Politecnico di Torino

Master's Degree in Mechanical Engineering
A.a. 2021/2022
Graduation Session April 2022

In-Hand Robotic Manipulation

Supervisors:
Prof. Massimo Sorli
Prof. Raúl Suárez Feijóo

Candidate:
Antonio Finelli

Ai miei genitori
e ai miei fratelli

Abstract

In-Hand dexterity is the ability to make coordinated movements of the fingers to grasp and manipulate objects. For humans, handling an object is an innate ability and a straightforward task, but the situation is completely reversed if a robot is performing the manipulation. Traditional industrial robot manipulators are typically designed as the combination of an arm and a simple gripper as an end-effector. Undoubtedly, this configuration is highly efficient when a high degree of on-site manipulation by the end-effector is not required. The challenge that fascinates researchers and robotics lovers is to create end-effectors that could provide greater possibilities for manipulation. To achieve this purpose, multi-fingered robotic hands are usually employed. Despite robotics making giant strides every day, dexterous manipulation can be considered one of the most uncertain topics of robotics since it requires precision and accuracy in controlling the variables involved that verge on perfection.

The main objective of the thesis is to contribute to the research and development of dexterous robotic manipulation. In this regard, the question behind the project is the following: is it possible to perform manipulation by executing rotation of the grasped object around any axis in space? Starting from this request, a manipulation strategy is implemented, based on the idea that the kinematic of the fingers is used to move the object from an initial to a final configuration while maintaining fingertip contacts.

The robotic device employed is the Allegro Hand, an anthropomorphic robotic hand made up of four fingers with sixteen degrees of freedom. The manipulated objects are rigid bodies, so it is assumed that they do not experience any deformation regardless of the forces applied. Moreover, the shapes of the objects are unknown. In fact, the procedure is based only on the tactile information obtained by tactile sensors located at the fingertips and on the kinematic structure of the hand. No other external feedback sources are considered.

In the first phase, the resulting movements from the strategy that the Allegro's fingers must execute are tested in a simulation environment without considering the grasped object. Then, the strategy is experimented in a real environment with the aid of the simulation environment to verify the forces involved in the manipulation and the points of contact between fingertips and the grasped object. The results indicate that the algorithm is accurate in specific axes of rotation.

Against this background, future research may be aimed at implementing a hybrid controller so that the optimal combination of force and position could be established. Additionally, more detailed research on the frictional forces involved in manipulation might lead to satisfactory results.

Acknowledgements

Desidero prima di tutto rivolgere un ringraziamento al Prof. Sorli per la gentilezza, la disponibilità e la professionalità dimostratami in questi mesi di lavoro non facili.

Gracias al profesor Suárez que me ha permitido realizar este proyecto de tesis de tanto interés.

Gracias a Leo por toda la ayuda que me ha prestado y todas las nociones que me ha enseñado, tanto en el aspecto académico, como en el personal.

Premetto che alla laurea triennale mi ero dilungato dedicando un pensiero a molti di voi. Questa volta sarò più coinciso e i ringraziamenti sono da considerarsi un'aggiunta ai precedenti.

La tesi è dedicata alla mia ragione di vita: i miei genitori e i miei fratelli. Il nostro rapporto non aveva bisogno di conferme, ciò nonostante è cresciuto a dismisura durante il periodo pandemico. Grazie perché senza di voi non avrei mai raggiunto questo obiettivo e non sarei mai diventato ciò che sono. Un grazie speciale va a te, mamma, che anche questa volta hai dovuto spendere giornate intere per darmi la forza per affrontare le difficoltà. Non hai bisogno che ti dica cosa provo per te, tu sai.

Grazie a tutti gli amici per essere rimasti accanto a me, ognuno a suo modo, in questo percorso intenso ed entusiasmante. Grazie a tutti i ragazzi della valle, ai ragazzi di Torino e ai ragazzi di Barcellona. Con ognuno condivido dei ricordi indimenticabili che hanno segnato la mia personalità. A parole non riesco ad esprimere il bene che vi voglio e quanto vi sono grato. Un grazie particolare ad Alessio, venuto in soccorso, ancora una volta, in un momento difficile.

Grazie a Catello, una persona unica e speciale. A Barcellona ci hanno preso in giro dicendo che siamo fidanzati e che non esiste Antonio senza Catello: io, un futuro senza di te, non oso immaginarlo. Grazie per essermi sempre vicino, soprattutto nei momenti di difficoltà. Te lo ripeto, senza il tuo aiuto non avrei mai continuato questo progetto di tesi.

Grazie a Marco Mannella e a Marco Maruffi per essere stati i colleghi perfetti e per lo stupendo rapporto di amicizia e stima che ne è conseguito.

Grazie a Francesco e Martina, per essere sempre presenti e per avermi accolto nuovamente a Rotondi come se non fossi mai andato via.

Grazie alla mia famiglia, nonna Lucia, zii e cugini per tutti gli insegnamenti e tutte i momenti condivisi.

Infine, grazie alle persone che non ci sono più, nonno Antonio, nonna Menechella e nonno Peppe. A te nonno Peppe, che sei andato via poco fa, un grazie speciale per ricordarmi che al mondo esistono persone buone ed oneste, per le quali vale la pena lottare. Mancate da morire, porterò sempre nel cuore il vostro ricordo.

Table of Contents

List of Figures

List of Tables

Acronyms

API Application Programming Interface

BFRF Base Finger Reference Frame

CAD Computer-Aided Design

CAN Controller Area Network

CPU Central Processing Unit

DOF Degrees of Freedom

FK Forward Kinematics

GUI Graphical User Interface

IOC Institute of Industrial and Control Engineering

IK Inverse Kinematics

KDL Kinematics and Dynamics Library

PID Proportional Integrative Derivative

PRF Planar Reference Frame

ROS Robot Operating System

RF Reference Frame

URDF Unified Robotic Description Format

VCP Virtual Contact Points

WTS Weiss Tactile Sensors

XLM eXtensible Markup Language

Chapter 1

Introduction

The desire of human beings to create artificial devices that can replace or help them in dealing with specific tasks dates back to ancient times. As Gasparetto and Scalera explains in the article "A Brief History of Industrial Robotics in the 20th Century" [7], «since the Greek-Hellenistic age, some devices, which were named *automata*, have been designed and created by ingenious inventors in order to amplify the process of civilization». And continuing, «the term *automata* mainly refers to human-like devices, while the term *robot* has a more general meaning. The origin of the term *robot* comes from the Czech word *robota* that means "hard work". The word appeared the first time in 1920, when the writer K. Čapek used it in his sci-fi drama R.U.R to designate the machines that work instead of workers».

Nowadays robots act and interact with humans in every field as if it were an ordinary occurrence. When the context of work is the execution of productive activities, it is possible to talk about industrial robots. Technological progress enabled the development of devices capable of carrying out almost any type of activity. In most cases, human intervention is unnecessary or merely reduced to programming and control tasks. But what lies behind the operation of a robot? An industrial robot is the functional union of four main systems (Figure 1.1):

- the *mechanical system* is merely the final device, the one that is commonly referred to as a robot;
- the *sensory system* acquires and interprets data (position, speed, force, etc.) from the mechanical system. The data could be detected by contact sensors, proximity sensors, tactile sensors, sensors for measuring distances, sensors for voice signals, etc.;
- the *control system* supervises the correct operation of the previous systems and, by means of algorithms, plans intermediate objectives and movements necessary for the task assigned to the robot;

- the *actuation system*, made of pneumatic, hydraulic, or electric drives, allow movement of the mechanical system monitored by the control system.

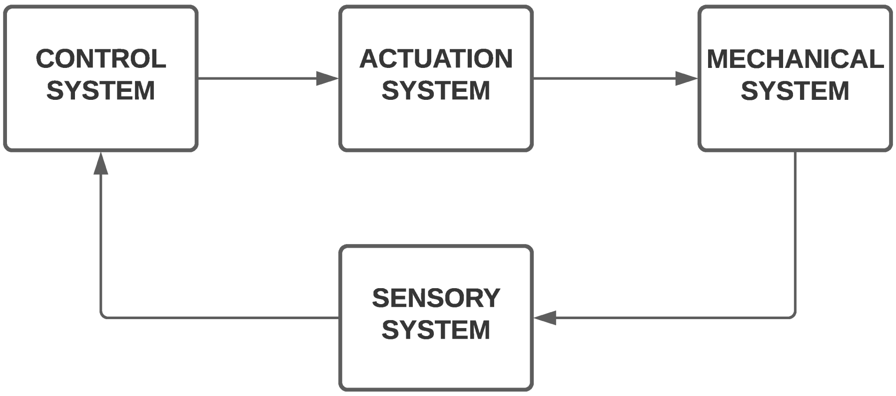


Figure 1.1: Industrial robots main systems.

Therefore, «it can be recognized that robotics is an interdisciplinary subject concerning the cultural areas of mechanics, control, computers, and electronics» [19].

Industrial manipulator robots are the backbone of the manufacturing sector. Usually, they consist of an open kinematic chain composed of a sequence of rigid bodies (links) interconnected through articulations (joints). The term end-effector is used to describe the last link that is interacting with the workpieces. Typically, end-effectors are simple grippers designed to execute specific tasks, such as pick and place operations. According to Darwin, without using the hands, the human being would never have achieved his predominant place in the world. Therefore, the only way to universalize the scope of the end-effector is to modify and shape them in such a way to resemble the human hand as much as possible. From this arise the creation of anthropomorphic multi-fingered robotic hands. Dexterous manipulation is the area of robotics that deals with research and development concerning multi-fingered robotic hands, so that fingers cooperate to grasp and manipulate objects. Dexterous manipulation is one of the most intriguing and undoubtedly one of the most challenging fields of robotic research because it requires precise and accurate control of forces and motions.

The current thesis project arises from a challenge proposed in “Object Manipulation Based on Tactile Information”, a Ph.D. Thesis drafted by Andrés Felipe Montaño Sarria and supervised by Raúl Suárez Feijóo [11]. Allegro Hand is one of the robotic devices employed in the above-mentioned project, while it is the only device used in the current thesis project. The aim is to implement in-hand manipulation strategies. In-hand manipulation is the capability to manipulate an object in the hand, namely moving the object from an initial position to a final position using only the fingers, without resorting to the support of the wrist. The majority in-hand of manipulation strategies rely on a priori knowledge of the object, which facilitates the possibility of predicting the final position to be reached by the object and the corresponding finger movement. Instead, Montaño's project starts from the assumption that the object to be manipulated is unknown. His approach to manipulation is quite interesting because it is based on the observation of the typical movements that a human being does to rotate an object. Manipulation is achieved by processing data from tactile sensors. «Tactile information is processed according to two different aims: object identification and manipulation control. On the one hand, the properties of the objects extracted from the robot's tactile sensors can be used to categorize the objects into different classes. On the other hand, the measurements obtained from the tactile sensors can also be applied to control the interaction force» [20]. Montaño in his Ph.D.thesis, regarding the section of the Allegro Hand, adopts the second approach. The limitation that he encounters is to obtain optimal experimental results only for rotations around a well-defined axis in space. The final goal of the current thesis is to generalize rotations around any axis.

The structure of the thesis with a brief summary of the chapters is presented below:

- *Chapter 2* is a detailed description of the tools employed for the experimental validation, both regarding the hardware part and the software part.
- *Chapter 3* outlines and comments on Montaño's manipulation strategy and then focuses on what has been developed anew. In addition, the fundamental aspects in common with the two approaches are highlighted in the initial section.
- *Chapter 4* presents the experimental results obtained.
- *Chapter 5* summarizes the content of the thesis and lists possible ways for future works.

Chapter 2

Equipment and Robotic System

2.1 Allegro Hand

Allegro Hand is a low-cost and highly adaptive robotic hand produced by Wonik Robotics. It is a perfect platform for grasp and manipulation research. The device is a left hand composed of four fingers (Figure 2.1).

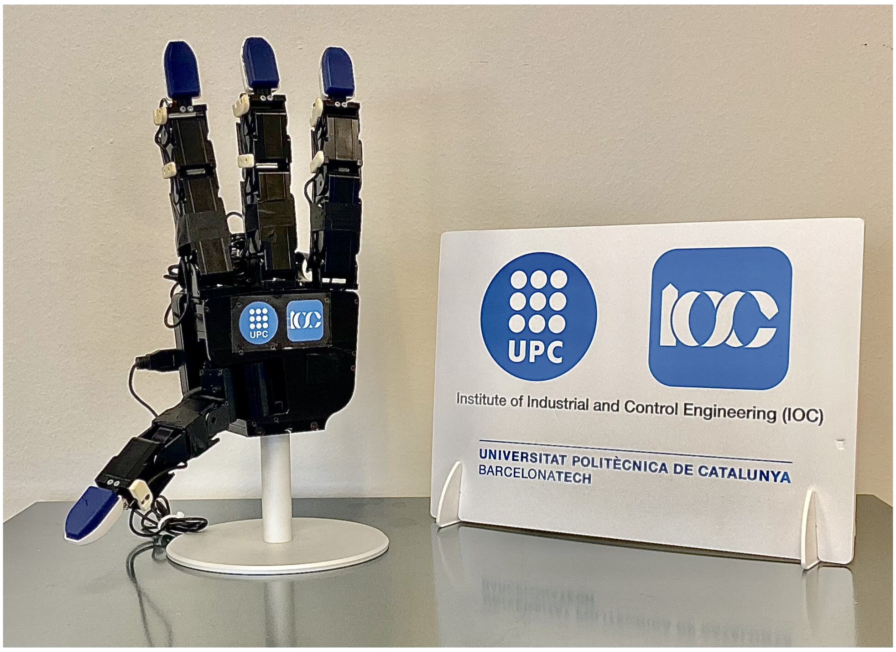


Figure 2.1: Allegro Hand.

Its main features are:

- lightweight and portable anthropomorphic design;
- low-cost dexterous manipulation with applications in research and industry;
- multiple ready-to-use sensorless grasping algorithms capable of handling a variety of object geometries;
- capable of holding up to 1.5 kg;
- 16 independent current-controlled joints (4 fingers x 4 DOF ea.).
- Allegro Hand Application Studio integration allows for simulation-based algorithm prototyping without ever changing your code;
- support for real-time control and online simulation.

More detailed technical specifications of the hand are shown in Table 2.1.

Number of Fingers	Four (4) fingers, including thumb
Degrees of Freedom	4 fingers x 4 = 16 (Active)
Actuation	Type: DC Motor Gear Ratio: 1:369 Maximum torque: 0.70 (Nm) Maximum joint speed: 0.11 (s/degree)
Weight	Finger: 0.17 (kg) Thumb: 0.19 (kg) Total: 1.08 (kg)
Joint Resolution	Measurement: Potentiometer Resolution (nominal): 0.002 (deg)
Communication	CAN (333 Hz)
Payload	5 (kg)
Power Requirement	7.4 VDC (7.0–8.1 V), 5 A minimum

Table 2.1: Allegro Hand technical specifications.

Figure 2.2 represents a technical drawing with the geometric dimensions of the Allegro Hand. The figure is extrapolated from the manual provided by the manufacturer [1]: for this reason, a right hand is depicted. It is worth underlining that the dimensions are the same for a left hand. All dimensions are displayed in millimeters (mm) and degrees.

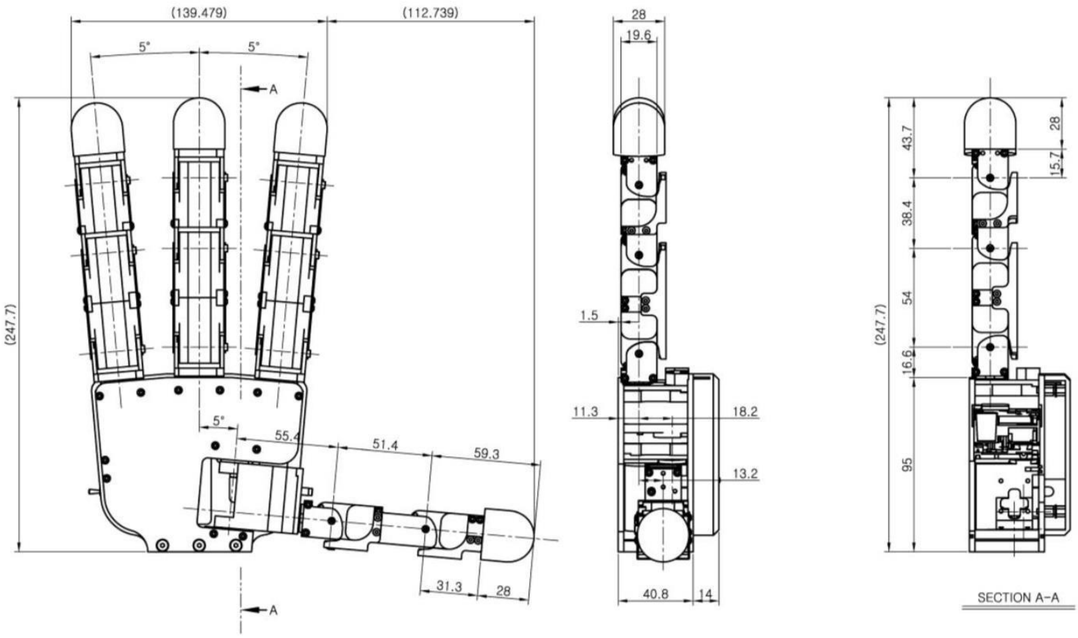


Figure 2.2: Allegro Hand dimensions.

The numeration of the joints is shown in Figure 2.3. The Index (I), Middle (M), and Ring (R) fingers have the same kinematic structure. To understand, focus the attention on Figure 2.4 and consider the Index: the joint 1 (located at the base of the finger) fixes the orientation of the working plane Π_I, while the other three DOF (the joints 2,3,4) are used to make the fingertip reach a point and an orientation in the working plane Π_I. A demonstration video is attached to the following link:

`https://youtu.be/fLJKyfJ8isc`.

In the case of the Thumb (T) (Figure 2.5), the first DOF (joint 13) produces the abduction movement, while the second DOF (joint 14) fixes the orientation of the working plane Π_T of the Thumb, leaving only two DOF (joint 15 and joint 16) to move the fingertip in the working plane Π_T. The related demonstration video is here posted:

`https://youtu.be/_Qxv-4oQX5A`.

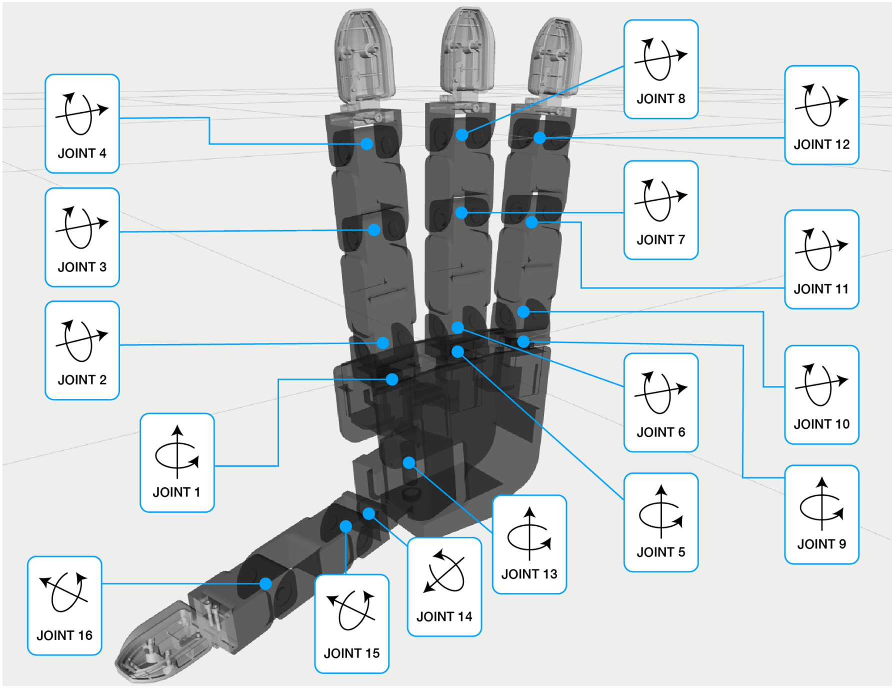


Figure 2.3: Allegro Hand joints.

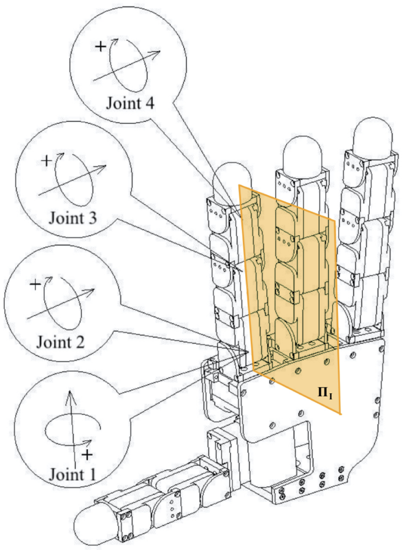


Figure 2.4: Kinematic structure of the Index with the corresponding working plane Π_I.

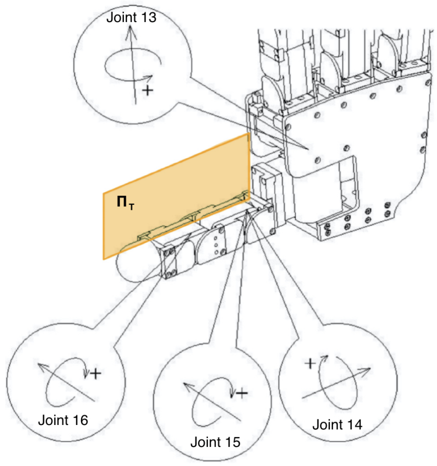


Figure 2.5: Kinematic structure of the Thumb with the corresponding working plane Π_T.

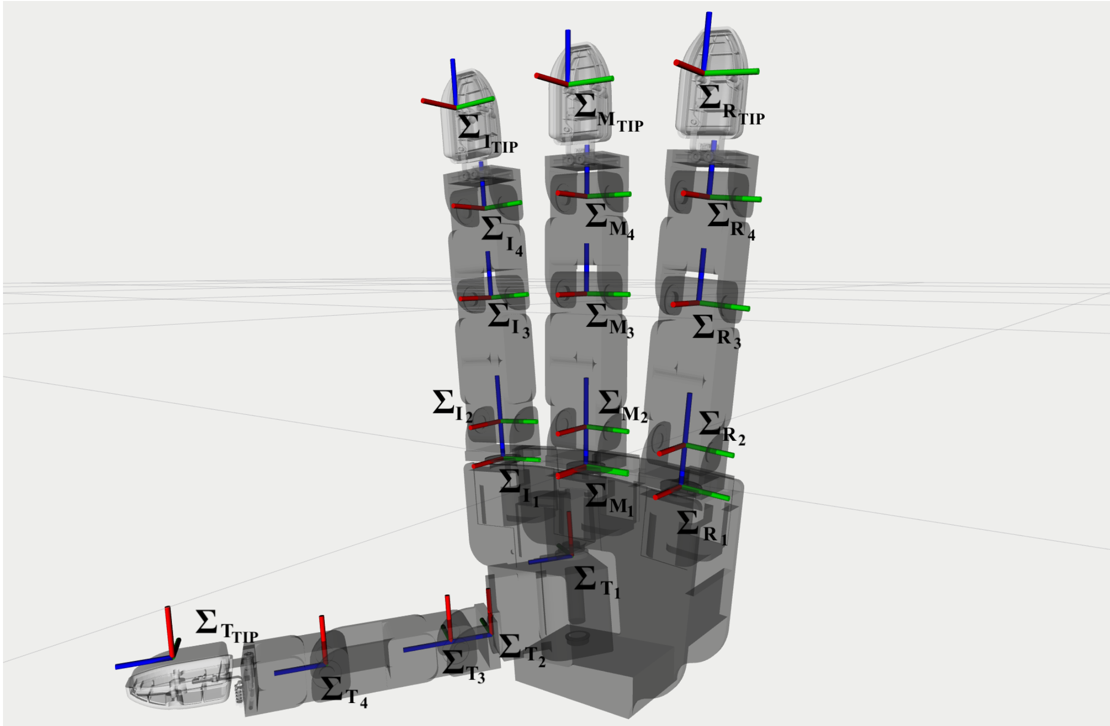


Figure 2.6: Allegro Hand in RVIZ with the systems reference frame.

As shown in Figure 2.6, each finger f_i, with $i \in \{I, M, R, T\}$, has a reference frame Σ_{i_j} located in the joints, where j indicates the number of joint under observation, $j \in \{1,2,3,4\}$; when a joint rotates, the corresponding reference frame rotates as well. The absolute reference frame $\mathcal{W}$ is located at the base of the middle finger f_M, in the intersection point with the palm. In Table 2.2 are set out the positions of the origins of the reference systems of the joints of the hand. It is worth underlining that the absolute reference frame $\mathcal{W}$ is always fixed and it is not coincident with Σ_{I_1}.

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

(a) Index.

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

(b) Middle.

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

(c) Ring.

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

(d) Thumb.

Table 2.2: Origin of the relative reference frames of the fingers Σ_{i_j} with respect to the world reference frame $\mathcal{W}$.

From a mechanical viewpoint, a finger f_i can be schematically modeled as a kinematic serial chain of rigid bodies (links) connected through joints. Each finger has n_j DOF. One end of the chain is constrained to a base, while an end-effector (the fingertip) is mounted to the other end. The number of links of a single finger is equal to four. Each finger i has j links ($j = \{1, ..., 4\}$) with length l_{ij}. The joint angle q_{i_j} is the angle between two consecutive links. As it can be noticed in Figure 2.6, for each finger, there is one more system reference frame (indicated as $\Sigma_{i_{TIP}}$) than those expected. To locate the position of the end-effectors (in our case the fingertips), it is defined a fixed joint located in the center of the tactile sensor. The relative position between $\Sigma_{i_{TIP}}$ and Σ_{i_4} (namely link l_{i4}) is indicated in Table 2.3.

	x [mm]	y [mm]	z [mm]	roll [°]	pitch [°]	yaw [°]
Index, Middle and Ring	0.0135	0	0.0408	0	0	0
Thumb	0.0135	0	0.0564	0	0	0

Table 2.3: Relative position between $\Sigma_{i_{TIP}}$ and Σ_{i_4}.

The configuration of the whole finger, or better, the resulting motion of the structure, is obtained by the composition of the elementary motions of each link with respect to the previous one. So, the finger configuration f_i is given by the composition of its joints angles as $\mathbf{q}_i = \{q_{i_1}, ..., q_{i_{n_j}}\}$.

The whole hand configuration is given by the concatenation of the configurations of the fingers as $\mathbf{Q} = \{\mathbf{q}_I, \mathbf{q}_M, \mathbf{q}_R, \mathbf{q}_T\}$. Figure 2.7 depicts the mechanical scheme of one of a finger that could be Index, Middle or Ring, while Figure 2.8 represents the mechanical scheme for the Thumb.

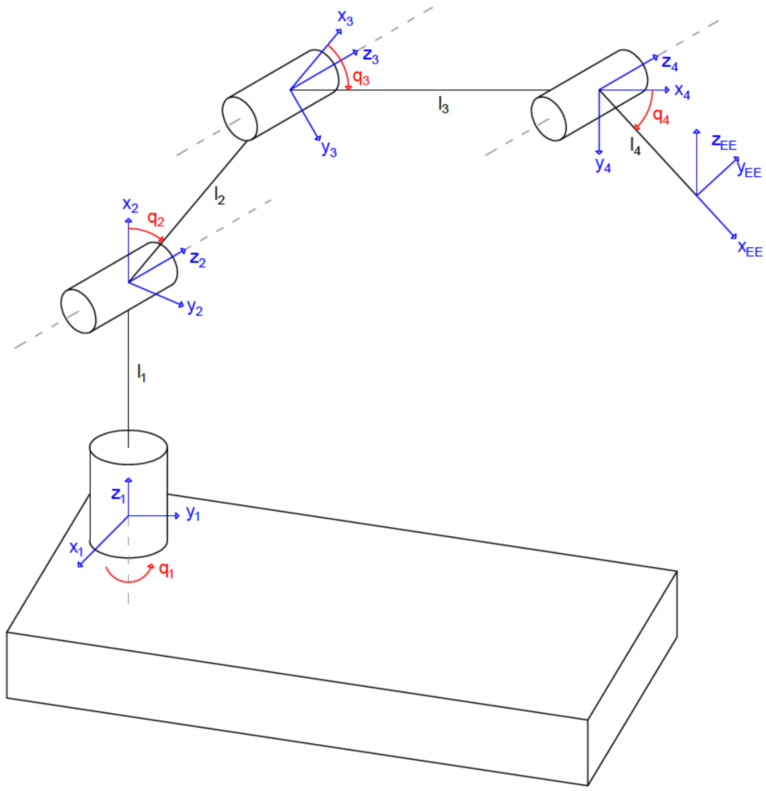


Figure 2.7: Mechanical scheme of Index, Middle and Ring.

Reporting what is shown in Table 2.1, the joints of the Allegro are actuated by DC motors and the joint resolution value is 0.002°, measured with a potentiometer. The hand is connected to a computer by a controller area network (CAN) bus.

The position of the fingers of the Allegro Hand is monitored by a position control system. The latter is based on the classical PID (proportional, integrative, derivative) controller:

$$\mathbf{u}(t) = K_P\mathbf{e}(t) + K_D\frac{d\mathbf{e}(t)}{dt} + K_I\int_0^t \mathbf{e}(t)dt \tag{2.1}$$

- $\mathbf{u}(t)$: necessary torque to reduce the position error;
- $\mathbf{e}(t)$: position error defined as the difference between the desired hand configuration $\mathbf{Q}_d$ and the current hand configuration $\mathbf{Q}_k$:

$$\mathbf{e}(t) = \mathbf{Q}_d(t) - \mathbf{Q}_k(t) \tag{2.2}$$

- K_P: proportional gain;

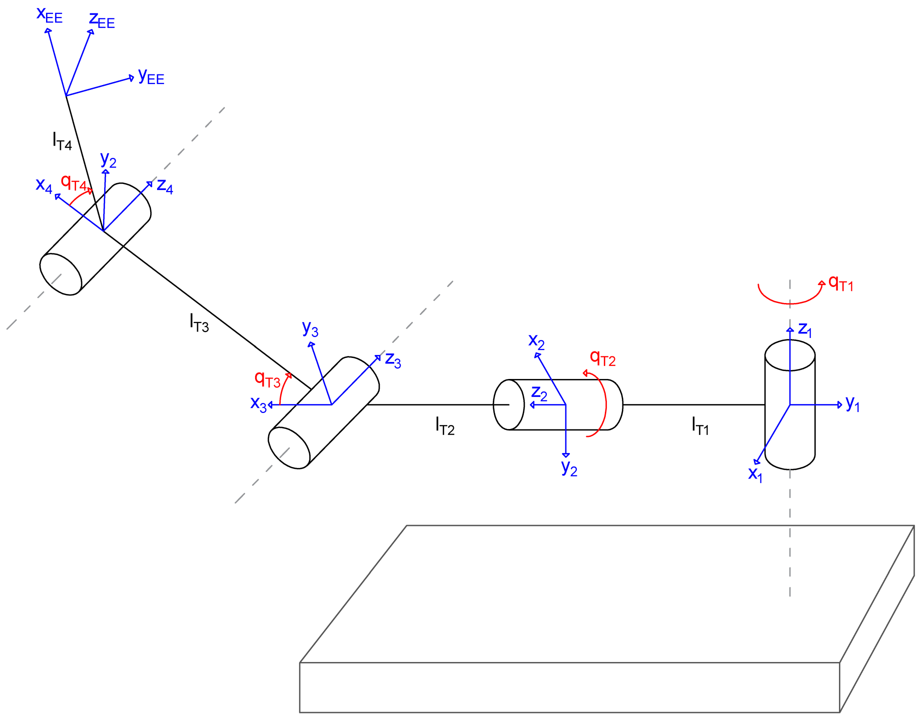


Figure 2.8: Mechanical scheme of Thumb.

- K_D: derivative gain;
- K_I: integral gain.

Considering Δt the sampling time, the integral term is approximated as:

$$\int_0^t \mathbf{e}(t)dt \cong \sum_{k=1}^{T} \mathbf{e}(t_k)\Delta t \tag{2.3}$$

while the derivative term is approximated as:

$$\frac{d\mathbf{e}(t)}{dt} \cong \frac{\mathbf{e}(t_k) - \mathbf{e}(t_{k-1})}{\Delta t} \tag{2.4}$$

In order to take into consideration the weight of the finger, the value of the torque $\mathbf{u}(t)$ derived from the equation (2.1) is modified according to:

$$\boldsymbol{\tau}(t) = \mathbf{u}(t) + \boldsymbol{\tau}_{gc} \tag{2.5}$$

where $\boldsymbol{\tau}_{gc}$ is the gravity compensation torque.

Algorithm 1, extrapolated from "Object manipulation based on tactile information"[11], shows the pseudo-code of the function *updateController* which computes the torque applied to the joints in each control cycle.

Algorithm 1 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 2.2
    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

2.2 Tactile Sensors

A tactile sensor is a device that measures information arising from physical interaction with its environment; it provides robots with information about physical contact, whereby autonomous robot hands can operate in unstructured environments and manipulate unknown objects.

The fingertips of the commercial version of the Allegro Hand do not have tactile sensors, thus, they are removed and substituted by Weiss Tactile Sensors WTS-FT 0408[1]. The WTS are sensorized fingertips specifically designed for handling technology: they increase the capabilities of the hand for dexterous manipulation. Despite the fully encapsulated design and its rugged construction, the WTS sensor modules feature a highly sensitive sensor matrix that allows a roughly precise detection of very light contact forces. The matrix is composed by arrays of tactile sensor cells (texels), with an associated reference frame Σ_{i_s} located in the left-upper corner of the pad. In the case considered, the sensing matrix has 4 by 8 texels of 3.8 mm by 3.8 mm, as shown in Figure 2.10a.

In this work, the contact between the fingertips and the manipulated object is modeled using the frictionless point-contact model (Appendix A). In the real experimentation, in general, the contact between the object and the tactile sensor takes place on a contact region including several texels. In order to respect the assumption of the point-contact model, the barycenter of the contact region ($\mathbf{b} \in \mathbb{R}^2$) is considered as the current effective contact point, and the current contact force F

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

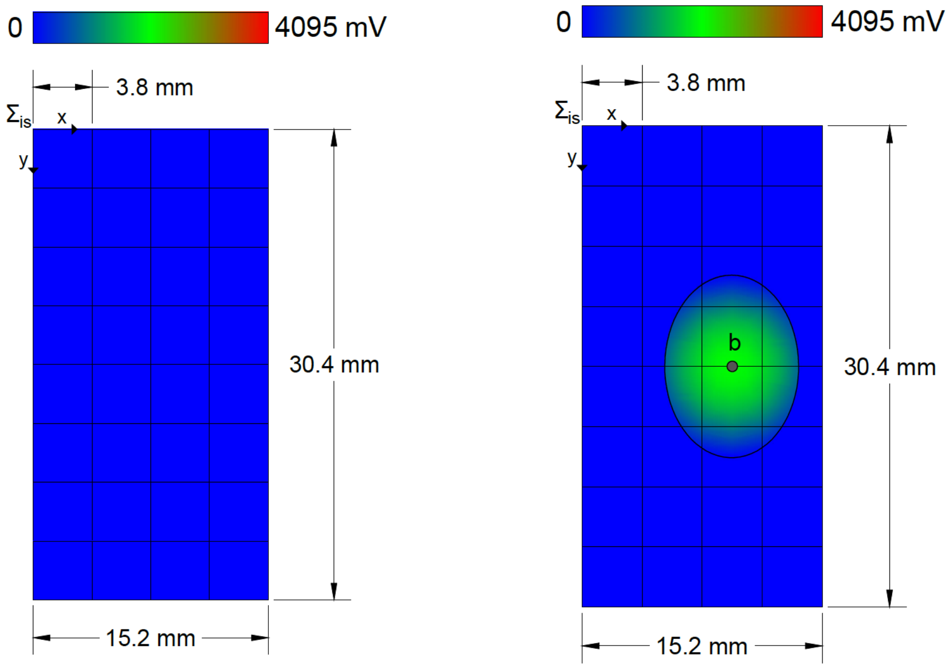
Figure 2.9: Picture of the tactile sensor WTS-FT 0480.

is considered as the summation of the forces sensed over all the texels. Figure 2.10b shows an example of a contact region highlighted with an ellipse. A colored bar at the bottom of Figure 2.10 indicates the electric signal measured by a textel (in mV). The range is between 0 mV, when no force is applied, and 4095 mV when the maximum force is applied. The latter corresponds to 1.23 N.

However, operating with the frictionless point contact model and with the WTS tactile sensors leads to a significant disadvantage: the tangential components of the grasping force are not sensed, so the actual contact force could be larger than the measured one. In experimentation, this problem is neglected and, for this reason, the results could be greatly affected.

The WTS-FT sensors are connected to a PC through a serial port with a transmission rate of 50 Hz.

The driver of the sensors provided by Weiss Robotic company has the basic functionalities for the configuration and the communication of the sensor. Moreover,



(a) Dimensions. **(b)** Elliptical contact region.

Figure 2.10: WTS tactile sensor matrix.

there is the possibility to read the pressure measurement on each texel. The pressure measurement is done using the function *getPressureOnTexel*. The members of the robotics section of IOC (Institute of Industrial and Control Engineering) - Universitat Politècnica De Catalunya developed a C++ library, called *WEISSlib*, to manage, configure and read information from the Weiss tactile sensors. *WEISSlib* provides complex functions that simplify the data management.

In the following part there is a description of *getContactInfo*, a fundamental function used in the manipulation strategy. *GetContactInfo* enables to obtain the barycenter, the contact force and the contact area between the object and the tactile sensors. *GetContactInfo* pseudo-code is shown in Algorithm 2 (extrapolated from «Dexterous manipulation of unknown objects using virtual contact points» [11]). The pressure of each texels of the sensor pad is read using the function $getPressureOnTexel(x, y)$. The output pressure must be larger than a threshold *pressureThreshold* in order to be considered as a valid value. In such a way, the measurement is acquired and 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 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:

$$b_x = \text{texel}_{\text{wide}} \frac{\sum_{x=1}^{x_{\max}} (x\ p)}{\sum p} \tag{2.6}$$

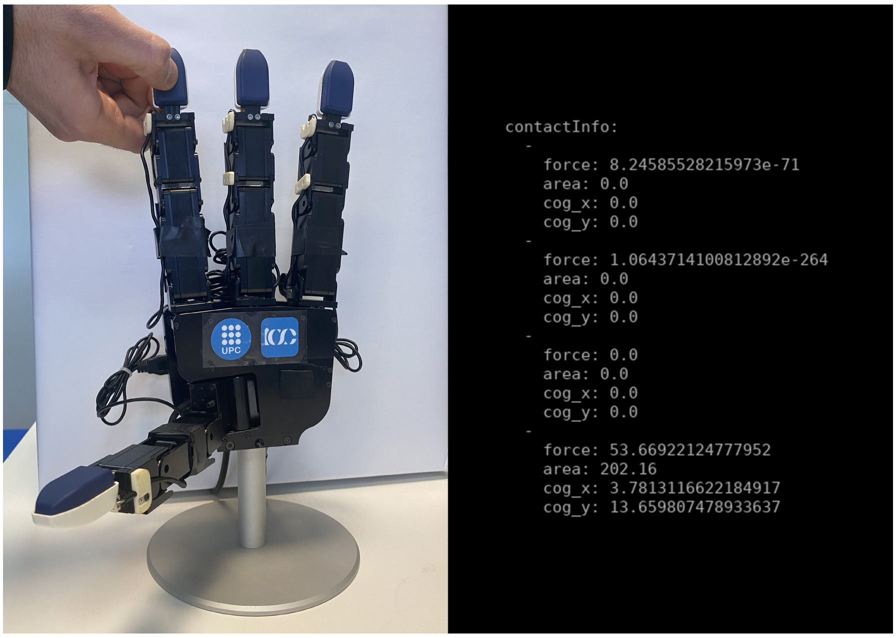
$$b_y = \text{texel}_{\text{high}} \frac{\sum_{y=1}^{y_{\max}} (y\ p)}{\sum p} \tag{2.7}$$

  - p is the measured pressure on the texel under investigation (current textel);
  - x is the current column, y is the current row of the textel considered;
  - $x_{\max}$ is the total number of columns and $y_{\max}$ is the total number and rows in the sensor pad.

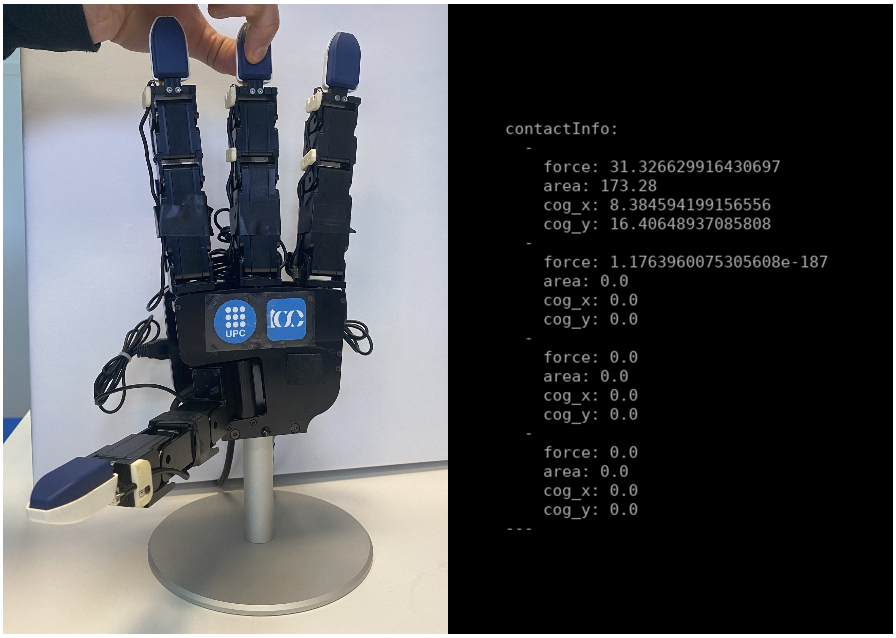
Algorithm 2 getContactInfo

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

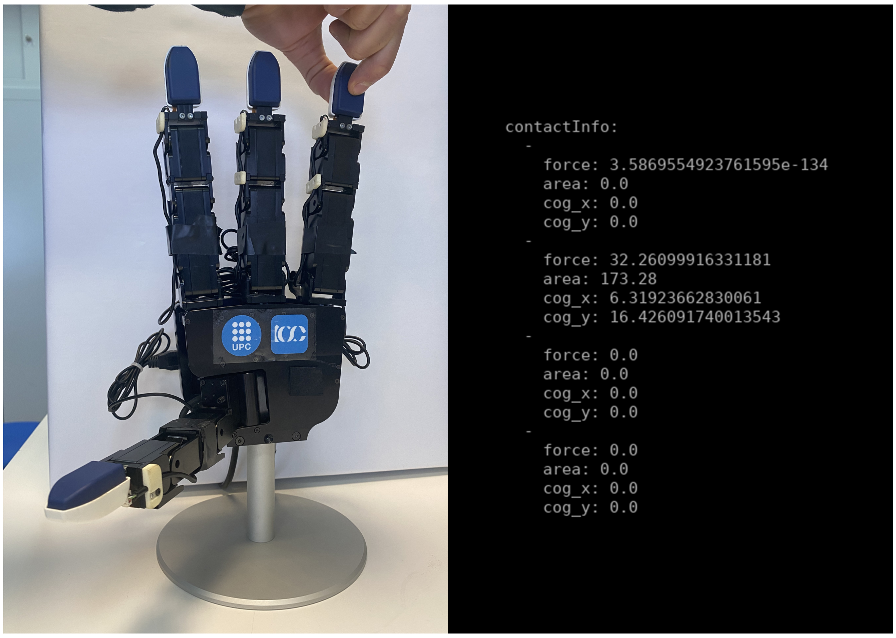
Figure 2.11 shows an example of the data obtained by Algorithm 2 according to the contact with the fingertips involved.



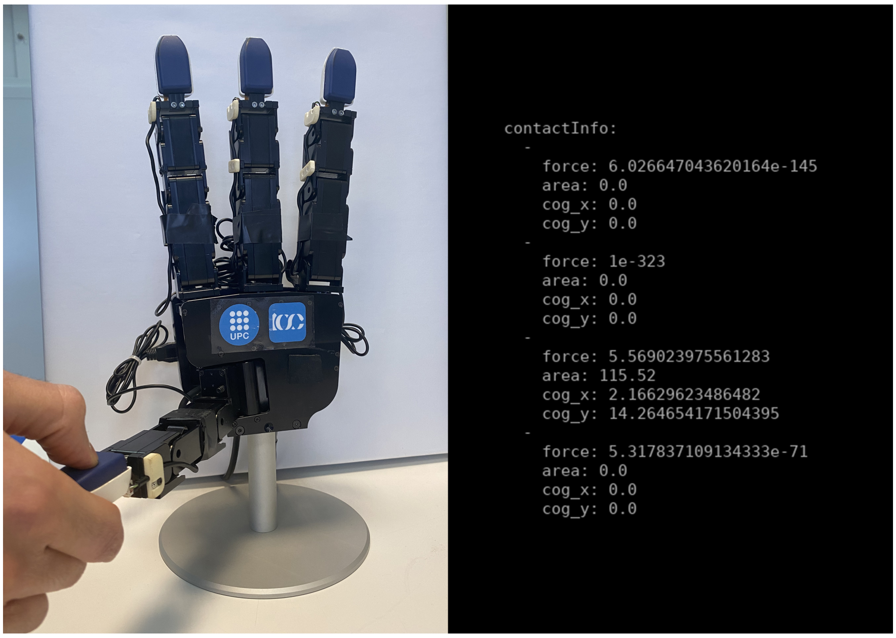
(a) Contact with Index fingertip.



(b) Contact with Middle fingertip.



(c) Contact with Ring fingertip.



(d) Contact with Thumb fingertip.

Figure 2.11: Data obtained from the contacts with the fingertips.

For the following explanation focus the attention on Figure 2.12. The coordinates of the barycenter of the contact region, obtained from Algorithm 2, are computed with respect to the sensor reference frame Σ_{i_s}. In order to include the contact information in the kinematics of the hand it is necessary to refer the contact point (CP) to $\Sigma_{i_{TIP}}$ reference frame by means of the following equations:

$$CP_{y-\Sigma_{i_{TIP}}} = (CP_{x-\Sigma_{i_s}} \cdot 0.001) - 0.0057; \tag{2.8}$$

$$CP_{z-\Sigma_{i_{TIP}}} = (CP_{y-\Sigma_{i_s}} \cdot 0.001) + 0.0137; \tag{2.9}$$

$CP_{x-\Sigma_{i_s}}$ and $CP_{y-\Sigma_{i_s}}$ are the coordinates of the contact point in Σ_{i_s}.
$CP_{y-\Sigma_{i_{TIP}}}$ and $CP_{z-\Sigma_{i_{TIP}}}$ are the coordinates of the contact point in $\Sigma_{i_{TIP}}$.
The multiplication for 0.001 is because the contact point coordinates must be converted from meters to millimeters.

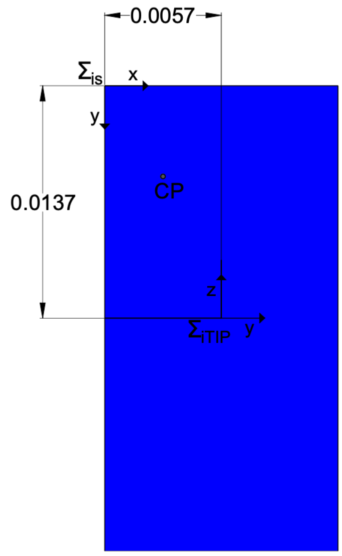


Figure 2.12: Relation between the $\Sigma_{i_{TIP}}$ and Σ_{i_s} in Weiss Tactile Sensor matrix.

2.3 Robot Operating System (ROS)

The following section describes the software tool that enables communication with the hardware devices. The topics behind ROS are innumerable. Consequently, only the fundamental concepts and the main tools employed in the thesis project are described below.

2.3.1 Definition and Key Issues

Robot Operating System (ROS)[2] is the core of the communication between user and robot. ROS is a flexible open-source framework for writing robot software. It is a collection of tools, libraries, and conventions used to facilitate the creation of complex and robust robot behavior, which can be managed by a wide variety of robotic platforms. ROS provides services such as hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, and package management.

Packages are the main structure for organizing ROS software. They contain processes, libraries, configuration files, datasets, and all the files files that are used by the system at run-time. They are the smallest structure possible to find within a ROS-based system. At the filesystem level, the package is represented by a folder. The folder structure includes some subfolders to manage the fundamental elements for its development, in particular:

- *include/package_name*: contains mostly C++ headers;
- *msg/*: folder containing files related to message types (message types);
- *src/package_name/*: folder that contains the source files;
- *srv/*: folder containing service types;
- *scripts/*: folder that contains scripts executable by the software;
- *CMakeLists.txt*: extremely important file for compiling the package using CMake;
- *package.xml*: file that contains the structure of the package in XML format;
- *CHANGELOG.rst*: inside the file are inserted changelogs related to updates. They will be used by ROS API inside binary files and during the creation of the Wiki page of the package.

In the ROS framework, the process occurs in nodes that can be run on the same or different CPU-cores or computers. Nodes are coordinated by a master node responsible for assigning a name and registering each node connected to the system as a publisher/subscriber (communication using topics) or service provider (communication using services).

[2] `https://wiki.ros.org/`

Communication Using Topics

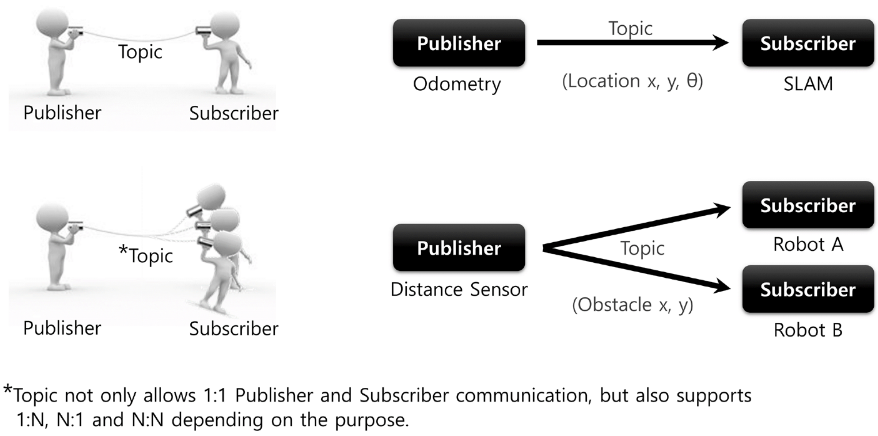


Figure 2.13: Communication using topics.

A topic is a channel for communication and for exchanging information between nodes. In function of the communication with a topic, a node can be:

- a publisher node, if it publish a message into a topic;
- a subscriber node, if it gets the information from a topic.

Topics handle information through messages. Message descriptions are stored in *.msg* files. There are lots of predefined message types in ROS, and also it is possible to define your message type and store it in the *msg/* subdirectory of a ROS package. Figure 2.14 and Figure 2.15 illustrate two general examples of communication with messages. In the main C++ code implemented are used two subscribers nodes:

- */allegro_hand_ros/joint_states* that gives the real time values of the joints;
- */weiss_sensor_ros/sensors_contactInfo* that obtains data from the sensors.

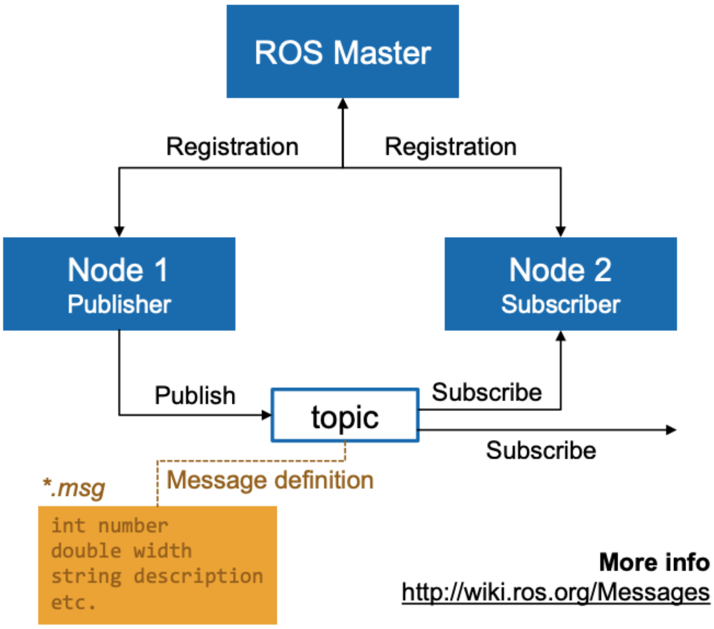


Figure 2.14: Example of communication using topics with messages definition.

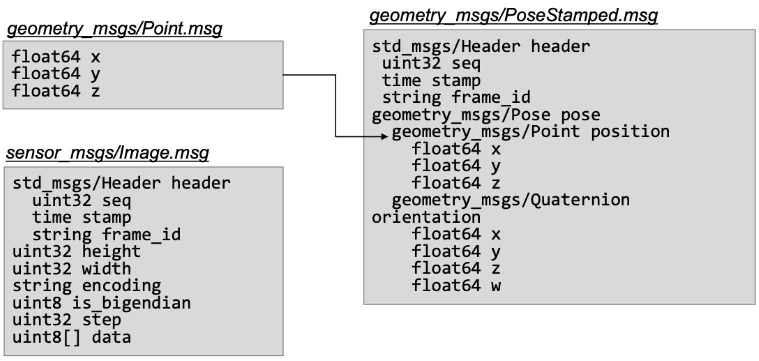


Figure 2.15: "PoseStamped" message example.

Communication Using services

Services are another method of communication for nodes in ROS. Services are based on a request/reply interaction. While topics allow nodes to subscribe to data streams and get continual updates, services only provide data when specifically called by a client. In Table 2.4 are indicated the services used in the main C++ code with a brief description.

Server	**Request**	**Response**	**Description**
JointLimits.srv	bool foo[3]	float64[ ] min float64[ ] max	- foo variable as request. - Range of motion of each joint as response.
SetController.srv	string tipo	bool succes	- A string "tipo" where is indicated the type of controller (for example PID) as request. - Boolean variable that indicates if the operation is done or not as a response.
SetDesPos.srv	float64[ ] pos	bool success	- Desired position for all the joints (in radiants) as request. - Boolean variable that indicates if the operation is done or not as response.
SetDesPosReflexxes.srv	float64[ ] pos float64 min_duration	bool success	- Desired position for all the joints (in degrees) and the relative time (in seconds) to reach the desired values as request. - Boolean variable that indicates if the operations done or not as response.
SetDesPosVel.srv	float64[ ] pos float64[ ] vel	bool success	- Desired position and velocities for all the joints (in degrees) as request. - Boolean variable that indicates if the operation is done or not as response.
SetDesTorque.srv	float64[ ] torque	bool success	- Desired torque for all the joints (in degrees) as request. - Boolean variable that indicates if the operation is done or not as response.
StopHand.srv	bool foo	bool success	- A foo variable as request (the real request is to stop the hand). - Boolean variable that indicates if the operation is done or not as response.

Table 2.4: Services used in the Manipulation node.

Nodes belonging at this thesis project are developed in C++ and are shown in Figure 2.16. Analyzing the functions of each node:

- Allegro Node is a module to command and read the current state of the allegro hand;
- PID controller is a module to control the communication with the hand;

[3] Foo is a term used by programmers as a placeholder for a value that can change, depending on conditions or on information passed to the program. Foo is formally known as metasyntactic variable.

- WTS-FT Nod is a module to receive the tactile sensors data;
- Manipulation Node is the principal module where is implemented the manipulation algorithm;
- GUI rviz is a module to visualize the hand (deepened in the next Section 2.3.2).

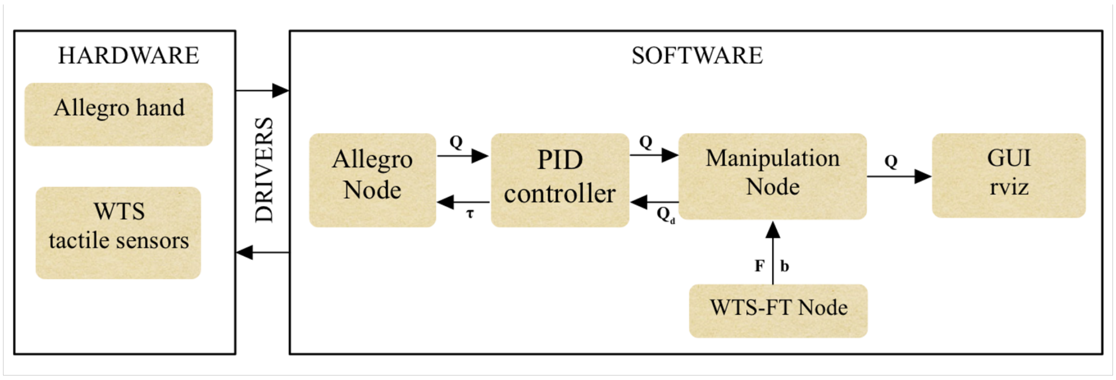


Figure 2.16: Scheme of the communication between software (nodes in ROS framework) and hardware.

2.3.2 ROS Tools

ROS not only provides a solid architecture for development, but also distributes tools for system analysis and debugging. ROS tools are extremely useful to programmers especially for testing procedure both in simulated and real environments.

URDF File

```
<!-- Index -->
<link name="link_0.0">
   <visual>
      <geometry>
         <mesh filename="package://allegro_hand_ros/conf/urdf/meshes/link_0.0.STL" />
      </geometry>
      <material name="black" />
   </visual>
   <collision>
      <geometry>
         <box size="0.0196 0.0275 0.0164" />
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0.0082" />
   </collision>
   <inertial>
      <mass value="0.0119" />
      <origin xyz="0.012303 -0.012311 0" />
      <inertia ixx="0.0" ixy="0.0" ixz="0.0" iyy="0.0" iyz="0.0" izz="0.0" />
   </inertial>
</link>
<joint name="joint_0.0" type="revolute">
   <axis xyz="0 0 1" />
   <limit effort="0.7" lower="-0.47" upper="0.47" velocity="3" />
   <parent link="base_link" />
   <child link="link_0.0" />
   <origin rpy="0.08726646255 0 0" xyz="0 -0.0435 -0.001542" />
   <dynamics damping="0.021307336803783977" friction="0.06754533316058077" />
</joint>
```

Figure 2.17: Portion of the URDF file of the Allegro Hand.

The Unified Robotic Description Format (URDF) is an XML[4] file format used in ROS to specify all the elements of a robot. Using XML tags, the URDF can represent:

- the kinematic description of the robot;
- the dynamic description of the robot;
- the visual representation of the robot;

[4]https://en.wikipedia.org/wiki/XML

- the collision model of the robot.

For example, in Figure 2.17 is shown a portion of the URDF file of the Allegro Hand where are specified the features of link_0.0 and joint_0.0, respectively l_{I1} and q_1 belonging to the Index. Moreover, in Figure 2.18 is shown the kinematic chain of Allegro Hand obtained from URDF file.

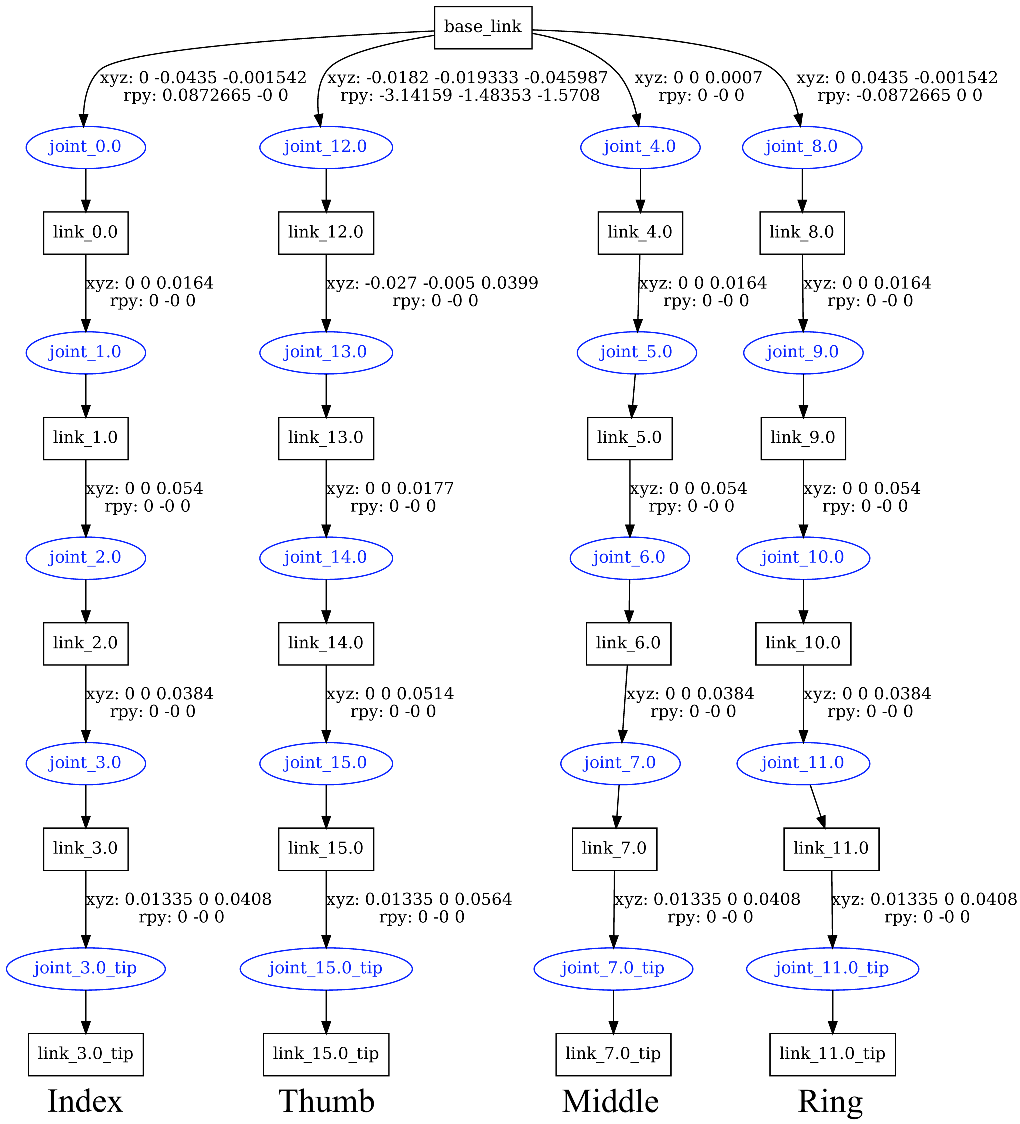


Figure 2.18: Allegro Hand kinematic chain.

Rviz Visualization Tool

Rviz is a 3D visualization tool for ROS. The GUI-rviz Node allows visualizing a rviz scene (Figure 2.19) 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) and the contact points among other elements. The scene is described using an URDF file with the inclusion of CAD models.

Since rviz is a mere visualization tool and not a simulation tool, an alternative solution had to be found to test the manipulation node without the real robotic device. The implemented solution by the researchers of IOC consists in creating a fake node of the Allegro Hand. This node is a straightforward node that simulates the hand and can be run without the communication with the real robot.

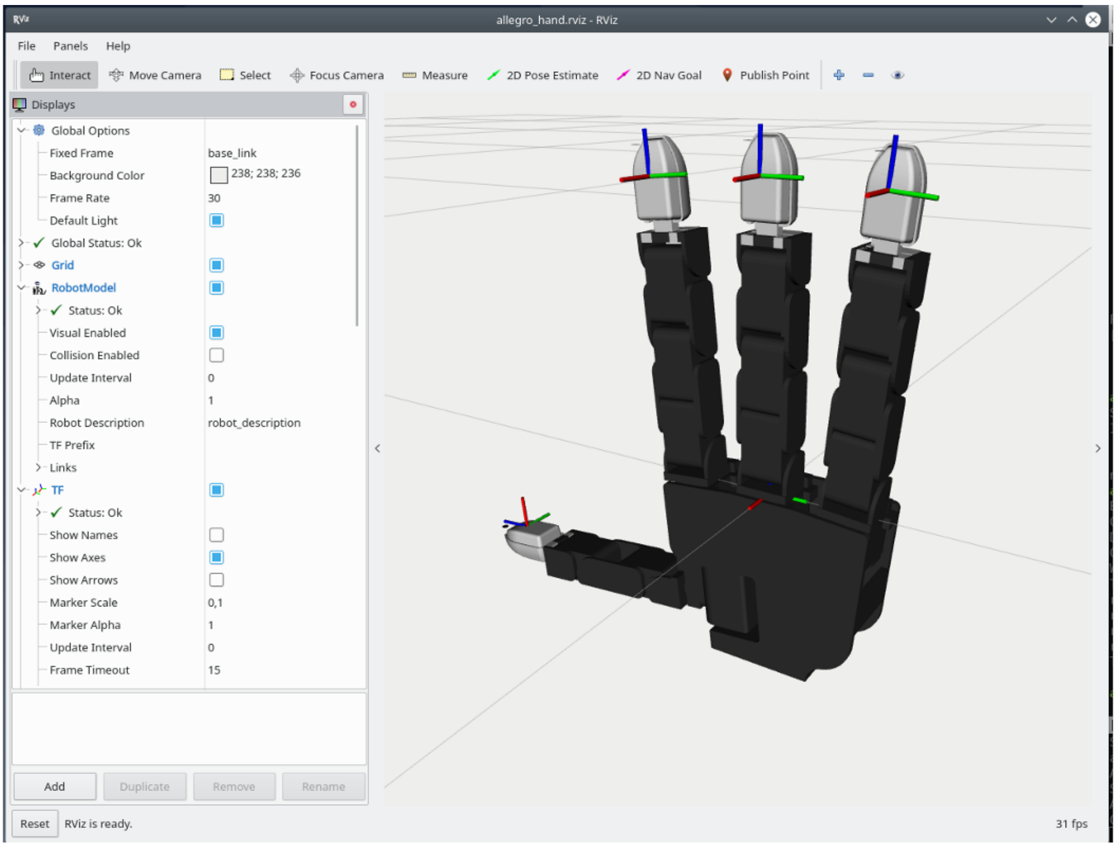


Figure 2.19: GUI - rviz.

The tf Tool

The tf library (transform library) lets the user keep track of multiple coordinate frames over time. It maintains the relationship between coordinate frames in a tree structure buffered in time (i.e. with transforms as edges and coordinate frames as nodes). The tree informs the user about transforms (Appendix C) between any two coordinate frames in any desired instant of time.

KLD Library

The Kinematics and Dynamics Library is a meta-package developed and distributed by Orocos Project[5]. KDL is widely employed in robotics projects because it is able to define tree structure to represent the kinematic and dynamic parameters of a robot mechanism. As for the GUI-rviz, KDL construct a KDL tree from an XML robot representation in URDF. Moreover, the library provides numerical solvers to compute forward/inverse position kinematics and dynamics.

[5] `https://www.orocos.org/index.html`

Chapter 3

Manipulation Strategies

From the June 2001 issue of "Mechanics of robotic manipulation", page 1, reference [10], «manipulation is the process of using one's hands to rearrange one's environment».

For humans, manipulating an object is an innate ability that is automatically performed regardless of the complex nature of the action. Instead, being able to program a robot to carry out any manipulation action requires being aware of the innumerable number of variables involved. Therefore, manipulation becomes an engineering science full of curiosities and aspects to be explored, which require to be consolidated through the implementation of scientific methods.

Manipulation in robotics is accomplished in many different ways. The main goal of this thesis is to set up a dexterous in-hand manipulation of an unknown object using only the movement possibilities supplied by the fingers, without resorting to the aid provided by the wrist and by the arm. The concept behind the project is to emulate the typical movements that a human being does to rotate an object. Therefore, just as in the case of a human hand, the fingers of an anthropomorphic robotic hand are used to rotate an object forward and backward around an arbitrary axis.

3.1 Hypotheses and Preliminary Remarks

As already mentioned in the introductory part, this project is a generalization of what was developed by Montaño and Suaréz, explained in the article "Dexterous Manipulation of Unknown Objects Using Virtual Contact Points" [12]. The two projects differ mainly in their manipulation strategy. The solution proposed by the researchers mentioned above consists in using the Thumb only and exclusively as a support finger while the motion of the object is caused by Index and Middle. The logical consequence of this choice is to allow the rotation of the object only around a specific axis. Instead, the solution proposed in the current thesis project (Section 3.3) assumes that the movement of the object is owed to all the finger of the Allegro Hand (including the Ring and the Thumb) and, consequently, is allowed the rotation of the object around any axis in the space.

There are basic hypothesis in common for both manipulation strategies:

- The procedure is based only on the tactile information obtained by the tactile sensor located at the fingertips and on the kinematic structure of the hand; no other external feedback sources, such as a vision system, are considered. As specified in the paragraph 2.2, it must be underlined that the tactile sensors are used only to know the position of the contact point on each fingertip and the module of the corresponding contact force.
- The contact between the object and the fingertips of the hand is modeled with the punctual contact model. The current contact point is the barycenter of the contact region resulting from the tactile sensors. Similarly, regarding the current contact force:
  - the magnitude is the summation of all the magnitudes of the forces sensed at each textel in the actual contact region;
  - the point of application is the current contact point;
  - the orientation is always perpendicular to the contact region (so perpendicular to the surface of the fingertip);
  - the sense is pointing towards the object.
- The manipulated objects are rigid bodies, so all the possible deformations are zero or so small that they can be neglected. Moreover, all the object's physical properties (i.e., shape, mass, center of mass, stiffness, and so on) are unknown.
- The friction coefficient is unknown neither identified during the manipulation. It is assumed to be above a minimum value that permits an object's grip.
- The controller behind the movement of the joints is a closed position controller. An external force disturbance in the controller is not directly considered, but as soon as both strategies are investigated in the following sections (Section 3.2 and Section 3.3), it will be evident that there is a relation between force (exchanged between object and fingertips) and position of the joints.
- The initial grasp algorithm optimization is not treated in either of the strategies described below.

In addition to the hypotheses, there are other aspects to consider in common to the two proceedings:

- The user decides the sense of the rotation movement, and the system autonomously determines the finger movements.

- The strategies are based on algorithms and geometric considerations performed in the operational space. Once the results are obtained, they are "translated" into the joint space using Inverse Kinematics. If the results are within the range of motion allowed by the hand structure, the movement will be performed; otherwise, the algorithm will stop.

- The contact points resulting from the geometric algorithm lie "inside" the object in order to apply a force on the object surface. If the fingertips are positioned exactly on the surface of the object, they will not produce grasping forces on it. For this reason, is adopted the "virtual contact points method" (which is explained in more detail later on in Section 3.2.2).

3.2 Previous Manipulation Strategy: Dexterous Manipulation of Unknown Objects Using Virtual Contact Points

3.2.1 Idea and Basic Concept

The current section is entirely extrapolated from the 2019 issue of Robotics Journal, volume 8, article number 86, «Dexterous Manipulation of Unknown Objects Using Virtual Contact Points», by Andrés Montaño and Raúl Suárez [12]. All the implementation, images, algorithms, and equations are merely reported with comments and observations, enabling the integration into the current thesis project. It is essential to expose them because it is the starting point from which the new strategy is elaborated.

The basic hypothesis of the manipulation strategy elaborated by Montaño and Suárez are reported in the previous Section 3.1. Moreover, it is essential to specify another time that only three fingers of the Allegro Hand, namely Index, Middle, and Thumb, are involved in the manipulation; Ring is not included. A tripod grasp strategy [4] is adopted, i.e. the Thumb works opposite to the other two fingers (abduction movement). 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.

3.2.2 Iterative Algorithm

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.

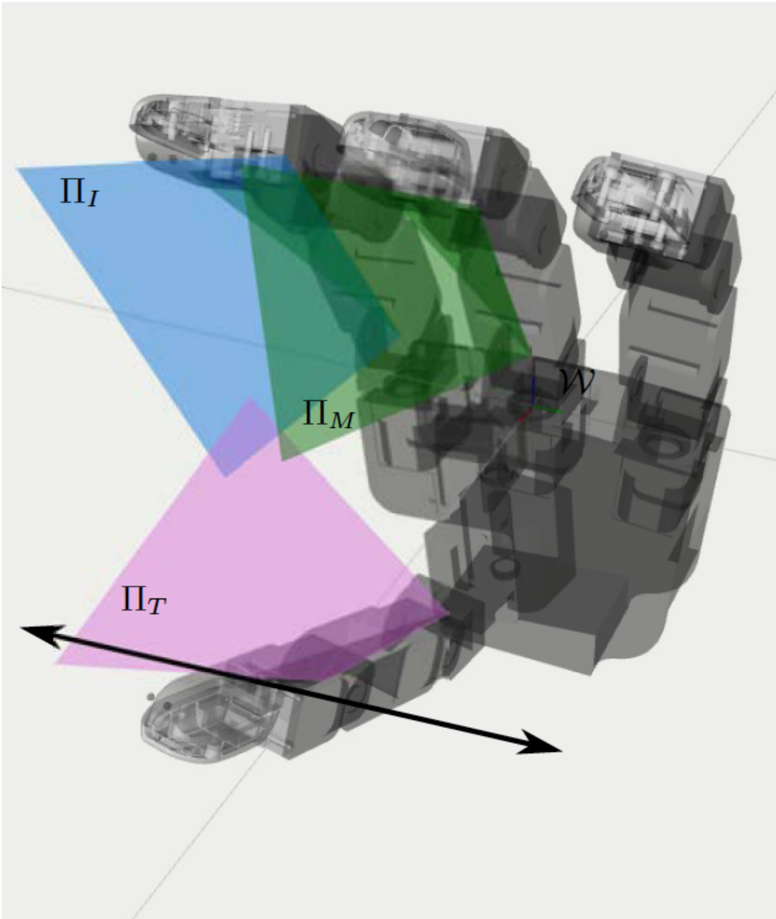


Figure 3.1: Allegro hand with the finger working planes Π_i for Index, Middle and Thumb and with the axis for the object rotation [Montaño and Suaréz 12].

First of all, it is worth remembering that $i \in \{I, M, T\}$ indicates the value of the finger under observation while with $\Sigma_{i_j}, j \in \{1, ..., n_i\}$ it is defined the position of each finger link with respect to the absolute reference frame $\mathcal{W}$.

The axis around which the object is rotated is parallel to the palm and passes through the point of contact between the thumb and the object. (Figure 3.1). In this way, the values of joint 1 and joint 5 (respectively the joint at the base of the Index and the joint at the base of the Middle) are not changing during the iteration procedure. To be more precise, the values are fixed to the initial values given by the initial closure. In the case of the Thumb, instead, not only the value of joint 14 is fixed, but also the value of joint 13 (abduction movement) is fixed.

The flexion/extension joints (i.e. joints 2,3,4 for Index, joints 6,7,8 for Middle, joints 15,16 for Thumb) of each finger i move the finger within a working plane Π_i. This working plane 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. Figure 3.2 clarifies on the joints involved.

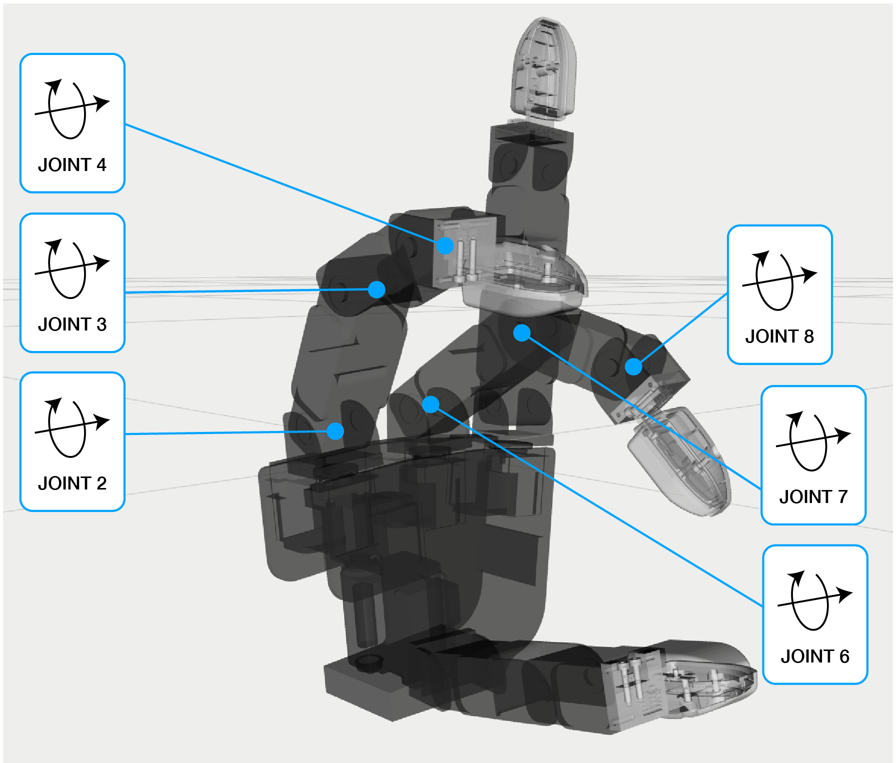


Figure 3.2: Joints involved in Montaño and Suárez strategy.

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. In this way, the fingers can perform cooperative movements, and the object can be rotated around the axis mentioned in the first part of this subsection.

Virtual Contact Points

The commanded fingertip positions is located "inside" the object and, since they are not physically reachable, they will be called "virtual contact points". 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 **case of the leading fingers (Index and Middle)** the computation is done as follows. From now on, let's pay attention to Figure 3.3. Supposing that the object is already grasped, define the contact points between the object and the fingertips as current contact points $\mathbf{P}_{i_k}$. The goal is to find the points at the next iteration $\mathbf{P}_{i_{k+1}}$. The indexes k and $k+1$ denote the current and next iteration, respectively.

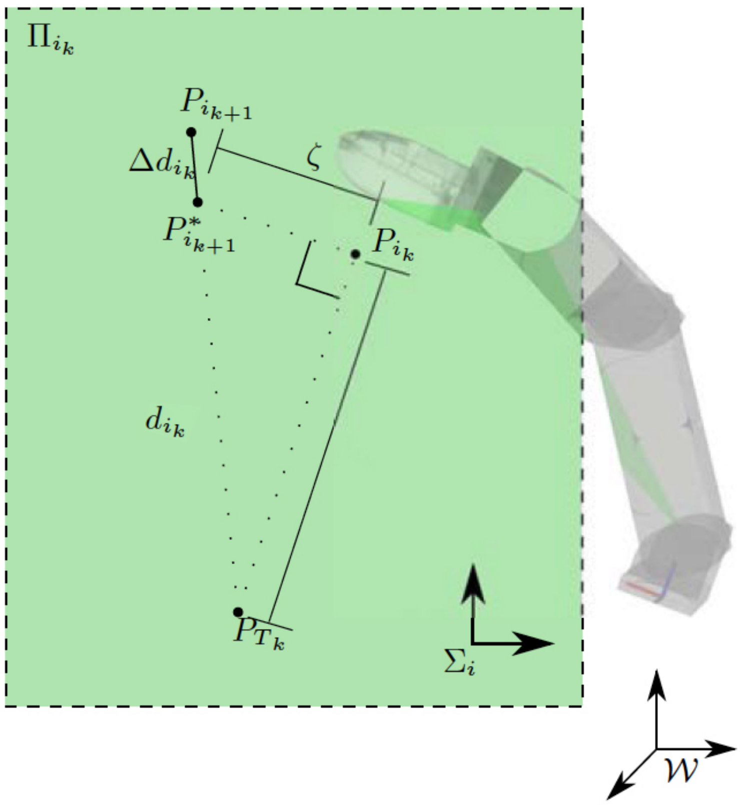


Figure 3.3: Example of the computation of $\mathbf{P}_{i_{k+1}}, i = \{I, M\}$, when the contact force F_{i_k} is larger than F_{i_d} (i.e., $e_{i_k} \leq 0$) [Montaño and Suaréz 12].

Two auxiliary points $\mathbf{P}^*_{i_{k+1}}, i = \{I, M\}$, called virtual contact points, are defined as the points resulting from a displacement $\pm\,\zeta$ of P_{i_k} along the line perpendicular to the segment between $\mathbf{P}_{i_k}$ and $\mathbf{P}_{T_k}$. The intention is to make the axis of rotation passing through $\mathbf{P}_{T_k}$ and perpendicular to the drawing (it is as if the axis were entering/exiting the sheet). It is fundamental to underline that the contact point between the object and the Thumb fingertip $\mathbf{P}_{T_k}$ is projected in the working plane of the Index or Middle Π_i. The sign of the displacement ζ depends on the desired sense of rotation for the current iteration. Thus,

$$\mathbf{P}^*_{i_{k+1}} = \mathbf{P}_{i_k} \pm \zeta \hat{\boldsymbol{p}} \tag{3.1}$$

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 current contact force F_{i_k}. The module of F_{i_k} must remain within a threshold around a desired value force 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. F_{i_d} is chosen equal to force applied between fingertip and object at the instant of the initial grasp, when the object is correctly grasped. In order control the value of the grasping forces, a force error e_{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.

$$e_{i_k} = F_{i_k} - F_{i_d} \tag{3.2}$$

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 $\mathbf{P}_T$,

$$d_{i_k} = \left| \mathbf{P}^*_{i_k} - \mathbf{P}_{T_k} \right| \tag{3.3}$$

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_{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:

$$\mathbf{P}_{i_{k+1}} = \mathbf{P}^*_{i_{k+1}} + \Delta d_{i_k} \hat{\boldsymbol{p}}^*_i \tag{3.4}$$

with

$$\hat{\boldsymbol{p}}^*_i = \frac{\mathbf{P}^*_{i_{k+1}} - \mathbf{P}_{T_k}}{\left| \mathbf{P}^*_{i_{k+1}} - \mathbf{P}_{T_k} \right|} \tag{3.5}$$

and

$$\Delta d_{i_k} = \begin{cases} \lambda(|e_{i_k}| + e^2_{i_k}) & \text{if } e_{i_k} \leq 0 \\ -\lambda e_{i_k} & \text{if } e_{i_k} > 0 \end{cases} \tag{3.6}$$

being λ a predefined constant, empirically obtained.
In the definition of the distance d_{i_k} are defined different gains depending on the sign of e_{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 an intermediate virtual point. $\mathbf{P}_{T_{k+1}}$ is computed considering an adjustment with respect to the Index and Middle fingers as,

$$\mathbf{P}_{T_{k+1}} = \mathbf{P}_{T_k} + \Delta d_{T_k} \hat{\boldsymbol{p}}_T \tag{3.7}$$

with Δd_{T_k} calculated as the medium of the delta of distances obtained for Index and Middle (respectively Δd_{I_k} and Δd_{M_k}):

$$\Delta d_{T_k} = -\frac{\Delta d_{I_k} + \Delta d_{M_k}}{2} \tag{3.8}$$

and

$$\hat{\boldsymbol{p}}_T = \frac{\hat{\boldsymbol{p}}_I^* + \hat{\boldsymbol{p}}_M^*}{|\hat{\boldsymbol{p}}_I^* + \hat{\boldsymbol{p}}_M^*|} \tag{3.9}$$

Finally, the new hand configuration $\mathbf{Q}_{k+1}$ is computed using Inverse Kinematics of $\mathbf{P}_{i_{k+1}}$, $i = \{I, M, T\}$.

It is essential to notice that the kinematic of the hand is computed using KDL (Section 2.3.2), and inverse kinematic is obtained exploiting one of the iterative algorithms contained in the library. 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 3 summarizes what has been explained so far. The following link allows visualizing the videos of the experiments performed by Montaño and Suaréz.

`https://bit.ly/2lLvbDY.`

For the comments and the considerations on the results obtained is possible to refer directly to the article «Dexterous Manipulation of Unknown Objects Using Virtual Contact Points» [12].

Algorithm 3 Manipulation algorithm [Montaño and Suaréz 12].

Require: F_d
  procedure MANIPULATE
    $k \leftarrow 0$
    repeat
      Read the direction of rotation s_k
      Compute finger working planes Π_{i_k}
      Project $\mathbf{P}_{i_k}$ onto Π_{i_k}
      for $i = \{I, M\}$ **do** ▷ Computation for Index and Middle
        Compute $\mathbf{P}^*_{i_{k+1}}$ according to s_k
        Compute Δd_{i_k}
        Adjust $\mathbf{P}^*_{i_{k+1}}$ to obtain $\mathbf{P}_{i_{k+1}}$
      end for
      Compute Δd_{T_k} ▷ Computation for the Thumb
      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

3.3 Actual Manipulation Strategy: In-Hand Robotic Manipulation

3.3.1 Idea and Basic Concept

As pointed out several times, the previously implemented manipulation strategy (Section 3.2) allows a rotation of the object only and exclusively around an axis parallel to the palm of the hand and passing through the point of contact between the object and the thumb. The goal of the new manipulation strategy is to enable rotations around any axis in space, obviously taking into account the limits imposed by the structure of the Allegro Hand.

The basic hypothesis in common to both strategies are listed in Section 3.1.

The idea behind the new project is based on a simple geometric reasoning: after the object has been grasped by the robotic hand, each point of contact between object and fingertip must travel on circular trajectories in space around an arbitrary axis (defined by the user).

Differently from the previous strategy, the actual one allows the movement of all the joints in order to reach the desired position. Consequently, adding movement possibilities to the Ring and joints 1, 5, 13, 14 (respectively orientation joint at the base of the Index, orientation joint at the base of the Middle, thumb abduction joint, and thumb orientation joint) implies adding 8 DOF more with respect to the previous strategy. From now on, the range of work is $i = \{I, M, R, T\}$, not anymore $i = \{I, M, T\}$ as in the previous strategy, and the reference figure where are shown the joints involved is Figure 2.3.

The basic idea behind any grasp manipulation algorithm is that the contact points between the object and fingertips define a rigid object in three-dimensional space. In the current case, since the Ring is also involved and, consequently, being the number of the contact points equal to four, a tetrahedron is defined (Figure 3.4). It is no longer feasible to consider a tripod grasp.

The algorithm implemented is a prediction of the possible configuration that the fingers will have to adopt. The computation is done in the operational space and by means of Inverse Kinematics it is switched to the joint space in order to command (or not command) the movement. For Index, Middle and Ring, the same Inverse Kinematics is adopted. For Thumb is exploited the potentiality of one of the iterative algorithms contained in KDL to solve the Inverse Kinematics.

Also in the actual manipulation scenario, as the previous one, it is considered that the commanded fingertip positions is located "inside" the object. Virtual contact point strategy is adopted but readjusting it and making some modifications based on the differences between the strategies.

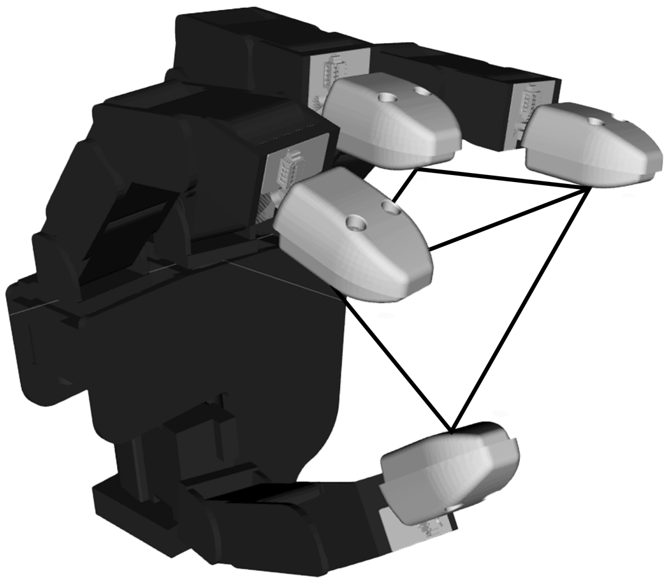


Figure 3.4: Drawing of a possible tetrahedron identified by the contact points between fingertips and object.

3.3.2 Iterative Algorithm

The algorithm implemented for the manipulation has been developed as a C++ code. The communication of the code with Allegro is established through ROS (Section 2.3). Figure 3.5 shows the algorithm flow chart. Each part of the flow chart it is analyzed in the following subsections. The nomenclature used is the one specified in the description of the Allegro Hand (Section 2.1).

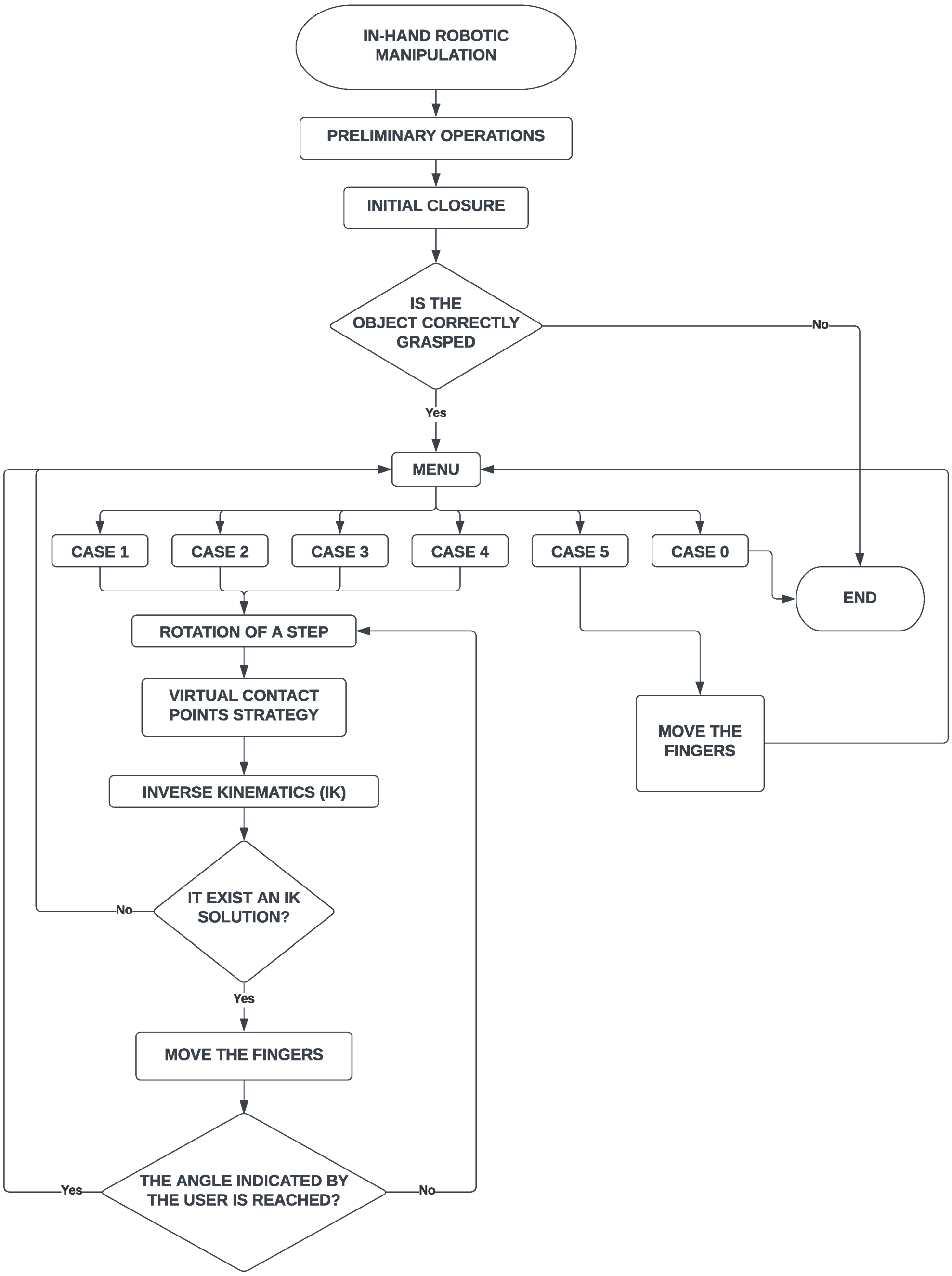


Figure 3.5: Actual manipulation strategy flow chart.

Preliminary Operations

The initial setup of the device is a fully open hand (Figure 3.6). In this situation the values of all joints are zero except the abduction joint of the thumb (joint 13) which is set at 0.92 rad. According to the actual strategy, for the implementation of the Inverse Kinematics of the Index, Middle and Ring it is essential to define for each finger a "Base Finger Reference Frame (BFRF)". BFRFs are reference systems that are located at the base of each finger or better, in the joints that define the orientation for Index, Middle and Ring and in the joint that defines abduction for the Thumb. BFRFs are fixed throughout the course of the experiments. Exploiting the potentiality of the tf library is possible to obtain the homogeneous transformation matrix (for each finger) of Base Finger Reference Frames w.r.t to Base Link Reference Frame: $\mathbf{T}^{\mathcal{W}}_{BFRF_i}$.

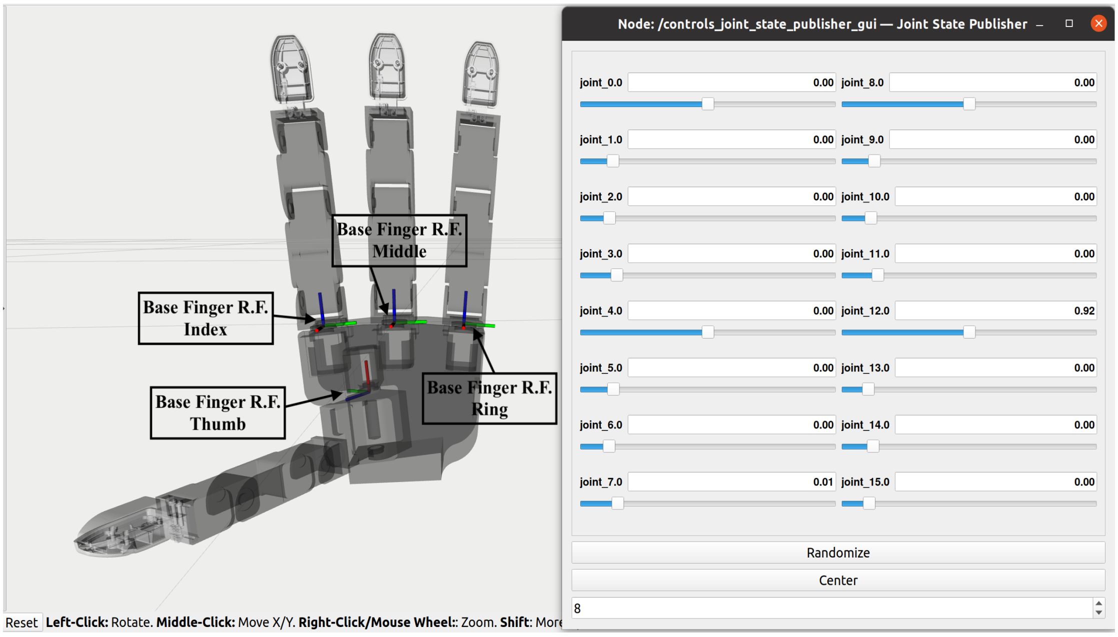


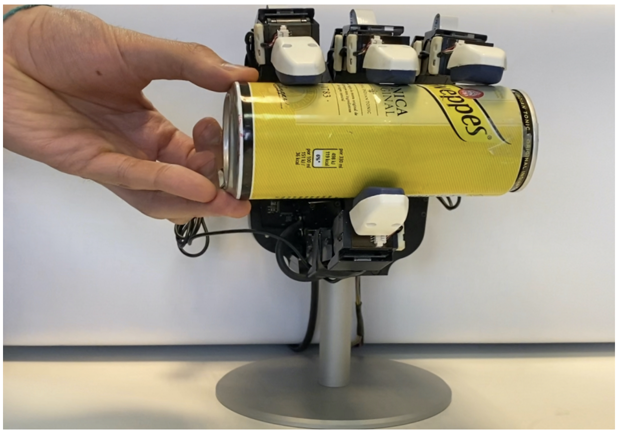
Figure 3.6: Locations of the Base Finger Reference Frames.

Initial Closure

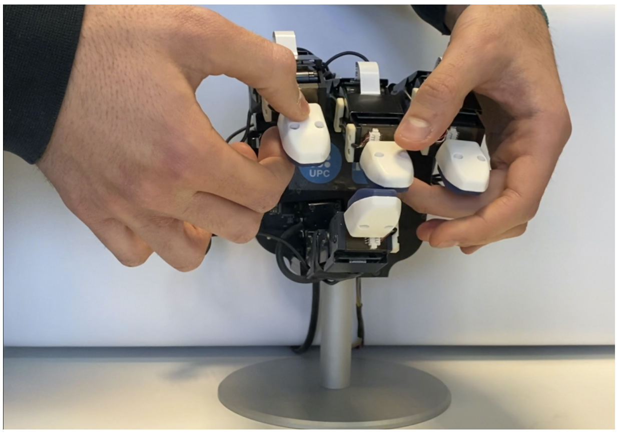
In a procedure independent from the current manipulation strategy, the values of the joints for the initial closure are computed resorting to a program included in the drivers of the Allegro Hand. The program allows some functionalities including the controller's setting of the robotic device. Once set the gravity controller, if the user manually moves the fingers in a certain position, the electric current sent to the joints allows the fingers to remain in that position balancing the gravitational force. Depending on the object to be grasped, the fingers are moved to a position that allows contact with the object so that it can be "measured". The object is then removed and Index, Middle and Thumb are moved closer to the thumb. Now, resorting to the same program, it is possible to read the current state of the hand, namely the values of the joints in the manually set position. In this way it should be guaranteed the necessary force which will allow the initial grasping. The actual values are saved and the program is shut down. Those values are the ones used for the initial closure of the current manipulation strategy and they do not belong to any criteria. The procedure goes against the assumption of an unknown object, but it is necessary to ensure the initial grasp. An example of the preliminary procedure for initial closure is shown in Figure 3.7. Figure 3.7a represents the "measurement" of the object; Figure 3.7b represents the instant when the fingers are moved closer to each other; Figure 3.7c represents the final position of the fingers to be saved. A demonstration video is attached at the following link:

`https://youtu.be/xI6b_BwpdfQ`.

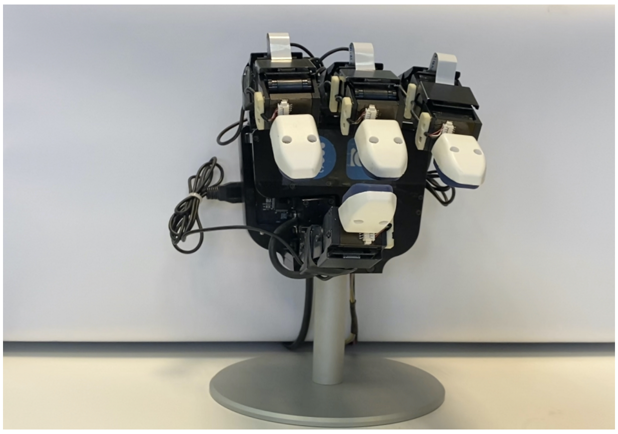
The initial closure algorithm consists merely of sending a close command to the fingers of the hand. The joints saved thanks to the procedure above explained are sent to the service "setDesiredPositionReflexxes". The device should reach the requested position, but once the sensors placed on the fingertips detect the presence of the object, the control system receives as feedback all the contact points and all the forces that the fingers are applying on the object (according to Algorithm 2 and to the considerations made in the final part of Section 2.2). Exploiting another time the tf2 library potentiality it is computed the homogeneous transformation matrix of the current contact point $\mathbf{P}_{i_k}$ (and its orientation) w.r.t World Reference Frame, so the homogeneous transformation matrix $\mathbf{T}^{\mathcal{W}}_{P_{i_k}}$. Moreover, the current contact force that each finger is applying on the object is, according to Section 3.2.2, the desired value force F_{i_d}.



(a) Measurement of the object to be handled.



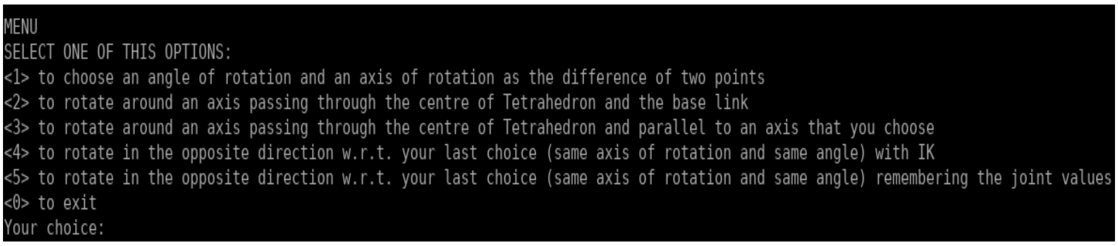
(b) Moving the fingers closer to each other.



(c) Final position of the fingers.

Figure 3.7: Measurement procedure for initial closure.

Menu



```
MENU
SELECT ONE OF THIS OPTIONS:
<1> to choose an angle of rotation and an axis of rotation as the difference of two points
<2> to rotate around an axis passing through the centre of Tetrahedron and the base link
<3> to rotate around an axis passing through the centre of Tetrahedron and parallel to an axis that you choose
<4> to rotate in the opposite direction w.r.t. your last choice (same axis of rotation and same angle) with IK
<5> to rotate in the opposite direction w.r.t. your last choice (same axis of rotation and same angle) remembering the joint values
<0> to exit
Your choice:
```

Figure 3.8: Menu.

The computer executes a printout of a menu (Figure 3.8) whose main objective is to have a simple interaction between the user and the program. Employing a *switch*[1] selection control mechanism, it allows the user to choose amongst the following cases:

- CASE 0: to end the program.
- CASE 1: an axis of rotation as the difference between two points in space and an angle of rotation. Moreover, it is possible to decide whether to rotate in accordance with the chosen axis or in the opposite direction.
- CASE 2: an axis of rotation passing through the centre of the Tetrahedron (individuated by the contact points between the fingertips and the object) and the origin of the World RF (coordinates $(x, y, z) = (0{,}0{,}0)$). Even in this case, the user chooses the angle of rotation. It is worth underlining that if for any of the fingers there is no point of contact with the object, the program will return an error.
- CASE 3: the direction of an axis of rotation passing through the center of the Tetrahedron. Even in this case, the user chooses an angle of rotation, and if for any of the fingers there is no point of contact with the object, the program will return an error.
- CASE 4: to rotate in the opposite direction with respect to the last choice (namely the same axis of rotation and the same angle of rotation) by means of Inverse Kinematics (explained further on). Of course, this case is limited by the existence of an axis and an angle already defined, otherwise, the program will return an error. Moreover, using Inverse Kinematics to rotate in the opposite direction means having different joint solutions with respect to the

[1] `https://en.wikipedia.org/wiki/Switch_statement`

previous phase. Therefore, most probably, the starting position will not be reached.

- CASE 5: identical to CASE 4, but with the goal of returning to the initial position. To do that, all the values of the joints explored in the previous rotation are saved and retraced just by commanding the hand in the same positions, without taking into account forces and geometries that come into play.

As seen on the flow chart (Figure 3.5), when one of cases 1,2,3,4 is selected, an internal loop computation starts. The fundamental passages of the loop are explained in the following sections.

Rotation of the Points Around the Axis

At this point, the angle chosen by the user is divided in a series of equals small angles θ_{step}. In this way, it is safeguarded that if the user chose a rotation angle too wide to be reached, the hand would move as far as its constraints would allow. Therefore, the point $\mathbf{P}_{i_k}$ is rotated of θ_{step} around the axis chosen (Figure 3.9).

The rotational function is based on quaternion's algebra. First of all, the point $\mathbf{P}_{i_k}$ is extrapolated from $\mathbf{T}^{\mathcal{W}}_{P_{i_k}}$ and obviously its coordinates are referred to World Reference Frame. Secondly, the axis chosen is normalized obtaining the unit axis $\vec{\mathbf{r}}$. To rotate $\mathbf{P}_{i_k}$ by an angle θ_{step} around the unit axis $\vec{\mathbf{r}}$ it is necessary to form:

- the quaternion associated to the point $\mathbf{P}_{i_k}$:

$$\mathbf{Q}_{P_{i_k}} = (0, P_{i_{k_x}}, P_{i_{k_y}}, P_{i_{k_z}}) \tag{3.10}$$

- the rotation quaternion that takes into account the angle of rotation θ_{step} and the rotation axis $\vec{\mathbf{r}}$:

$$\mathbf{Q}_{rotation} = (\cos(\frac{\theta_{step}}{2}), r_x \sin(\frac{\theta_{step}}{2}), r_y \sin(\frac{\theta_{step}}{2}), r_z \sin(\frac{\theta_{step}}{2})) \tag{3.11}$$

The new hypothetical virtual contact point $\mathbf{P}^*_{i_{k+1}}$ is the last three components of the resultant quaternion:

$$\mathbf{Q}_{res} = \mathbf{Q}_{rotation} \cdot \mathbf{Q}_{P_{i_k}} \cdot \mathbf{Q}^{-1}_{rotation} \tag{3.12}$$

Note that the rotation axis must pass through the origin; otherwise, it must be translated to pass through the origin before applying the rotation and then translated back to its original position.

The orientation (rotational matrix) of $\mathbf{P}^*_{i_{k+1}}$ is computed as:

$$R_{P^*_{i_{k+1}}} = R_{P_{i_k}} \cdot R_{rotation} \tag{3.13}$$

LEGEND
$x - y - z$: World Reference Frame
$x_k - y_k - z_k$: pose of the contact point at iteration k
$x_{k+1} - y_{k+1} - z_{k+1}$: pose of the contact point at iteration $k+1$
θ_{STEP}: rotation angle
$\vec{r}$: axis of rotation

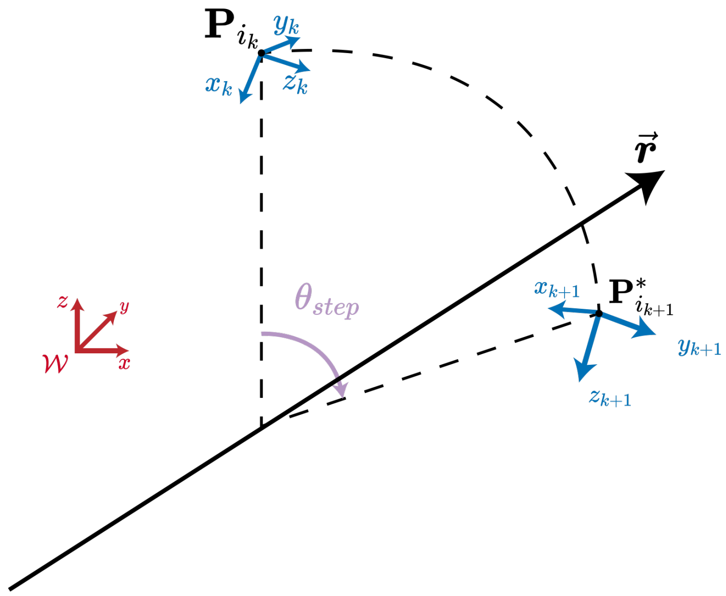


Figure 3.9: $\mathbf{P}_{i_k}$ rotation around $\vec{\mathbf{r}}$ of θ_{step}.

- $R_{P_{i_k}}$ rotational matrix derived from the homogeneous transformation matrix of $\mathbf{P}_{i_k}$ w.r.t. World RF
- $R_{rotation}$ rotational matrix derived from $\mathbf{Q}_{rotation}$.

Virtual Contact Points and Force Compensation Error

The idea of Virtual Contact Point and Force Compensation Error described in Section 3.2.2 is adapted to the case under consideration.

The new hypothetical virtual contact point $\mathbf{P}^*_{i_{k+1}}$ (derived from the rotation) is adjusted for each finger as a function of force error. The equation that permits to obtain the final hypothetical contact point $\mathbf{P}_{i_{k+1}}$ is:

$$\mathbf{P}_{i_{k+1}} = \mathbf{P}^*_{i_{k+1}} + \Delta d_i \hat{\boldsymbol{p}}^*_i \tag{3.14}$$

- $\mathbf{P}^*_{i_{k+1}}$ is the virtual contact point.
- Δd_i is the adjustment that allows changing the grasping force. Its value is the same considered in Section 3.2.2. To simplify the comprehension of the current treatment are reported again the equations 3.6 and 3.2:

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

$$e_{i_k} = F_{i_k} - F_{i_d}$$

λ is a predefined constant empirically obtained $\left[\frac{m}{N}\right]$.
e_{i_k} is the force error.
F_{i_k} is the current force measured by the sensor.
F_{i_d} is the desired force (chosen equal to the force sensed at the initial closure).

- $\hat{\boldsymbol{p}}^*_i$ is the direction of the adjustment and, differently to the equation 3.5, is equal to:

$$\hat{\boldsymbol{p}}^*_i = \begin{cases} \frac{\mathbf{P}^*_{i_{k+1}} - \mathbf{P}^*_{T_{k+1}}}{\left|\mathbf{P}^*_{i_{k+1}} - \mathbf{P}^*_{T_{k+1}}\right|} & \text{if} \quad i = \{I, M, R\} \\ \frac{\frac{\left(\mathbf{P}^*_{I_{k+1}} + \mathbf{P}^*_{M_{k+1}} + \mathbf{P}^*_{T_{k+1}}\right)}{3} - \mathbf{P}^*_{T_{k+1}}}{\left|\frac{\left(\mathbf{P}^*_{I_{k+1}} + \mathbf{P}^*_{M_{k+1}} + \mathbf{P}^*_{T_{k+1}}\right)}{3} - \mathbf{P}^*_{T_{k+1}}\right|} & \text{if} \quad i = \{T\} \end{cases} . \tag{3.15}$$

Regarding the orientation, it is assumed that $\mathbf{P}_{i_{k+1}}$ maintains the same orientation of $\mathbf{P}^*_{i_{k+1}}$ because they are chosen value of λ in such a way that the adjustment done is relatively small. Figure 3.10 represents an example of the virtual contact point strategy applied to Index and Thumb when $e_{i_k} \leq 0$ for both. It is worth to precise that the planes $\Pi_{(\perp \text{ to } \vec{\boldsymbol{r}})-Index}$ and $\Pi_{(\perp \text{ to } \vec{\boldsymbol{r}})-Thumb}$ are the planes where the rotation occurs, so perpendicular to $\vec{\boldsymbol{r}}$. They should be not confused with the working planes Π_I and Π_T mentioned in Section 2.1. Moreover, notice that Δd_T is not on the same direction of Δd_I.

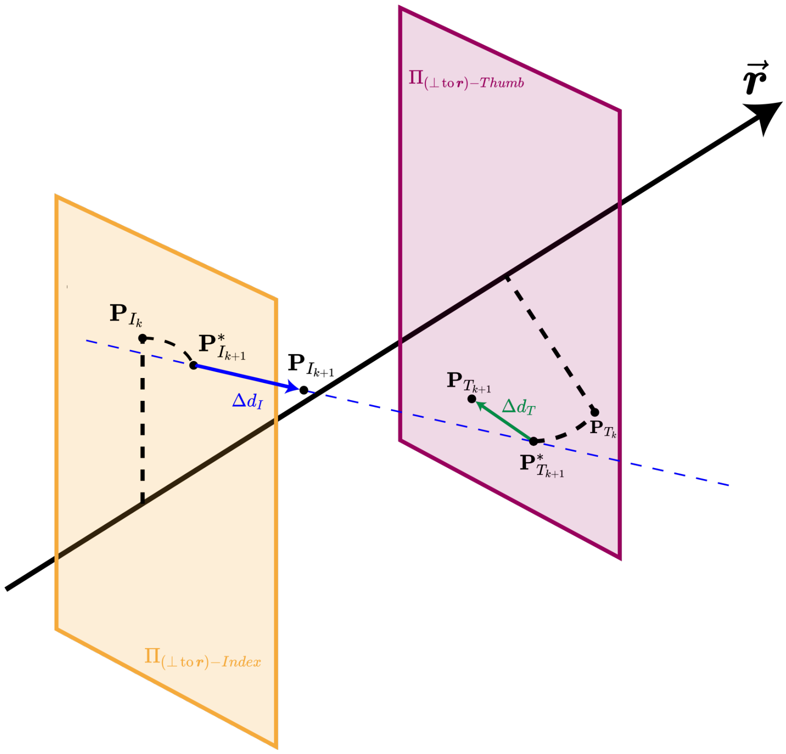


Figure 3.10: Virtual Contact Point and Force Compensation Error applied to actual strategy when $e_{I_k} \leq 0$ and $e_{T_k} \leq 0$.

Inverse Kinematics (IK)

Once the homogeneous transformation matrix of the hypothetical new contact point $\mathbf{P}_{i_{k+1}}$ with respect to the World RF has been obtained, it is necessary to compute Inverse Kinematics to pass from operational space to joint space and verify if is it possible or not to command the joints to reach the new position.

Montaño and Suaréz exploited the KDL to obtain an IK solution. As specified in Section 2.3.2, KDL resorts to numerical solution techniques; these clearly have the advantage of being applicable to any kinematic structure, but in general they do not allow computation of all admissible solutions. Consequently, if is it possible, is always preferred to compute IK based on algebraic and geometric intuition. For the actual manipulation strategy, KDL is adopted only for the Thumb, whereas for Index, Middle and Ring a geometrical and algebraic strategy is considered.

Index, Middle, and Ring have the same kinematic structure as already indicated in the description of the Allegro Hand (Section 2.1, Figure 2.4). First of all is computed the value of the joint located at the base of the finger referring $\mathbf{P}_{i_{k+1}}$ to the Base Finger Reference Frame (defined in the Section 3.3.2 - "Preliminary

Operations"). To accomplish this, it is computed the value of the homogeneous transformation matrix of point $\mathbf{P}_{i_{k+1}}$ w.r.t. Base Finger Reference Frame:

$$\mathbf{T}^{BFRF}_{P_{i_{k+1}}} = (\mathbf{T}^{\mathcal{W}}_{BFRF_i})^{-1} \cdot \mathbf{T}^{\mathcal{W}}_{P_{i_{k+1}}} \tag{3.16}$$

The structure of resulting the matrix is always the same:

$$\mathbf{T}^{BFRF}_{P_{i_{k+1}}} = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & (P_{i_{k+1,x}})_{BFRF} \\ R_{yx} & R_{yy} & R_{yz} & (P_{i_{k+1,y}})_{BFRF} \\ R_{zx} & R_{zy} & R_{zz} & (P_{i_{k+1,z}})_{BFRF} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The values $(P_{i_{k+1,x}})_{BFRF}, (P_{i_{k+1,y}})_{BFRF}, (P_{i_{k+1,z}})_{BFRF}$ identify the position vector of $\mathbf{P}_{i_{k+1}}$ in BFRF.

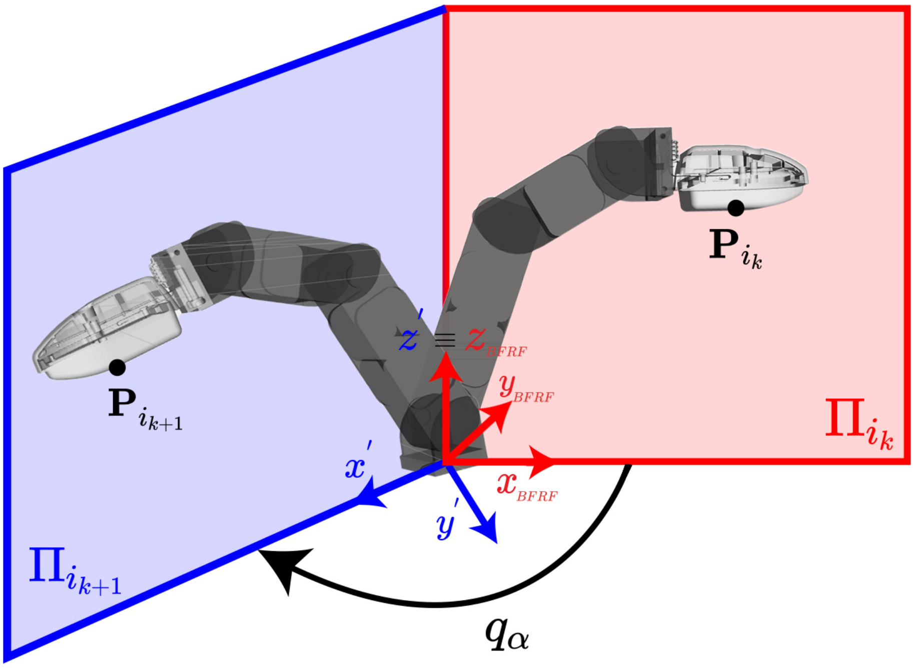


Figure 3.11: From $\mathbf{P}_{i_k}$ to $\mathbf{P}_{i_{k+1}}$ to compute the value of q_α .

Looking at Figure 3.11, it is quite simple to understand that projecting $\mathbf{P}_{i_{k+1}}$ in plane $x_{BFRF} - y_{BFRF}$ locates the angle between x' and x_{BFRF}, which corresponds exactly to the value of the rotation of the joint at the base of the finger:

$$q_\alpha = \arctan\left(\frac{(P_{i_{k+1,y}})_{BFRF}}{(P_{i_{k+1,x}})_{BFRF}}\right) \quad \text{with} \quad \alpha = \begin{cases} 1 & \text{if} \quad i = I \\ 5 & \text{if} \quad i = M \\ 9 & \text{if} \quad i = R \end{cases} \tag{3.17}$$

To find the values of the others subsequent three joints of one finger (q_2, q_3, q_4 for Index, q_6, q_7, q_8 for Middle, q_{10}, q_{11}, q_{12} for Ring) one resorts to the "Solution of Three-link Planar Arm" [19]. From now on, let's define:

$$q_\beta \quad \text{with} \quad \beta = \begin{cases} 2 & \text{if} \quad i = I \\ 6 & \text{if} \quad i = M \\ 10 & \text{if} \quad i = R \end{cases}$$

$$q_\gamma \quad \text{with} \quad \gamma = \begin{cases} 3 & \text{if} \quad i = I \\ 7 & \text{if} \quad i = M \\ 11 & \text{if} \quad i = R \end{cases}$$

$$q_\epsilon \quad \text{with} \quad \epsilon = \begin{cases} 4 & \text{if} \quad i = I \\ 8 & \text{if} \quad i = M \\ 12 & \text{if} \quad i = R \end{cases}$$

The Three-link planar arm solution is computed in a plane, as can be understood from its name. In the case under consideration, knowing the value of the joint at the base of the finger q_α is possible to work in the flexion/extension plane Π_i where are included the three points corresponding to the positions of the reference frames Σ_{ij} of the three phalanges of the finger. Therefore it is needed to refer $\mathbf{P}_{i_{k+1}}$ w.r.t. a system reference frame fixed in the joint that is consecutive at finger base joint (q_β) and rotated w.r.t. the Base Finger RF of an angle exactly equal to q_α. This reference frame is called "Planar Reference Frame (PRF)" (Figure 3.12).

$$\mathbf{T}^{PRF}_{P_{i_{k+1}}} = (\mathbf{T}^{BFRF}_{PRF})^{-1} \cdot \mathbf{T}^{BFRF}_{P_{i_{k+1}}} \tag{3.18}$$

- $\mathbf{T}^{BFRF}_{P_{i_{k+1}}}$ is obtained from eq. (3.16); .
- $\mathbf{T}^{BFRF}_{PRF} = \begin{pmatrix} cos(q_\alpha) & -sin(q_\alpha) & 0 & 0 \\ sin(q_\alpha) & cos(q_\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0{,}0164 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ that is the composition of:
  - the rotation matrix which takes into account the value of joint at the base of the finger

$$\mathbf{R}_{q_\alpha} = \begin{pmatrix} cos(q_\alpha) & -sin(q_\alpha) & 0 \\ sin(q_\alpha) & cos(q_\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

  - the position vector that indicates the position of q_β w.r.t. q_α

$$\overrightarrow{\mathbf{p}}_{q_\beta q_\alpha} = \begin{pmatrix} 0 \\ 0 \\ 0{,}0164 \end{pmatrix}.$$

Note that this value corresponds exactly to the length of the link between the first two consecutive joints of a finger, and it is equal for all the fingers (except for the Thumb).

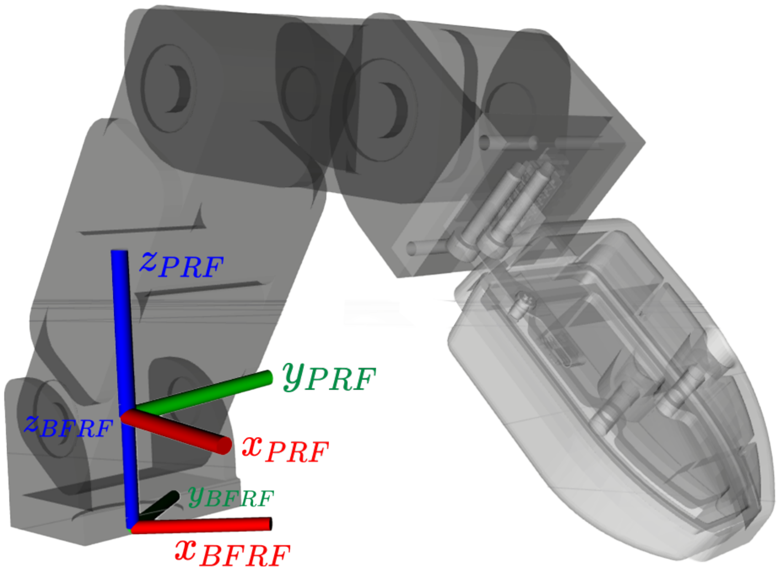


Figure 3.12: Transition from Base Finger RF to Planar RF

In mathematical terms, the position of point $\mathbf{P}_{i_{k+1}}$ w.r.t. planar RF is obtain with the relation: The resulting homogeneous transformation matrix $\mathbf{T}^{PRF}_{P_{i_{k+1}}}$ has the following structure confirming the fact that it has switched to working in the $x_{PRF} - z_{PRF}$ plane:

$$\mathbf{T}^{PRF}_{P_{i_{k+1}}} = \begin{bmatrix} (R_{xx}) & 0 & (R_{xz})_{PRF} & (P_{i_{k+1,x}})_{PRF} \\ 0 & 1 & 0 & 0 \\ (R_{zx})_{PRF} & 0 & (R_{zz})_{PRF} & (P_{i_{k+1,z}})_{PRF} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The resulting structure to consider now is showed in Figure 3.13. It is possible to refer to direct kinematics specifying position and orientation in terms of a minimal number of parameters: the two coordinates $(P_{i_{k+1,x}})_{PRF}$, $(P_{i_{k+1,z}})_{PRF}$ and

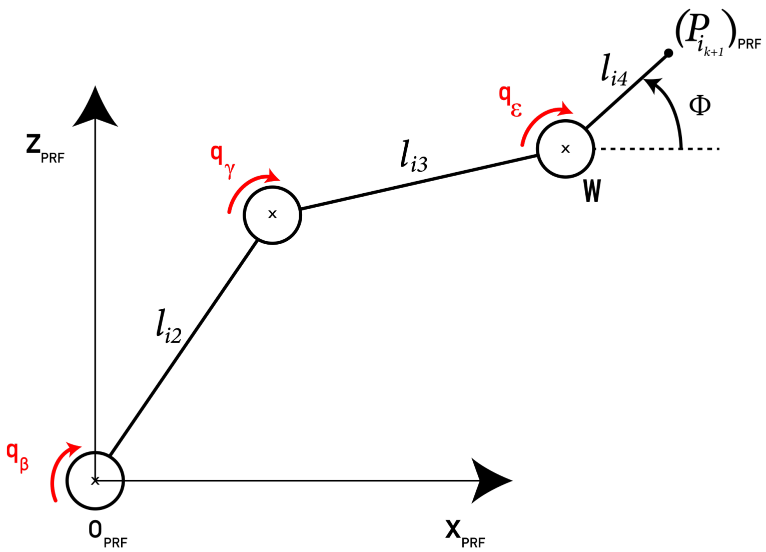


Figure 3.13: Inverse Kinematics in Planar Reference Frame.

the orientation angle ϕ that is between the position $(\mathbf{P}_{i_{k+1}})_{PRF}$ and axis x_{PRF}. The direct kinematic equation is:

$$(\mathbf{P}_{i_{k+1}})_{PRF} = \begin{bmatrix} (P_{i_{k+1,x}})_{PRF} \\ (P_{i_{k+1,z}})_{PRF} \\ \phi \end{bmatrix} \begin{bmatrix} l_{i2}c_{\beta} + l_{i3}c_{\beta\gamma} + l_{i4}c_{\beta\gamma\epsilon} \\ l_{i2}s_{\beta} + l_{i3}s_{\beta\gamma} + l_{i4}s_{\beta\gamma\epsilon} \\ q_{\beta} + q_{\gamma} + q_{\epsilon} \end{bmatrix} \tag{3.19}$$

$$\begin{aligned} c_{\beta} &= \cos(q_{\beta}) \\ c_{\beta\gamma} &= \cos(q_{\beta} + q_{\gamma}) \\ c_{\beta\gamma\epsilon} &= \cos(q_{\beta} + q_{\gamma} + q_{\epsilon}) \\ s_{\beta} &= \sin(q_{\beta}) \\ s_{\beta\gamma} &= \sin(q_{\beta} + q_{\gamma}) \\ s_{\beta\gamma\epsilon} &= \sin(q_{\beta} + q_{\gamma} + q_{\epsilon}) \end{aligned}$$

It is essential to underline that the value of link l_{i4} is computed in function of the contact point $\mathbf{P}_{i_{k+1}}$ accordingly to Section 2.2. Consequently, the module of l_{i4} changes at each iteration k.

Knowing the orientation angle:

$$\phi = \arctan\left(\frac{(R_{xz})_{PRF}}{(R_{zz})_{PRF}}\right) \tag{3.20}$$

and defining W the position of the joint q_ϵ in PRF is it possible to write:

$$P_{Wx} = (P_{i_{k+1,x}})_{PRF} - l_{i4}c_\phi = l_{i2}c_\beta + l_{i3}c_{\beta\gamma} \tag{3.21}$$

$$P_{Wz} = (P_{i_{k+1,z}})_{PRF} - l_{i4}s_\phi = l_{i2}s_\beta + l_{i3}s_{\beta\gamma} \tag{3.22}$$

The position of W depends only on the first two angles q_β and q_γ. Squaring and summing (3.21), (3.22) yields

$$P_{Wx}^2 + P_{Wz}^2 = l_{i2}^2 + l_{i3}^2 + 2l_{i2}l_{i3}c_\gamma$$

from which

$$c_\gamma = \frac{P_{Wx}^2 + P_{Wz}^2 - l_{i2}^2 - l_{i3}^2}{2l_{i2}l_{i3}}.$$

A solutions exits only if -1 $\leq c_\gamma \leq$ 1, otherwise the given point would be outside the arm reachable workspace. Then,

$$s_\gamma = \pm\sqrt{1 - c_\gamma^2},$$

where the positive sign is relative to the elbow-down posture and the negative sign to the elbow-up posture. Of course in the current case it is taken in consideration only the elbow-up posture. The angle q_γ is computed as:

$$q_\gamma = \mathrm{Atan2}(s_\gamma, c_\gamma).$$

Once the value of q_γ is obtained, the angle q_β can be found substituting q_γ into eq. (3.21) and in eq. (3.22). In such a way is achieved an algebraic system of two equations in the two unknowns s_β and c_β, whose solution is

$$s_\beta = \frac{(l_{i2} + l_{i3}c_\gamma)P_{Wz} - l_{i3}s_\gamma P_{Wx}}{P_{Wx}^2 + P_{Wz}^2}$$

$$c_\beta = \frac{(l_{i2} + l_{i3}c_\gamma)P_{Wx} - l_{i3}s_\gamma P_{Wz}}{P_{Wx}^2 + P_{Wz}^2}.$$

It is easy to find the value:

$$q_\beta = \mathrm{Atan2}(s_\beta, c_\beta).$$

Finally, the angle q_ϵ is found from (3.20) as

$$q_\epsilon = \phi - q_\beta - q_\gamma.$$

Notice that in the examined algebraic solution is not considered the particular case of *kinematic singularity*, when $s_\gamma = 0$ and consequently $q_\gamma = 0$. At this point the new possible configuration of the joints of Index, Middle and Ring are known (respectively $\mathbf{q}_I, \mathbf{q}_M, \mathbf{q}_R$).

Thumb has a particular kinematic structure (Figure 2.5). For this reason, instead of computing the Inverse Kinematics resorting to algebraic and geometrical considerations, it is preferred to use KDL. The kinematic chain created for KDL starts from the World R.F and ends in the possible new contact point $\mathbf{P}_{T_{k+1}}$. The IK iterative method selected is the "Damped Least Squared Method", also called Levenberg - Marquardt method (Appendix B). Finally, if a solution exists, it is computed the joint configuration of the Thumb $\mathbf{q}_T$.

Move the Fingers

In the computation of IK for all the fingers there is a constraint that takes into account if the point would be outside the arm reachable workspace:

- if a solution exists for all the fingers, the movement will be triggered. All joints will assume the values of the new configuration and the procedure will be repeated for the next step. Moreover, if the actual iteration is the last one, or better, when the requested rotation indicated by the user is wholly exploited, there will be a message of completion of the operation and the way to proceed is postponed to the user redirecting to the main menu;

- if a solution does not exist for at least one finger, all the joints remain in the old configuration, and the way to proceed is postponed to the user redirecting, as the case before, to the main menu.

The movement is done by means of the service «setDesiredPosition» sending the value of the joints in the new configuration $\mathbf{Q} = \{\mathbf{q}_I, \mathbf{q}_M, \mathbf{q}_R, \mathbf{q}_T\}$.

Chapter 4

Applications and Experimental Results

4.1 Simulation Environment

Before performing the experiments with the real device, it is necessary to verify the correctness of the implemented algorithm. As explained in Section 2.3.2, rviz is a mere visualization tool so, to have a simulation environment, the fake node is exploited. To model the object to be manipulated in rviz is not possible since no CAD model and URDF file are provided neither created. Moreover, to be in line with the purpose of the thesis, the object must be unknown. Therefore, there is no contact between object and the fingertips and the following considerations are required:

- the sensors are switched off;
- a completely random position, such as the one in Figure 4.1, is assumed as the initial position;
- the contact points between the fingertips and the object are considered coincident with the origin of the Tip Reference Frame $\Sigma_{i_{TIP}}$. Of course, also the orientation of the contacts points is the same of the Tip Reference Frame;
- no forces come into play, so "Virtual Contact Point and Force Compensation Error" strategy (Section 3.3.2) is not applied.

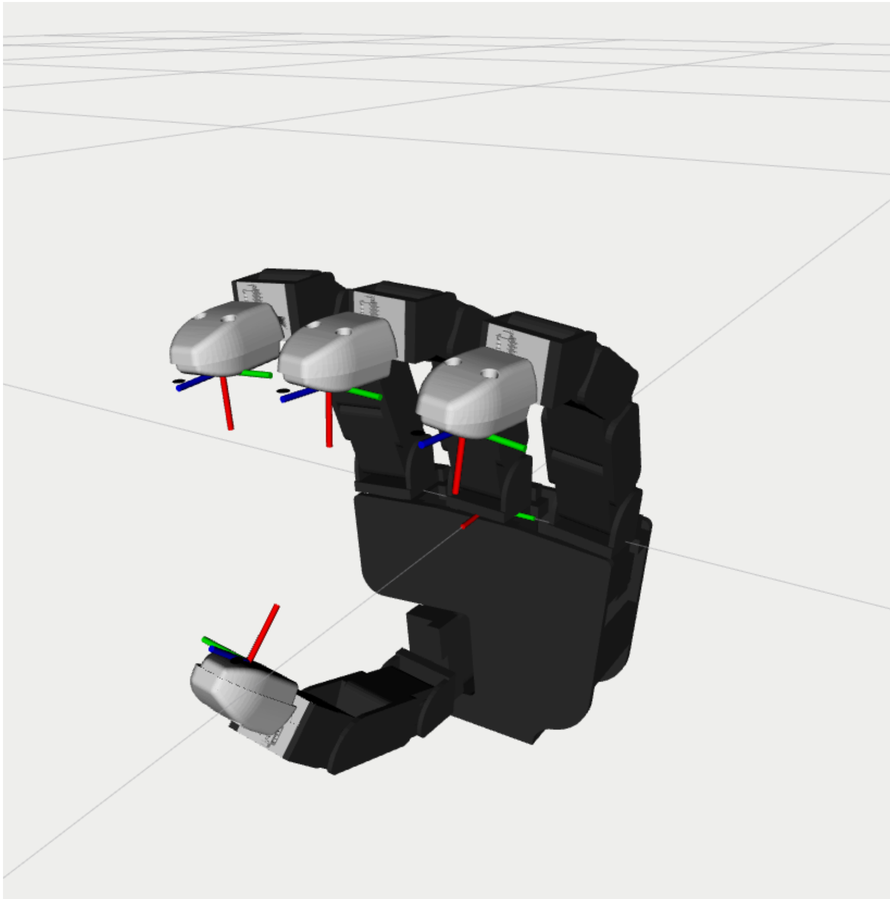


Figure 4.1: Example of a starting position position in simulation environment.

From all of the above, it can be stated that the simulation allows verifying the correctness of the rotation of the points around the user-defined rotation axis and the correctness of the Inverse Kinematics. A demonstration video of the effectiveness of the simulation is attached at the following link:

`https://youtu.be/luvyRGqCJuQ`.

4.2 Real Environment

Once it has been demonstrated that the strategy works in the simulation environment, the practical application is performed. The assumptions of the previous Section 4.1 are not valid anymore and the manipulation strategy is applied to the real device. The following subsections show the experiments classified in function of the manipulated object. For each experiment are provided the starting values and a table with the initial values of the forces, i.e. the values detected by the sensors when the object is correctly grasped. To avoid rendering the discussion excessively copious, the data representing the values of the current forces are not reported in tables, but they are graphed and compared with the initial force for each finger. Each test is performed with a forward phase chosen by the user between case 1,2,3, followed by a return phase operated with case 5, i.e. without resorting to the use of Inverse Kinematics for the returning phase, namely case 4 (see Section 3.3.2 - Menu). Moreover, the angle and the step chosen are the same for all the tests. It will be observed that the values of the forces are very close to zero, therefore a measurement will be considered non-zero only if its degree of precision is greater than the third decimal digit (i.e. only if it is ≥ 0.001). Finally, for all the test are provided the corresponding demonstration video and considerations.

The initial phase of testing has highlighted the limitations of handling. In most cases, there is no Inverse Kinematics solution because the contact between the fingertip and object is lost immediately in the first step of the rotation. The points to be calculated require extremely high precision and the PID controller is particularly unstable when the fingers come into contact with the object (as can be seen from attached videos in the following subsections). Since the final goal is the movement of the object, it is decided to add a further condition to the program: if the contact is assured for the Thumb and for at least one finger between Index, Middle and Ring, the movement takes place anyway. In this situation, the fingers that do not detect the contact assume as contact point the point that is at the center of the fingertip.

4.2.1 Manipulated Object: Plastic Tea Bottle

The manipulated object is a plastic tea bottle. Tests 1,2,3 are all conducted with the same initial parameters for testing repeatability. Test 4 is performed increasing λ.

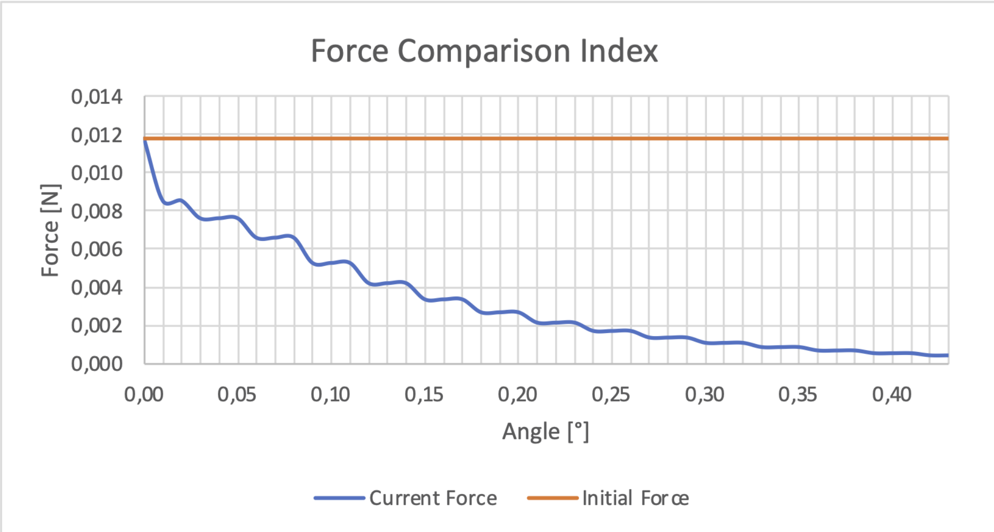
Test 1

- Axis of rotation: parallel to the y-axis of the World Reference Frame.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}005 \ \frac{m}{N}$.
- Video link: `https://youtu.be/yj285zeFx7Av`.

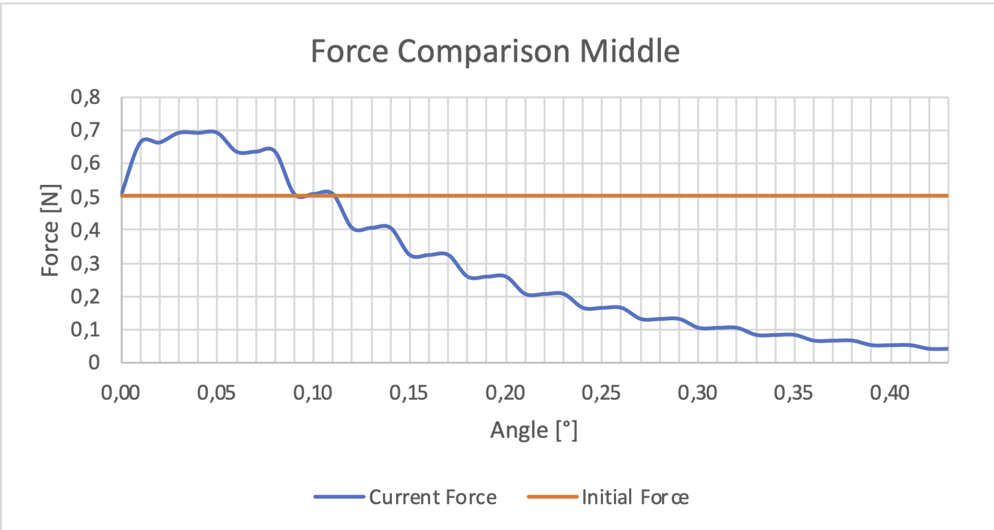
Initial Forces	
Index	0,0118 N
Middle	0,5038 N
Ring	0,0442 N
Thumb	0,0081 N

Table 4.1: Initial forces - Test 1

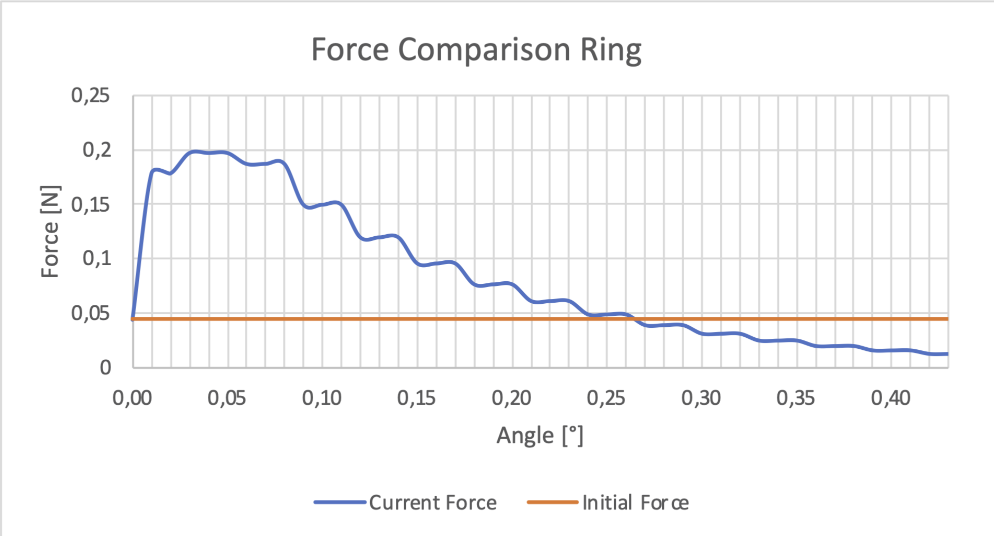
For the Middle, Ring and Thumb, the force compensation error algorithm works until a different angle for each finger; for the Index the algorithm turns out to be a failure. The current forces tend to zero as the rotation step increases for all the fingers. As can be noticed in the video, there is no contact between the object and fingertips at the end of the forward phase. The object is held in place only because it is in contact with a non-sensitive part of the fingertips. It can be argued that the rotation is performed fortuitously only due to the structure of the hand. The force error seem The angle of rotation achieved is 24.63°. The return phase is successful.



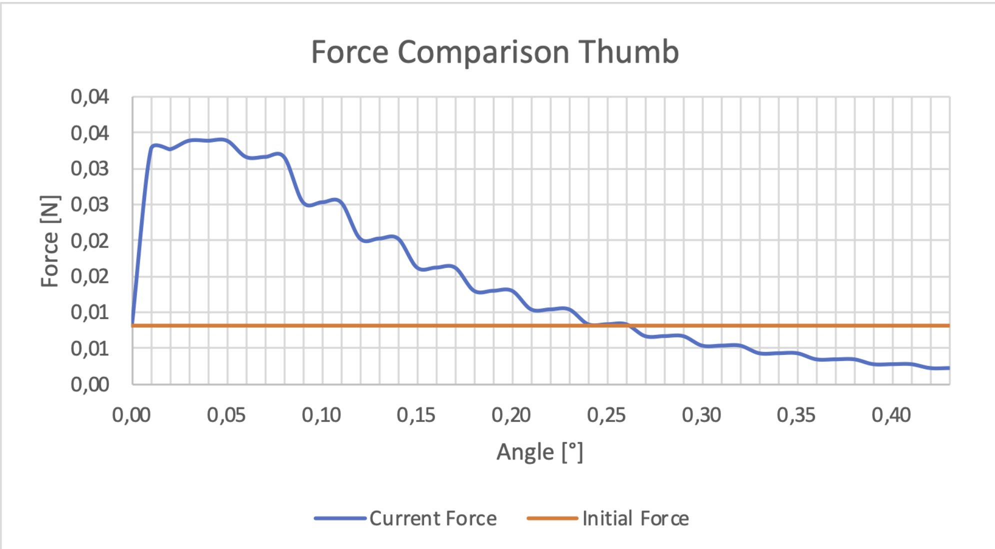
(a) Test 1 - Index force comparison.



(b) Test 1 - Middle force comparison.



(c) Test 1 - Ring force comparison.



(d) Test 1 - Thumb force comparison.

Figure 4.2: Force comparison - Test 1.

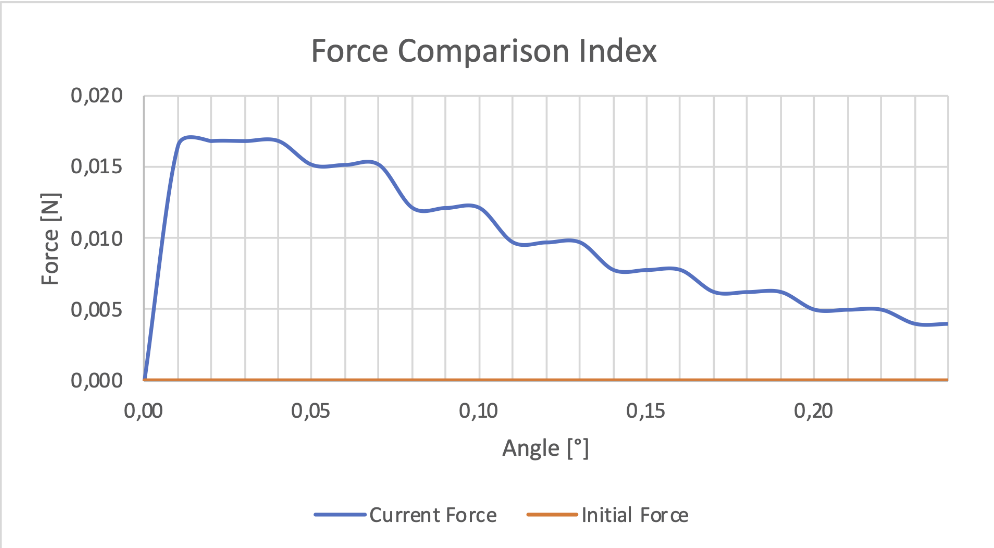
Test 2

- Axis of rotation: parallel to the y-axis of the World Reference Frame.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}005\ \frac{m}{N}$
- Video link: `https://youtu.be/nDV7HcauAUc`.

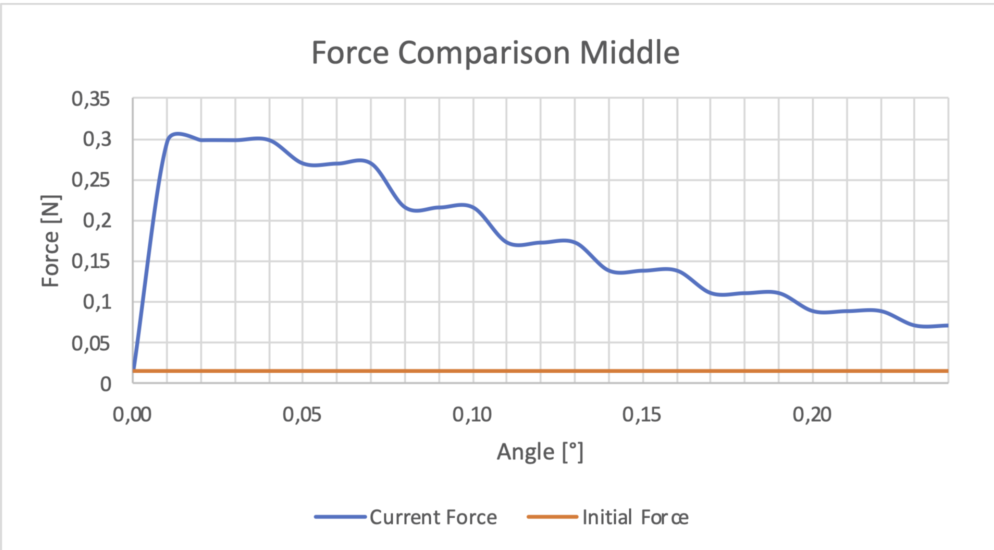
Initial Forces	
Index	$0\ N$
Middle	$0{,}0148\ N$
Ring	$0\ N$
Thumb	$0\ N$

Table 4.2: Initial forces - Test 2

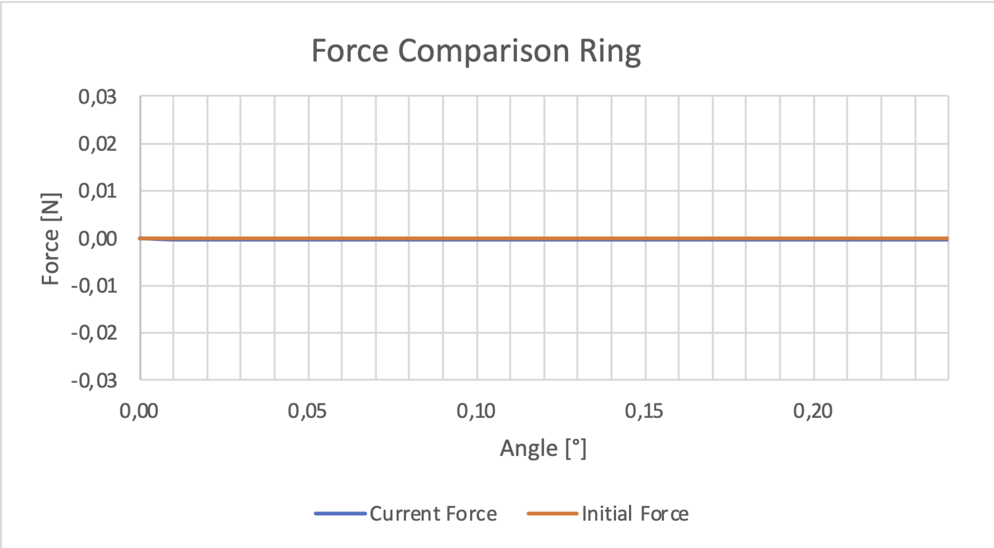
The initial forces are zero for the Index, Ring, and Thumb. There is undoubtedly a measurement error by the sensors since the rotation is successful, as seen in the video. In this case, having an initial measurement error is not a negative aspect. On the contrary, since the rotation is satisfactory, it leads to the conclusion that the points obtained from Inverse Kinematics are accurate, and they do not need force compensation. This consideration can be made even for the Middle the initial force is approximated zero. The angle of rotation achieved is 13.75°. The return phase is successful.



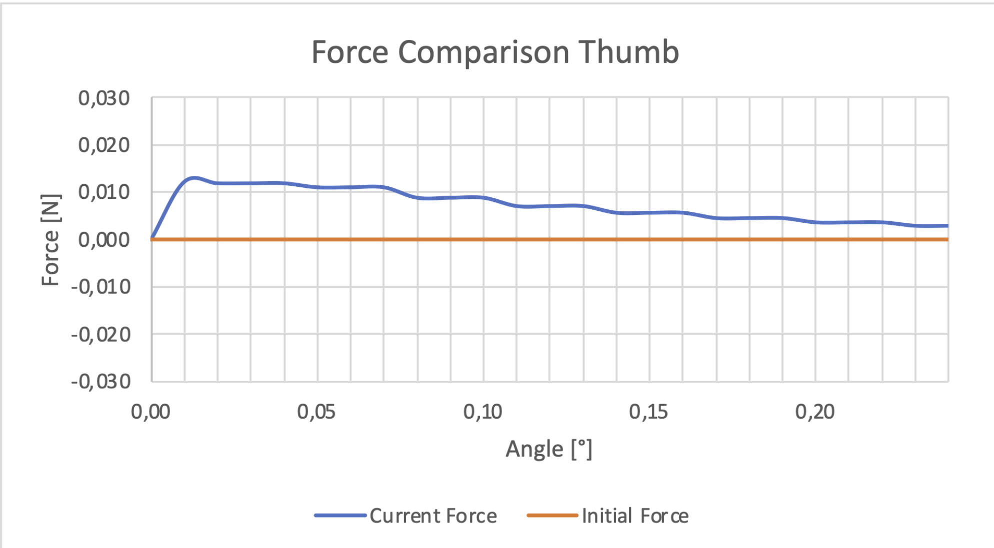
(a) Test 2 - Index force comparison.



(b) Test 2 - Middle force comparison.



(c) Test 2 - Ring force comparison.



(d) Test 2 - Thumb force comparison.

Figure 4.3: Force comparison - Test 2.

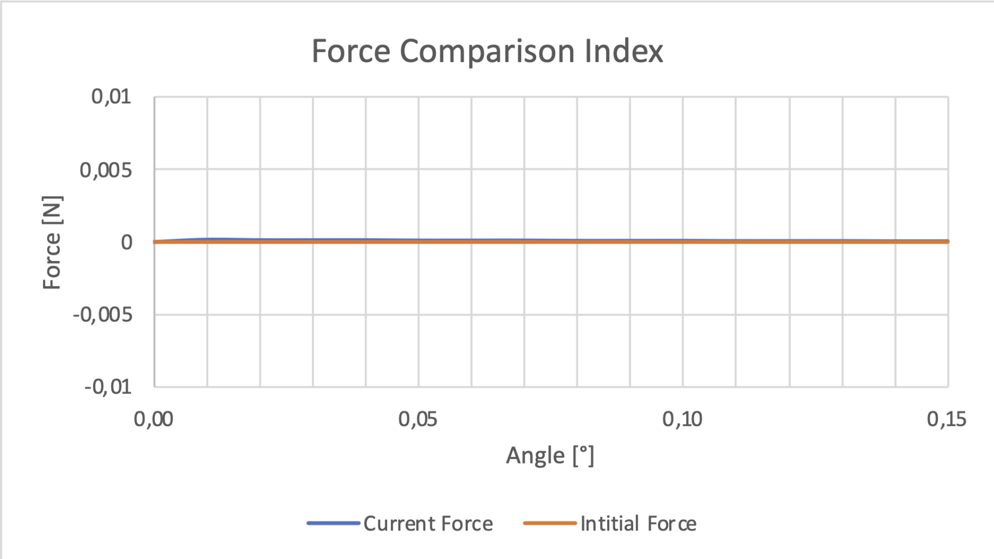
Test 3

- Axis of rotation: parallel to the y-axis of the World Reference Frame.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}005\ \frac{m}{N}$
- Video link: `https://youtu.be/IpMBoY-so0Q`.

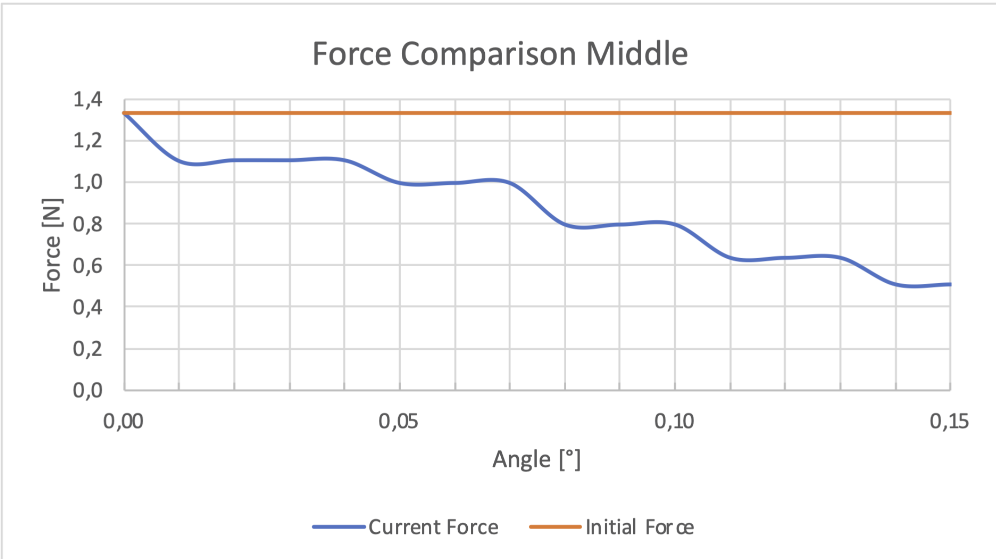
Initial Forces	
Index	0 N
Middle	1,332 N
Ring	0,2315 N
Thumb	0,00515 N

Table 4.3: Initial forces - Test 3

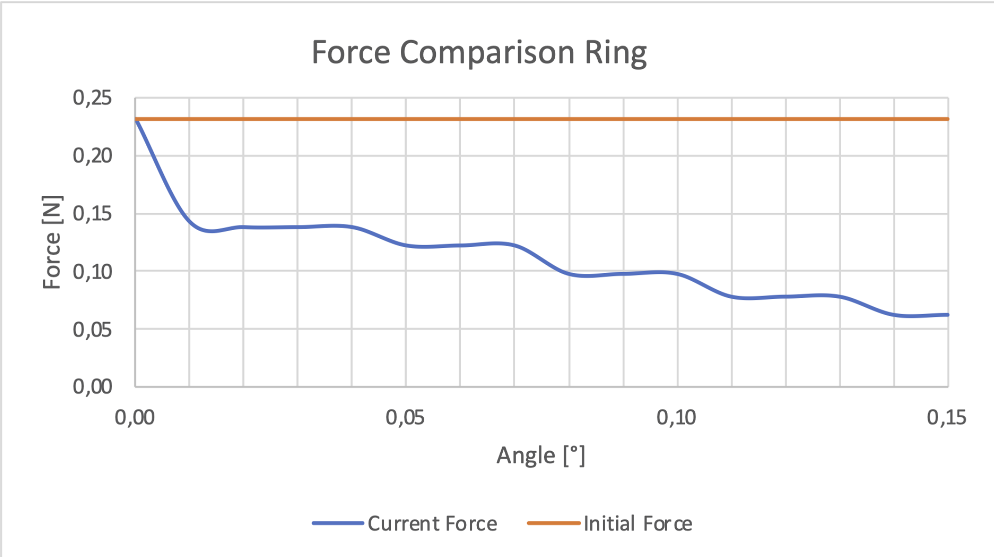
For the Index, the initial and the current forces are zero, so it does not contribute to the manipulation. Even if the goal is achieved, the force error tends to increase for the Middle, Ring, and Thumb at each step. The angle of rotation achieved is 8.59°. The return phase is successful.



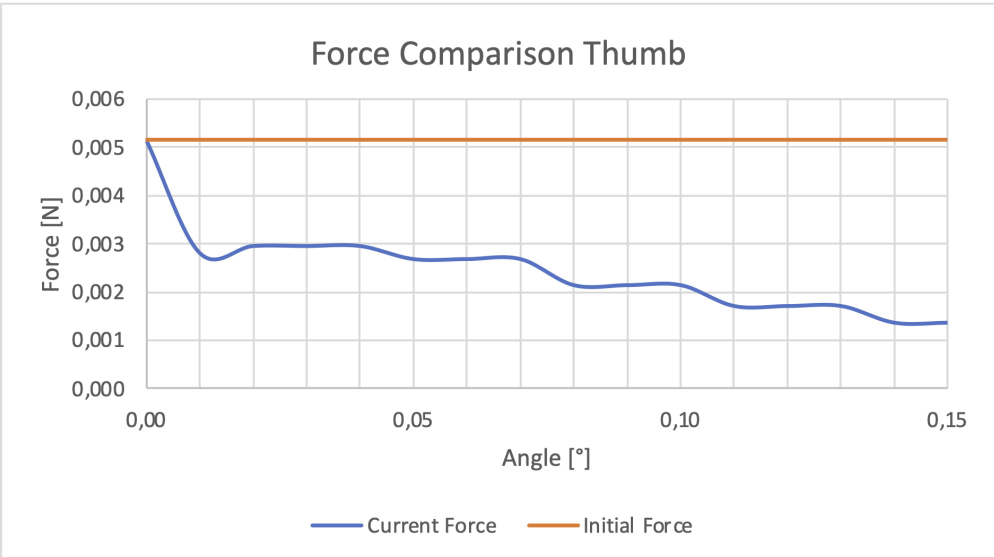
(a) Test 3 - Index force comparison.



(b) Test 3 - Middle force comparison.



(c) Test 3 - Ring force comparison.



(d) Test 3 - Thumb force comparison.

Figure 4.4: Force comparison - Test 3.

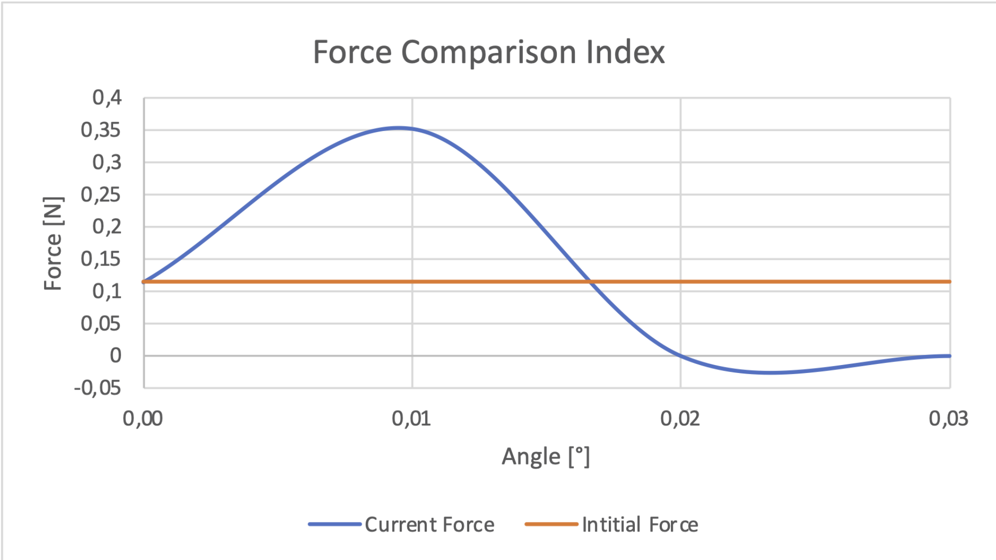
Test 4

- Axis of rotation: parallel to the y-axis of the World Reference Frame.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}05\ \frac{m}{N}$
- Video link: `https://youtu.be/IRgg5TUfx6g`.

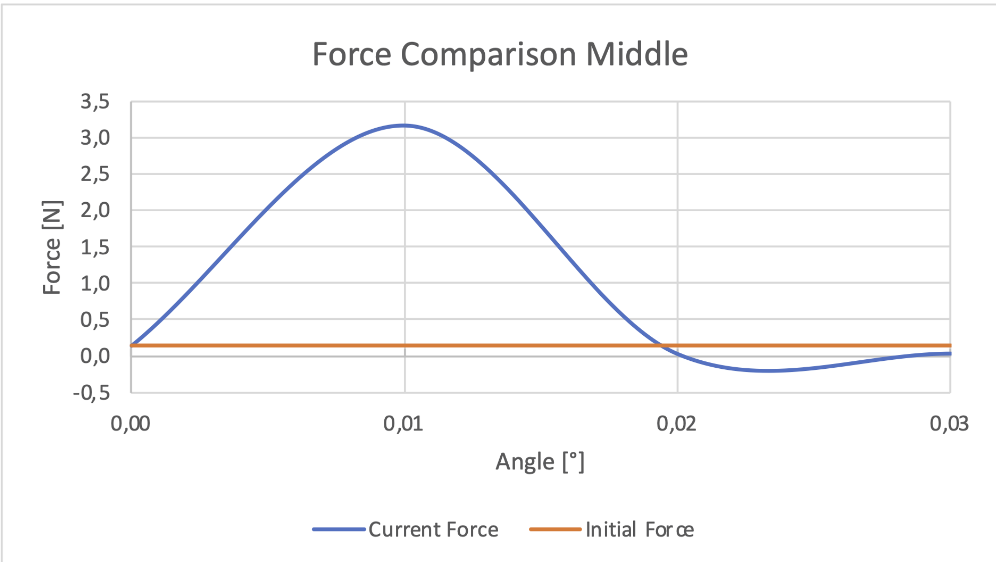
Initial Forces	
Index	0,1142 N
Middle	0,1395 N
Ring	2,7708 N
Thumb	0,3100 N

Table 4.4: Initial forces - Test 4

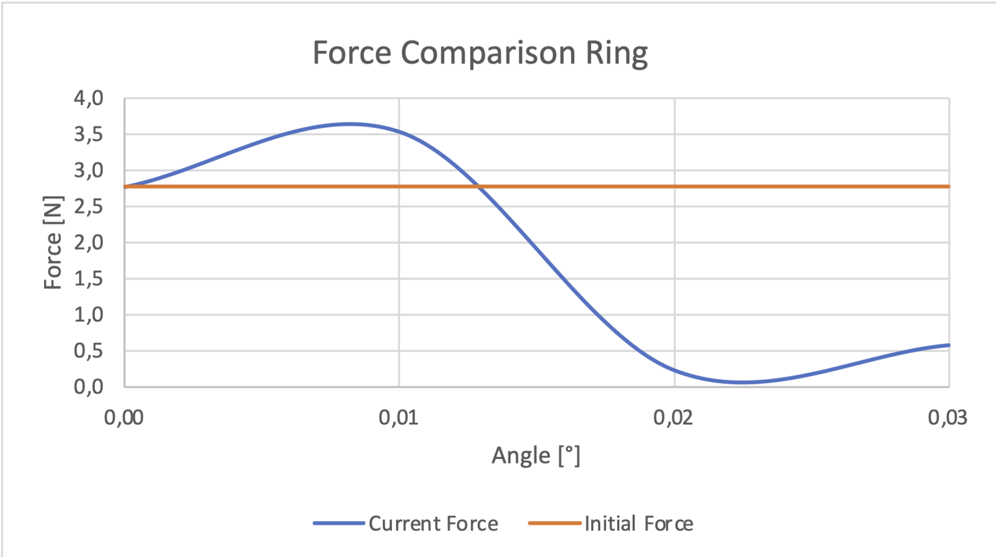
Test 4 is performed in the same conditions of Test 1,2,3, but the object falls by increasing λ by one order of magnitude.



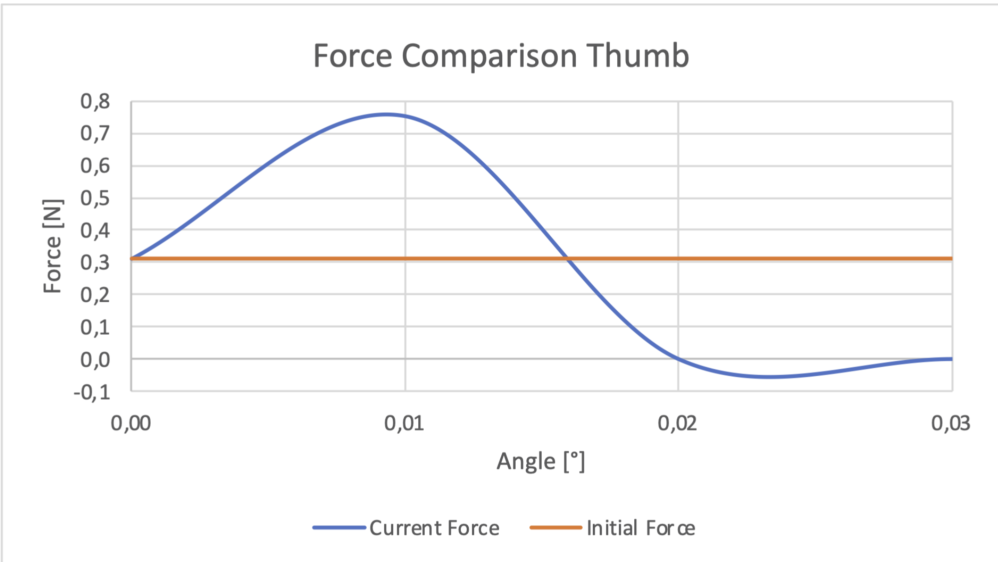
(a) Test 4 - Index force comparison.



(b) Test 4 - Middle force comparison.



(c) Test 4 - Ring force comparison.



(d) Test 4 - Thumb force comparison.

Figure 4.5: Force comparison - Test 4.

4.2.2 Manipulated Object: Aluminum Can

The manipulated object is an aluminum can. Tests 5 and 6 are performed using the same axis of rotation, but with different constants λ. For Tests 7 and 8, the same conditions are considered between them, but the axis of rotation and λ are different from Tests 5 and 6.

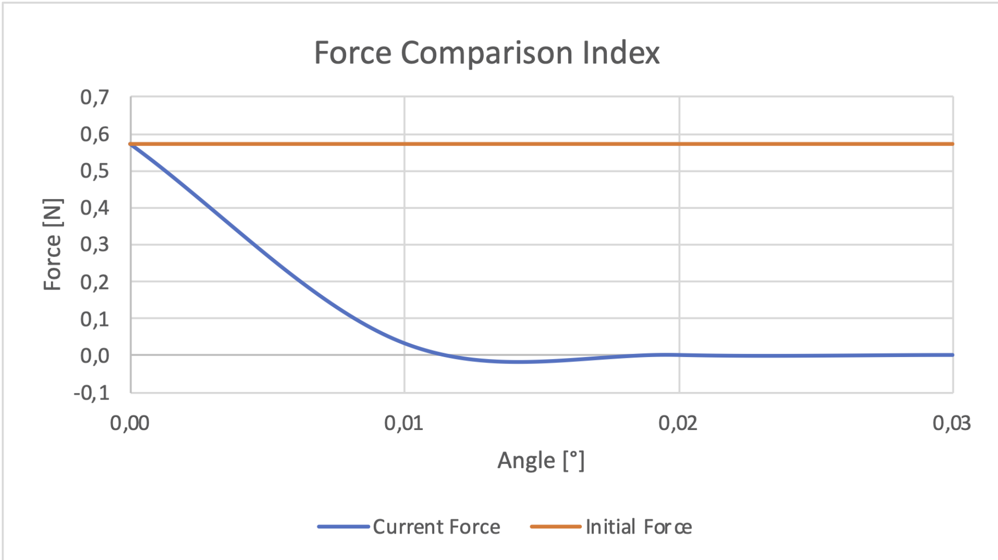
Test 5

- Axis of rotation: passing through the center of the tetrahedron and parallel to the y axis of the base link.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}01\ \frac{m}{N}$
- Video link: `https://youtu.be/45QJpZagVdM`.

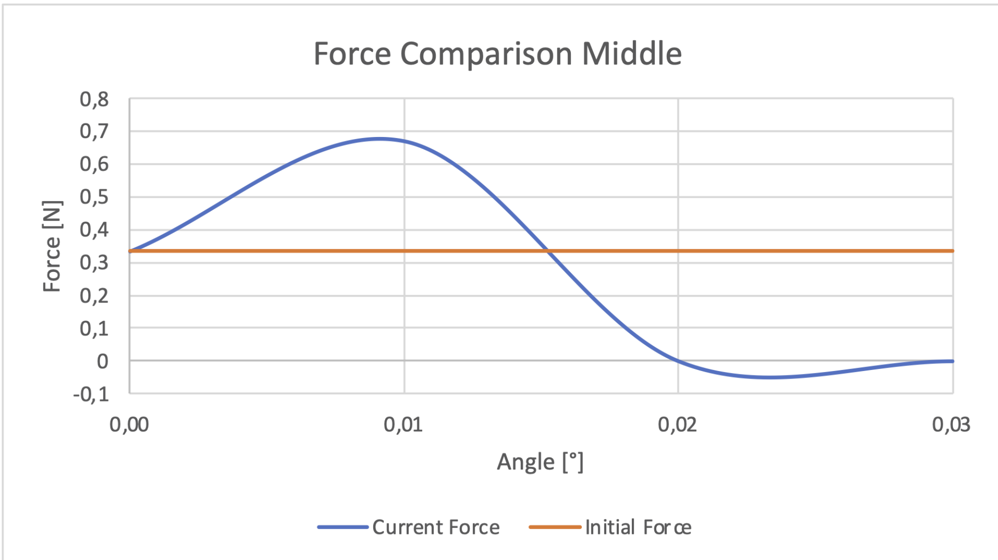
Initial Forces	
Index	0,571 N
Middle	0,334 N
Ring	1,792 N
Thumb	0,236 N

Table 4.5: Initial forces - Test 5

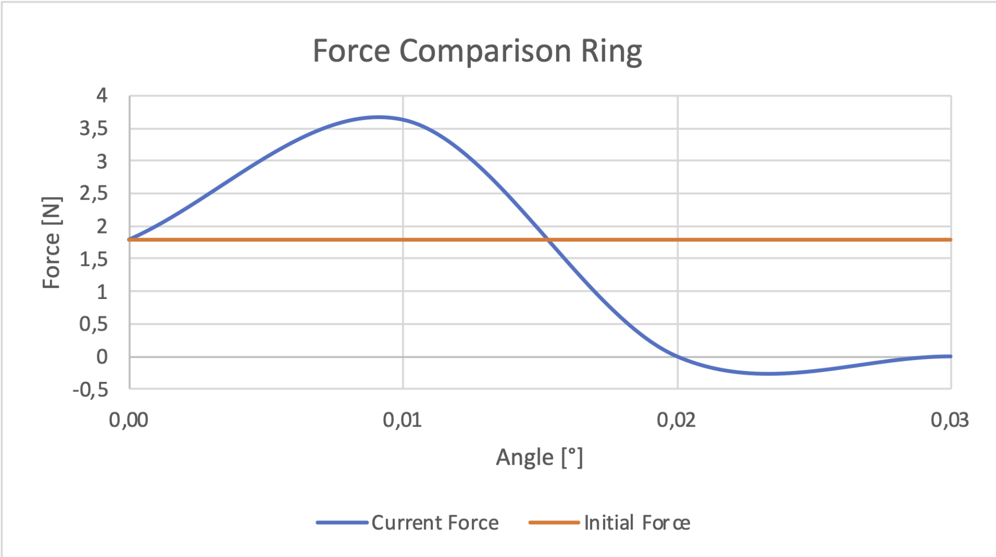
The angle of rotation achieved is 1.72°, so the process performed is minimal. However, graphs and data emphasize how inefficient the strategy is for the Index and Thumb. Instead, for Ring and Middle, the force error tends to zero in a first phase until 0.85°. The returning phase is successful.



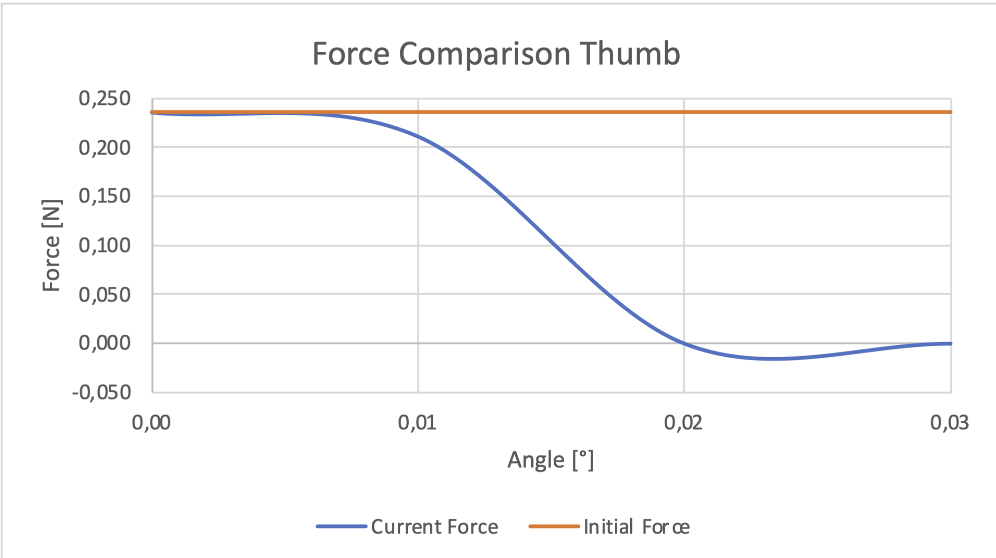
(a) Test 5 - Index force comparison.



(b) Test 5 - Middle force comparison.



(c) Test 5 - Ring force comparison.



(d) Test 5 - Thumb force comparison.

Figure 4.6: Force comparison - Test 5.

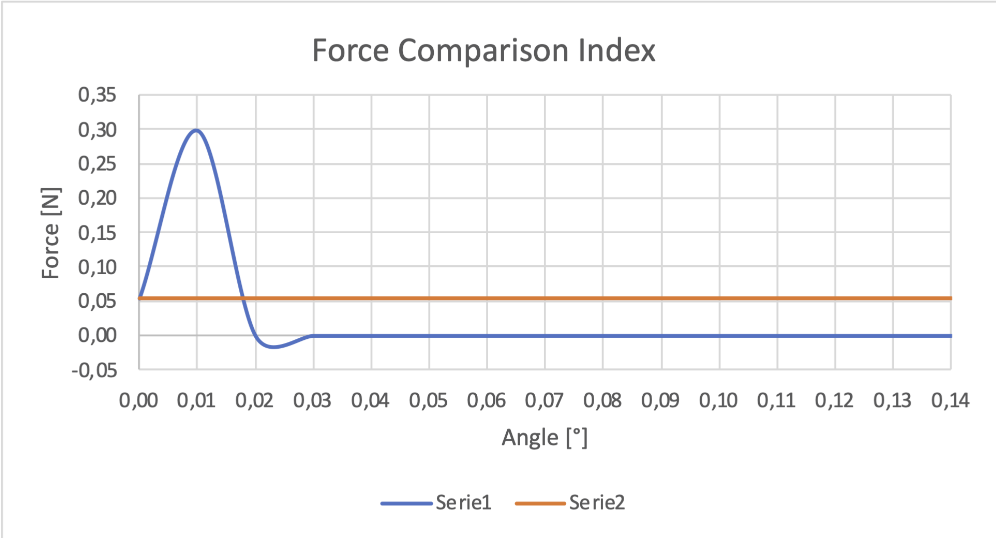
Test 6

- Axis of rotation: passing through the center of the tetrahedron and parallel to the y axis of the base link.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}15\ \frac{m}{N}$
- Video link: `https://youtu.be/aB7Y44MlPyI`.

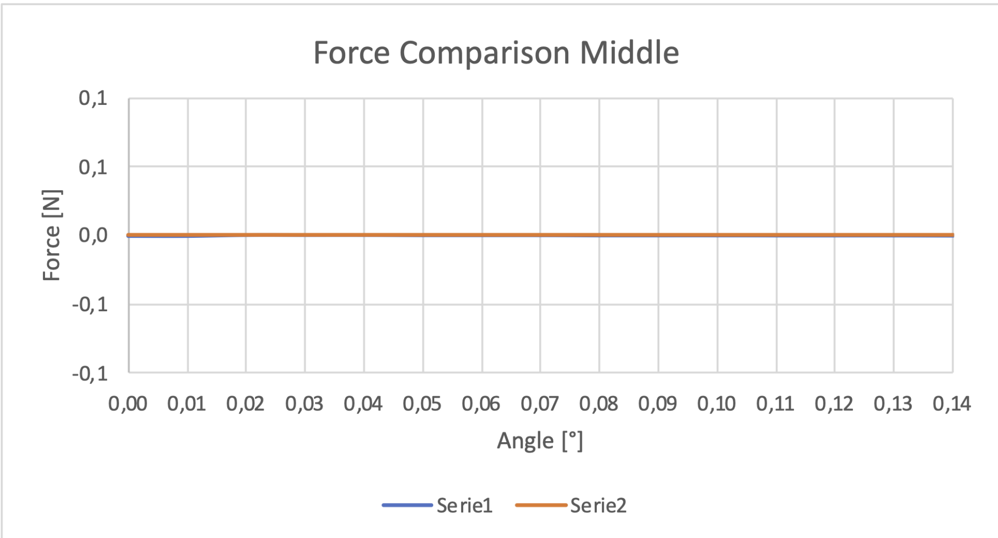
Initial Forces	
Index	0,05406 N
Middle	0 N
Ring	0 N
Thumb	0,0753 N

Table 4.6: Initial forces - Test 6

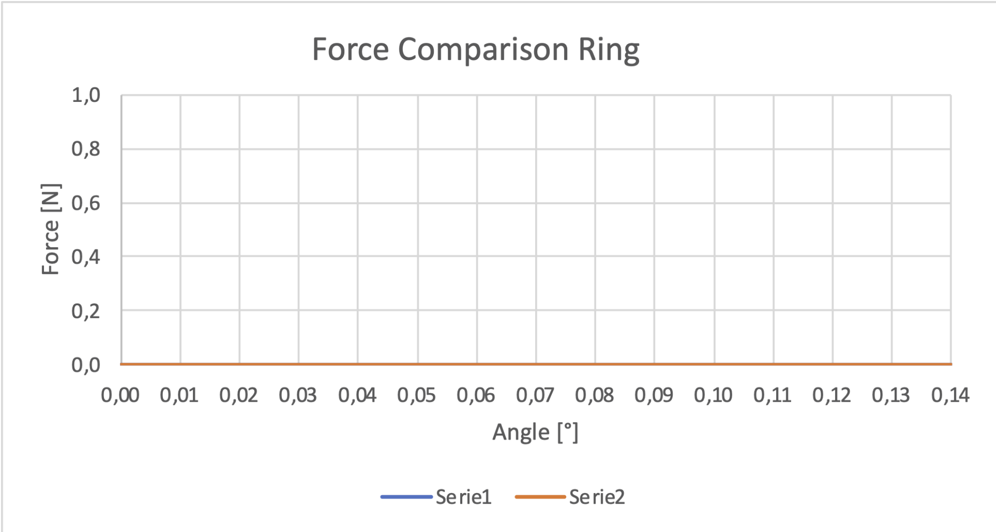
Increasing λ w.r.t the previous test (Test 5) leads to wrong results, and the object falls.



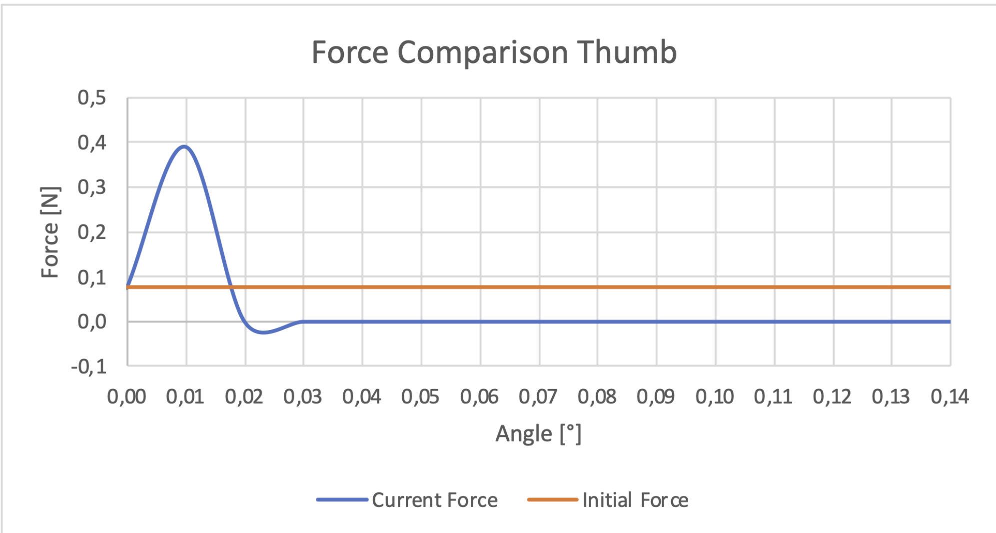
(a) Test 6 - Index force comparison.



(b) Test 6 - Middle force comparison.



(c) Test 6 - Ring force comparison.



(d) Test 6 - Thumb force comparison.

Figure 4.7: Force comparison - Test 6.

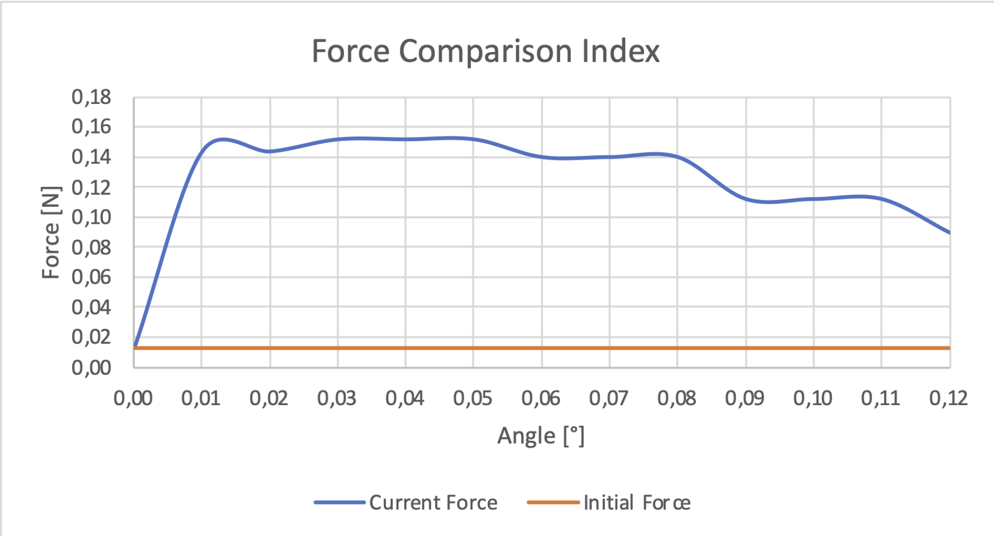
Test 7

- Axis of rotation: Axis of rotation: passing through the center of the tetrahedron and parallel to the x axis of the base link.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}001\ \frac{m}{N}$
- Video link: `https://youtu.be/GFFCF8uwtbg`.

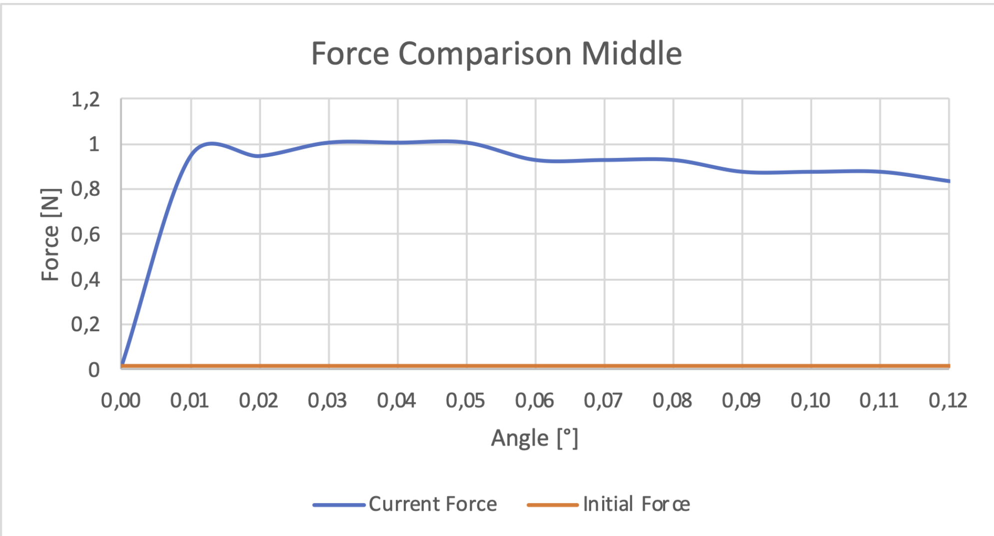
Initial Forces	
Index	0,0125 N
Middle	0,0178 N
Ring	0,004 N
Thumb	0,1701 N

Table 4.7: Initial forces - Test 7

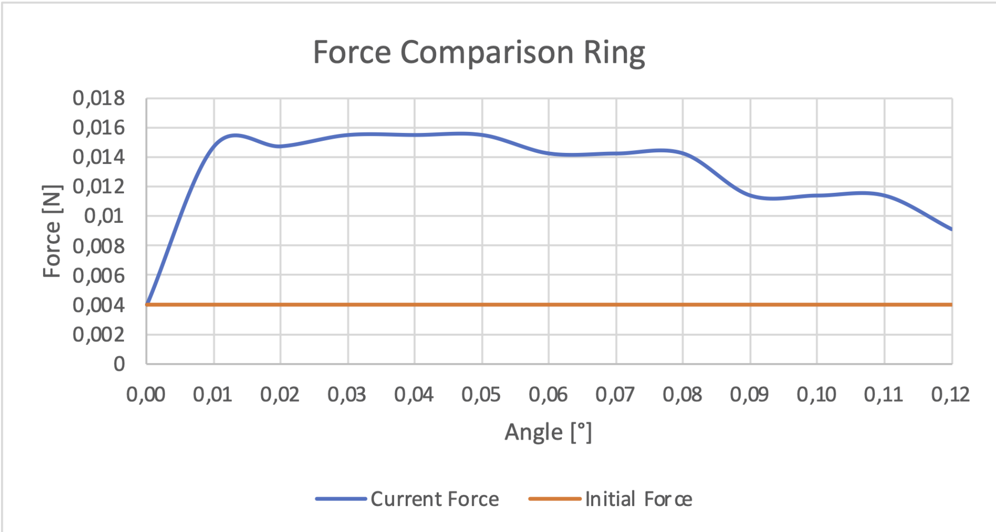
The force errors tend to decrease for all fingers. Even in this case, as in Test 2, the rotation is satisfactory, and the initial forces are approximately zero. Therefore, the points obtained from Inverse Kinematics are accurate; they do not need force compensation. The rotation performed is minimal, i.e. 6.88°. The returning phase is successful.



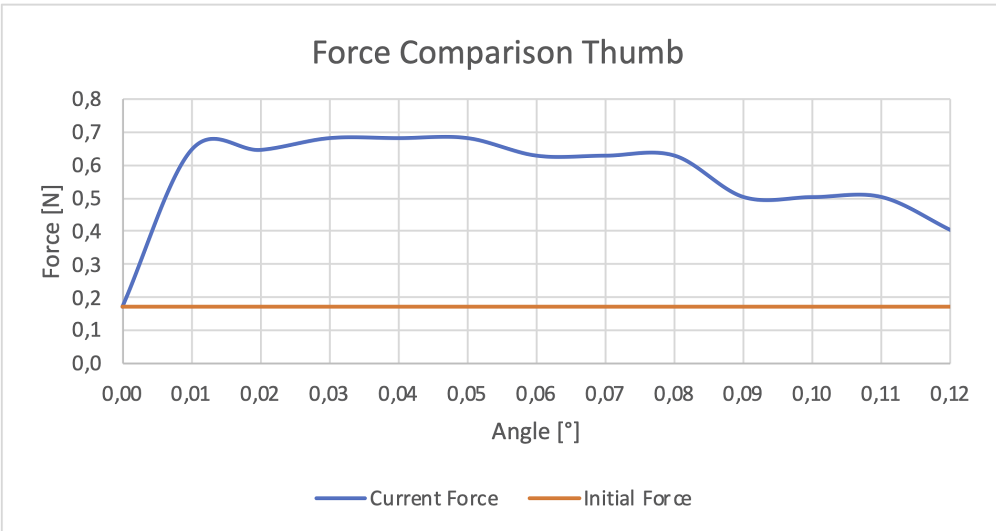
(a) Test 7 - Index force comparison.



(b) Test 7 - Middle force comparison.



(c) Test 7 - Ring force comparison.



(d) Test 7 - Thumb force comparison.

Figure 4.8: Force comparison - Test 7.

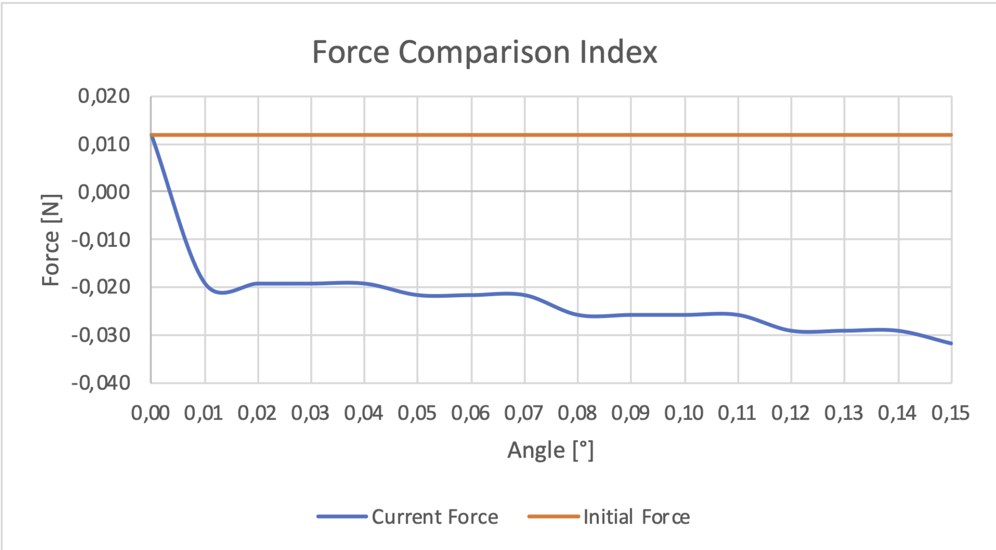
Test 8

- Axis of rotation: Axis of rotation: passing through the center of the tetrahedron and parallel to the x axis of the base link.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}001\ \frac{m}{N}$
- Video link: `https://youtu.be/x0DyDduYIHg`.

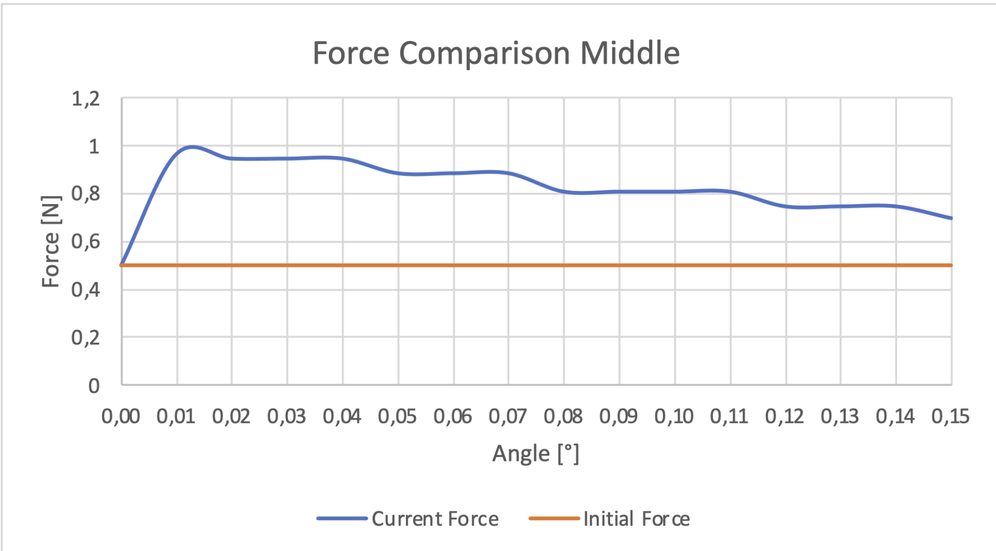
Initial Forces	
Index	0,0118 N
Middle	0,5038 N
Ring	0,0442 N
Thumb	0,0081 N

Table 4.8: Initial forces - Test 8

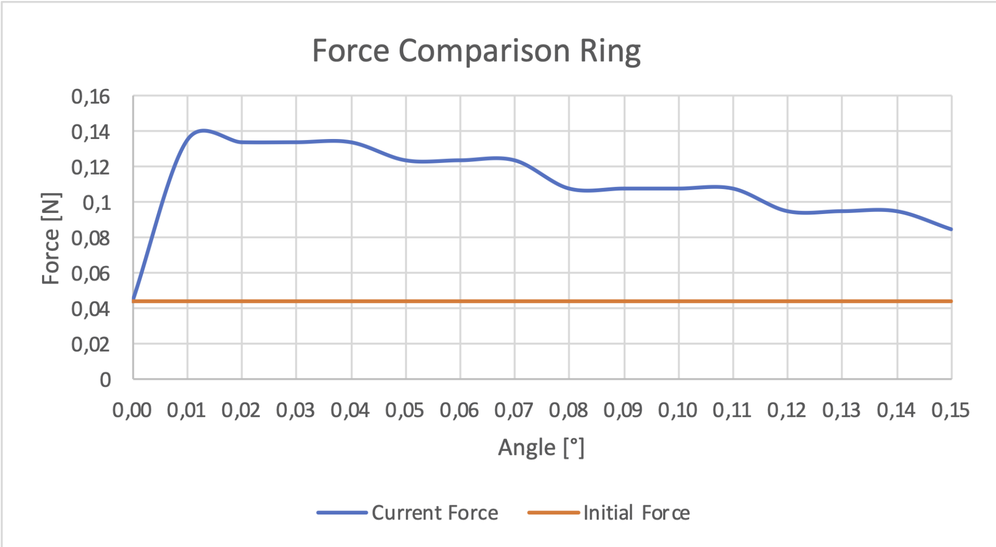
Test 8 is performed under the same conditions as Test 7, and from the video, it is possible to affirm that the rotation is broader and clearer in this case. The angle of rotation reached is 8.59°. Unfortunately, there is a phenomenon of instability in the return phase, and the hand loses its grip.



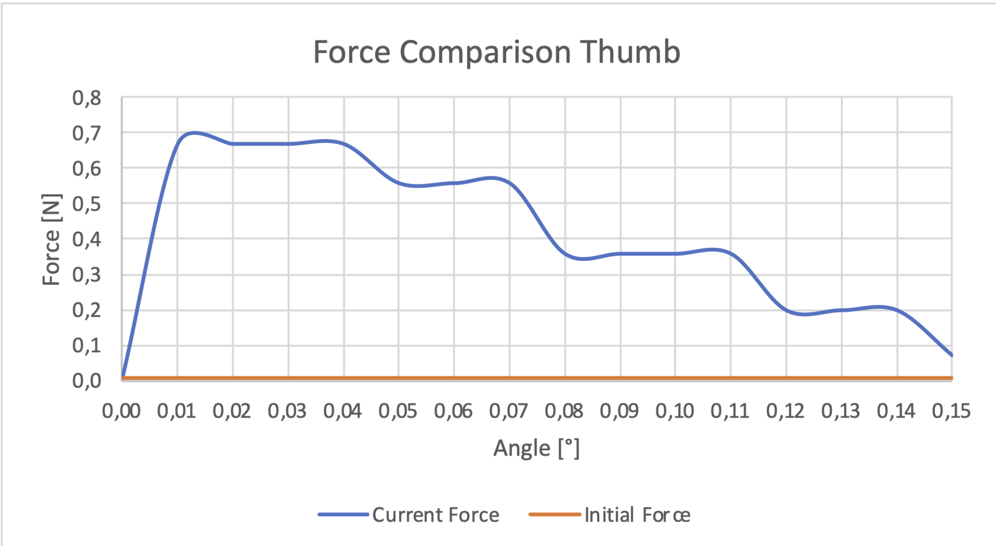
(a) Test 8 - Index force comparison.



(b) Test 8 - Middle force comparison.



(c) Test 8 - Ring force comparison.



(d) Test 8 - Thumb force comparison.

Figure 4.9: Force comparison - Test 8.

4.2.3 Manipulated Object: Polyester Sphere

The manipulated object is a polyester sphere. Tests 9 and 10 are performed using the same axis of rotation, but with different constants λ.

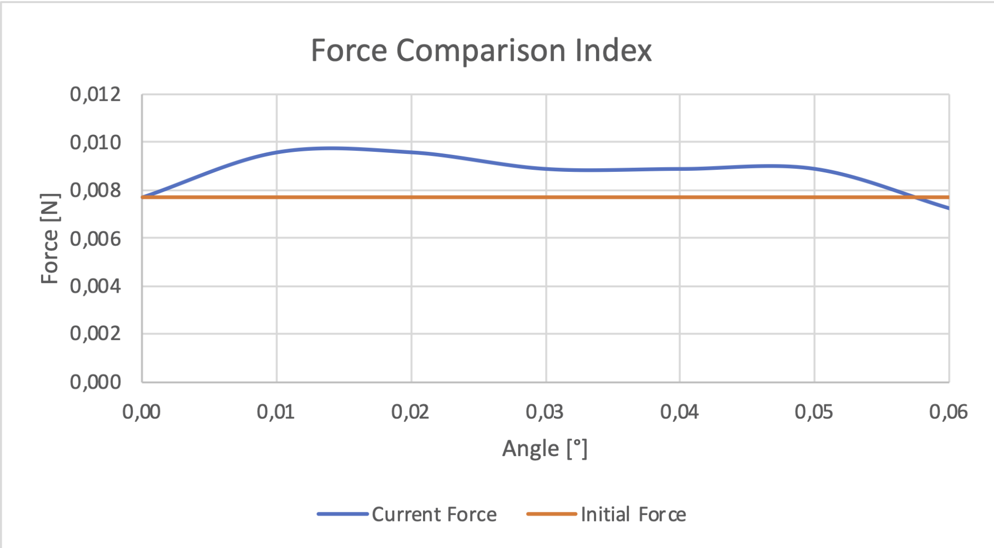
Test 9

- Axis of rotation: passing through the center of the tetrahedron and parallel to y axis of the base link.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}001\ \frac{m}{N}$
- Video link: `https://youtu.be/VMw4q68n-5c`.

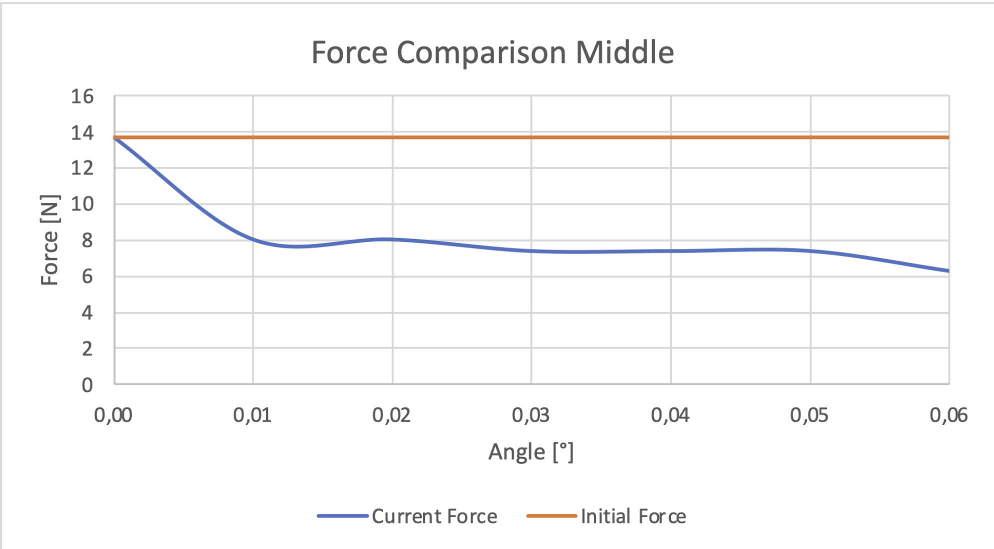
Initial Forces	
Index	0,0077 N
Middle	13,6804 N
Ring	0,0958 N
Thumb	0,7751 N

Table 4.9: Initial forces - Test 9

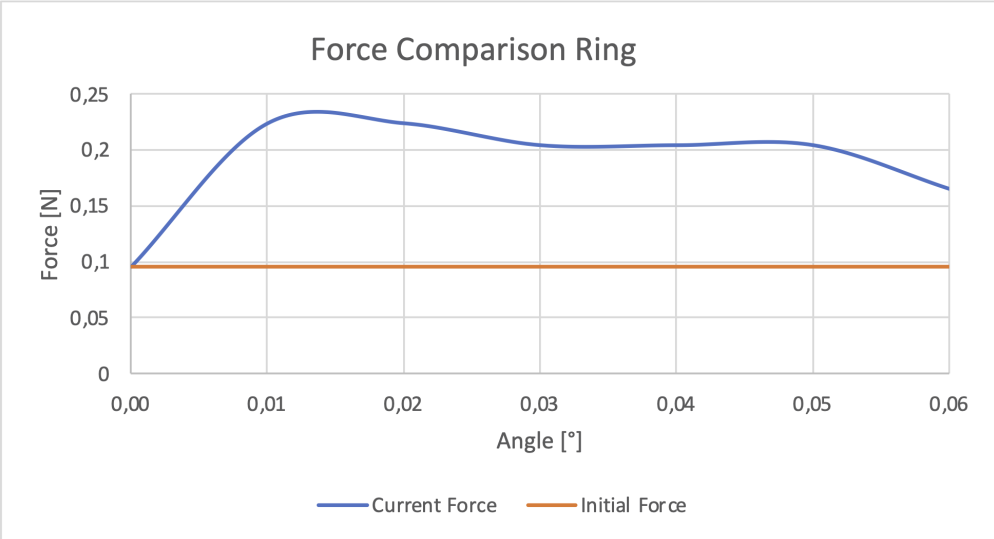
The Index immediately loses contact with the object. The force error tends to increase during rotation for the Middle and Thumb. The force error tends to decrease for the ring, but during the stall phase, i.e. when the device is waiting for the command to start the return phase, the contact is lost. The return phase is unsuccessful.



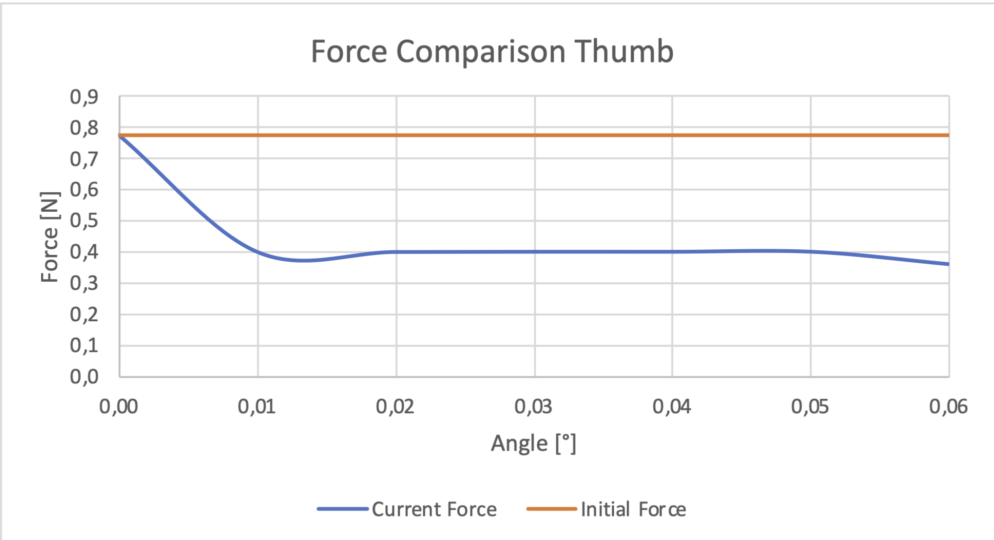
(a) Test 9 - Index force comparison.



(b) Test 9 - Middle force comparison.



(c) Test 9 - Ring force comparison.



(d) Test 9 - Thumb force comparison.

Figure 4.10: Force comparison - Test 9.

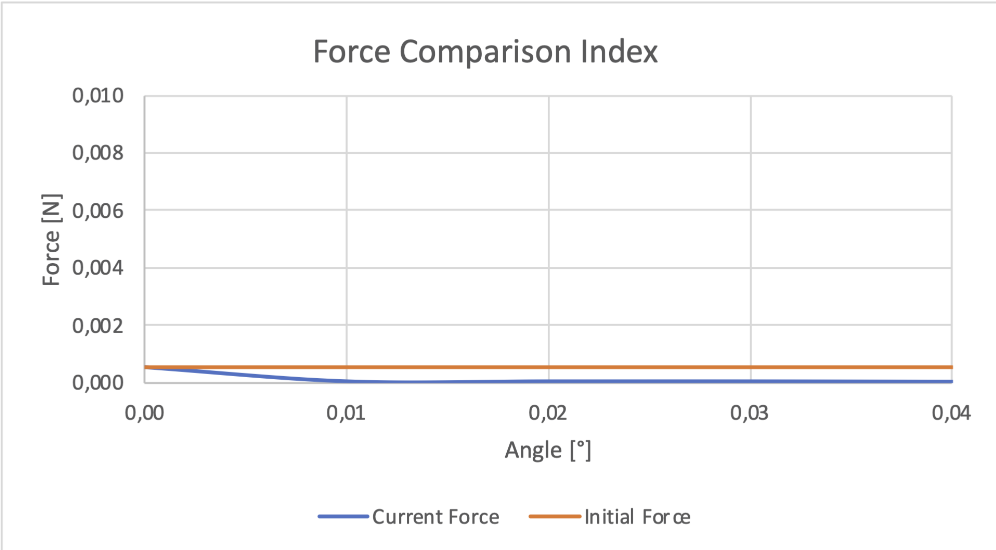
Test 10

- Axis of rotation: passing through the center of the tetrahedron and parallel to y axis of the base link.
- Angle of rotation $\theta = 100°$.
- Step of rotation $\theta_{step} = 0.01°$.
- Direction of rotation: positive direction of the axis of rotation.
- Predefined constant: $\lambda = 0{,}1\ \frac{m}{N}$
- Video link: `https://youtu.be/2sFeilaNkP8`.

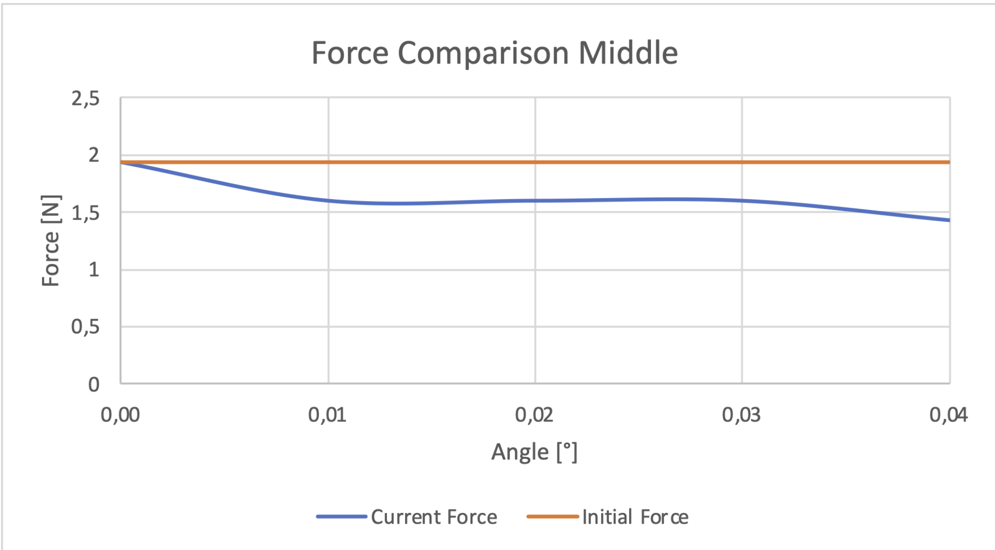
Initial Forces	
Index	0 N
Middle	1,935 N
Ring	0 N
Thumb	1,02 N

Table 4.10: Initial forces - Test 10

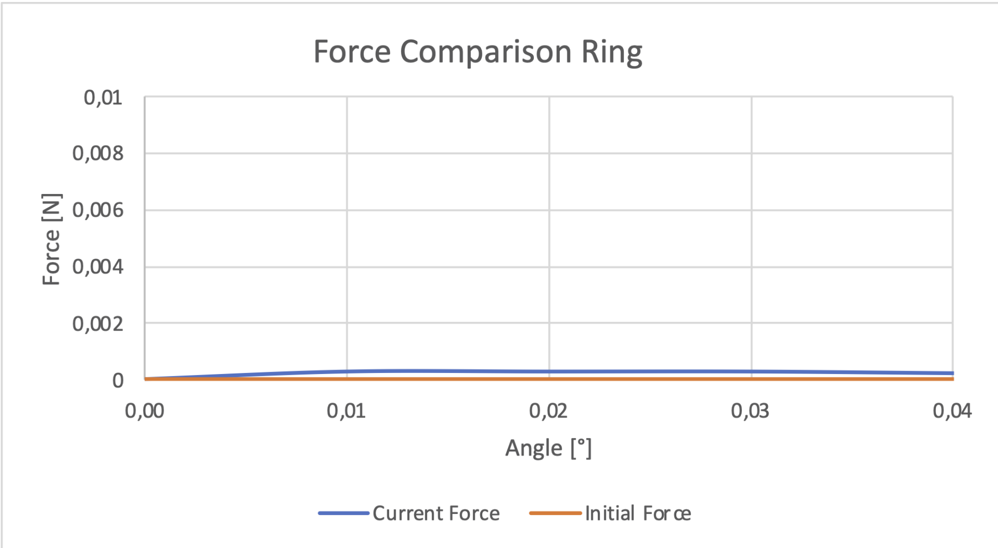
For the Index and Ring, since the initial forces are zero, there is a measurement error in the detection by the sensors. Moreover, the Index and Ring lose contact immediately, so they are not involved in the manipulation. The rotation is satisfactory since the force error is relatively small for the Middle and Thumb. The return phase is unsuccessful.



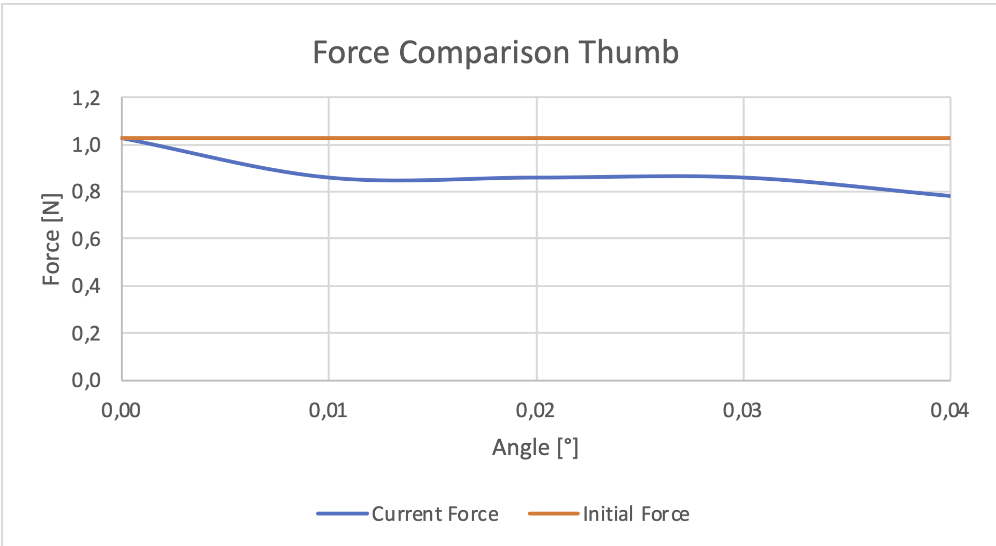
(a) Test 10 - Index force comparison.



(b) Test 10 - Middle force comparison.



(c) Test 10 - Ring force comparison.



(d) Test 10 - Thumb force comparison.

Figure 4.11: Force comparison - Test 10.

Chapter 5

Conclusions and Future Research

The thesis is focused on the in-hand manipulation of an object. The manipulated objects are rigid bodies, and their physical properties are unknown. The manipulation is exploited through the Allegro Hand, a multi-fingered robotic hand device composed of four fingers (Index, Middle, Ring, and Thumb) and sixteen degrees of freedom. Weiss Tactile sensors FT 0408 are installed on the fingertips, from which the only information available to implement a manipulation strategy is obtained. The communication with the robotic device is through ROS (Robot Operating System), and the implemented algorithm is developed in C++ language.

The developed manipulation strategy and corresponding algorithm are inspired by experiments illustrated in the paper «Dexterous manipulation of unknown objects using virtual contact points», drafted by A. Montaño and R. Suárez. The researchers mentioned above operate with only three fingers of the hand (Index, Middle, and Thumb) to manipulate the object. The algorithm at the base of their project returns out to be verified only for a specific axis, i.e., the axis parallel to the palm and passing through the point of contact between Thumb and object. This thesis aims to generalize the manipulation to every axis in space involving all the fingers of the Allegro Hand. The basic idea is that all contact points between object and fingertips must travel circular trajectories around an axis arbitrarily chosen by the user. The possible positions each contact point can explore are obtained through the algebra of quaternions. After that, the possible contact points are adjusted according to the current force sensed by the sensor. It is defined a linear law that brings into play an experimental constant λ to transform the force error into a displacement. The results, i.e., the new contact points between object and fingertips, must be converted from operational space to joint space. This is done through the use of Inverse Kinematics. For Index, Middle, and Ring, Inverse

Kinematics is an algorithm based on geometric considerations. Since its kinematic structure is different from the other fingers, the Damped Least Square Method (KDL) is used for the Thumb. If Inverse Kinematics returns solutions within the structural limits of the hand, the movement is performed, and the desired positions are achieved. In case there are no solutions, the program stops. The initial closure algorithm is not explored in depth during the thesis.

The manipulation strategy is tested and verified in the simulation environment. However, in this case, since it is not possible to simulate the object to be manipulated, the contact points are assumed at the center of the sensors, and the forces involved are not taken into account. The experimental phase is performed with different objects to be manipulated: a plastic tea bottle, an aluminum can, and a polyester sphere. Four tests are performed for the plastic bottle. The first three tests (i.e., Test 1,2,3) are repeated under the same conditions. The results highlight that although the object being manipulated is the same, each test is unique because the forces involved are different. In the last test (Test 4), the parameter λ is tried to increase, but the result is unsatisfactory, and the object falls. Even for the aluminum can, four tests are performed. The first two (Tests 5 and 6) have the same axis of rotation but different λ between them. Even if small, the rotation is successful in the first test (Test 5). In the second test (Test 6), λ is increased, and the object is not manipulated. The remaining two tests (Tests 7 and 8) are performed under the same conditions among them, but the axis of rotation and λ are different from the first two tests (different from Tests 5 and 6). Tests 7 and 8 confirm that each test is unique. Moreover, it is possible to state that rotation occurs under the same test conditions despite the different forces involved. For the polyester sphere, two tests are performed with the same axis of rotation, but different lambda between them. It is noticed that by increasing lambda, the rotation happens anyway in this case, but only for the forward phase. The return phase occurs without success showing its limitations. From the set of all the tests performed, it is possible to state that the algorithm works properly in finding the new contact points and computing the Inverse Kinematics. The whole strategy, including the force error procedure, operates relatively well for rotation axes that are parallel to the palm of the hand. As expected, the kinematics of the hand allow very small or nonexistent rotations around some particular axis. In addition, the proposed solution that relates position to force is limited to a few cases. At the following link, it is possible to consult the web site in which all the videos are enclosed, both those demonstrative and those relative to the tests, and the code in C++ used for the implementation:

`http://inhandmanipulation.freecluster.eu/`

The experimental results demonstrate that, as described in the introduction (Section 1), Dexterous Manipulation is an area of robotics full of uncertainties since it requires extreme accuracy in considering the variables involved. Below are listed some proposals that could contribute to the improvement of the project:

- Develop a proper initial closure algorithm that can allow the initial forces to be the same each time the experiment is repeated for the same object.
- Try to limit the instability phenomenon due to current peaks by adjusting the gains of the PID controller.
- Consider the curvature of the Weiss tactile sensors and do not treat them as a flat surface. Also, one could test the strategy by considering different types of tactile sensors.
- Consider the tangential forces between object and fingertips.
- Find a better position-force relationship, maybe implementing a hybrid controller.
- Implement a geometrical approach to solve the Thumb Inverse Kinematics instead of an iterative algorithm.
- Elaborate a strategy for the returning phase that considers the forces involved.

Appendix A

Contact Models

Studying robotic grasping it is mandatory to summarize the most common contact models. First of all they can be considered as kinematic pairs, in which the objects involved can interact through contact points, lines or planes. Additionally, friction can be considered too, obtaining many combinations and therefore many pairs. Considering the field of study taken into account, the more used ones are the frictionless point contact, the frictional point contact and the soft-finger contact, which will be briefly described below.

For first one mentioned, the frictionless point contact, the contact occurs between objects without involving friction phenomena. Furthermore, the contact takes place in a point, C_i, in which the finger applies a force perpendicular to the object surface. It is easy to understand that this model represent an over-simplified situation of the actual grasp.

More realistic model is the frictional point contact, in which the contact still takes place in a point, but besides the normal force mentioned above, there is a tangential component applied to the surface. Among the several empirical models proposed to describe the phenomenon, the Coulomb's friction model is the most used. It states that the force opposes to the motion, but its amplitude does not depend on velocity nor on contact surface. Motion does not occur when $F_t \leq \mu F_n$, where F_t is the tangential component of the force, F_n is the normal component and μ is the static friction coefficient, which depends on the materials of the objects involved in the contact. This constrain implies the fact that the force vector stays inside a cone centered at the contact point.

The last model mentioned is the soft-finger contact. It is the only model, among the ones mentioned, where the contact occurs on a surface. It also considers the friction phenomenon previously stated, with an additional torque normal to the surface. It is easy to understand that this model is the most accurate, but due to its complexity, it is preferred to use the frictional point contact in the robotic grasping, obtaining results with an acceptable accuracy.

Appendix B

Damped Least-Squares Method

Before starting with the explanation of the iterative method, it is important to do a brief introduction to frame the problem and define the notation.

A robotic device is modeled as a set of link connected by joints. For simplicity, the algorithm below is regarding only rotational joints, but it can be applied to arbitrary joints. The whole configuration of a robot is defined by the scalars $\theta_1, ..., \theta_n$, with n number of joints. Moreover, it is possible to write the joints angles as a column vector: $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)^T$.

The positions of the end effectors is identified as $\mathbf{s}_i$, with $i = \{1, ..., k\}$ and with k number of the end effectors. The column vector $\vec{\boldsymbol{s}} = (\mathbf{s}_1, ..., \mathbf{s}_k)^T$ can be viewed as a column vector either with $m = 3k$ scalar entries or with k entries from $\mathbb{R}^3$.

To be able to switch from one joints configuration to another, it is necessary to define the target position vector: $\vec{\boldsymbol{t}} = (t_1, ..., t_k)^T$, where $\mathbf{t}_i$ is the target position for the ith end effector.

Now, let's define the desired change in position as $\boldsymbol{e_i} = \boldsymbol{t_i} - \boldsymbol{s_i}$ (moving to the desired ith target), also written as $\vec{\boldsymbol{e}} = \vec{\boldsymbol{t}} - \vec{\boldsymbol{s}}$.

End effector position is a function of the joints angles, so it is possible to define the *Forward Kinematics (IK) problem*:

$$\vec{\boldsymbol{s}} = f(\theta) \tag{B.1}$$

Equation B.1 can be also written as:

$$\boldsymbol{s_i} = f(\theta_i) \quad \text{with} \quad i = \{1, ..., k\} \tag{B.2}$$

The goal of *Inverse Kinematics (IK)* is to find a vector $\boldsymbol{\theta}$ such that $\vec{\boldsymbol{s}}$ is equal to a given desired configuration $\vec{\boldsymbol{s}}_{\boldsymbol{d}}$:

$$\theta = f^{-1}(\vec{\boldsymbol{s}}_{\boldsymbol{d}}) \tag{B.3}$$

where f is a highly non linear operator difficult to invert. For this reason iterative methods are used.

Iterative methods are all based on the Jacobian matrix:

$$J(\theta) = \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j)}\right)_{i.j} \tag{B.4}$$

where $i = \{1, ..., k\}$ and $j = \{1, ..., n\}$.

Equation B.1 can be rewritten, in function of the Jacobian, as:

$$\dot{\vec{\mathbf{s}}} = J(\theta)\dot{\boldsymbol{\theta}}. \tag{B.5}$$

Assuming that the values $\boldsymbol{\theta}$, $\vec{\mathbf{s}}$ and $\vec{\mathbf{t}}$ are known, thanks to Equation B.4 is possible to compute the value of the Jacobian matrix.

At this point the goal is to find a value $\Delta\boldsymbol{\theta}$ for incrementing the joint angles $\boldsymbol{\theta}$ by $\Delta\boldsymbol{\theta}$:

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \Delta\boldsymbol{\theta}. \tag{B.6}$$

The last equation (Equation B.6) leads a change in joints angle and, consequently, a change in end effector position that can be estimated as:

$$\vec{\mathbf{s}} \approx J\Delta\boldsymbol{\theta} \tag{B.7}$$

The idea is that the value of $\Delta\boldsymbol{\theta}$ should be chosen so that $\vec{\mathbf{s}} \approx \vec{\mathbf{e}}$. Consequently, the FK problem can be expressed as:

$$\vec{\mathbf{e}} = J\Delta\boldsymbol{\theta} \tag{B.8}$$

and the IK problem can be rewritten as:

$$\Delta\boldsymbol{\theta} = J^{-1}\vec{\mathbf{e}} \tag{B.9}$$

In most cases, the IK equation cannot be solved uniquely because the Jacobian J may not be square or invertible, and even if it is invertible, J may work poorly as it may be nearly singular[1]. Several approaches have been proposed to overcome these problems such Jacobian Pseudo-inverse, Jacobian Transpose, Singular Value Decomposition, Damped Least Squares, Selectively Damped Least Squares and so on. In this appendix only the Damped Least Squares method is analyzed. For more information about the other methods it is possible to refer to articles such that [3] and [2].

[1]Singularities occur when no change in joint angle can achieve a desired change in chain end position.

The damped least squares method (DLS), also called Levenberg-Marquardt method, give a numerically stable method of selecting $\Delta\theta$. DLS starts finding the value of $\Delta\theta$ that minimizes the quantity (objective function):

$$\|J\Delta\theta - \vec{e}\|^2 + \lambda^2\|\Delta\theta\|^2, \tag{B.10}$$

where $\lambda \in \mathbb{R}$ is a non-zero damping constant. This is equivalent to minimizing:

$$\left\| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta - \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix} \right\|. \tag{B.11}$$

The previous equation (Eq. B.11) can be rewritten as:

$$\left(J^T J + \lambda^2 I\right) \Delta\theta = J^T \vec{e} \tag{B.12}$$

It can be shown that $J^T J + \lambda^2 I$ is non-singular (Section 6 of reference [3]). Thus, the damped least squares solution is equal to:

$$\Delta\theta = \left(J^T J + \lambda^2 I\right)^{-1} J^T \vec{e}. \tag{B.13}$$

Now $J^T J$ is an $n \times n$ matrix, where n is the number of degrees of freedom. Using matrix algebra, is it easy to find that:

$$\left(J^T J + \lambda^2 I\right)^{-1} J^T = J^T \left(J J^T + \lambda^2 I\right)^{-1}.$$

Substituting in Equation B.13:

$$\Delta\theta = J^T \left(J J^T + \lambda^2 I\right)^{-1} \vec{e}. \tag{B.14}$$

The advantage of Equation B.14 is that the matrix being inverted is only $m \times m$, where $m = 3k$ is the dimension of the space of target positions, and m is often much less than n. Additionally, the equation can be computed without needing to carry out the matrix inversion.

The damping constant depends on the details of the multibody and the target positions and must be chosen carefully to make Equation B.14 numerically stable. The damping constant should large enough so that the solutions for $\Delta\theta$ are well-behaved near singularities, but if it is chosen too large, then the convergence rate is too slow.

Appendix C

Pose of a Rigid Body

A rigid body is completely described in space by its position and orientation (in brief pose) with respect to a reference frame. *O - xyz* (base link R.F.) is the orthonormal reference frame and **x,y,z** are the unit vectors of the frame axes. The position of a point **o**' on the rigid body with respect to the coordinate frame *O - xyz* is expressed by the relation:

$$\mathbf{o}' = o'_x\mathbf{x} + o'_y\mathbf{y} + o'_z\mathbf{z} \tag{C.1}$$

The compact form of **o**' is:

$$\mathbf{o}' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix} \tag{C.2}$$

The orientation is described by an orthonormal frame *O' - x'y'z'* attached to the body and its unit vectors are **x',y',z'**. Expressing this vector in the base link R.F. *O - xyz* :

$$\mathbf{x}' = x'_x\mathbf{x} + x'_y\mathbf{y} + x'_z\mathbf{z} \tag{C.3}$$

$$\mathbf{y}' = y'_x\mathbf{x} + y'_y\mathbf{y} + y'_z\mathbf{z} \tag{C.4}$$

$$\mathbf{z}' = z'_x\mathbf{x} + z'_y\mathbf{y} + z'_z\mathbf{z} \tag{C.5}$$

Figure C.1 shows the pose of a rigid body with respect to an orthonormal reference frame.

The components of each unit vector are the direction cosines of the axes of *O' - x'y'z'* with respect to the reference frame *O - xyz*. By adopting a compact notation is possible to define the rotation matrix that describes the orientation of the body as follows:

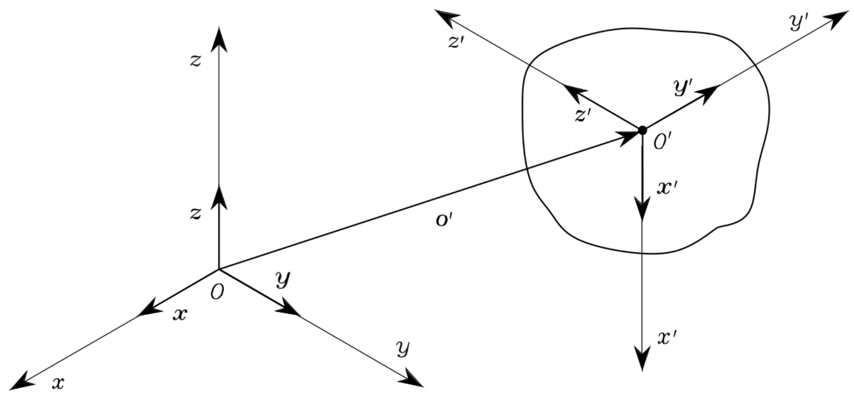


Figure C.1: Pose and orientation of a rigid body.

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \tag{C.6}$$

In order to manage transformations in the space it is possible to define the homogeneous transformation matrix:

$$\mathbf{T}_{EE}^{BaseLink} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & o'_x \\ r_{yx} & r_{yy} & r_{yz} & o'_y \\ r_{zx} & r_{zy} & r_{zz} & o'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.7}$$

Bibliography

[1] *Allegro hand user's manual v4.0* (cit. on p. 6).

[2] A. Aristidou and J. Lasenby, «Inverse kinematics: A review of existing techniques and introduction of a new fast iterative solver», Sep. 2009 (cit. on p. 92).

[3] S. Buss, «Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods», *IEEE Transactions in Robotics and Automation*, vol. 17, May 2004 (cit. on pp. 92, 93).

[4] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, «The grasp taxonomy of human grasp types», *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016. DOI: 10.1109/THMS.2015.2470657 (cit. on p. 30).

[5] O. Fuentes and R. C. Nelson, «Virtual tool approach to dextrous telemanipulation», vol. 2, IEEE, 1996, pp. 1700–1705. DOI: 10.1109/robot.1996.506957.

[6] W. Fukui, F. Kobayashi, H. Nakamoto, and F. Kojima, «Object manipulation based on tactile information of multi-fingered robot hand», vol. 37, IOS Press BV, 2012, pp. 185–195, ISBN: 9781614990918. DOI: 10.3233/978-1-61499-092-5-185.

[7] A. Gasparetto and L. Scalera, «A brief history of industrial robotics in the 20th century», *Advances in Historical Studies*, vol. 08, pp. 24–35, Jan. 2019. DOI: 10.4236/ahs.2019.81002 (cit. on p. 1).

[8] C. Jara, J. Pomares, F. Candelas Herias, and F. Torres, «Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors' feedback», *Sensors (Basel, Switzerland)*, vol. 14, pp. 1787–804, Jan. 2014. DOI: 10.3390/s140101787.

[9] O. Kerpa, K. Weiss, and H. Wörn, «Development of a flexible tactile sensor system for a humanoid robot», vol. 1, 2003, pp. 1–6. DOI: 10.1109/iros.2003.1250596.

[10] M. T. Mason, *Mechanics of Robotic Manipulation*. Cambridge, MA, USA: MIT Press, 2001, ISBN: 0262133962 (cit. on p. 28).

[11] A. Montaño, «Object manipulation based on tactile information», *TDX (Tesis Doctorals en Xarxa)*, Mar. 2021. [Online]. Available: `http://www.tesisenred.net/handle/10803/672089` (cit. on pp. 2, 11, 14).

[12] A. Montaño and R. Suárez, «Dexterous manipulation of unknown objects using virtual contact points», *Robotics*, vol. 8, no. 4, p. 86, 2019, ISSN: 2218-6581. DOI: `10.3390/robotics8040086`. [Online]. Available: `https://www.mdpi.com/2218-6581/8/4/86` (cit. on pp. 28, 30, 31, 33, 35, 36).

[13] A. Montaño and R. Suárez, «Manipulation of unknown objects to improve the grasp quality using tactile information», *Sensors (Switzerland)*, vol. 18, 5 May 2018, ISSN: 14248220. DOI: `10.3390/s18051412`.

[14] Montaño, Andrés and Suárez, Raúl, «Improving grasping forces during the manipulation of unknown objects».

[15] R. M. Murray, Z. Li, and S. Sastry, *A mathematical introduction to robotic manipulation*. CRC Press, 1994, p. 456, ISBN: 9780849379819.

[16] A. Okamura, N. Smaby, and M. Cutkosky, «An overview of dexterous manipulation», in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, 255–262 vol.1. DOI: `10.1109/ROBOT.2000.844067`.

[17] S. T. Puente, L. Más, F. Torres, and F. A. Candelas, «Virtualization of robotic hands using mobile devices», *Robotics*, vol. 8, 3 Sep. 2019, ISSN: 22186581. DOI: `10.3390/robotics8030081`.

[18] C. Rosales, L. Ros, J. M. Porta, and R. Suárez, «Synthesizing grasp configurations with specified contact regions».

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010, ISBN: 1849966346 (cit. on pp. 2, 49).

[20] T. Yoshikawa, «Multifingered robot hands: Control for grasping and manipulation», *Annual Reviews in Control*, vol. 34, pp. 199–208, Dec. 2010. DOI: `10.1016/j.arcontrol.2010.09.001` (cit. on p. 3).