. Suárez (2013b). Object shape reconstruction based on the object manipulation. In *International Conference on Advanced Robotics*, pp. 1–6.

Montaño, A. and R. Suárez (2014). Getting comfortable hand configurations while manipulating an object. In *IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1–8.

Montaño, A. and R. Suárez (2015). Unknown object manipulation based on tactile information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5642–5647.

Montaño, A. and R. Suárez (2016). Commanding the Object Orientation Using Dexterous Manipulation. In L. Reis, A. Moreira, P. Lima, L. Montano, and V. Muñoz-Martinez (Eds.), *Robot 2015: Second Iberian Robotics Conference. Advances in Intelligent Systems and Computing*, pp. 69–79. Springer.

Montaño, A. and R. Suárez (2017). Robust dexterous telemanipulation following object-orientation commands. *Industrial Robot 44*(5), 648–657.

Montaño, A. and R. Suárez (2018a). Improving Grasping Forces During the Manipulation of Unknown Objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3490–3495.

Montaño, A. and R. Suárez (2018b). Manipulación de Objetos Desconocidos Analizando Localmente su Forma para Optimizar las Fuerzas de Prensión. In *XXXIX Jornadas de Automática*, pp. 276–282.

Montaño, A. and R. Suárez (2018c). Manipulation of Unknown Objects to Improve the Grasp Quality Using Tactile Information. *Sensors 18*(5), 1412.

Montaño, A. and R. Suárez (2019a). Dexterous Manipulation of Unknown Objects Using Virtual Contact Points. *Robotics 8*(4), 86.

Montaño, A. and R. Suárez (2019b). Herramienta para visualización gráfica de fuerzas de contacto y de movimientos de una mano robótica con sensores táctiles. In *XL Jornadas de Automática*, pp. 749–755.

Montaño, A. and R. Suárez (2019c). Manipulación Diestra de Objetos Desconocidos Usando Puntos de Contacto Virtuales. In *Jornadas Nacionales de Robótica (Spanish National Robotics Conference)*, pp. 221–228.

Montaño, A. and R. Suárez (2019d). Model-free in-hand manipulation based on commanded virtual contact points. In *IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 586–592.

Montaño, A., R. Suárez, C. I. Aldana, and E. Nuño (2020). Bilateral telemanipulation of unknown objects using remote dexterous in-hand manipulation strategies. In *IFAC Proceedings Volumes*, pp. 1–8. Elsevier.

Morgan, A. S., K. Hang, W. G. Bircher, and A. M. Dollar (2019). A Data-Driven Framework for Learning Dexterous Manipulation of Unknown Objects. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 8273–8280.

Murray, R. M., Z. Li, and S. Sastry (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press LLC.

Nadon, F., A. Valencia, and P. Payeur (2018). Multi-Modal Sensing and Robotic Manipulation of Non-Rigid Objects: A Survey. *Robotics 7*(4), 74.

Natale, L. and G. Cannata (2017). Tactile Sensing. In *Humanoid Robotics: A Reference*, pp. 1–24. Springer.

Navarro, S. E., N. Gorges, H. Worn, J. Schill, T. Asfour, and R. Dillmann (2012). Haptic object recognition for multi-fingered robot hands. In *IEEE Haptics Symposium*, pp. 497–502.

Nuño, E., M. Arteaga-Pérez, and G. Espinosa-Pérez (2018). Control of bilateral teleoperators with time delays using only position measurements. *International Journal of Robust and Nonlinear Control 28*(3), 808–824.

Okada, T. (1979). Object-Handling System for Manual Industry. *IEEE Transactions on Systems, Man and Cybernetics 9*(2), 79–89.

Okamura, A. M., N. Smaby, and M. R. Cutkosky (2000). An overview of dexterous manipulation. In *IEEE International Conference on Robotics and Automation*, pp. 255–262.

Or, K., M. Tomura, A. Schmitz, S. Funabashi, and S. Sugano (2016a). Interpolation control posture design for in-hand manipulation. In *IEEE/SICE International Symposium on System Integration*, pp. 187–192.

Or, K., M. Tomura, A. Schmitz, S. Funabashi, and S. Sugano (2016b). Position-force combination control with passive flexibility for versatile in-hand manipulation based on posture interpolation. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 2542–2547.

Ozawa, R., S. Arimoto, S. Nakamura, and Ji-Hun Bae (2005). Control of an object with parallel surfaces by a pair of finger robots without object sensing. *IEEE Transactions on Robotics 21*(5), 965–976.

Ozawa, R., S. Arimoto, M. Yoshida, and S. Nakamura (2004). Stable grasping and relative angle control of an object by dual finger robots without object sensing. In *IEEE International Conference on Robotics and Automation*, pp. 1694–1699.

Palli, G. and S. Pirozzi (2019). A Tactile-Based Wire Manipulation System for Manufacturing Applications. *Robotics 8*(2), 46.

Park, H. and D. Kim (2020). An open-source anthropomorphic robot hand system: HRI hand. *HardwareX 7*, e00100.

Park, Y. C. and G. P. Starr (1992). Grasp Synthesis of Polygonal Objects Using a Three-Fingered Robot Hand. *The International Journal of Robotics Research 11*(3), 163–184.

Pastor, P., M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal (2011). Skill learning and task outcome prediction for manipulation. In *IEEE International Conference on Robotics and Automation*, pp. 3828–3834.

Pellerin, C. (1991). Salisbury hand. *Industrial Robot 18*(4), 25–26.

Perkins, W. A. (1978). A Model-Based Vision System for Industrial Parts. *IEEE Transactions on Computers c-27*(2), 126–143.

Pozo-Espín, D. (2012). Identificación y posicionamiento de objetos mediante manipulación con una mano robótica sensorizada. Master's thesis, Universitat Politècnica de Catalunya.

Prats, M., P. J. Sanz, and Á. P. del Pobil (2010). Reliable non-prehensile door opening through the combination of vision, tactile and force feedback. *Autonomous Robots 29*(2), 201–218.

Prattichizzo, D., M. Pozzi, and M. Malvezzi (2020). Dexterous Manipulation. In M. H. Ang, O. Khatib, and B. Siciliano (Eds.), *Encyclopedia of Robotics*, pp. 1–8. Springer.

Prattichizzo, D. and J. C. Trinkle (2008). Grasping. In *Springer Handbook of Robotics*, pp. 671–700. Springer.

Quigley, M., B. Gerkey, and W. Smart (2015). *Programming robots with ROS*. O'Reilly Media, Inc.

Righetti, L., M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal (2014). An autonomous manipulation system based on force control and optimization. *Autonomous Robots 36*(1-2), 11–30.

Roa, M. A. and R. Suárez (2009). Regrasp planning in the grasp space using independent regions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1823–1829.

Roa, M. A. and R. Suárez (2014). Grasp quality measures: review and performance. *Autonomous Robots 38*(1), 65–88.

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.

Rosell, J., R. Suárez, and A. Pérez (2014). Safe teleoperation of a dual hand-arm robotic system. In M. Armada, A. Sanfeliu, and M. Ferre (Eds.), *Robot 2013: First Iberian Robotics Conference. Advances in Intelligent Systems and Computing*, pp. 615–630. Springer.

Rubert, C., B. León, and A. Morales (2014). Grasp quality metrics for robot hands benchmarking. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 761–766.

Sadigh, M. J. and H. Ahmadi (2009). Safe grasping with multi-link fingers based on force sensing. In *IEEE International Conference on Robotics and Biomimetics*, pp. 1796–1802.

Salisbury, J. K. and J. J. Craig (1982). Articulated Hands. *The International Journal of Robotics Research 1*(1), 4–17.

Salisbury, J. K. and B. Roth (1983). Kinematic and Force Analysis of Articulated Mechanical Hands. *Journal of Mechanisms, Transmissions, and Automation in Design 105*, 35–41.

Saut, J.-P., A. Sahbani, S. El-Khoury, and V. Perdereau (2007). Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2907–2912.

Schneider, A., J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard (2009). Object identification with tactile sensors using bag-of-features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 243–248.

Shadow Robot Company (2015). Shadow Dexterous Hand. https://www.shadowrobot.com/products/dexterous-hand/.

Shaw-Cortez, W., D. Oetomo, C. Manzie, and P. Choong (2018). Tactile-Based Blind Grasping: A Discrete-Time Object Manipulation Controller for Robotic Hands. *IEEE Robotics and Automation Letters 3*(2), 1064–1071.

Sherwood, C. C., F. Subiaul, and T. W. Zawidzki (2008). A natural history of the human mind: Tracing evolutionary changes in brain and cognition. *Journal of Anatomy 212*(4), 426–454.

Shi, J., J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch (2017). Dynamic In-Hand Sliding Manipulation. *IEEE Transactions on Robotics 33*(4), 778–795.

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

Shirafuji, S. and K. Hosoda (2014). Detection and prevention of slip using sensors with different properties embedded in elastic artificial skin on the basis of previous experience. *Robotics and Autonomous Systems 62*(1), 46–52.

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.

Su, Z., K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal (2015). Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 297–303.

Suárez, R., L. Basañez, and J. Rosell (1995). Using configuration and force sensing in assembly task planning and execution. In *IEEE International Symposium on Assembly and Task Planning*, pp. 273–279.

Tahara, K., S. Arimoto, and M. Yoshida (2010). Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In *IEEE International Conference on Robotics and Automation*, pp. 4322–4327.

Takahashi, T., T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, Masatsugu Iribe, Tetsuharu Fukushima, and Masahiro Fujita (2008). Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control. In *IEEE International Conference on Robotics and Automation*, pp. 264–271.

Tischler, C. R., A. E. Samuel, and K. H. Hunt (1998). Dextrous robot fingers with desirable kinematic forms. *The International Journal of Robotics Research 17*(9), 996–1012.

Toh, Y. P., S. Huang, J. Lin, M. Bajzek, G. Zeglin, and N. S. Pollard (2012). Dexterous telemanipulation with a multi-touch interface. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 270–277.

Velasco, E., B. S. Zapata-Impata, P. Gil, and F. Torres (2020). Clasificación de objetos usando percepción bimodal de palpación única en acciones de agarre robótico. *Revista Iberoamericana de Automática e Informática industrial 17*(1), 44–55.

Ward-Cherrier, B., N. Rojas, and N. F. Lepora (2017). Model-Free Precise in-Hand Manipulation with a 3D-Printed Tactile Gripper. *IEEE Robotics and Automation Letters 2*(4), 2056–2063.

Weigl, A. and M. Seitz (1994). Vision assisted disassembly using a dexterous hand-arm-system: An example and experimental results. In *IFAC Proceedings Volumes*, pp. 315–320. Elsevier.

Wöhlke, G. (1990). A programming and simulation environment for the karlsruhe dextrous hand. *Robotics and Autonomous Systems 6*(3), 243–263.

Wonik Robotics (2018). Allegro Robotic Hand. http://www.simlab.co.kr/Allegro-Hand.htm.

Xu, J., T. J. Koo, and Z. Li (2007). Finger gaits planning for multifingered manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2932–2937.

Xue, Z., J. M. Zollner, and R. Dillmann (2008). Dexterous manipulation planning of objects with surface of revolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2703–2708.

Yashima, M. (2004). Manipulation Planning for Object Re-orientation based on Randomized Techniques. In *IEEE International Conference on Robotics and Automation*, pp. 1245–1251.

Yoshikawa, T. (1985). Manipulability and redundancy control of robotic mechanisms. In *IEEE International Conference on Robotics and Automation*, pp. 1004–1009.

Zhang, H. and N. N. Chen (2000). Control of contact via tactile sensing. *IEEE Transactions on Robotics and Automation 16*(5), 482–495.

Zou, L., C. Ge, Z. Wang, E. Cretu, and X. Li (2017). Novel Tactile Sensor Technology and Smart Tactile Sensing Systems: A Review. *Sensors 17*(11), 2653.