

# An Autonomous Procedure for Scoop-Grasp Exploiting Hand Reconfigurability



By

**Valerio Bo**

Supervisor

**Prof. Alessandro Rizzo**

---

Co-supervisor

**Prof. Domenico Prattichizzo**

Università di Siena

**Ing. Enrico Turco**

Università di Pisa

---

M.Sc. in Mechatronic Engineering

Department of CONTROL AND COMPUTER ENGINEERING (DAUIN)

School of Mechanical and Manufacturing Engineering (SMME)

Politecnico di Torino

Torino, Italy

April 2020

This thesis is dedicated to *my beloved parents Isabella and Giuseppe, my dear sister Viviana and my closest friends*

# Abstract

Grasping in unstructured environments requires highly adaptable and versatile hands together with strategies to exploit their features to get robust grasps. This thesis presents a method to grasp objects using the novel Soft Scoop Gripper (SSG). The main features of this hand are represented by the scoop and by the reconfigurability of its fingers. The first acts as an embodied environmental constraint (EC), it allows to facilitate the grasp and improves its robustness while also reducing the planning effort. The second allows adopting a single strategy for grasping objects of different sizes.

The thesis proposes three fingers' configurations that are designed to deal with various objects. These configurations are the results of a preliminary grasp analysis of 12 initial settings. The considered grasp strategy is the so-called scoop grasp, where the scoop adapts its orientation to the surface where it slides, while the soft fingers establish reliable contact with the object.

The hand positioning for grasping the object is implemented using a recently proposed functional model for soft hands, the closure signature. Besides, some criteria are defined to discriminate the optimal fingers' setting to approach a given object exploiting the position of the closure application point. Given the object to be grasped, the proposed grasp planner chooses the best configuration of the fingers depending on object geometry and orientation, and then suitably aligns the hand to it. Whereas, a hybrid force-velocity controller is implemented to execute the sliding motion on the surface.

The results of 250 grasping trials with 13 different objects using the SSG attached to a robot arm confirm the validity of the criteria used for choosing the best fingers' setting. Besides, it was shown that the deliberate exploitation of environmental (surface on which the object lies) and embedded (scoop) constraints leads to robust grasps and that fingers' reconfigurability allows using one strategy (scoop grasp) for several different objects.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>State-of-the-art</b>	<b>4</b>
<b>3</b>	<b>Methods</b>	<b>11</b>
3.1	Grasp Analysis . . . . .	13
3.2	Closure Signature . . . . .	17
3.3	Grasp Planner . . . . .	23
3.3.1	Vision Algorithm . . . . .	24
3.3.2	Trajectory Planner . . . . .	28
3.3.3	Hybrid Force-Velocity Control . . . . .	32
3.3.4	Hand Control . . . . .	36
<b>4</b>	<b>Results</b>	<b>39</b>
4.1	Experimental Setup . . . . .	39
4.2	Success Rate and Scoop Rate . . . . .	41
4.3	Motor Torques . . . . .	44
<b>5</b>	<b>Conclusions</b>	<b>46</b>
5.1	Discussions . . . . .	46
5.2	Future works . . . . .	48
	<b>Appendices</b>	<b>52</b>

## CONTENTS

<b>A</b>	<b>Hardware descriptions</b>	<b>53</b>
A.1	Kinect One . . . . .	53
A.2	KUKA LBR iiwa . . . . .	54
A.3	KUKA LBR iiwa . . . . .	56
<b>B</b>	<b>Software descriptions</b>	<b>57</b>
B.1	ROS . . . . .	57
B.2	MATLAB . . . . .	58
B.3	GAZEBO . . . . .	59
	<b>References</b>	<b>61</b>

# List of Figures

1.1	Sequence of grasping with the Scoop strategy . . . . .	2
2.1	Contact models commonly used in robotics: a) Points without Friction contact; b) Hard Finger contact; c) Soft finger contact . . . . .	4
2.2	From the left: RBO Hand 2, SDM Hand, Pisa/IIT Soft Hand . . . . .	5
2.3	Edge Grasping using two different strategies . . . . .	7
2.4	Human actuation of the Soft ScoopGripper . . . . .	9
2.5	Example of an autonomous robot using two collaborative arms . . . . .	10
3.1	CAD of the Soft ScoopGripper . . . . .	11
3.2	Grasp analysis . . . . .	15
3.3	Rotation of left and right dovetail joint whose axes of rotation point towards the palm of the hand: (a) clockwise rotation for the left finger (in blue) and a counterclockwise rotation for the right one (in orange) (b) opposite clockwise rotations . . . . .	16
3.4	Configurations estimated from the grasp analysis: (a) Configuration 1, (b) Configuration 2, (c) Configuration 3 . . . . .	17
3.5	Closure Signature of Configuration 1 . . . . .	19
3.6	Closure Signature of Configuration 2 . . . . .	20
3.7	Closure Signature of Configuration 3 . . . . .	21

3.8	Sides of the bounding box of an object laying on the table surface. $h$ is the side normal to the surface, $l$ the longest side on the surface plane and $w$ the shortest. . . . .	21
3.9	Selection criteria. The threshold $h_{thold}$ is about 9.7 cm, while $l_{thold}$ measures around 11.6 cm . . . . .	22
3.10	Scene with Kinect, Marker and Base Frame . . . . .	26
3.11	Workspace with the bounding box and the table array . . . . .	27
3.12	Scene with the Realtime URDF filter applied . . . . .	28
3.13	Pre-grasp pose aligned with x-axis of the object . . . . .	29
3.14	Pre-grasp pose aligned with y-axis of the object . . . . .	30
3.15	Initial (a) and final (b) pose of KUKA LBR iiwa . . . . .	31
3.16	Force function depending on the height of the end-effector . . . . .	34
3.17	Angle followed by the end-effector to reach the CS inclination for each configuration . . . . .	34
3.18	Alignment of the CS with each configuration . . . . .	35
3.19	Fingers' and scoop's motor torque . . . . .	37
3.20	Finite State Machine of the Soft ScoopGripper . . . . .	38
4.1	Experimental setup and tested objects. Soft ScoopGripper (A), ATI F/T sensor (B), KUKA iiwa LBR (C), and Kinect One (D) are shown. Objects are indicated with IDs and their properties are reported in Table 4.1. . . . .	39
4.2	Successful grasp without the use of the scoop (a) and exploiting the scoop (b) . . . . .	41
4.3	Success and scoop rates of Configuration 1 sorted in ascending order by weight and approached side (width) . . . . .	43
4.4	Success and scoop rates of Configuration 2 sorted in ascending order by weight and approached side (length) . . . . .	43
4.5	Success and scoop rates of Configuration 3 sorted in ascending order by weight and approached side (length) . . . . .	44

## LIST OF FIGURES

4.6	Motor torques for each configuration. Above: fingers' torque. Below: scoop's torque . . . . .	45
5.1	Difference between the scoop rates of Configurations 2 and 3 in ascending order by objects length. . . . .	48
5.2	Object recognition of multiple objects placed on a table . . . . .	49
5.3	Sigma7 haptic device . . . . .	51
A.1	Kinect One Sensor Components . . . . .	53
A.2	Schematics of the ATI sensor . . . . .	56
B.1	Part of the Ros Node Graph . . . . .	58
B.2	Closure Signature estimated on MATLAB . . . . .	59
B.3	Model of the Soft ScoopGripper in GAZEBO . . . . .	60

# List of Tables

3.1	Technical features of the Soft ScoopGripper . . . . .	12
3.2	Grasp Isotropy Index (GII) for the 12 initial configurations . . . . .	13
3.3	Closure Signature's properties . . . . .	23
4.1	Objects properties. The second last column reports the size of the bounding box of the object, indicating the values of $w$ , $l$ , and $h$ when the object was configured as in Fig. 4.1. The last column reports the configurations that were used to grasp the object. Some lines contain two numbers because the pose of the object was changed in order to test different configurations. . . . .	40
5.1	Comparison between C2 and C3 rates sorted in ascending order by objects' length . . . . .	47
A.1	Technical specifications of Kinect One . . . . .	54
A.2	Technical specifications of KUKA LBR iiwa . . . . .	55

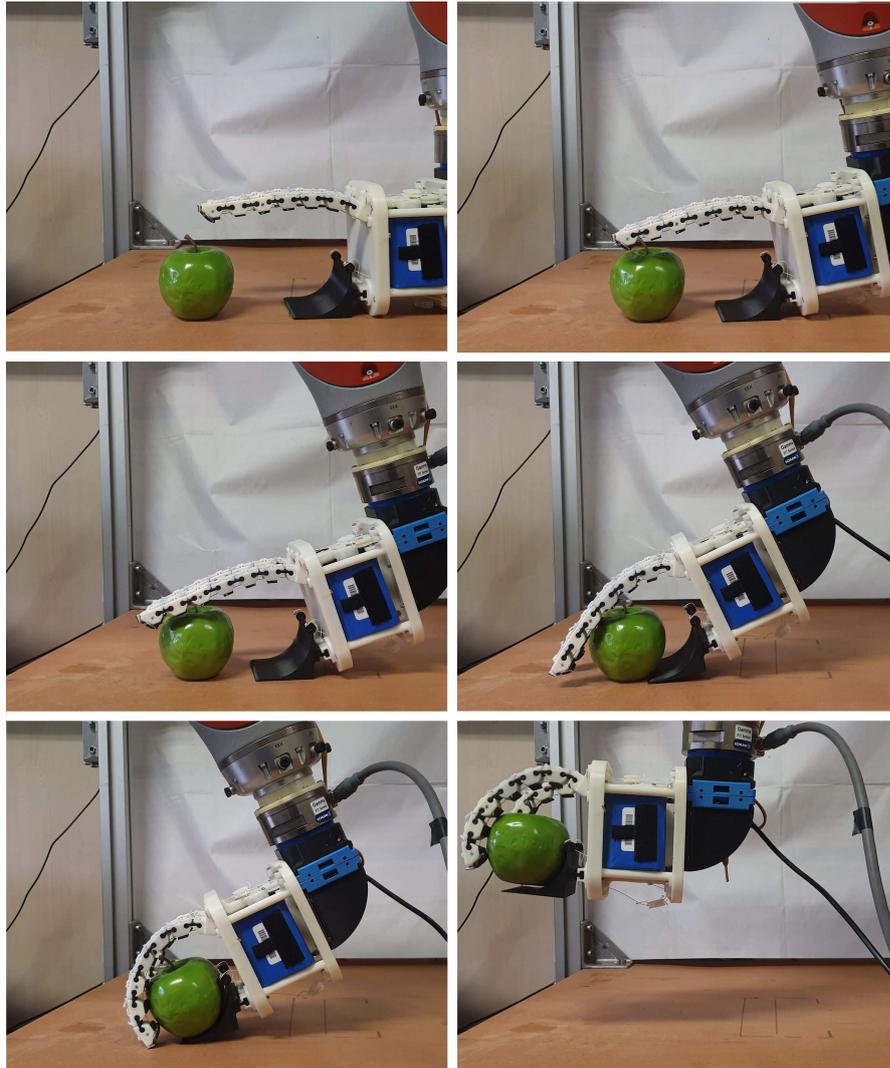
## CHAPTER 1

# Introduction

In this chapter, a brief introduction to the subject of the thesis is exposed. An explanation of what is contained in each chapter is given, including the structure of the paragraphs. The thesis aims to prove the development of the scoop-grasp, using an algorithm that suggests which configuration of the gripper should be applied. This process is intended to make autonomous the grasping of the Soft ScoopGripper, a novel soft hand. The method consists in the development of an automatic procedure whose purpose is to grasp multiple objects placed on a surface within the workspace. The steps necessary to achieve a successful grasp will be explained more in detail later, but briefly are:

1. Object detection through a vision system;
2. Trajectory planning to a pre-grasp pose;
3. Object-hand-environment interaction using a hybrid force-velocity control;
4. Closure of the gripper to grasp the object;
5. Return to the robot's initial pose;

Firstly, a preliminary grasp analysis has been performed in simulation to analyze the hand's capabilities: 12 fingers configurations were studied. Hence, three of them have been selected for a more in-depth analysis. Exploiting the recent Closure Signature method, it was possible to choose the best hand configuration for a given object and the best alignment over it. A set of objects whose properties are not known a priori has been used. Lastly, the analysis of the results has been carried out. Starting from the State-Of-The-Art, the content of the chapters is briefly introduced below.



**Figure 1.1:** Sequence of grasping with the Scoop strategy

- In Chapter 2 (State-of-the-Art), the study of the related papers was conducted. The discussion moves from a general definition of grasping deepen insight into the concepts of the novel Soft ScoopGripper. Lastly, the process behind the development and the automation of the grasping action is unfolded.
- In Chapter 3 (Methods), three paragraphs are described. In the first paragraph, the specifications of the Soft ScoopGripper and the process behind the choice of the three fingers' configuration are exposed. In the second paragraph, the process behind the calculation and the purpose of the Closure Signature are explained. In the third paragraph, the steps required to set up the experiment are listed. Every subparagraph explains in detail how the algorithms were developed and on which

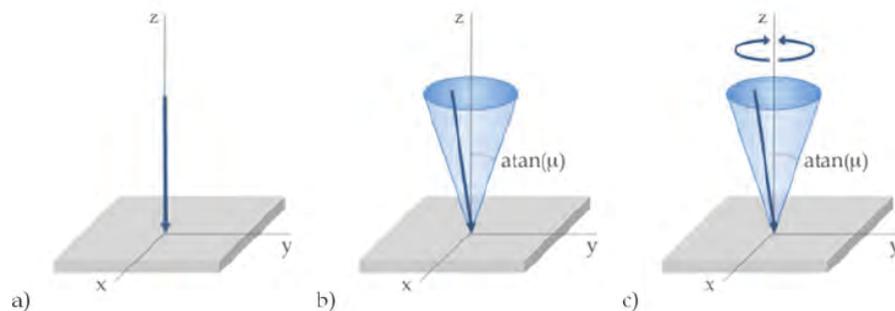
theoretical principles they are based.

- In Chapter 4 (Results), the set of the objects and the experimental results are reported.
- In Chapter 5 (Conclusions), the obtained results are commented on, and the possible future developments are shown.

# State-of-the-art

Grasping is the task of restraining objects [1]. Grasp analysis consists in defining a model that predicts the behavior of a gripper and an object under different loading. The purpose is to achieve a secure grasp without contact separation or sliding. Contact modeling is essential to ensure a secure grasp, and it varies based on the type of robotic hand being used [2]. Mainly, there are three contact models, as shown in Fig. 2.1:

1. Points without Friction (PwoF): this condition occurs when only the forces acting normally on the contact point of the surface are considered;
2. Hard Finger (HF): all the forces that lie within the friction cone around the surface normal can be exerted in the presence of friction;
3. Soft Finger (SF): it arises when a torque can be applied in the contact point around the surface normal;



**Figure 2.1:** Contact models commonly used in robotics: a) Points without Friction contact; b) Hard Finger contact; c) Soft finger contact

The most fundamental requirements in grasping and dexterous manipulation are the abilities to hold an object and control its position and orientation relative to the palm of the hand. Given the model of the hand and the object to grab, the two most useful characterizations of grasp restraint are *form closure* and *force closure*:

- Form Closure consists of creating a case for the object using a specific configuration of the hand. It relies on unilateral, frictionless contact constraints to inhibit the motion of the object.
- Force Closure uses large contact forces (exerted by the gripper) so that there will be no sliding or separation between the contact points.

According to the studies, the distinction between the two closures is that no friction model is considered in Form Closure. In contrast, the frictional contact forces are considered in Force Closure.

Form Closure usually implies that fewer contact forces are applied, so the development of soft grippers has been widely spread. This kind of soft gripper is more adaptable and compliant, reaching a higher level of form closure than other rigid grippers. There are different kinds of soft grippers. Remarkable examples of under-actuated compliant hands, as shown in Fig. 2.2, are:

- RBO Hand 2 [3]
- SDM Hand [4]
- Pisa/IIT Soft Hand [5]



**Figure 2.2:** From the left: RBO Hand 2, SDM Hand, Pisa/IIT Soft Hand

Common features of the soft hand are *softness*, i.e., their embodied ability to comply and adapt to features of the environment, and *underactuation*, i.e., the number of active joints is less than the degree of freedom of the gripper. Underactuation allows the use of less complex control of the hand. Moreover, it also ensures robust grasping.

The robustness of the grasping depends on the pre-grasp pose of the hand. The pre-grasp pose can be evaluated for both rigid and soft hands. However, the evaluation processes are as under:

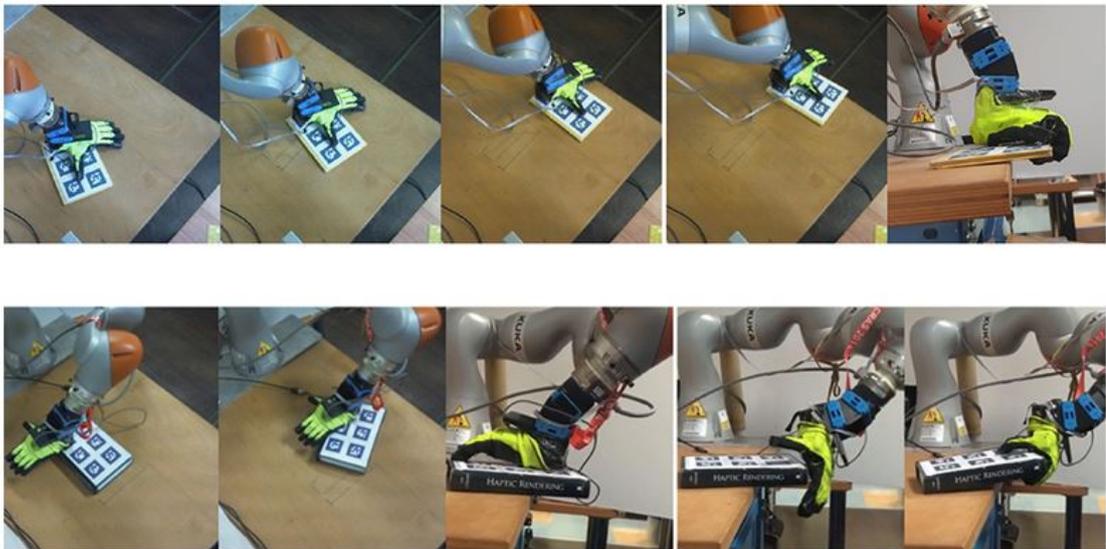
- For rigid hands, this process is simple: the transformation from the joint space to the task space is well known, using forward kinematics (from  $\mathbb{R}^n$  to  $\mathbb{R}^6$ ). Usually, these hands are redundant, so they allow multiple ways to approach the object, depending on the kinematics of the gripper. It is possible to analyze the forces acting on the contact points, merely knowing the exact position of the fingers. In this way, an estimation of the quality of the grasp can be extracted. Different methods are available, some based on algebraic properties of the grasp matrix  $G$  and others based on geometric relations [6].
- As regards soft hands, this process could be complex. The conventional analysis (the one applied to rigid hands) can be used with some limitations [7]. A recently proposed method for modeling the closure of robotic hands called *closure signature* (CS) [8] can estimate the direction of the maximum closure of the hand. This method consists of defining a set of suitable reference points on the hand and then tracking them from the initial to the final position. Thanks to the tracking, a point cloud is defined. Then, to describe the average motion of all the reference points, a homogeneous transform matrix can be extracted from the point cloud itself. A preferred closing direction can be retrieved from the analysis of the transformation. This direction and its application point constitute the Closure Signature. This process can be performed in a simulation environment such as Gazebo or SynGrasp [9], a specific toolbox in MATLAB.

The capability to interact with the environment is another characteristic that differentiates soft hands from rigid ones. This interaction is more developed in the soft hands, due to their compliance, and it makes possible the exploitation of new concepts regarding the environmental constraints. The interaction between the soft gripper and the environment was investigated in several works. One of them is the Environmental

Constraint Exploitation (ECE) [10] that focuses on the possible grasps obtained using the object's surroundings (e.g., wall grasp, edge grasp). This approach refers to human manipulation, where the arm trajectory planner is not always precise, and the grasp is achieved using tactile information.

Moreover, there are some situations when the hand cannot grasp the object, and the subject needs to interact with the environment to achieve the operation. When a robotic hand is performing these tasks, a precise strategy must be studied. In some cases, multiple procedures are necessary. They can be considered complementary to grasp objects depending on the dimensions of the objects themselves, as explained in [11] and shown in Fig. 2.3.

According to this research, the grasping strategy could be based on the exploitation of environmental constraints, which could be a support for the robotic hand, rather than an obstacle to be avoided. The unveiled innovation consists of building simple hands capable of achieving anthropomorphic grasps instead of design hands similar to the human ones. During the development of the gripper, the level of reachable interaction should be defined. It is possible to use a-priori analysis as a simulation tool to test the different levels of compliance. One of the most interesting studies available in the literature to examine the concepts relative to ECE is well explained in [12].



**Figure 2.3:** Edge Grasping using two different strategies

Starting from the main concepts of ECE, together with studies about the development of the grippers, the research community explored the possibilities of building a proper

constraint on the gripper itself. A noteworthy example is the Soft ScoopGripper [13]. The Soft ScoopGripper is a novel non-anthropomorphic robotic soft hand composed by:

- Two fingers, mono-actuated through the use of tendons. They are modular, made of some rigid and flexible parts. Both fingers can be passively rotated to adapt the hand to different kinds of objects. An application of these fingers is exposed in [14].
- One mono-actuated flat surface (scoop). It is connected through a flexible hinge to the hand palm. It can be actuated to raise the robustness of the grasp.

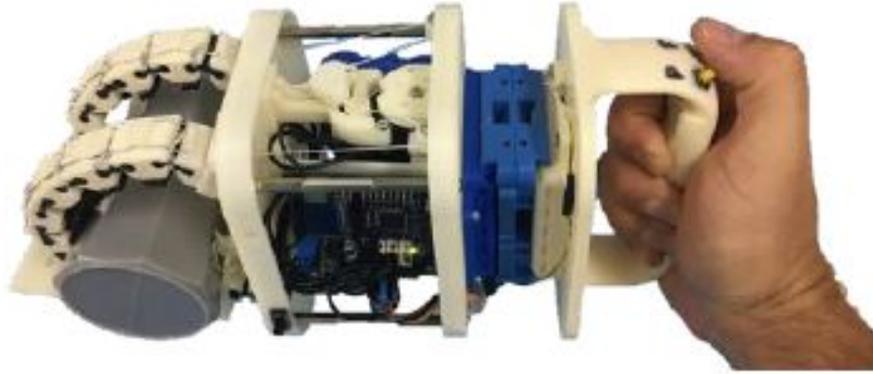
Instead of using just the soft fingers to wrap the objects, the gripper embodies the constraint in its design. This characteristic allows finding the primitives, i.e., the strategies of grasping with a certain gripper, independently of the environmental constraints. Thanks to the flexible hinge of the scoop, the flat part can easily slide on a surface, assuming the right orientation, enabling a grasp with a surface-constrained strategy as also done in [15, 16], where a shape-adaptation aware approach to grasp unknown objects is presented.

In such a grasp, the contact between the palm and a surface is used to level the hand with the object, and then the object is restrained on the palm by the fingers. In this case, the way of exploiting environmental constraints is facilitated by the actuated scoop that provides an additional constraint with respect to the palm. Indeed, shape match between hand, object, and environment can further improve grasp success.

Its purpose is no more to obtain an anthropomorphic grasp, but to avoid the detection and the usage of the constraints [17] in the surroundings, applying the concepts behind ECE from a new perspective.

These capabilities make the Soft ScoopGripper different from the previous soft hands. Besides, handling the rotation of the passive dovetail joints, it is possible to obtain several hand configurations. This feature enables the use of the same strategy for objects that usually needed different approaches, depending on the objects' pose or dimensions.

At the date, the Soft ScoopGripper was always managed by human operators, as can be seen in Fig. 2.4. Through the application of a handle, the human operator moves the gripper towards the object, and then actuates the motors pressing a button.



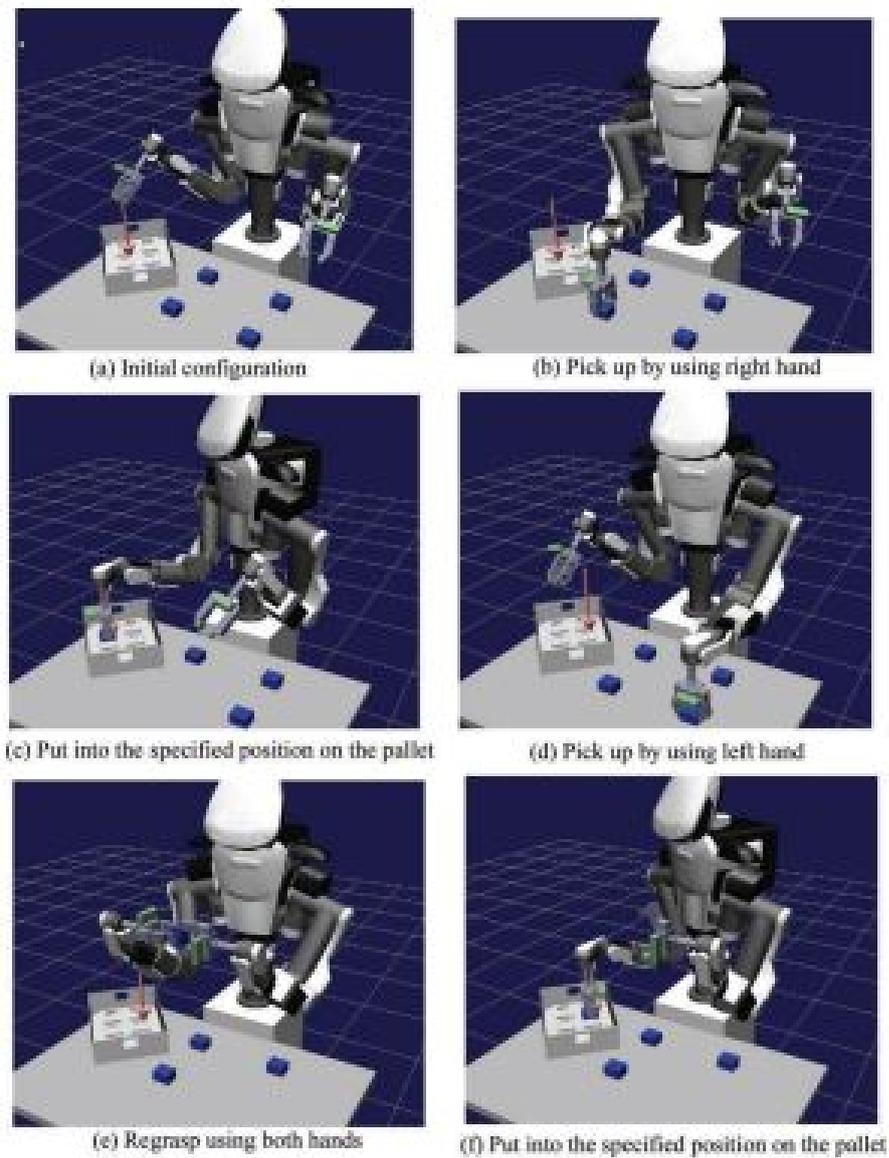
**Figure 2.4:** Human actuation of the Soft ScoopGripper

So, the next step for the further development of the Soft ScoopGripper is to make an automatic movement. Indeed, in literature and industry, some examples of gripper's automation are already available; the following actions are considered necessary to achieve a complete autonomous grasping:

1. Recognition of the object to be grasped in the workspace: it is crucial to recognize the object through the use of depth segmentation algorithm, as explained in [18];
2. The positioning of the robotic arm in a pre-grasp pose: it is necessary to decide which pre-grasp pose is the most appropriate according to the information given by the vision algorithm. There are different approaches to evaluate the pre-grasp pose:
  - The application of the Minimum Volume Bounding Box technique, such as in [19].
  - The usage of information about the local surface properties of the object to be grabbed as explained in [20];
3. Closing of the gripper: the closure of the gripper and the consequent successful grasp of the object can be verified with sensors when available;
4. Placing of the object in a goal position: the object is successfully moved to a predefined location.

A possible example of the entire online process can be found in [21]: the method discussed in this reference is more challenging due to the presence of two collaborative

arms, as can be seen in Fig. 2.5.

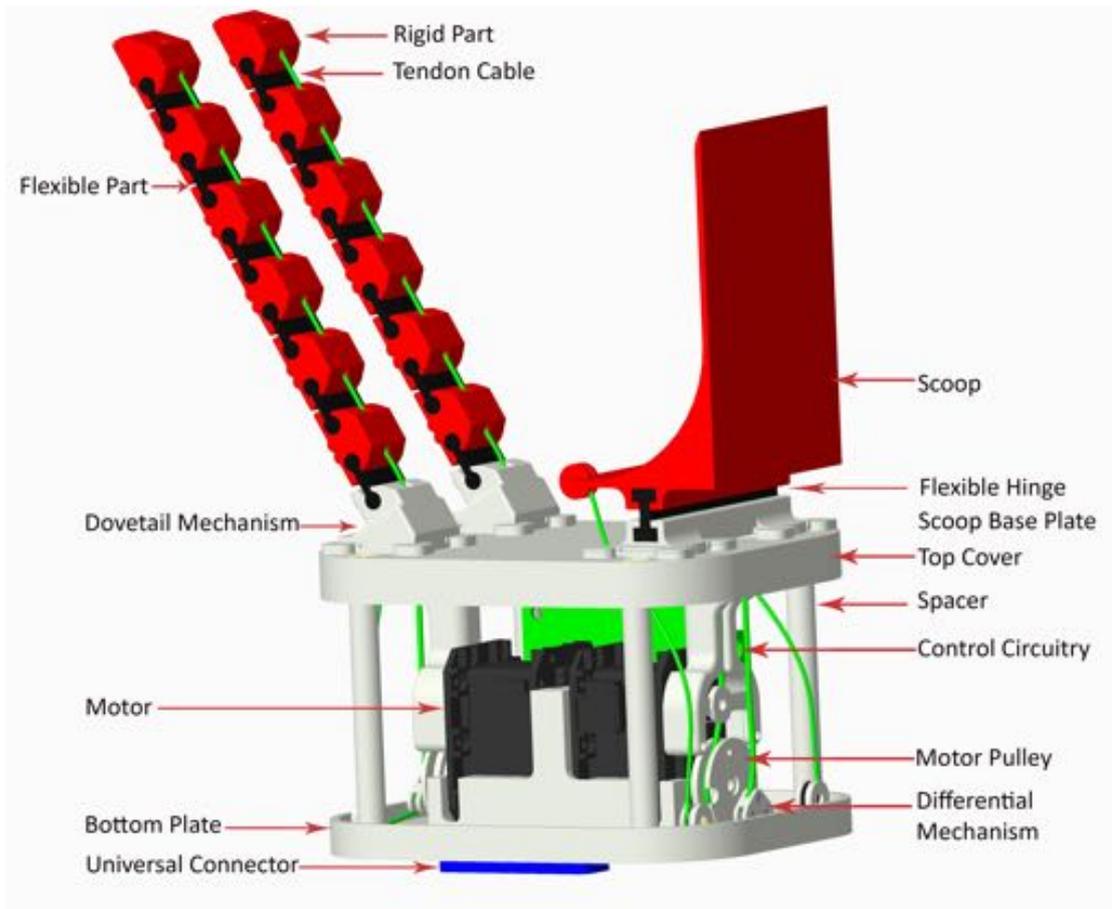


**Figure 2.5:** Example of an autonomous robot using two collaborative arms

## CHAPTER 3

# Methods

The Soft ScoopGripper consists of two modular fingers and a scoop connected to the gripper wrist by a flexible hinge.



**Figure 3.1:** CAD of the Soft ScoopGripper

Each module of the fingers consists of a rigid part 3D printed using ASA material and

a flexible part 3D printed in thermoplastic polyurethane. Table 3.1 shows the main technical features and parameters of the gripper.

<b>Technical features</b>		
Weight (including motors)	500g	
Max actuator torque	3.1 Nm @ 12 V	
Max current	2.8 A @ 12 V	
Continuous operating time	3.5 h @ stall torque	
Dimension of the wrist	130 mm x 105 mm x 85 mm	
Dimension of a finger	144 mm x 20 mm x 15 mm	
Dimension of the scoop	101 mm x 70 mm	
<b>Material Parameters</b>	<b>Flexible Part</b>	<b>Stiff Part</b>
Modulus of elasticity (E)	15.2 MPa	29 MPa
Shore Hardness	85 A	80 D
Density	1200 kg/m <sup>3</sup>	1070 kg/m <sup>3</sup>
<b>Geometric Parameters</b>		
width	20 mm	20 mm
length	17.5 mm	23 mm
height	2.5 mm	15 mm

**Table 3.1:** Technical features of the Soft ScoopGripper

The rigid parts contain holes to allow the passage of a cable that provides the tendon driven actuation. The actuation of the device is achieved by using two actuators and four tendons running in parallel, each pair of tendons connected to one actuator with two tendons running through the modular fingers and two running through the scoop. The actuators used are two *Dynamixel MX-28T*, each having a maximum torque of 3.1 Nm and a maximum angular speed of 684 deg/s. An *Arbotix-M* controller is used to control the actuators. This solution for Dynamixel motors incorporates an *AVR micro-controller*, a socket for an *Xbee* wireless radio, and the motor driver. The differential mechanism plays an essential role in the adaptation of fingers' configurations to the specific geometric features of the grasped object. If one of the fingers meets the object, the other finger can continue its flexion motion. The closure is the result of the winding

of the tendon cable on the pulley while the opening is achieved thanks to the elastic force stored in the flexible part of the modules. The CAD model of the gripper is shown in Fig. 3.1.

As already explained in Chapter 2, the modular fingers of the hand can be passively rotated. This characteristic allows the Soft ScoopGripper to assume different orientations of the fingers, determining a set of configurations.

### 3.1 Grasp Analysis

First, an analysis of the grasping was carried out on the MATLAB Toolbox SynGrasp. This preliminary analysis was performed computing the quality of 12 different grasps to evaluate which was the best fingers' configuration for grasping objects from the side exploiting the scoop.

Mainly, three elementary-shaped objects like a sphere, a cylinder and a cuboid were considered whose size is reported in Table 3.2.

	$\theta_L$	$\theta_R$	GII
Sphere	<b>0</b>	<b>0</b>	<b>0.0304</b>
(r=35mm)	$\pi/36$	$-\pi/36$	0.0265
	$\pi/18$	$-\pi/18$	0.0256
	$\pi/12$	$-\pi/12$	0.0147
Cylinder	$\pi/2$	$-\pi/2$	0.0013
(h=160mm	$7\pi/12$	$-5\pi/12$	0.0013
r=20mm)	$2\pi/3$	$-\pi/6$	0.011
	<b><math>3\pi/4</math></b>	<b><math>-\pi/4</math></b>	<b>0.014</b>
Cuboid	<b><math>\pi/2</math></b>	<b><math>-\pi/2</math></b>	<b>0.0106</b>
(w=30mm	$7\pi/12$	$-5\pi/12$	0.0094
l=150mm	$2\pi/3$	$-\pi/6$	0.093
h=130mm)	$3\pi/4$	$-\pi/4$	0.009

**Table 3.2:** Grasp Isotropy Index (GII) for the 12 initial configurations

The left and right dovetail joint angles can be denoted by  $\theta_l$  and  $\theta_r$ , respectively, whose axes of rotation point towards the palm of the hand.

Starting from the configuration ( $\theta_l = 0$  and  $\theta_r = 0$ ), the grasp matrix  $G$  was computed. The grasp matrix  $G$  is of the utmost importance in grasp analysis. The transpose of the grasp matrix maps the object twist to the contact frames. To derive the grasp matrix, let us denote  $\omega_{obj}^N$  the angular velocity of the object expressed in  $\{N\}$  and  $v_{i,obj}^N$  the velocity of the point on the object coincident with the origin of  $\{C\}_i$ , where  $\{N\}$  indicates the inertial frame and  $\{C\}_i$  the frame at contact  $i$ . These velocities can be obtained from the object twist referred to  $\{N\}$  as

$$\begin{pmatrix} v_{i,obj}^N \\ \omega_{obj}^N \end{pmatrix} = P_i^T v \quad (3.1.1)$$

where

$$P_i = \begin{bmatrix} I_{3 \times 3} & 0 \\ S(c_i - p) & I_{3 \times 3} \end{bmatrix} \quad (3.1.2)$$

Let  $R_i = (\hat{n}_i \hat{t}_i \hat{o}_i) \in \mathfrak{R}^{3 \times 3}$  represent the orientation of the  $i$ -th contact frame  $\{C\}_i$  w.r.t. the inertial frame and  $\bar{R}_i = \text{Blockdiag}(R_i, R_i)$ . Then, the grasp matrix  $G_i^T \in \mathfrak{R}^{6 \times 6}$ , that maps the object twist from  $\{N\}$  to  $\{C\}_i$  is

$$G_i^T = \bar{R}_i^T P_i^T \quad (3.1.3)$$

Once the grasp matrix has been estimated, the Grasp Isotropy Index (GII), i.e., the ratio between the minimum and the maximum singular value of  $G$ , was evaluated, as shown in Fig. 3.2.

The GII approaches one when the grasp is isotropic, i.e., the contact forces uniformly contribute to the total wrench applied on the object (optimal case), and approaches zero when the grasp is close to a singular configuration. This geometrical quality index was adopted because it is straightforward and sufficient to validate the preliminary analysis.

Computing more complex quality measures, such as an index of force closure, would have required additional information on the hand configuration not available during experiments.

For the sphere (Fig. 3.2a), it was considered that the fingers close from the top and thus

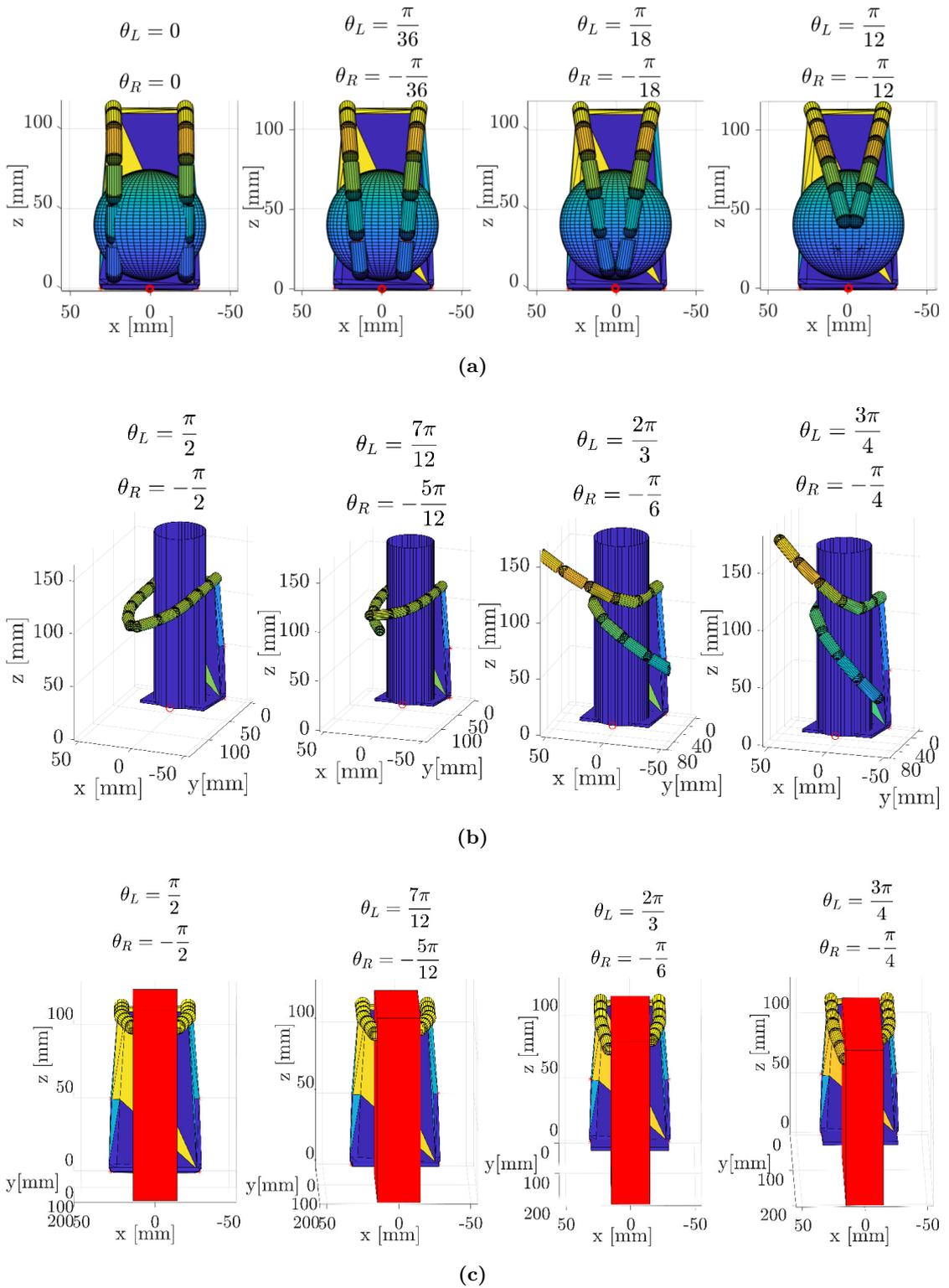
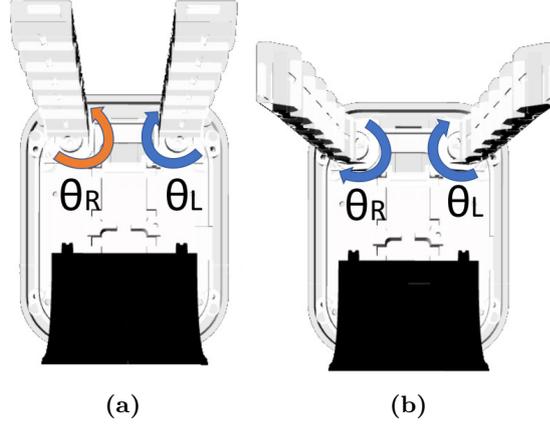


Figure 3.2: Grasp analysis



**Figure 3.3:** Rotation of left and right dovetail joint whose axes of rotation point towards the palm of the hand: (a) clockwise rotation for the left finger (in blue) and a counterclockwise rotation for the right one (in orange) (b) opposite clockwise rotations

simulated a clockwise rotation for the left finger and a counterclockwise rotation for the right one (Fig. 3.3a).

Steps of  $\pi/36$  rad for both fingers simultaneously were implemented until the final configuration ( $\theta_L = \pi/12$ ,  $\theta_R = -\pi/12$ ) was reached.

For the cylinder (Fig. 3.2b) and the cuboid (Fig. 3.2c), the configurations in which the object is grasped from the side were analyzed and the fingers' angles start from  $\theta_L = \pi/2$  and  $\theta_R = -\pi/2$  (Fig. 3.3b), and then rotate in opposite directions with steps of  $\pi/12$  rad.

Out of the 12 studied configurations, those with the highest GII index were selected for a more in-depth analysis.

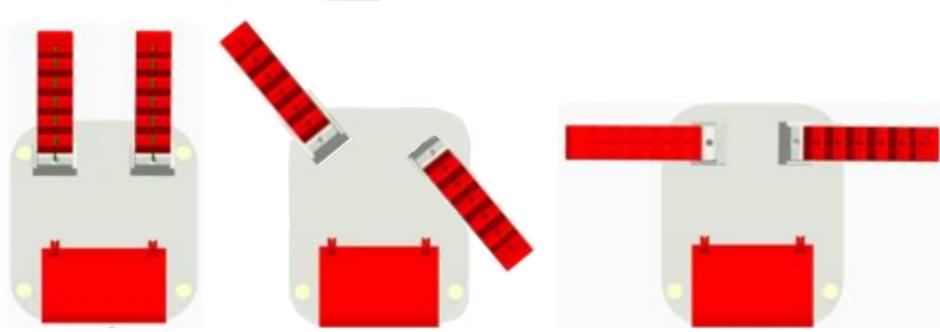
The selected configurations were:

- Configuration 1(C1):  $\theta_L = 0$ ,  $\theta_R = 0$
- Configuration 2(C2):  $\theta_L = 3\pi/4$ ,  $\theta_R = -\pi/4$
- Configuration 3(C3):  $\theta_L = \pi/2$ ,  $\theta_R = -\pi/2$

These resulting configurations can also be seen in Fig. 3.4.

Since the rotation of the fingers is not motor-driven, it is needed to rotate them by hand. In general, this led inaccuracies in positioning. Nonetheless, the adopted configurations

are easy to reproduce with a low error in the positioning of the fingers and at the same time with the highest grasp quality index.



**Figure 3.4:** Configurations estimated from the grasp analysis: (a) Configuration 1, (b) Configuration 2, (c) Configuration 3

## 3.2 Closure Signature

The analysis of the hand closure for each selected configuration allowed us to understand its behavior and to estimate a pre-grasp pose. This analysis has been performed evaluating the Closure Signature (CS) [8]. The study focused on the estimation of the orientation of the gripper such that the grasp is robust. The main feature of the Closure Signature is that it represents how the gripper deforms when wrapping an object. As a consequence, the Closure Signature is useful to obtain robust grasps, without taking into account how the gripper moves the object.

Three different calculations were necessary, one for each configuration. The steps to obtain the Closure Signature were the same for all the configurations:

1. The models of the configurations of the Soft ScoopGripper were developed in Gazebo. They were actuated to represent the same motion of the real gripper;
2. A reference point was placed on each module of the gripper, another one on the scoop and the last on the palm;
3. Hence, these reference points were tracked during simulated the motion of the SSG;
4. The gathered data were therefore analyzed in MATLAB, where the transformation between the initial and the final pose of the closing motion is found;

5. Ultimately, the closing direction and its application point (the CS) were extracted.

The Closure Signature is computed considering that  $p_f = \mathbf{T}p_i$  where  $p_f$  and  $p_i$  are the initial and final pose of the reference points. The matrix  $\mathbf{T}$  should be estimated using a precise method. Moreover, this transform cannot be calculated using typical tools because the transform is not purely rigid:

$$\mathbf{T} = \mathbf{T}_{def}\mathbf{T}_{rb} \quad (3.2.1)$$

where  $\mathbf{T}_{def}$  is the transform for the non rigid part and  $\mathbf{T}_{rb}$  is the rigid body transform. A method to solve this problem has been proposed in [22]. Considering  $p_k$  as a generic reference point, its augmented vector can be defined as  $\hat{p}_k = [p_k \ 1]$ .

Then, given the initial and final reference points  $\hat{p}_{k_i}$  and  $\hat{p}_{k_f}$ , the transformation  $\mathbf{T}$  can be evaluated by solving the linear system

$$\hat{p}_{k_f} = Mt \quad (3.2.2)$$

where  $t \in \mathbb{R}^{12}$  contains the components of the transformation  $\mathbf{T}$

$$\begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the system matrix  $M \in \mathbb{R}^{3n \times 12}$  is defined as

$$M = \begin{bmatrix} M_1 \\ \dots \\ M_n \end{bmatrix} \quad (3.2.3)$$

The generic matrix  $M_k \in \mathbb{R}^{3 \times 12}$  is given by

$$M_k = \begin{bmatrix} \hat{p}_{k_i}^T & 0_{1,4} & 0_{1,4} \\ 0_{1,4} & \hat{p}_{k_i}^T & 0_{1,4} \\ 0_{1,4} & 0_{1,4} & \hat{p}_{k_i}^T \end{bmatrix} \quad (3.2.4)$$

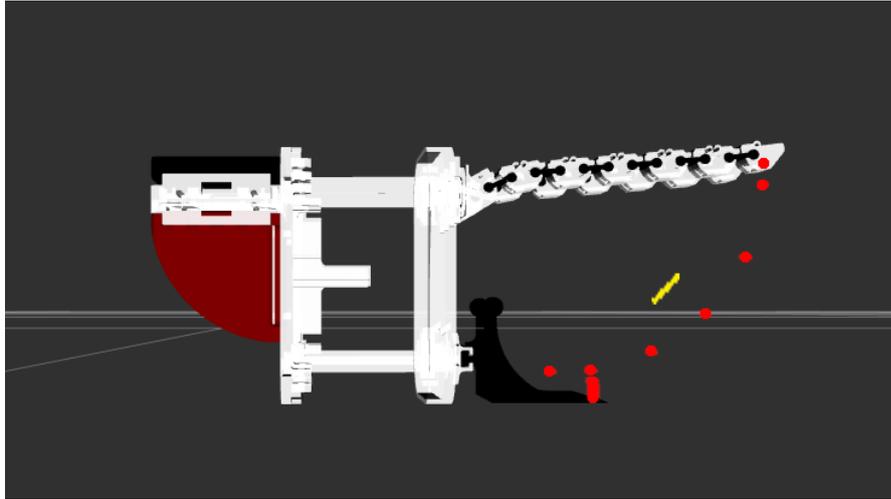
Once obtained, the matrix  $\mathbf{T}$  is a  $4 \times 4$  homogeneous matrix which can be written as follows:

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}$$

where  $\mathbf{A}$  is a  $3 \times 3$  matrix representing the linear map and  $\mathbf{b}$  is the vector representing the translation in the transformation. The matrix  $\mathbf{A}$  can be written using the polar decomposition as  $\mathbf{A} = \mathbf{UR}$  where  $\mathbf{U}$  is an Hermitian semi-positive matrix representing the non rigid deformation, while  $\mathbf{R}$  is orthogonal and contains the rigid rotation. Then, the following step is to estimate the eigenvectors of the matrix  $\mathbf{U}$ .

The eigenvector associated with the largest eigenvalue is the direction of the Closure Signature. Lastly, the centroid (application point) of the CS is estimated. It is calculated as the mean of all the points belonging to the closure trajectory.

The reference points were chosen depending on which parts of the hand play a fundamental role during grasping [8], as observed from the simulations described above. In Configurations 1 (Fig. 3.5) and 3 (Fig. 3.7), the two fingertips and the central point of the scoop edge were considered. In Configuration 2, depicted in Fig. 3.6, the trajectory of the fourth phalanxes of the fingers was considered.

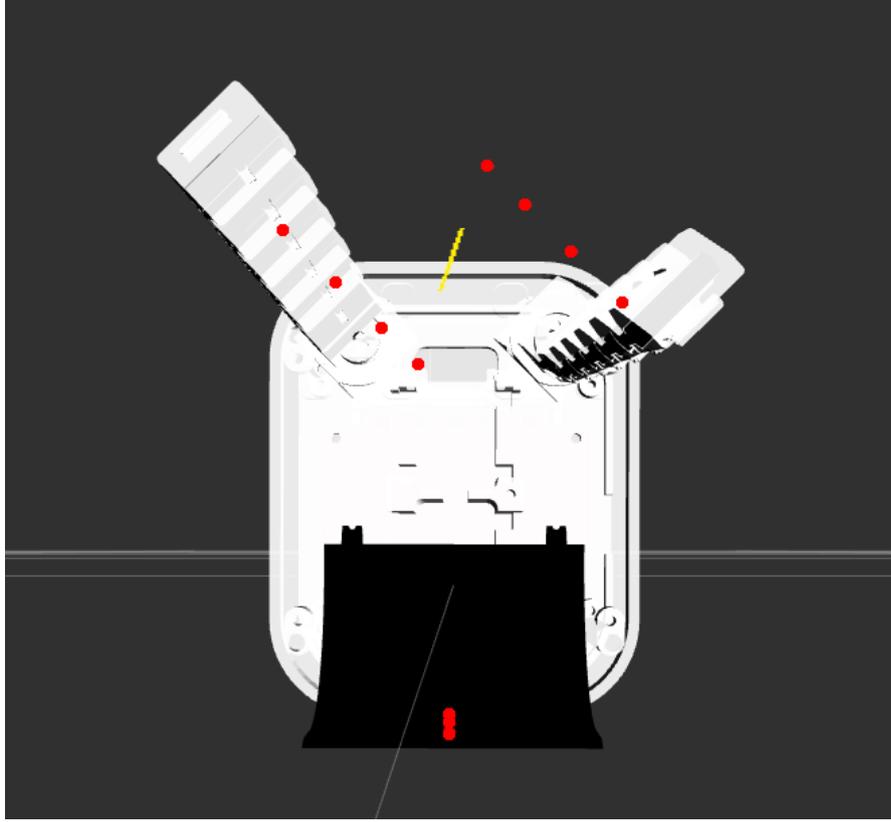


**Figure 3.5:** Closure Signature of Configuration 1

The information extracted from the CS was used for two purposes:

- To define a strategy to select the best fingers configuration according to the object to be grasped;
- To well align the SSG to the object.

Concerning the foregoing purpose, the approach presented in this work is based on geometric relations between the application point ( $o_h$ ) of the CS and the object's center



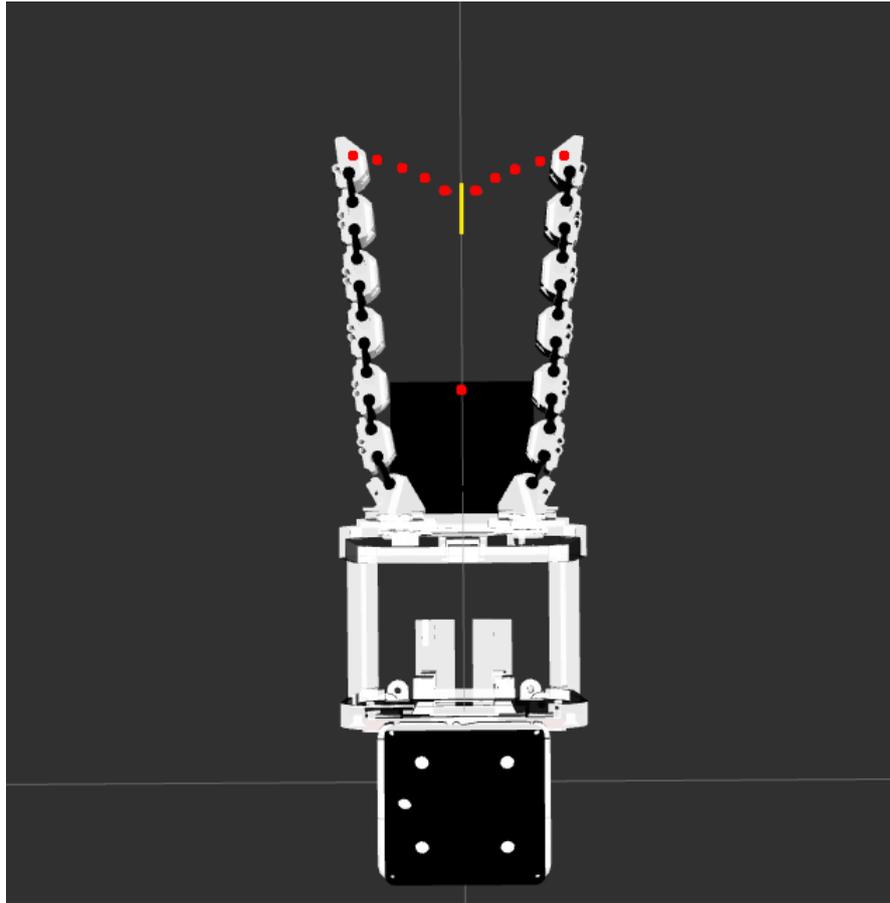
**Figure 3.6:** Closure Signature of Configuration 2

of mass (CM). In particular, the distance between  $o_h$  and CM was considered and used as an index of grasp quality:

$$Q_{DC} = D(CM, o_h) \quad (3.2.5)$$

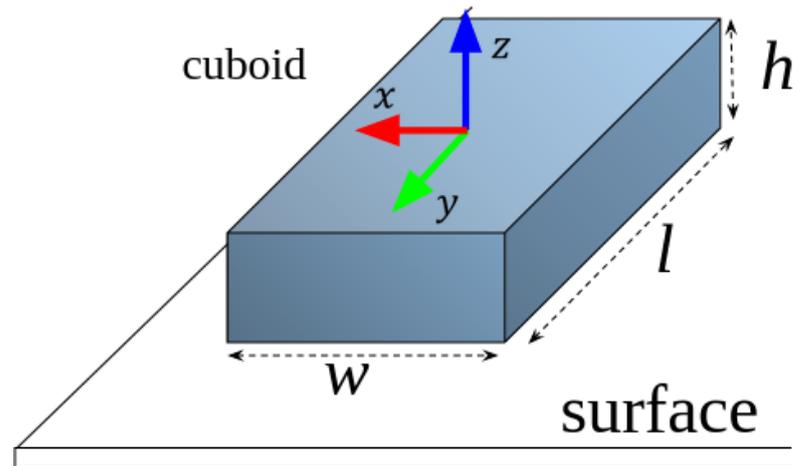
This is similar to what was done in previous works considering the distance between the centroid of the contact polyhedron and the object CM [6]. The closer the two points are, the more the effect of inertial and gravitational forces on the grasp is minimized [23], [24]. Index  $Q_{DC}$  has a simple physical interpretation, but it might be difficult to compute it in case the object CM is not known. To overcome this issue, it was assumed to have objects with homogeneous density, and the position of CM was approximated at the center of each of them.

Let us consider the frame  $\{H\}$  centered in the hand palm and let us denote with  $o_{h_1}$ ,  $o_{h_2}$ , and  $o_{h_3}$  the CS application points expressed in  $\{H\}$  belonging to the three fingers' configurations, respectively. Consider, for instance, a simple cuboid object, that could be the result of the approximation of an object with a bounding box procedure. Let us indicate the object sides with  $w$ ,  $l$ , and  $h$ , where  $h$  is the side normal to the surface,  $w$



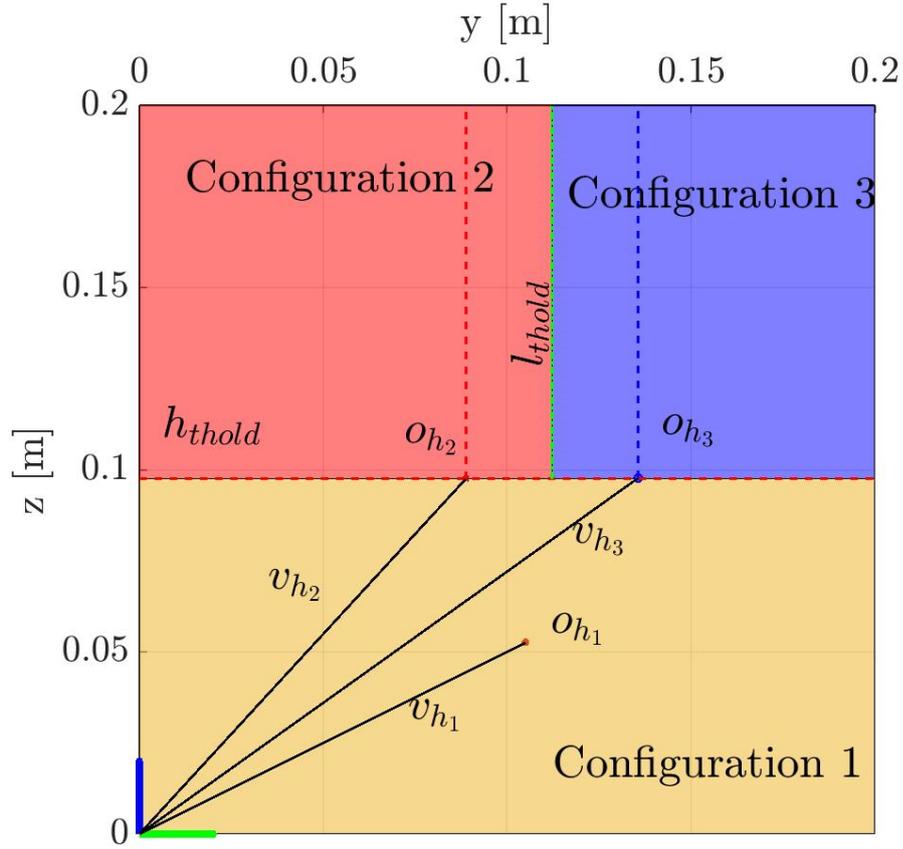
**Figure 3.7:** Closure Signature of Configuration 3

the shortest side on the surface plane and  $l$  the longest one, as shown in Fig. 3.8.



**Figure 3.8:** Sides of the bounding box of an object laying on the table surface.  $h$  is the side normal to the surface,  $l$  the longest side on the surface plane and  $w$  the shortest.

Note that the values of  $w$ ,  $l$ , and  $h$  change according to the object orientation on the



**Figure 3.9:** Selection criteria. The threshold  $h_{thold}$  is about 9.7 cm, while  $l_{thold}$  measures around 11.6 cm

table. The selection criteria are based on comparing the dimensions of the object to be grasped with the components of  $o_{h_1}$ ,  $o_{h_2}$ , and  $o_{h_3}$ . The object height allows to discriminate the first configuration from the other two.

Consider the  $z$  components of the centroids  $o_{h_2}$  and  $o_{h_3}$  which in this case coincide in  $h_{thold}$ . The criteria establish that objects whose height  $h$  is smaller than  $h_{thold}$  have to be grasped with Configuration 1. In other words, Configurations 2 and 3 are more tailored for tall objects, as, due to the physical limitations of the hand, it is not possible to grasp an object taller than a certain value with Configuration 1. When Configuration 1 has to be discarded, the threshold to decide between Configurations 2 and 3 is set in the midpoint between the  $y$  components of  $o_{h_2}$  and  $o_{h_3}$ . Fig. 3.9 summarizes the selection criteria: Configuration 1 was used for objects shorter than ( $h_{thold} = 9.7$  cm); vice versa, Configuration 2 or 3 were the eligible ones. Tall objects were then classified looking at their length ( $l$ ): if it exceeded the predetermined threshold ( $l_{thold} = 11.6$  cm), Configuration 3 was used, otherwise Configuration 2. These criteria ensure that among

the configurations that lead to a feasible grasp, the chosen one has the centroid  $o_h$  closer to the object CM. The information extracted from the Closure Signature are listed in the following Table 3.3 for each configuration.

Closure Signature		
	$v_{CS}$	$o_h$
Configuration 1	[0.0467;-0.9163;-0.3978]	[0.026;0.1;0.05]
Configuration 2	[0.1962;-0.5248;-0.8283]	[0.034;0.09;0.097]
Configuration 3	[0;-1;-0.0213]	[0.036;0.135;0.097]

**Table 3.3:** Closure Signature's properties

All the characteristics mentioned above were then translated in coding for the control algorithms necessary to perform the grasp. This part will be better explained in Chapter 3.3.

### 3.3 Grasp Planner

The goal of the experiment is to grasp different objects using the scoop-grasp technique. This was obtained using:

- Vision algorithm: it recognizes the object on the workspace and assigns a bounding box to the object with its pose and its dimensions
- Trajectory planner: it generates a trajectory to move the robotic arm from the initial pose to a final pre-grasp pose
- Hybrid Force-Velocity control: it permits the robotic arm to meet a surface and to slide on it keeping a desired force and velocity
- Hand control: it allows the hand to grasp the objects with the proper torque without damaging the motors and the objects

Some of the algorithms and the controls (trajectory planner, hybrid control) were developed considering that three different configurations of the hand exist, as already explained in Chapter 3.1.

### 3.3.1 Vision Algorithm

The first step consists in detecting objects placed over a surface like a table and in drawing a bounding box around them. The detection is achieved using a library for ROS called *tabletop* [25]. The sensor data used consists of a depth point cloud generated by the Kinect One camera. Then, the algorithm uses cluster segmentation:

- a surface is detected by finding the dominant plane in the point cloud using RANSAC (RANdom SAMple Consensus);
- points above the surface are considered to belong to graspable objects. A clustering algorithm is used to identify individual objects. The points corresponding to a single object are referred to as clusters.

Using the clusters obtained by the tabletop algorithm, a bounding box is created, taking the edges of the clusters belonging to the same object through the use of the *PCL* library [26]. A bounding box (or bounding volume) for a set of objects is a closed volume that completely contains the union of the objects in the set. Bounding volumes are used to improve the efficiency of geometrical operations by using simple volumes to include more complex objects. In many applications, the bounding box is aligned with the axes of the coordinate system, and it is then known as an axis-aligned bounding box (AABB). To distinguish the general case from an AABB, an arbitrary bounding box is sometimes called an oriented bounding box (OBB), or an OOBB when an existing object's local coordinate system is used. AABBs are much simpler to test for intersection than OBBs but have the disadvantage that, when the model is rotated, they cannot be simply rotated with it, but need to be recomputed. From now on, the term bounding box will refer to an OBB since it is the type used. These bounding boxes were oriented such that two faces are parallel to the table plane. The algorithm was developed considering that the bounding box should be created even if the object is suspended on the table surface. In this way, the condition of a successful grasp can be discerned, looking at the object pose.

Moreover, a Statistical Outlier Removal Filter was applied to the PCL Point Cloud to improve the quality of the bounding boxes. Their pose and their dimensions define the bounding boxes. Both are described with respect to the reference frame of the camera. However, it is needed that the pose and the dimensions of the bounding box are defined

in the frame of the robotic arm to be useful for the trajectory planner. In order to transform the reference frame, a set of fiducial markers was placed on the workspace. This set was located at a known distance from the base of the robot such that the transform from this reference to the one at the base was easily obtainable. In this way, the final reference frame was obtained using the two following transforms:

1. From the camera reference frame to the markers reference system:

- Linear

$$x = 0.942784375934$$

$$y = -0.177612673106$$

$$z = 0.705886756056$$

- Angular (in quaternion)

$$x = 0.69519360973$$

$$y = 0.483767117401$$

$$z = -0.332356035532$$

$$w = -0.415415098033$$

2. Transformation from the markers reference frame to the robot base frame:

- Linear

$$x = -0.478$$

$$y = 0.014$$

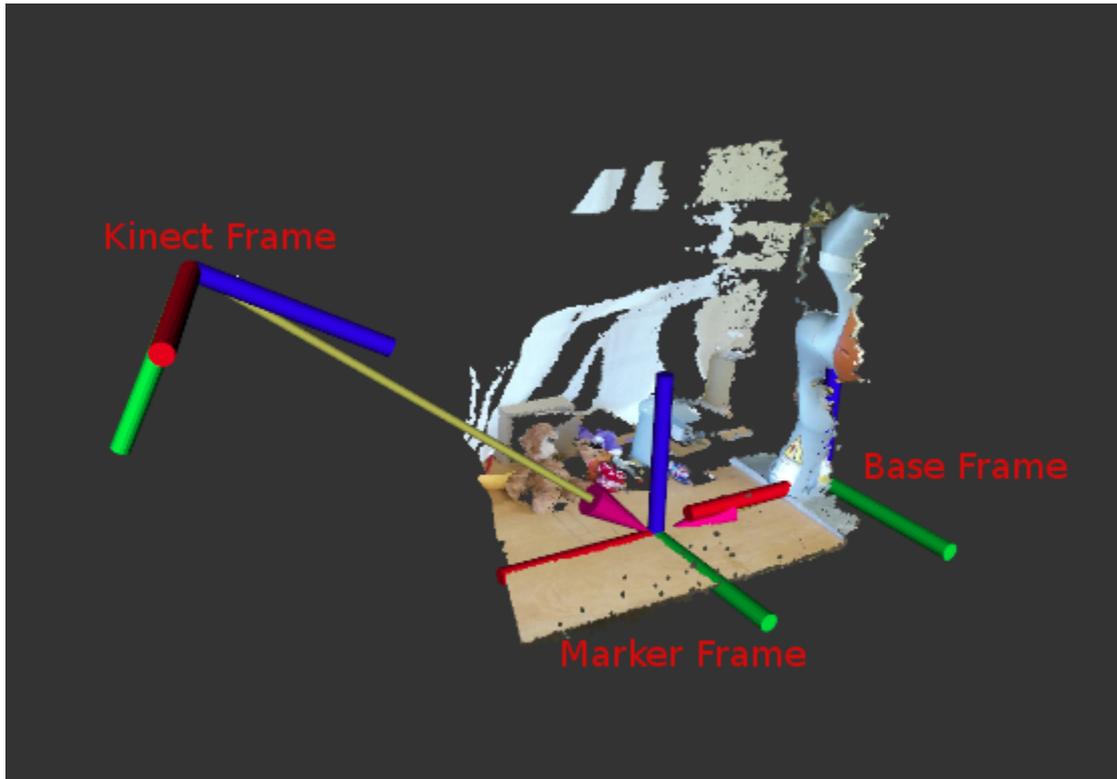
$$z = 0.045$$

- There is no angular component in the transform

The three reference frames can be seen in Fig. 3.10.

Then, a coordinate frame was placed at the center of the bounding box, such that:

1. The basis vectors are aligned with the edges of the bounding box
2. The z-axis is always perpendicular to the surface of the workspace. This means that the z-axis of the bounding box is the one that results the highest dot product with the normal to the surface. The pseudo-algorithm can be:



**Figure 3.10:** Scene with Kinect, Marker and Base Frame

```

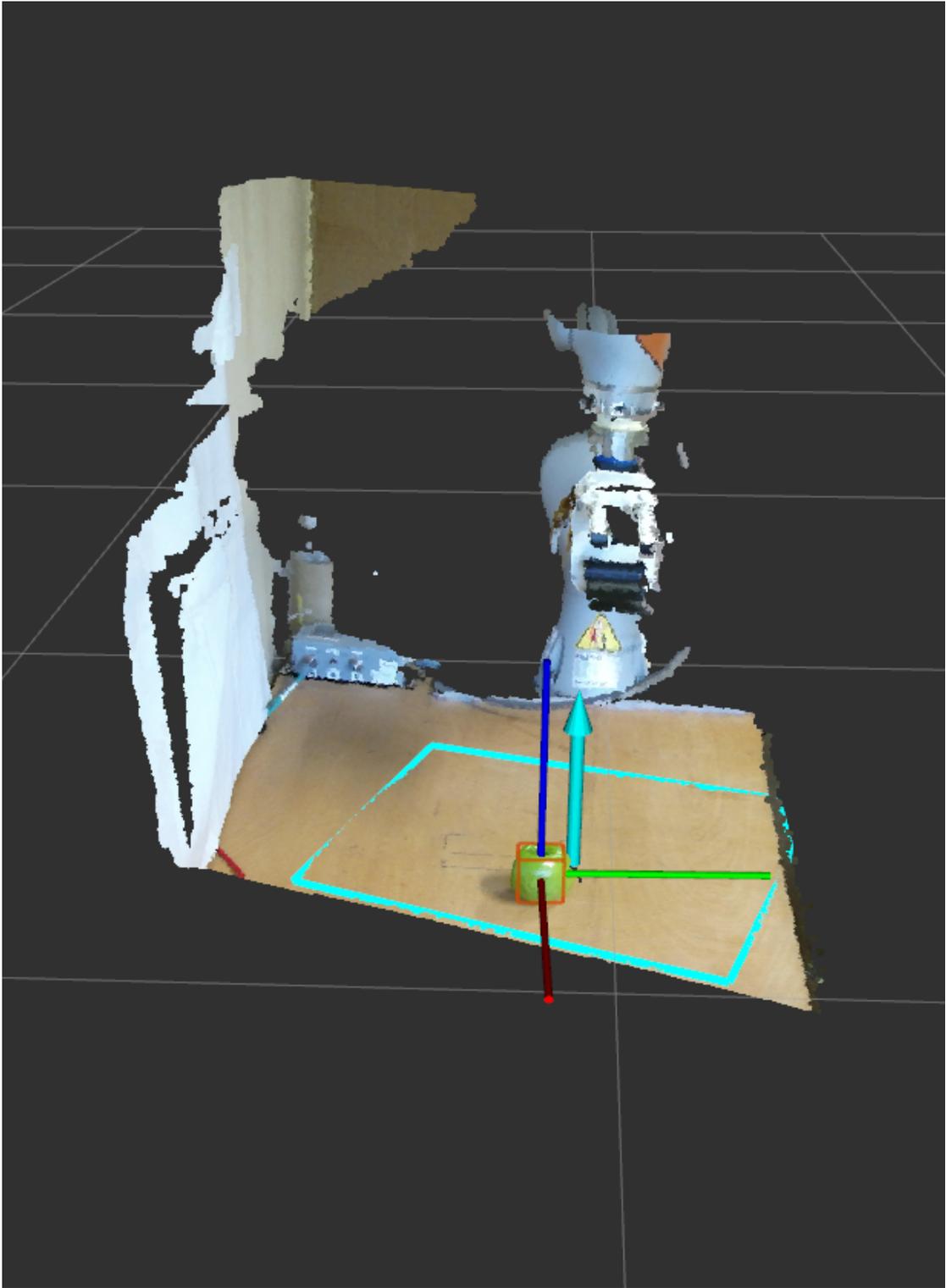
for (axis in axes)
    dot product = table normal.dot(axis)
    if (dot product > max dot product)
        z axis = axis

```

3. The x-axis should be parallel to the shortest side of the bounding box plane belonging to the surface, while the y-axis should be on the longest side. After having obtained the x-axis, the y-axis direction is calculated by the cross-product of the other two axes to keep the right-hand rule  $Y = Z \times X$ . If the difference between two edges is inferior to a certain threshold, the x-axis and the y-axis are predefined, avoiding uncertainties.

The obtained result is shown in Fig. 3.11, where the pose and the dimensions of the bounding box (in cm) are

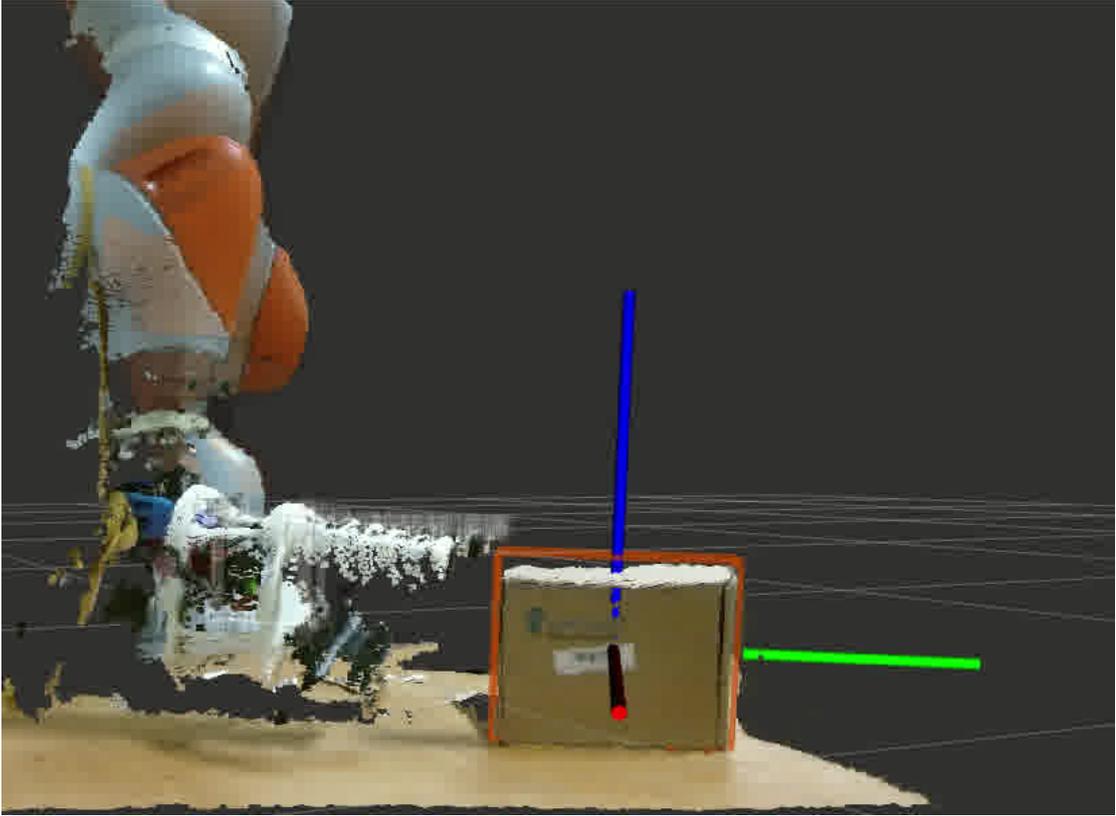
$$\begin{aligned}
 x &= 0.73 & y &= -0.09 & z &= 0.01 \\
 x &= -0.01 & y &= -0.01 & z &= 0.08 & w &= 0.99 \\
 l_x &= 0.05 & l_y &= 0.06 & l_z &= 0.06
 \end{aligned}$$



**Figure 3.11:** Workspace with the bounding box and the table array

Therefore, the *Realtime URDF Filter* [27] is applied to the registered point cloud. This filter takes as input a URDF (Unified Robot Description Format) and erases it from the scene. This is useful to not confuse the end-effector with an object positioned on the

workspace when the robotic arm is close to the surface, as can be seen in Fig. 3.12.



**Figure 3.12:** Scene with the Realtime URDF filter applied

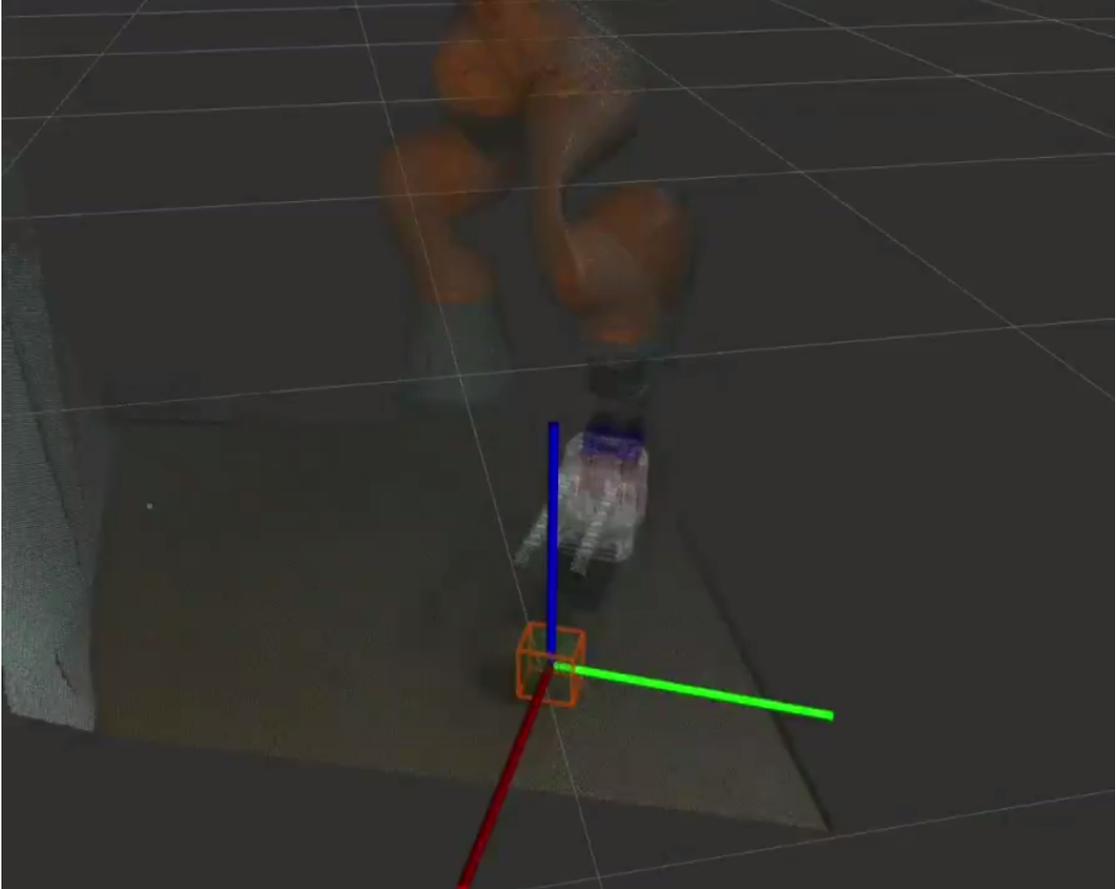
### 3.3.2 Trajectory Planner

In literature, a trajectory planner is defined as creating motion for an end-effector from one point to another while avoiding collisions. The movement that the end-effector must perform is from an initial pose, corresponding to the joint position (in radians):

$$q_0 = 0.0 \quad q_1 = 0.0 \quad q_2 = 0.0 \quad q_3 = -1.69 \quad q_4 = 0.0 \quad q_5 = 1.46 \quad q_6 = 0.76$$

to the pre-grasp pose, which depends on where the object is placed. As already said in the introduction of this chapter, there are three different fingers' configurations. These configurations were planned to widen the set of objects graspable from the SSG. Their application was studied on MATLAB through the analysis of the Closure Signature. Before the trajectory planning starts, an algorithm suggests which is the best fingers' configuration to grab an object depending on its dimension. The key features are the height and length of the object.

- If the height of the object is less than 9.7 cm, Configuration 1 must be used. This threshold is the projection of  $o_{h_2}$  on the  $z$ -axis, and it was extracted from the study of the Closure Signature (explained in Chapter 3.2).
- If the length of the object is lower than 11.6 cm, Configuration 2 should be used. In this case, the threshold depends on the application point of the Closure Signature, as mentioned above.



**Figure 3.13:** Pre-grasp pose aligned with x-axis of the object

Based on the chosen configuration, the end-effector approaches the object from a different pre-grasp pose. The possible cases are the following:

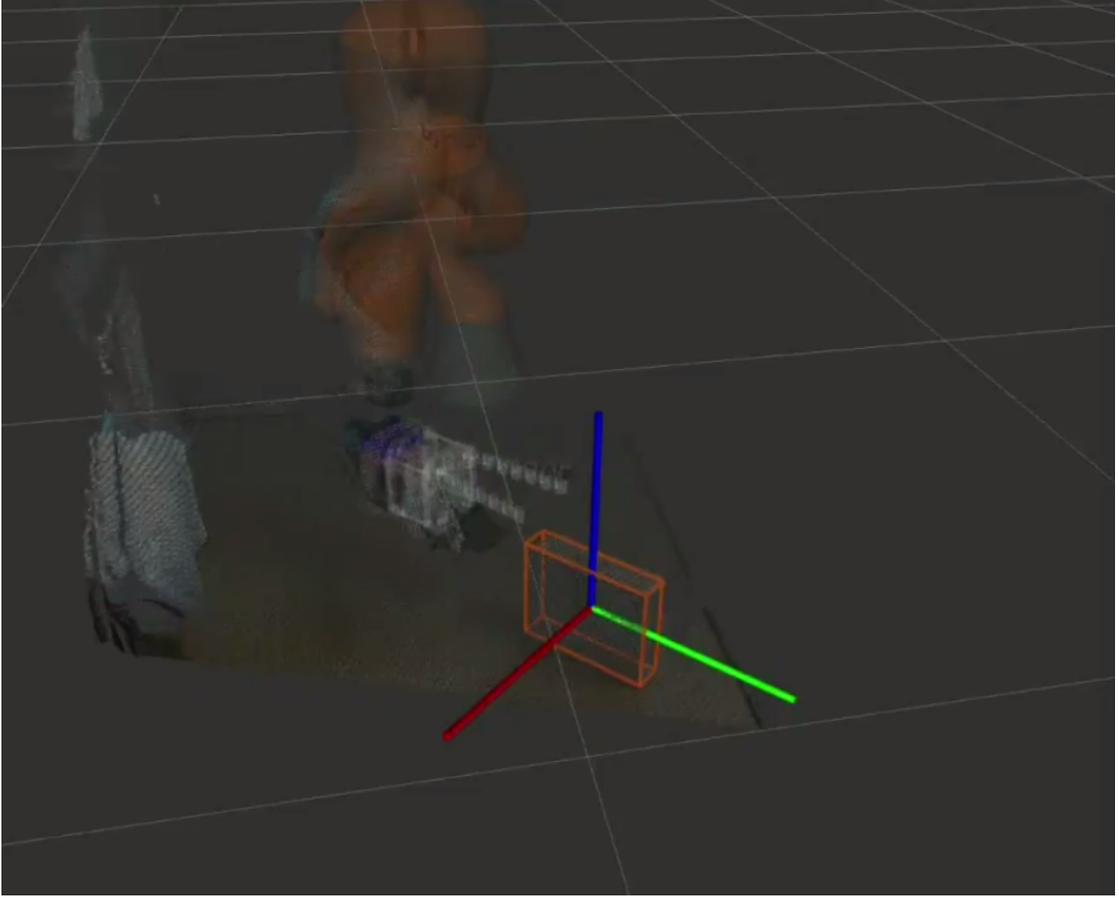
- Pre-grasp pose along the x-axis of the object: this strategy is used when Configuration 1 is suggested. It consists in placing the hand in the pose of the object applying an offset of

$$offset = [-(x_{box}/2 + 0.1) \quad 0.0 \quad 0.05] \quad (3.3.1)$$

and a transformation

$$R = R_z(-\pi/2)R_x(-\pi/2) \quad (3.3.2)$$

In this way the scoop of the gripper is aligned with the x-axis of the object as shown in Fig. 3.13.



**Figure 3.14:** Pre-grasp pose aligned with y-axis of the object

- Pre-grasp pose along the y-axis of the object: this strategy is used when Configuration 2 or Configuration 3 are suggested. It consists in placing the hand in the pose of the object applying an offset of

$$offset = [0.0 \quad -(y_{box}/2 + 0.1) \quad 0.05] \quad (3.3.3)$$

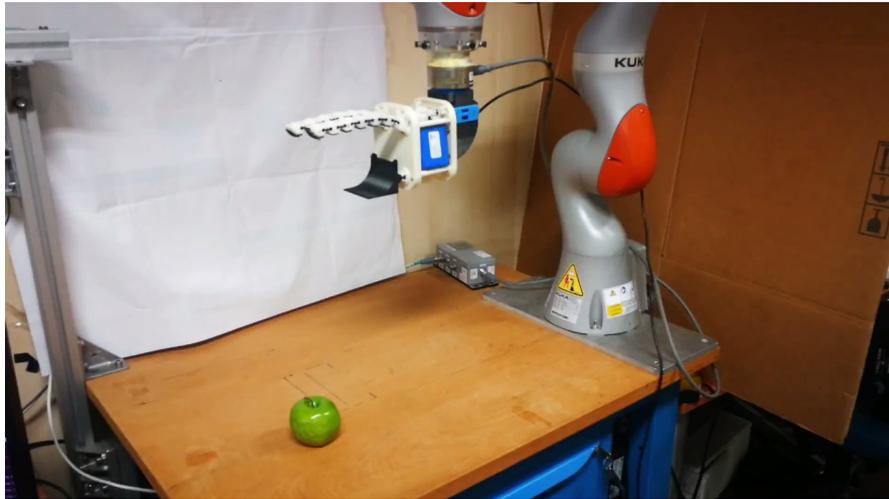
and a transformation

$$R = R_x(-\pi/2) \quad (3.3.4)$$

In this way the scoop of the gripper is aligned with the y-axis of the object as shown in Fig. 3.14.

The transformations and the offsets, mentioned before in (3.3.1),(3.3.2),(3.3.3),(3.3.4), were referred to the object's frame and then transformed into the robot base frame. Besides, these alignments were chosen to minimize the distance between the object's center of mass and  $o_h$  of CS for each case.

Once the gripper configuration has been settled, the trajectory planner starts. For this work, a robotics manipulation platform was used, called *MoveIt* [28]. It incorporates some planner libraries like the Open Motion Planning Library (OMPL [29]) or the Stochastic Trajectory Optimization for Motion Planning (STOMP). The library used is OMPL.



(a)



(b)

**Figure 3.15:** Initial (a) and final (b) pose of KUKA LBR iiwa

It consists of many state-of-the-art sampling-based motion planning algorithms. From these many, there is a class of planners called Single-query planners: these planners typically grow a tree of states connected by valid motions. They differ in the heuristics they use to control where and how the tree is expanded. Some tree-based planners grow two trees: one from the start and one from the goal. Such planners will attempt to connect a state in the start tree with another state in the goal tree, as in the case of the Rapidly-exploring Random Trees Connect (RRTconnect) [30].

This algorithm was preferred over the others because it tends to outperform them when the problem is awkward.

To improve the capability of the program, if the pre-grasp pose is not reachable by the robotic arm, the trajectory planner tries to plan another trajectory from the opposite direction of the axis to be approached. Definitely, all the successive steps were developed in a modular way always to approach the object correctly.

Besides, the trajectory planner is applied at the end of the experiment. After having grasped the object, the end-effector is moved from the actual pose, which depends on the current object's pose, to the following final pose in the joint space:

$$q_0 = 0.0 \quad q_1 = -0.34 \quad q_2 = 0.0 \quad q_3 = -1.97 \quad q_4 = 0.0 \quad q_5 = 1.0 \quad q_6 = 0.76$$

This last motion of the robot was implemented to demonstrate that the grasp is robust if the gripper can hold an object while some motion is being performed.

The initial and the final pose of the robotic arm can be seen in Fig. 3.15.

### 3.3.3 Hybrid Force-Velocity Control

The hybrid force-velocity controller acts when the end-effector has reached the pre-grasp pose until the object is approached. Once the SSG is aligned with the axis of the object to be contacted, the end-effector start to descend until it touches the surface of the workspace.

To avoid possible breakdown and to be sure that the scoop is lying on the surface, a controller over the force exerted by the robotic arm was necessary. Then, the end-effector starts to slide towards the object, keeping a constant force against the surface and velocity directed along with the object. As shown in the Appendix, an ATI sensor is placed on the wrist of the robotic arm, and the KUKA LBR iiwa itself is endowed

with torque sensors on each joint.

The force applied on the surface is  $F_n = 1N$ . Hence, it was necessary to estimate the force applied to the end-effector. A small amount of the vertical force applied becomes a torque due to the geometry of the adapter for the Soft ScoopGripper. Since this torque was negligible, the forces and torques applied on the end-effector are obtained as the difference between the force/torque sensor data and the inertial forces and torques of the robotic arm:

$$ft_{ext} = ft_{sensor} - ft_{inertial} \quad (3.3.5)$$

This difference is to be intended as the difference of every component (force:  $f_x, f_y, f_z$ , and torque:  $\tau_x, \tau_y, \tau_z$ ) of the two wrenches. The inertial forces and torques are estimated, starting from the acceleration and the velocity of the end-effector:

$$a_{ee} = \dot{J}\dot{q} + J\ddot{q} + a_{grav} \quad (3.3.6)$$

$$v_{ee} = J\dot{q} \quad (3.3.7)$$

where  $a_{grav} = [0 \ 0 \ 9.8 \ 0 \ 0 \ 0]$ .

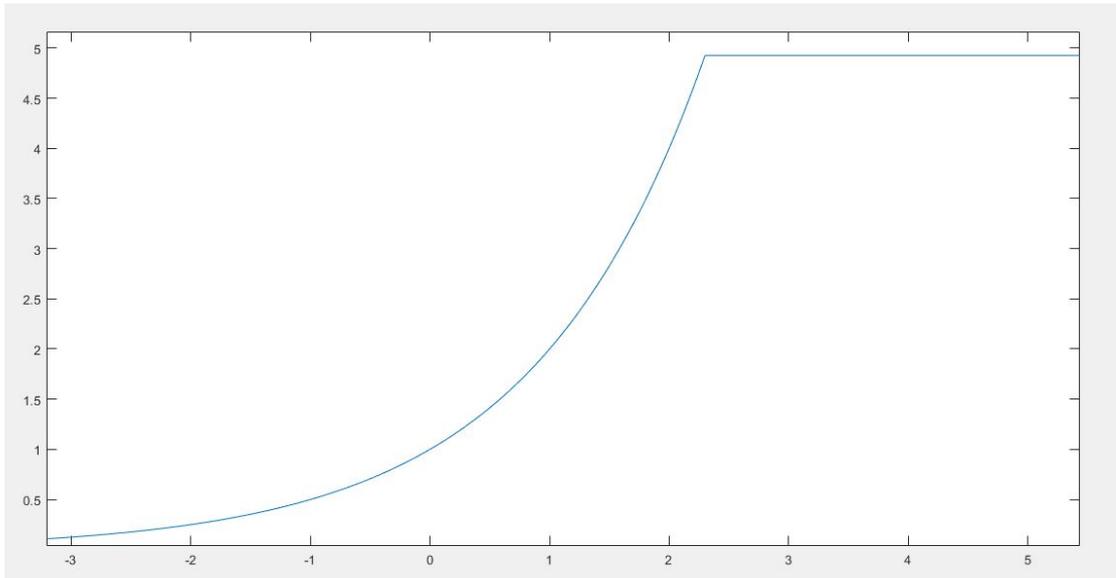
Then, the rotation matrix  $T$  from the base of the robotic arm and its end-effector is calculated using the Forward Kinematics. As  $T_{inv}$  is the inverse of the rotation matrix  $T$  calculated before, the inertial forces and torques are:

$$ft_{inertial} = inertia_{tool}(T_{inv}a_{ee}) + (T_{inv}v_{ee})(inertia_{tool}(T_{inv}v_{ee})) \quad (3.3.8)$$

where  $inertia_{tool}$  is the rotational inertia of the Soft ScoopGripper expressed in the end-effector reference frame.

After obtaining the force perceived on the end-effector, a function to adapt the descending velocity of the end-effector has been implemented. As the velocity is related to the force, the motion of the end-effector takes too much time if the same force is applied during the entire descent. The developed function consists of adapting the force depending on the position of the end-effector with respect to the distance on the table surface. The function should have the following characteristics:

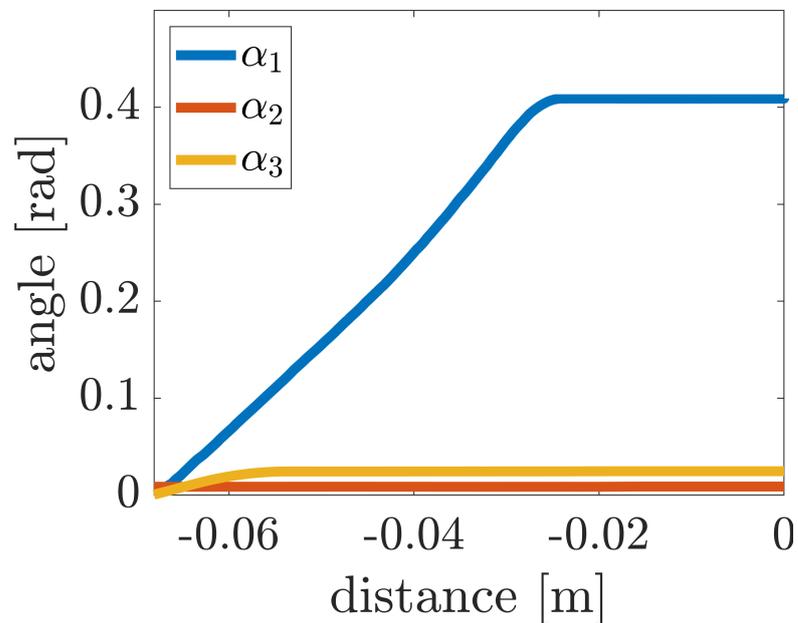
1. At the position of  $z = 0$ , the force should be  $1N$
2. The maximum applied force should be around  $5N$



**Figure 3.16:** Force function depending on the height of the end-effector

The function used is  $f = 2^h$  with saturation if  $h > 2.3$ , as shown in Fig. 3.16.

Once the surface has been reached, the end-effector starts moving towards the object, keeping the same force against the surface. In this case, the velocity applied is no more directed along the surface normal but along the approaching axis of the object. The controller moves the end-effector until it meets the object.

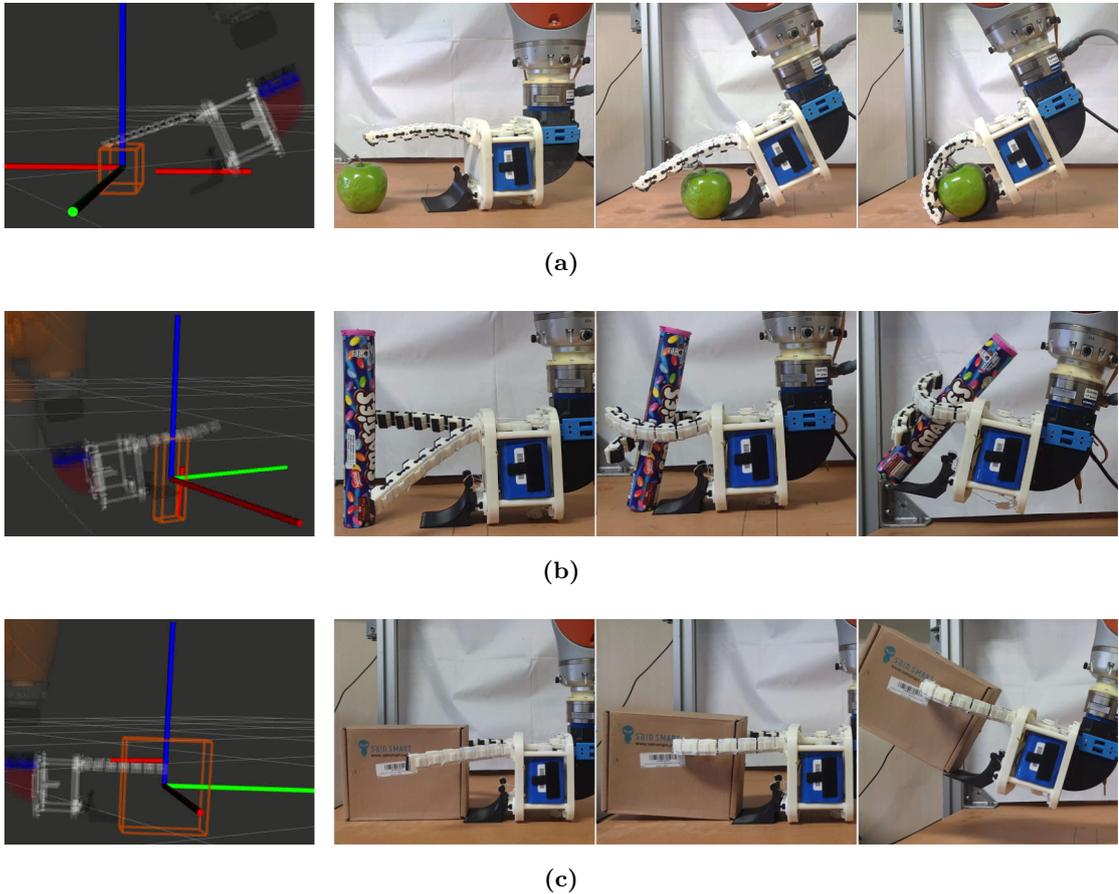


**Figure 3.17:** Angle followed by the end-effector to reach the CS inclination for each configuration

Then, the sliding continues for two seconds in order to facilitate the grasp of the object. The velocity applied to the end-effector is directly proportional to the applied force.

During this motion, angular velocity is applied to the end-effector. This velocity is meant to position the CS of the SSG with a certain angle concerning an axis of the object, as can be seen in Fig. 3.17. This angle depends on the configuration of the hand, and it corresponds to the angle needed to align the Closure Signature of the hand with one of the axes of the object.

For the sake of clarity, let us refer to the cuboid mentioned above (Fig. 3.8) to explain the alignment for each configuration. When the scoop is approaching objects whose height  $h$  is less than  $h_{thold}$  (Configuration 1), the CS is aligned with  $w$ . This is because the center of mass of the object should lay on the scoop once it is grabbed.



**Figure 3.18:** Alignment of the CS with each configuration

For tall and wide objects (Configuration 3), instead, the CS is aligned with  $l$ . In this way, the fingers can wrap the object pulling it towards the hand palm.

Lastly, tall and narrow objects are approached (Configuration 2), aligning the CS to their height  $h$ .

Independently of the selected CS, the application point  $o_h$  was always placed as close as possible to the object's center of mass to minimize Eq. (3.2.5). Such alignment can be seen in Fig. 3.18.

It is necessary to find a relationship between the angular and the linear motion of the end-effector to obtain the angular velocity needed to reach the desired angle. The solution found in the following:

$$\omega = \frac{v\alpha}{d} \quad (3.3.9)$$

where  $d$  is the distance on the x-y plane between the pose of the hand and the pose of the bounding box, and the angle is given from the a-priori analysis on the Closure Signature. Moreover, to lower the possibility of errors, the angle has been put on feedback.

The complete sequence of the hybrid controller can be seen in Fig. 3.18a, which is the case where the end-effector needs to be inclined the most.

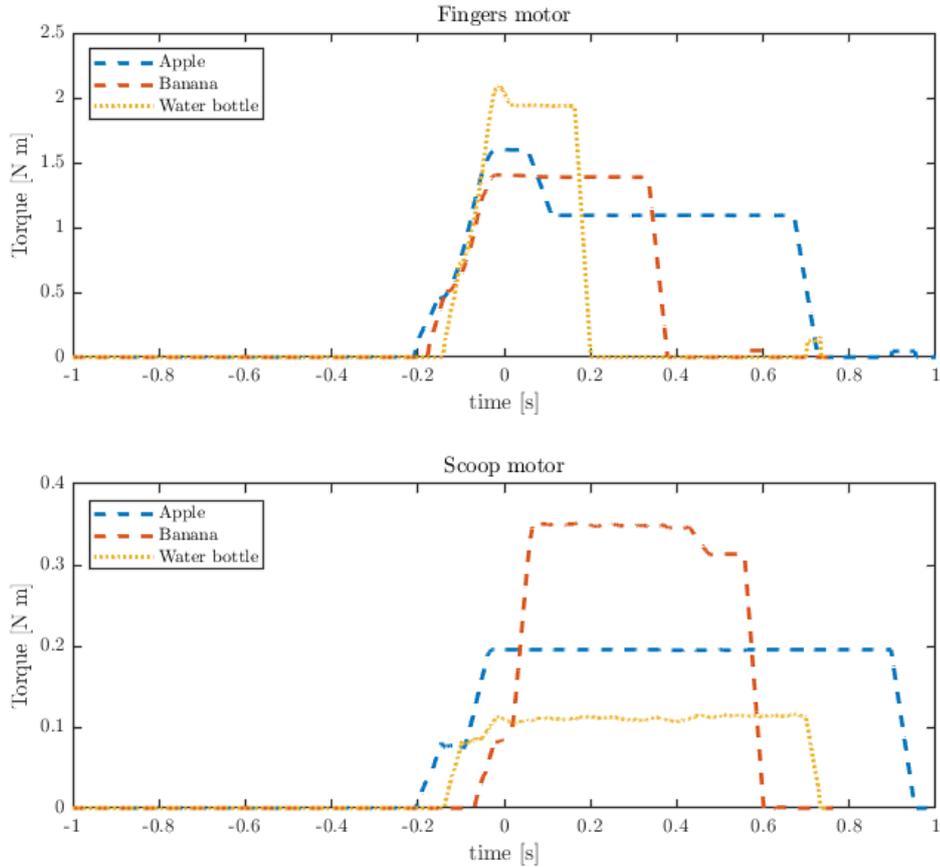
### 3.3.4 Hand Control

The control of the hand starts working as soon as the object has been approached, while the robotic arm is slowly dragging the object. The grasp begins even if the end-effector is still moving in order to help the fingers to pull the object on the scoop.

The control of the hand was developed in the Arduino environment because the Arbotix-M Robocontroller is an Arduino compatible microcontroller. For this purpose, it was used an Arduino library called *ax12* that is necessary to interface the motors to Arduino. The main variables are used to define the maximum torque, the torques, the velocities, and the positions of the engines. This algorithm was interfaced with ROS using the library *rosserial* [31]. This library allows the program on the microcontroller to work with the *ROS topic*, so the microcontroller can publish and subscribe to topics already present in the ROS network. In this case, the microcontroller subscribes to a topic that corresponds to the command of opening and closing the hands.

Meanwhile, the Arbotix-M publishes the torques of the motors on a topic, which is useful to know for multiple objects the torque necessary to perform the grasp. An example of the published torques of both the fingers and the scoop can be seen in Fig. 3.19. The

objects taken as an example are a fake apple, a fake banana, and a water bottle. The



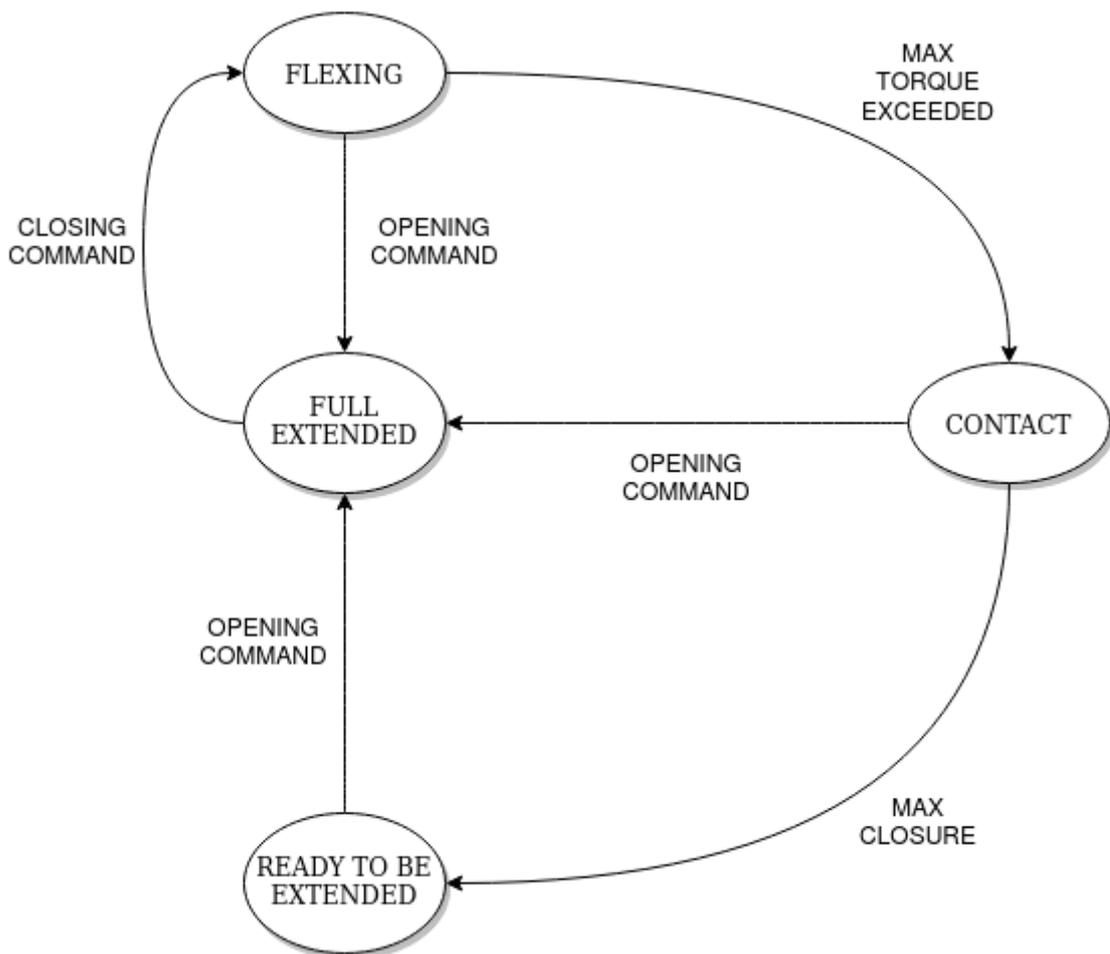
**Figure 3.19:** Fingers' and scoop's motor torque

algorithm written on the microcontroller works in the following way:

1. There is a main loop, waiting for receiving a command from the `/cmd-gripper` topic and continuously publishing the motor torques on the `/motor-status` topic. When the loop is executed, the positions and the torques of the motors are written in the respective variables. In the beginning, the state of the program is *Full Extended*.
2. If the program is in the state *Full Extended*, there has not been any contact of the fingers, and the closing command is received, the motor linked to the fingers starts to wind the tendons trying to reach the closing position. The state changes to *Flexing*.
3. If the program is in the state *Flexing* and the torque of the motor linked to the fingers exceeds a threshold, it means that the fingers met the object. The

threshold was estimated, looking at the value of the motor torque when no object was present. The state passes from *Flexing* to *Contact*.

4. If the program is in the state of *Contact*, the motor linked to the fingers start to increment the torque until the closing position is reached or the maximum torque is exceeded. After having reached the closing condition, the motor linked to the scoop starts to wind the tendons to increase the robustness of the grasp. The state changes in *Ready To Be Extended*, and the program waits in this state.
5. If the program is in any state and the command of the opening is received, both the motors start to rewind, completely opening the fingers and the scoop. The state changed again to *Full Extended*.

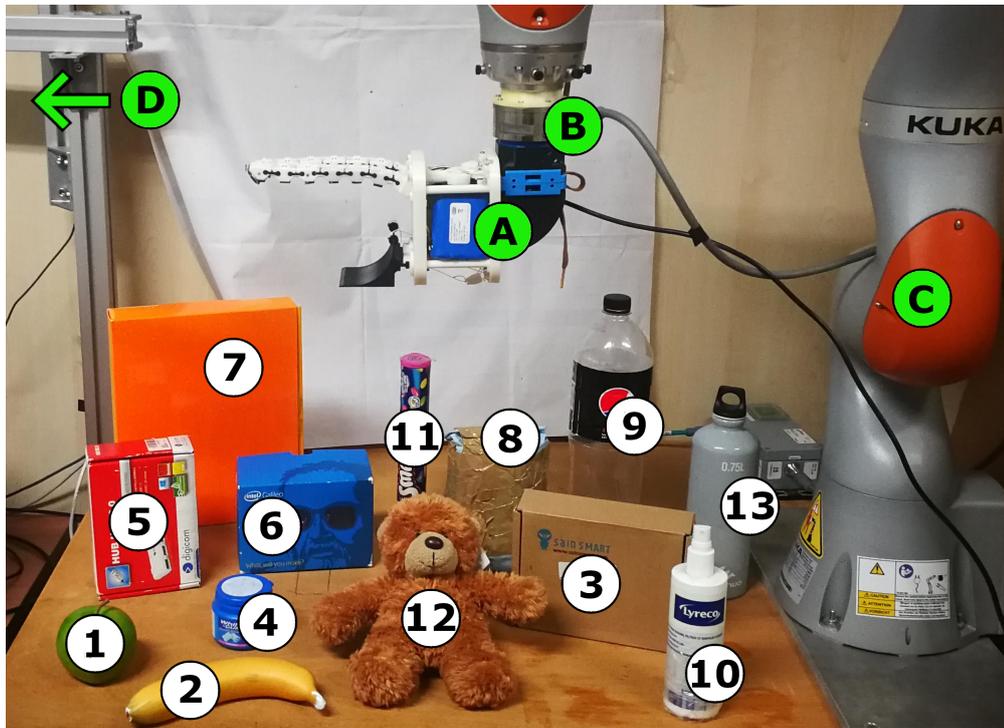


**Figure 3.20:** Finite State Machine of the Soft ScoopGripper

The diagram of the Finite State Machine describing the control of the Soft ScoopGripper is shown in Fig. 3.20.

# Results

## 4.1 Experimental Setup



**Figure 4.1:** Experimental setup and tested objects. Soft ScoopGripper (A), ATI F/T sensor (B), KUKA iiwa LBR (C), and Kinect One (D) are shown. Objects are indicated with IDs and their properties are reported in Table 4.1.

The experimental setup, as shown in Fig. 4.1, included a KUKA LBR LightWeight robot arm (KUKA iiwa 7 - LBR), an ATI Gamma 6-axis force-torque sensor and the Soft ScoopGripper attached to the end-effector. A Microsoft Kinect One RGB-D camera

was used to detect objects. All the hardware is detailed in Chapter A.

ID	Object	Weight (g)	$w \times l \times h$ (cm)	Conf.
1	Apple	60	$7.5 \times 7.5 \times 6.3$	1
2	Banana	48	$18.5 \times 3.5 \times 3.3$	1
3	Box	63	$14 \times 18 \times 5.1$	1, 3
4	CBox	300	$6.5 \times 6.5 \times 5.7$	1
5	Red Box	61	$5.7 \times 12.6 \times 15.9$	3
6	Blue Box	93	$7.4 \times 15.3 \times 12.5$	3
7	Orange Box	118	$4.4 \times 22.5 \times 28.5$	3
8	Pasta Pack	510	$6.5 \times 11.4 \times 18.2$	2
9	Plastic Bottle	43	$9 \times 9 \times 34.2$	2
10	Spray Bottle	94	$5 \times 5 \times 18.5$	2
11	Candy Tube	12	$3.6 \times 3.6 \times 22.8$	2
12	Teddy Bear	134	$20 \times 22 \times 17.3$	3
13	Water Bottle	127	$7.2 \times 23.7 \times 7.2$	1, 2

**Table 4.1:** Objects properties. The second last column reports the size of the bounding box of the object, indicating the values of  $w$ ,  $l$ , and  $h$  when the object was configured as in Fig. 4.1. The last column reports the configurations that were used to grasp the object. Some lines contain two numbers because the pose of the object was changed in order to test different configurations.

During the experiments, the camera was standing opposite to the base of the robot, aiming to get a top view of the workspace. In the first stage, the table is detected by finding the dominant plane in the scene. Hence, as explained in details in Chapter 3.3.1 a table-top object detector identifies and extracts a cluster of 3D points belonging to the object.

Each cluster on top of this plane was segmented and treated as a separate object. The bounding box for each object was obtained using the oriented bounding box method available in the Point Cloud Library (PCL) [26].

Comparing the size of the bounding box with the thresholds computed as described in Chapter 3.2, the configuration of the fingers were appropriately modified. The bounding boxes must be oriented such that one of the faces coincides with the table plane. Then,

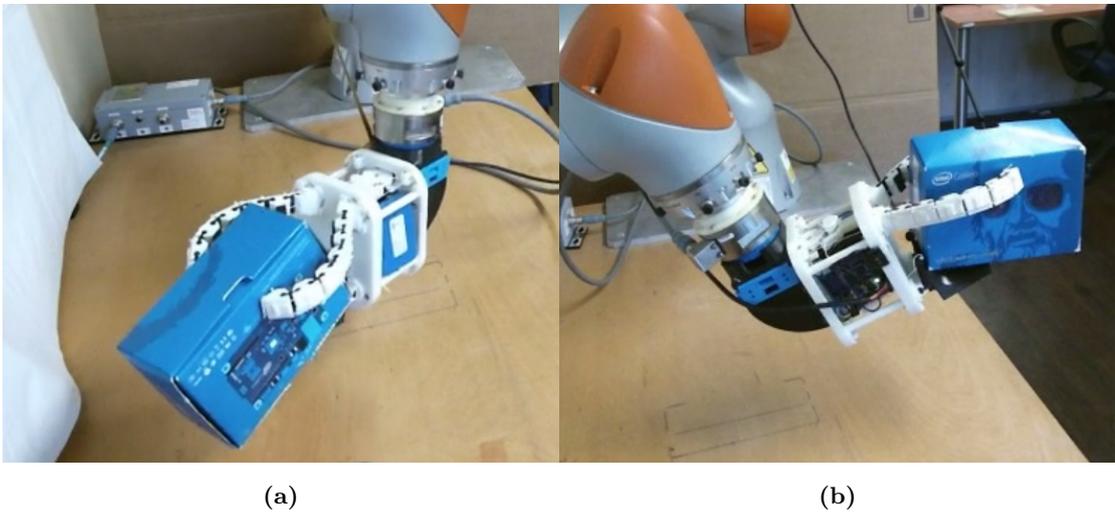
a local coordinate frame was placed at the center of the bounding box, such that the basis vectors are aligned with the edges of the bounding box. The  $z$ -axis is always perpendicular to the surface of the workspace. The  $x$ -axis is chosen such that it is parallel to the shortest side of the bounding box  $w$ , while the  $y$ -axis is coincident with the longest side  $l$ .

Table 4.1 summarizes the properties of each object and the adopted configuration.

Ten trials were made for each object using its related fingers setting. In the used SSG model, the rotation of the fingers is not motor-driven, and must be done manually. In general, this can introduce inaccuracies in positioning, but signs were put on the gripper to reproduce configurations with a low error.

## 4.2 Success Rate and Scoop Rate

Success rates for each configuration are reported in terms of the objects' features, sorted in ascending order. The selected properties are the weight of the object and the size of the object in the approach direction: width  $w$  for Condition 1, length  $l$  for the other two<sup>1</sup>.



**Figure 4.2:** Successful grasp without the use of the scoop (a) and exploiting the scoop (b)

<sup>1</sup>Note that the approach direction of the scoop is defined according to the CS and as shown in Fig. 3.18: for Configuration 1, the scoop moves along the  $x$ -axis of the object local frame, whereas for Configurations 2 and 3 the scoop moves in the  $y$ -axis.

To evaluate the performance of the proposed grasping system two metrics were considered: grasp success and scoop rate. A grasp is considered successful if the object is grasped and moved to the final position without falling at a velocity of about 10 cm/s. Otherwise, it will be unsuccessful. The scoop rate measures the level of usage of the scoop.

From empirical observations it could be noticed that grasps achieved exploiting the scoop appeared more robust, while if the scoop did not help the gripper to keep the object, the grasp was in many cases successful but not very robust (see for example the grasp in Fig. 3.18c, that mainly relies on the force exerted by the fingers). The scoop rate is calculated as:

$$SR = \frac{n_{scoop}}{n_{succ}} 100 \quad (4.2.1)$$

where  $n_{scoop}$  is the number of times the scoop was effectively used and  $n_{succ}$  the number of successes.

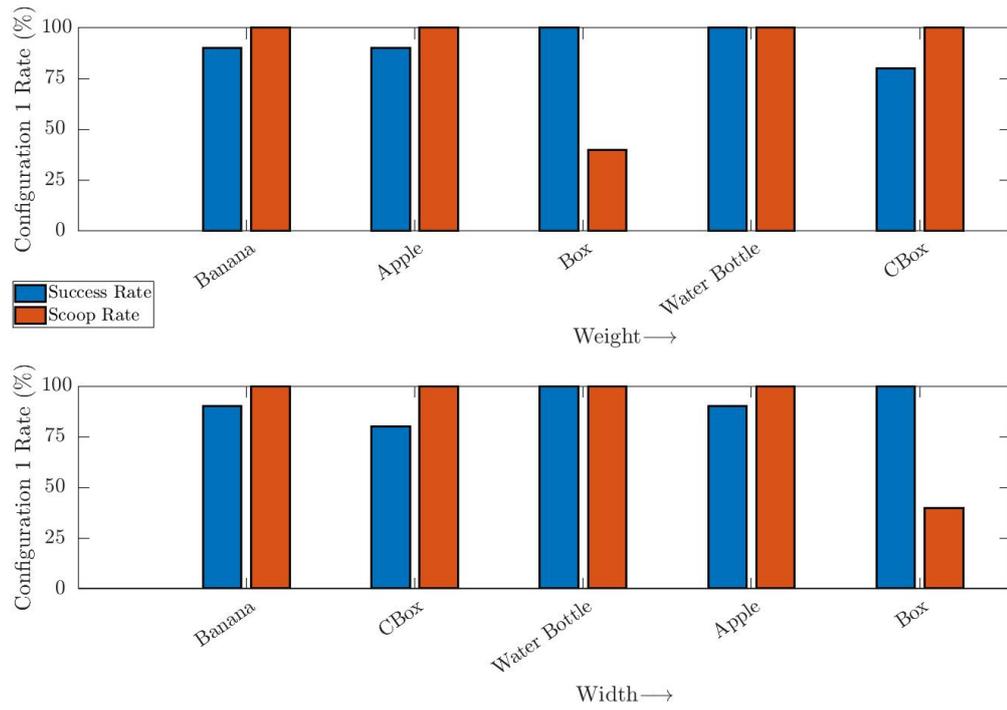
For the sake of clarity, example of grasps obtained with and without the exploitation of the scoop are shown in Fig. 4.2.

Five objects were tested per configuration. Water bottle and Box have been used for two configurations by changing their pose on the table.

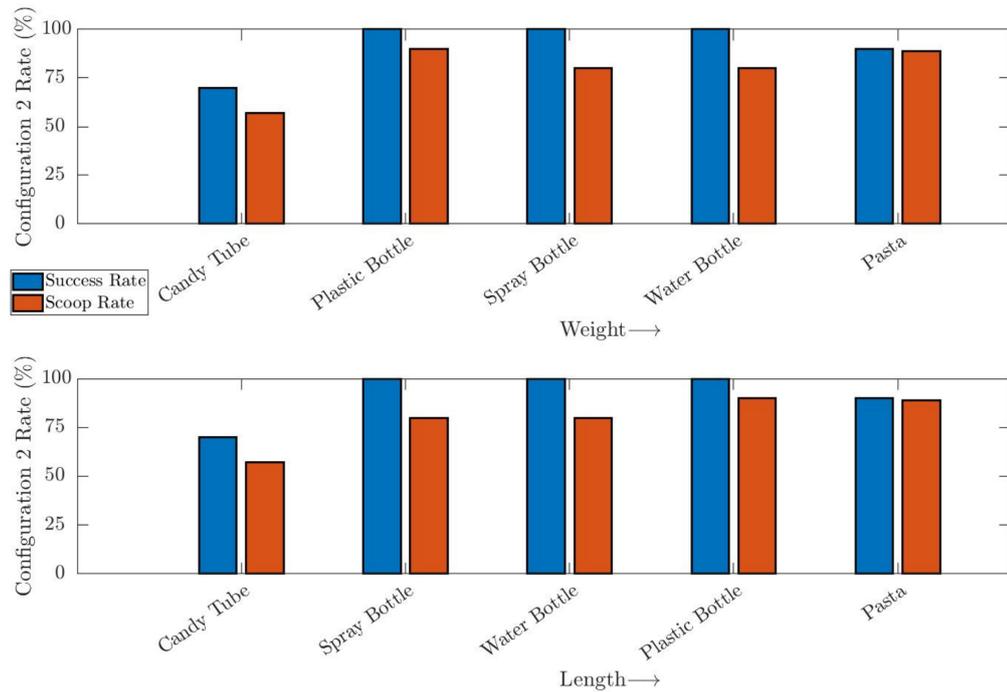
As depicted in Fig. 4.3, it can be noticed that the success of Configuration 1 does not depend on the weight of the object as there is not a clear trend in the metrics. In general, however, as the weight increases, inevitably, the success rate will tend to zero once the payload value supported by the scoop is exceeded. Another thing to notice is that large objects, such as the Box, are challenging to grab using Configuration 1. In fact, despite having been 100% successful, less than 50% of the time, the object was placed on the scoop.

Regarding Configuration 2, in Fig. 4.4, success and scoop rates seem to have a similar trend either for weight and length. Narrow and light objects (such as the Candy Tube) are less successful than objects similar in size but heavier (Spray Bottle). This is mainly due to the misalignment of the fingers producing a moment experienced by the object: the lighter it is, the easier it will tend to be rotated missing the scoop's surface.

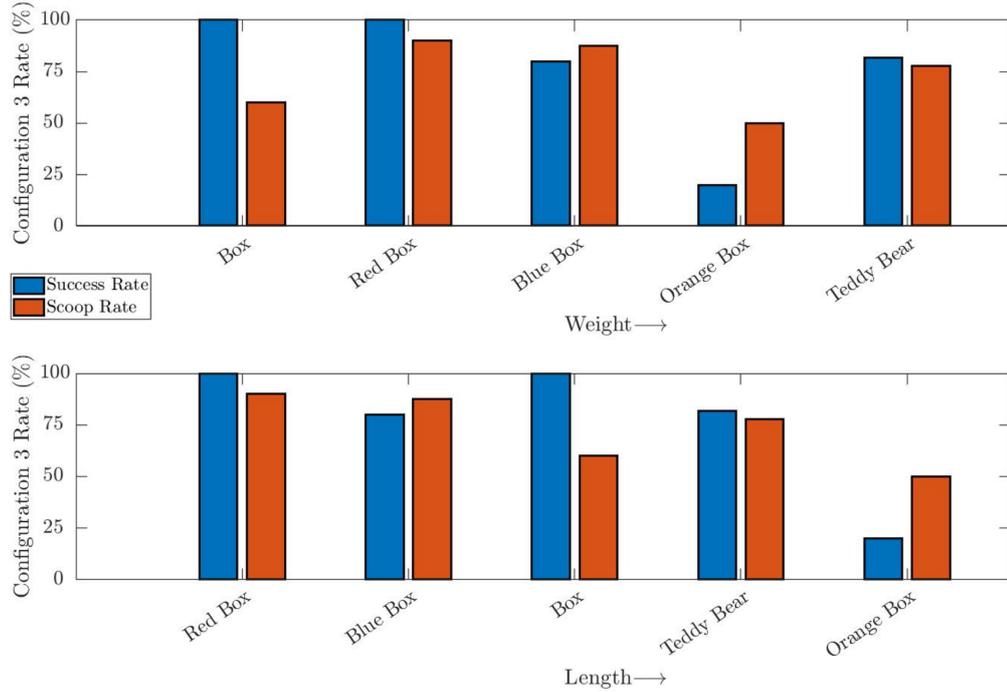
In Configuration 3 (Fig. 4.5, it can be noticed that as the length increases, the performance deteriorates in terms of either success rate and scoop rate. Whereas, there is



**Figure 4.3:** Success and scoop rates of Configuration 1 sorted in ascending order by weight and approached side (width)



**Figure 4.4:** Success and scoop rates of Configuration 2 sorted in ascending order by weight and approached side (length)



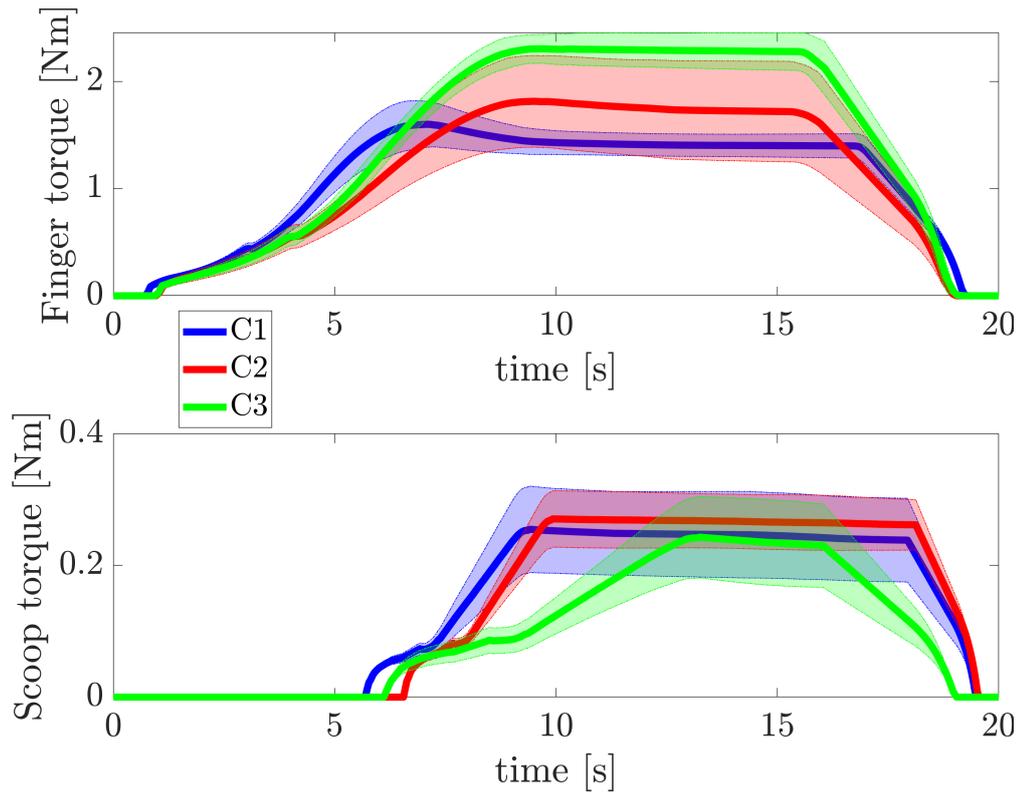
**Figure 4.5:** Success and scoop rates of Configuration 3 sorted in ascending order by weight and approached side (length)

no trend concerning weight. However, this happened only for objects which possess all the dimensions more significant than the threshold on the z-axis. In other cases, it was necessary to place the object in a different orientation.

### 4.3 Motor Torques

The motor torques were analyzed for both the fingers and the scoop. In Fig. 4.6 the mean values of the torques for the three configurations and the corresponding standard deviations were reported. During experiments the scoop is activated when a threshold value of about 1.4 Nm is exceeded on the fingers motor, i.e. when a contact is detected. In Configuration 1, the maximum torque of the finger motor is around 1.5 Nm. Unlike the other two configurations, it makes remarkable use of the scoop as evidenced by the high percentage of scoop rates (as shown in Fig. 4.3), allowing less effort to be delivered to the finger motor. Looking at the standard deviations, it was noticed that scoop torque values has a greater excursion, an evident sign of greater use of the scoop.

In Configurations 2 and 3, the steady-state torque in the finger motor is higher.



**Figure 4.6:** Motor torques for each configuration. Above: fingers' torque. Below: scoop's torque

The reason is that grasping is mostly executed using the fingers: the scoop does not play a significant role as in Configuration 1. On the other hand, it helps the gripper to keep objects once the grasp is achieved.

The different excursions from the scoop torque mean value in the three configurations are related to the objects' weights. Indeed, objects grabbed with Configuration 2 are averagely the heaviest in the dataset, while the lighter ones were used for Configuration 3.

# Conclusions

## 5.1 Discussions

The previously presented results for each strategy indicate that the SSG is capable of grasping a wide range of objects with the same strategy exploiting the hand reconfigurability.

Considering all objects (15 in total, since 2 were used twice), grasped with the configuration chosen with the criteria and with the procedure explained in Chapter 3.2, 130 out of 150 grasps were successful, with an overall grasp success rate of about 87%. Besides, 109/130 successful grasps were evaluated as robust as the object was firmly held over the scoop.

The fact that in around the 84% of successful grasps the scoop is exploited, confirms the usefulness of embedded constraints.

To test the correctness of the estimated thresholds, other experimental trials were conducted. As  $h_{threshold}$  is strictly related to the hand physical limits, i.e. fingers height, it was decided to focus our study on  $l_{threshold}$ . Additional experiments has been conducted grasping the objects previously grasped with Configuration 2 using Configuration 3, and vice versa.

Results are reported in Table 5.1.

It can be noticed that for objects near the threshold (e.g., Plastic Bottle, Pasta Pack, Red Box) success rates do not significantly differ between the two configurations. The effective use of the scoop, measured with the scoop rate, instead, follows a trend.

	Objects	Succ C2	Succ C3	Scoop C2	Scoop C3
← Length ( $l$ )	Candy Tube	70%	20%	57%	0%
	Spray Bottle	100%	0%	80%	0%
	Water Bottle	100%	100%	80%	60%
	Plastic Bottle	100%	100%	90%	70%
	Pasta Pack	90%	70%	89%	71%
	Red Box	100%	100%	90%	90%
	Blue Box	60%	80%	67%	87%
	Box	80%	100%	37%	60%
	Teddy Bear	60%	80%	50%	78%
	Orange Box	0%	20%	0%	50%

**Table 5.1:** Comparison between C2 and C3 rates sorted in ascending order by objects' length

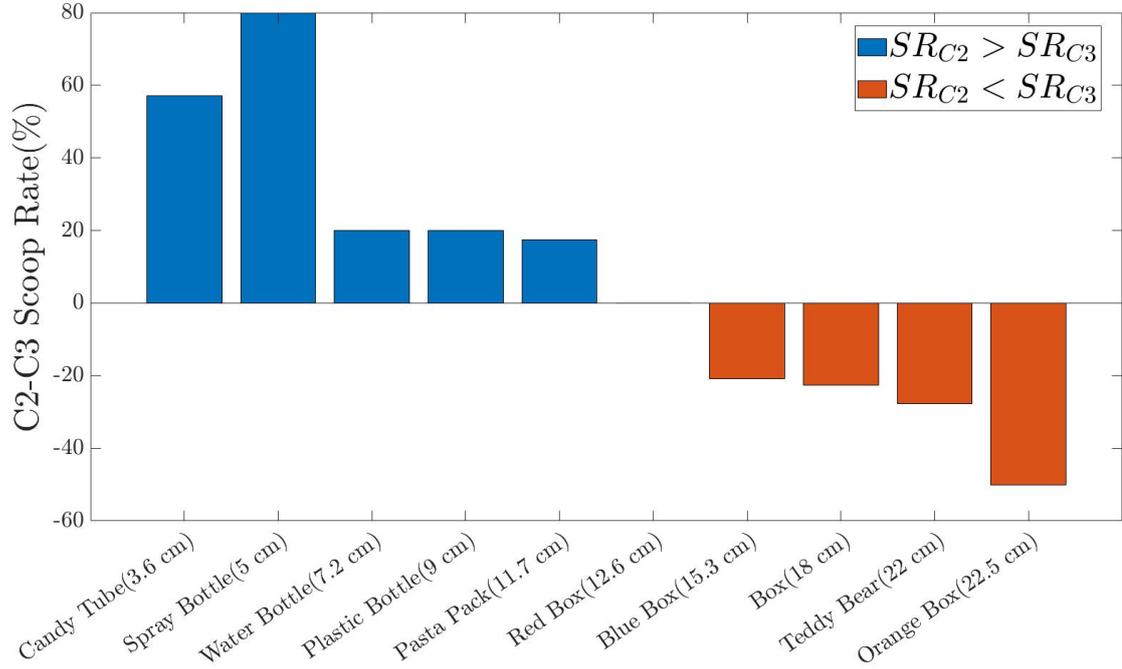
In Fig. 5.1, the difference between the scoop rate of the two configurations is showed visually. Undoubtedly, it is simpler to grasp thin objects with Configuration 2 rather than using Configuration 3, as in the case of the Candy Tube where the fingers wrap around it, allowing a firm grip.

Vice versa, using Configuration 3 the grasping is carried out exclusively using the fingers, not enabling to place the object on the scoop. Besides, objects with smaller width require that the fingers are placed more accurately, in order to perform a pinch grasp. Otherwise, small errors in the rotation of the fingers usually result in grasp failures. Furthermore, narrow and heavy objects (Spray Bottle) are challenging to grasp as the fingers alone do not provide enough torque to firmly keep the object.

Lengthy objects (e.g., Orange Box) are less tricky to grab using Configuration 3 because the fingers are oriented such that they tend to pull the objects on the scoop. In contrast, in Configuration 2, they will tend to rotate the object due to their misalignment during the closure motion.

The threshold obtained from the empirical results is around 12.6 cm (Red Box) against 11.6 cm estimated at the beginning (Chapter 3.2). However, because of the uncertainty caused by the camera depth sensor during size detection, the results can be considered satisfying.

Hence, it can be affirmed that, within the studied range of objects length, these two configurations are notably complementary.



**Figure 5.1:** Difference between the scoop rates of Configurations 2 and 3 in ascending order by objects length.

## 5.2 Future works

In this work, an approach to grasp objects exploiting the embodied reconfigurability of the Soft ScoopGripper was presented. A strategy, the so-called scoop grasp, where the soft fingers are exploited to cage the object on the inherent environmental constraint was implemented.

Following the approach of Environmental Constraint Exploitation, the scoop was used to simplify the grasping strategy.

After a preliminary analysis of possible fingers' configurations conducted in simulation, the three of them recognized as the best ones to grasp were selected to carry out grasps that are tailored for different kinds of objects.

Through the analysis of the hand closure, criteria were found that allow choosing the best hand configuration for a given object, and the best alignment over it.

First, each configuration was tested with 5 different objects, evaluating the success and

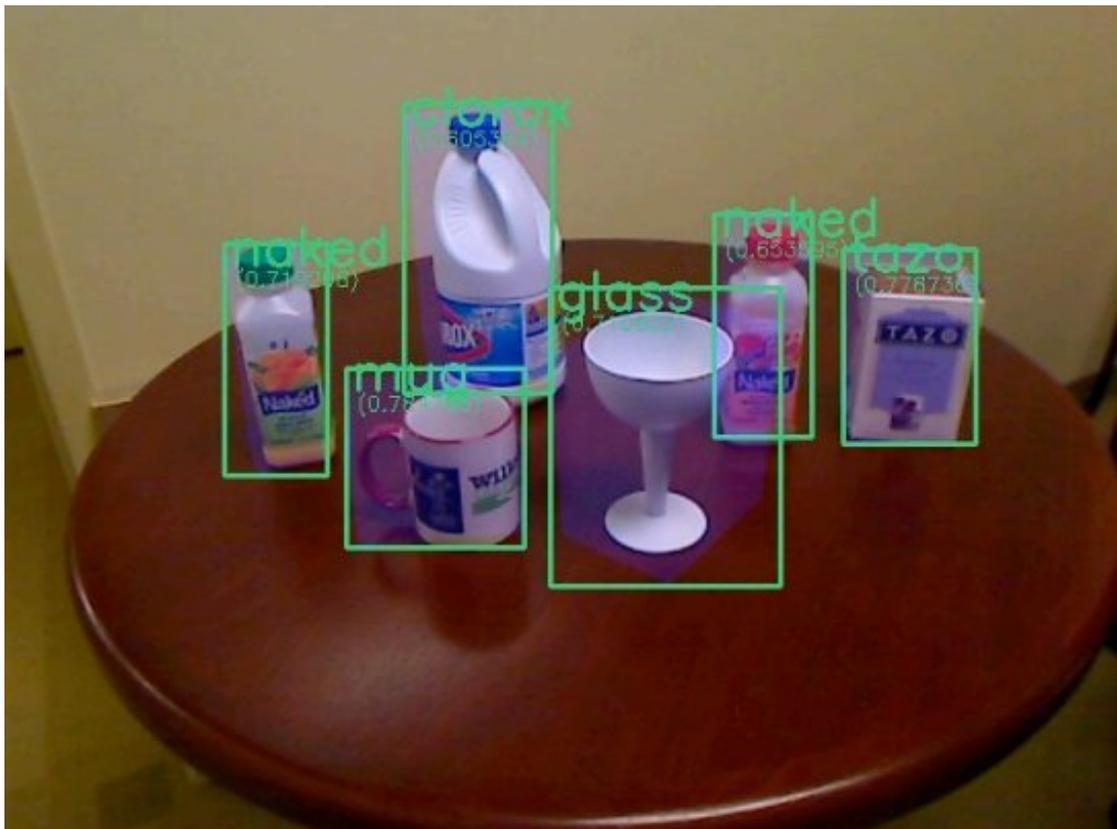
robustness of the grasps, obtaining an overall success rate of about 87%.

Then, the correctness of the proposed selection criteria was verified by carrying out additional experiments for two configurations.

Experimental results showed interesting correlations between the success of a grasp and the object length. Indeed, it was noticed that the two configurations under review are notably complementary.

Improvements in the hand design can be made in the future to enhance the performance and enable the possibility to actively rotate the dovetail joint on fingers' bases allowing for precise positioning of the fingers. This possibility can be obtained just by adding two-step motors, one for each finger.

Firstly, this will allow spanning the space of all the possible configurations that could be obtained independently rotating the fingers. This improvement will undoubtedly open new scenarios, e.g., it would be interesting to find customized fingers' settings related to a presented object.



**Figure 5.2:** Object recognition of multiple objects placed on a table

Indeed, having the possibility of using various configurations, it can be possible to compute the CS online and to find the best settings based on the dimensions of the object.

Other improvements can be made on the vision algorithm and the trajectory planner. The vision algorithm can be developed to recognize a specific object between multiples placed on the workspace, as shown in Fig. 5.2. This has already been obtained in other works through the use of machine learning. In this case, the trajectory planner would need to be remodeled to prevent collisions with the objects placed on the table. The task would be to compute a path to avoid unwanted objects instead of the shortest to reach the object.

As the SSG is a modular gripper, different add-ons can be modeled depending on the purpose of the gripper. Different fingers can be mounted with more or fewer modules, and longer scoops can replace the one used in this work. Also, the shape of the scoop can be different, satisfying the needs of the client.

As regards the scoop, it can be made in a "soft" fashion. This means that the upper surface can be designed in a soft material, in order to keep the features to slide over a surface but at the same time to be more compliant with the grasped objects.

In this work, a grasping strategy, the so-called scoop-grasp, was presented, but undoubtedly other possible ways to grasp objects using the SSG may exist. In literature, the way to grab an object with a specific gripper is called primitive of the gripper itself.

A further step to obtaining other primitives could be to teleoperate with haptic devices (Fig. 5.3), a robotic arm with the SSG attached to it. Hence, after having collected enough data, deep learning algorithms could be used to analyze the behavior of the users to understand if some primitives are more effective with this gripper.

Lastly, one useful feature that can be inserted in this work should be the complete automation of the process through the implementation of a Finite State Machine. In this way, even if the object falls from the gripper, the algorithm is capable of detecting a failure and of re-planning a grasping action.



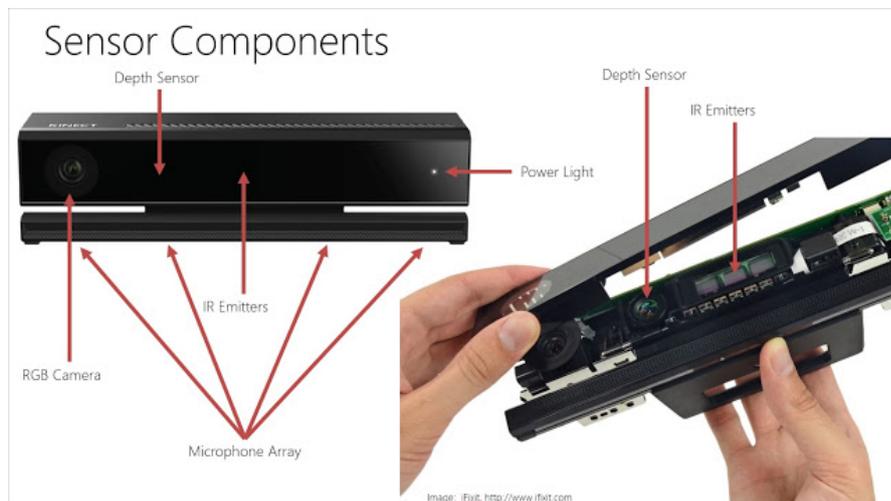
**Figure 5.3:** Sigma7 haptic device

# Appendices

# Hardware descriptions

## A.1 Kinect One

As an RGB-D sensor, the Kinect v2 (or Kinect One) is the second-generation Kinect by Microsoft, released in July 2014. Equipped with a color camera, a depth sensor (including an infrared camera and IR projector), and a microphone array, the Kinect v2 can be used to sense color images, depth images, IR images, and audio information. The hardware structure is shown in Fig. A.1.



**Figure A.1:** Kinect One Sensor Components

The detailed technical specifications are summarized in Table A.1.

To interface the Kinect to the ROS environment, it is necessary to use a specific library. In this case, it is decided to use `iai_kinect2` [32]. This is a collection of tools and libraries

<b>Color</b>	<b>Camera Resolution Framerate</b>	1920 × 1080 pixels 30 frames per second
<b>Depth</b>	<b>Camera Resolution Framerate</b>	512 × 424 pixels 30 frames per second
<b>Field of VIEW (depth)</b>	<b>Horizontal Vertical</b>	70 degrees 60 degrees
<b>Operative Measuring Range</b>		from 0.5 m to 4.5 m
<b>Depth Technology</b>		Time-of-flight (ToF)
<b>Tilt Motor</b>		No
<b>USB Standard</b>		3.0

**Table A.1:** Technical specifications of Kinect One

that contains:

- a calibration tool for calibrating the IR sensor of the Kinect One to the RGB sensor and the depth measurement
- a library for depth registration with OpenCL support
- a viewer for the images/point clouds
- the bridge between libfreenect2 and ROS

The libfreenect2 [33] is a library containing the driver for Kinect for Windows.

## A.2 KUKA LBR iiwa

The KUKA LBR iiwa is a lightweight industrial robotic arm with seven axes. Each of its joints is equipped with torque sensors as well as a position sensor. Sensory data enables the use of impedance control in addition to position control, thus making it possible to implement compliant behaviors. Highly accurate measurements, with down to millisecond update intervals, enable the robot to react very quickly to process forces and makes it particularly suitable for interaction with humans. The specifications of the robot can be seen in Table A.2. The LBR iiwa can be programmed for a variety of tasks through “KUKA Sunrise control technology”. This comprises “KUKA Sunrise OS” control software, which can execute programs in JAVA as the programming language on “KUKA Sunrise Cabinet” control hardware. Although Java is a flexible and common language, in-depth knowledge about the Sunrise system is required for programming the

		<b>LBR iiwa 7 R800</b>
Rated payload	_____	7 kg
Number of axes	_____	7
Wrist variant	_____	In-line wrist
Mounting flange A7	_____	DIN ISO 9409-1-A50
Installation position	_____	any
Repeatability (ISO 9283)	_____	±0.1 mm
Axis-specific torque accuracy (of maximum torque)	_____	±2 %
Weight	_____	22.3 kg
Protection rating of the robot	_____	IP54
		<b>KUKA Sunrise Cabinet</b>
Processor	_____	Quad-core processor
Hard drive	_____	SSD
Interfaces	_____	USB, EtherNet, DVI-I
Protection rating	_____	IP20
Dimensions (DxWxH)	_____	500 mm x 483 mm x 190 mm
Weight	_____	23 kg
<b>Power supply connection</b>		
Rated supply voltage	_____	AC 110 V – 230 V
Permissible tolerance of rated voltage	_____	±10 %
Mains frequency	_____	50 Hz ± 1Hz oder 60 Hz ± 1Hz
Mains-side fusing	_____	2 x 16 A slow-blowing

**Table A.2:** Technical specifications of KUKA LBR iiwa

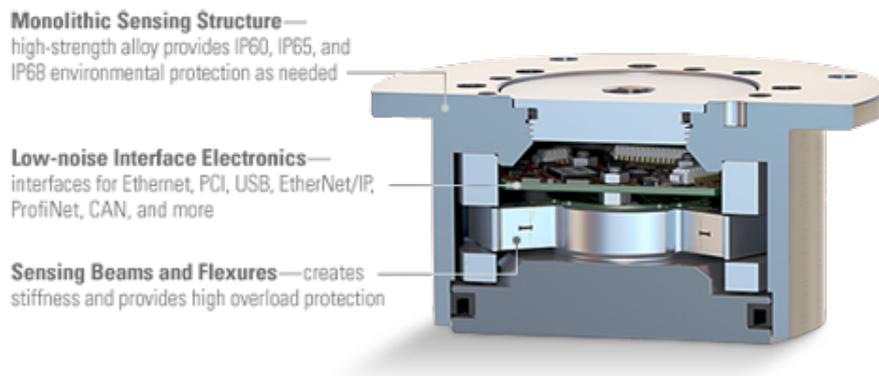
robot and utilizing its functionality. For this reason, it is necessary to use an interface to ROS to program the robot in C++ or Python. In this case, the library chosen is iiwa stack [34]. It provides an integration of KUKA's Smart Servo and PTP motions:

- joint position, velocity and cartesian position control via simple ROS messages
- online configuration of Joint Impedance, Cartesian Impedance, Desired Force and Sine Pattern via ROS service
- online configuration of joint/cartesian velocity and acceleration via ROS services
- full MoveIt! Integration
- Gazebo support

### A.3 KUKA LBR iiwa

It measures all six components of force and torque, and it is applied to the robot end-effector. It publishes topics on ROS using the netft rdt driver package [35]. The sensor consists of a transducer, interface electronics, and cabling. The compact and rugged monolithic transducer uses silicon strain gages to sense forces, as shown in Fig. A.2. The transducer's silicon strain gages provide high noise immunity and allow high overload protection, which is standard on all models. The Net F/T, DAQ F/T, Controller F/T, and TWE F/T each provide a variety of powerful functions:

- Tool transformations translate and/or rotate the F/T reference frame.
- Demo software allows configuration and basic data logging capabilities.
- Biasing provides a convenient way to offset tool weight.
- Increased system throughput is possible by reducing the number of axes of output.
- Threshold detection eases integration into industrial applications (Net F/T and Controller F/T only).
- Integral temperature compensation ensures accuracy over a wide temperature range.



**Figure A.2:** Schematics of the ATI sensor

# Software descriptions

## B.1 ROS

ROS [36] is an open-source, meta-operating system for robots. It provides the services that would be expected from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

ROS implements several different styles of communication, including synchronous RPC-style communication over services, asynchronous streaming of data over topics, and storage of data on a Parameter Server. ROS is not a real-time framework, though it is possible to integrate ROS with real-time code.

Its architecture is based on Peer-to-Peer (P2P) communication between different nodes. These nodes can be defined as programs that perform various tasks and are running on one or more computers being part of a network.

A part of the map of the ROS nodes working in the project is shown in Fig. B.1.

It was decided to use ROS instead of another robotics software platform because the ROS framework is:

- Thin such that code written for ROS can be used with other robot software frameworks
- Built using ROS-agnostic libraries with clean functional interfaces

- Language independent such that it is easy to implement in any modern programming language
- Easy to scale such that large project are easy to develop

The employed ROS distribution for this project is ROS Kinetic, released on May 2016.

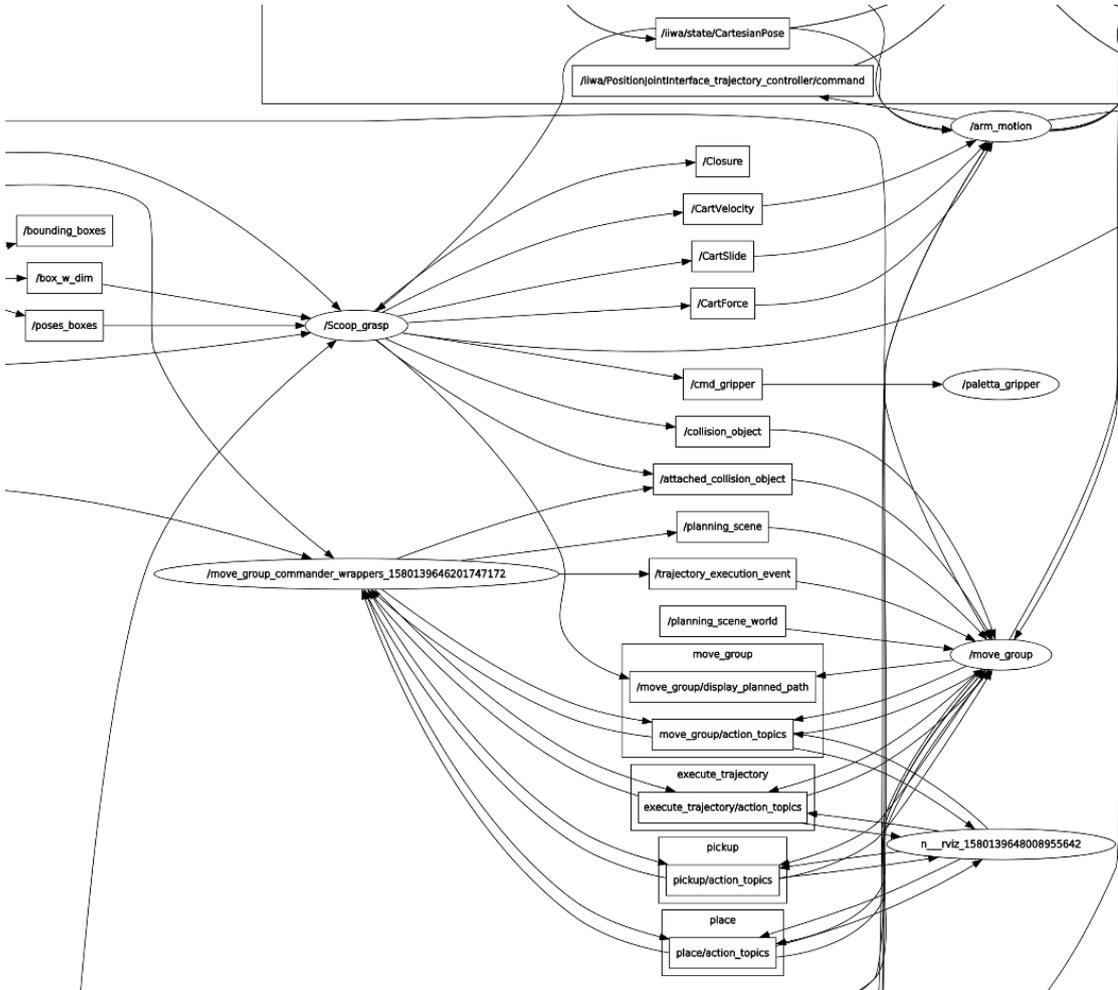


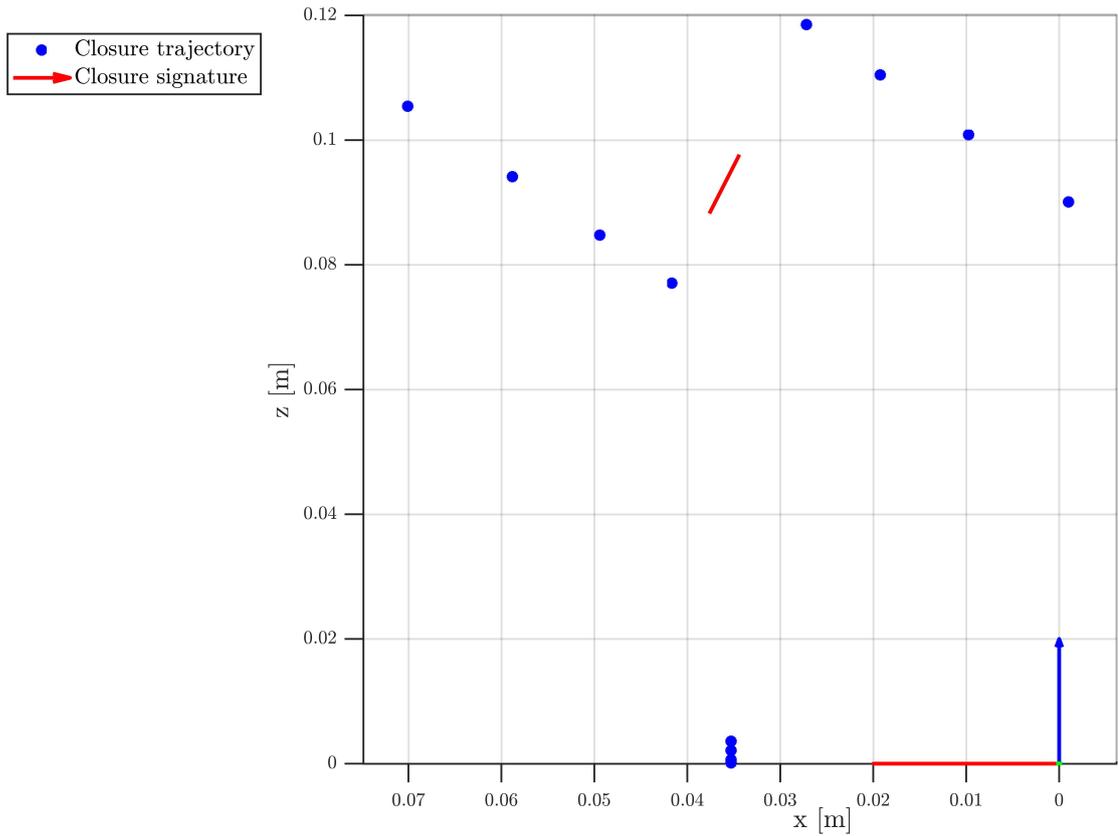
Figure B.1: Part of the Ros Node Graph

## B.2 MATLAB

MATLAB (Matrix Laboratory) is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks.

MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other

languages.



**Figure B.2:** Closure Signature estimated on MATLAB

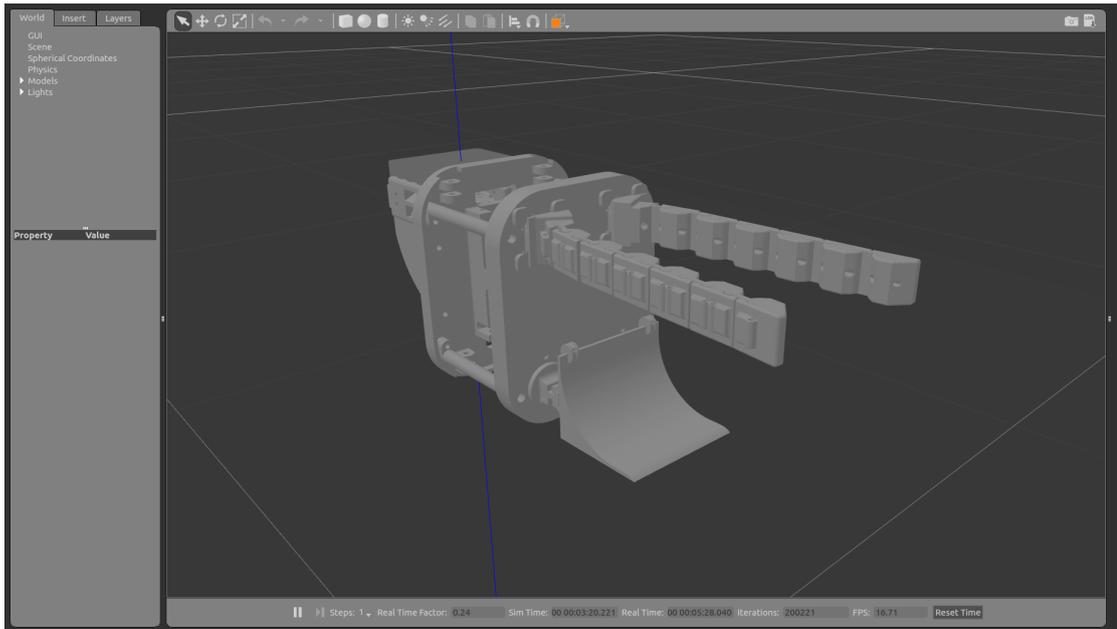
MATLAB was preferred with respect to other software because it's one of the most powerful computing environment existing at this moment. It was useful to analyze the initial fingers' configurations and to estimate the Closure Signature of the Soft ScoopGripper. The estimated Closure Signature with MATLAB is shown in Fig. B.2.

### B.3 GAZEBO

Gazebo is an open-source 3D robotics simulator. Gazebo was a component in the Player Project from 2004 through 2011. Gazebo integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control.

It was chosen because it is a well-designed simulator that is able to rapidly test algorithms, design robots, and perform regression testing.

Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It is a physics engine, with high-quality



**Figure B.3:** Model of the Soft ScoopGripper in GAZEBO

graphics. It was used together with MATLAB for the estimation of the Closure Signature with the models represented in Fig. B.3.

# References

- [1] Domenico Prattichizzo and Jeffrey C. Trinkle. *Grasping*, pages 955–988. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1\_38. URL [https://doi.org/10.1007/978-3-319-32552-1\\_38](https://doi.org/10.1007/978-3-319-32552-1_38).
- [2] Imin Kao, Kevin Lynch, and Joel W. Burdick. *Contact Modeling and Manipulation*, pages 647–669. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-30301-5. doi: 10.1007/978-3-540-30301-5\_28. URL [https://doi.org/10.1007/978-3-540-30301-5\\_28](https://doi.org/10.1007/978-3-540-30301-5_28).
- [3] Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35, 08 2015. doi: 10.1177/0278364915592961.
- [4] Aaron M. Dollar and Robert D. Howe. The sdm hand: A highly adaptive compliant grasper for unstructured environments. In Oussama Khatib, Vijay Kumar, and George J. Pappas, editors, *Experimental Robotics*, pages 3–11, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-00196-3.
- [5] M.G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi. Adaptive synergies for the design and control of the pisa/iit soft hand. *The International Journal of Robotics Research*, 33(5):768–782, 2014. doi: 10.1177/0278364913518998. URL <https://doi.org/10.1177/0278364913518998>.
- [6] Máximo A. Roa and Raúl Suárez. Grasp quality measures: Review and performance. *Auton. Robots*, 38(1):65–88, January 2015. ISSN 0929-5593. doi: 10.1007/s10514-014-9402-3. URL <https://doi.org/10.1007/s10514-014-9402-3>.
- [7] M. Pozzi, A. M. Sundaram, M. Malvezzi, D. Prattichizzo, and M. A. Roa. Grasp

- quality evaluation in underactuated robotic hands. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1946–1953, Oct 2016. doi: 10.1109/IROS.2016.7759307.
- [8] M. Pozzi, G. Salvietti, J. Bimbo, M. Malvezzi, and D. Prattichizzo. The closure signature: A functional approach to model underactuated compliant robotic hands. *IEEE Robotics and Automation Letters*, 3(3):2206–2213, July 2018. ISSN 2377-3774. doi: 10.1109/LRA.2018.2810946.
- [9] M. Malvezzi, G. Gioioso, G. Salvietti, and D. Prattichizzo. Syngrasp: A matlab toolbox for underactuated and compliant hands. *IEEE Robotics Automation Magazine*, 22(4):52–68, Dec 2015. ISSN 1558-223X. doi: 10.1109/MRA.2015.2408772.
- [10] Clemens Eppner, Raphael Deimel, José Álvarez Ruiz, Marianne Maertens, and Oliver Brock. Exploitation of environmental constraints in human and robotic grasping. *The International Journal of Robotics Research*, 34(7):1021–1038, 2015. doi: 10.1177/0278364914559753. URL <https://doi.org/10.1177/0278364914559753>.
- [11] Joao Bimbo, Enrico Turco, Mahdi Ghazaei Ardakani, Maria Pozzi, Gionata Salvietti, Valerio Bo, Monica Malvezzi, and Domenico Prattichizzo. Exploiting robot hand compliance and environmental constraints for edge grasps. *Frontiers in Robotics and AI*, 6:135, 2019. ISSN 2296-9144. doi: 10.3389/frobt.2019.00135. URL <https://www.frontiersin.org/article/10.3389/frobt.2019.00135>.
- [12] G. Salvietti, M. Malvezzi, G. Gioioso, and D. Prattichizzo. Modeling compliant grasps exploiting environmental constraints. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4941–4946, May 2015. doi: 10.1109/ICRA.2015.7139885.
- [13] G. Salvietti, Z. Iqbal, M. Malvezzi, T. Eslami, and D. Prattichizzo. Soft hands with embodied constraints: The soft scoopgripper. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2758–2764, May 2019. doi: 10.1109/ICRA.2019.8793563.
- [14] G. Salvietti, I. Hussain, M. Malvezzi, and D. Prattichizzo. Design of the passive joints of underactuated modular soft hands for fingertip trajectory tracking. *IEEE*

- Robotics and Automation Letters*, 2(4):2008–2015, Oct 2017. ISSN 2377-3774. doi: 10.1109/LRA.2017.2718099.
- [15] Raphael Deimel, Clemens Eppner, José Álvarez-Ruiz, Marianne Maertens, and Oliver Brock. Exploitation of environmental constraints in human and robotic grasping. In *Robotics Research*, pages 393–409. Springer, 2016.
- [16] Oliver Brock, Jaeheung Park, and Marc Toussaint. *Mobility and Manipulation*, pages 1007–1036. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1\_40. URL [https://doi.org/10.1007/978-3-319-32552-1\\_40](https://doi.org/10.1007/978-3-319-32552-1_40).
- [17] Clemens Eppner and Oliver Brock. Visual detection of opportunities to exploit contact in grasping using contextual multi-armed bandits. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 273–278. IEEE, 2017.
- [18] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping novel objects with depth segmentation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2578–2585, Oct 2010. doi: 10.1109/IROS.2010.5650493.
- [19] Manuel Bonilla, D. Resasco, M. Gabiccini, and Antonio Bicchi. Grasp planning with soft hands using bounding box object decomposition. pages 518–523, 09 2015. doi: 10.1109/IROS.2015.7353421.
- [20] S. Jain and B. Argall. Grasp detection for assistive robotic manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2015–2021, May 2016. doi: 10.1109/ICRA.2016.7487348.
- [21] K. Harada, T. Foissotte, T. Tsuji, K. Nagata, N. Yamanobe, A. Nakamura, and Y. Kawai. Pick and place planning for dual-arm manipulators. In *2012 IEEE International Conference on Robotics and Automation*, pages 2281–2286, May 2012. doi: 10.1109/ICRA.2012.6224780.
- [22] G. Salvietti, M. Malvezzi, G. Gioioso, and D. Prattichizzo. On the use of homogeneous transformations to map human hand movements onto robotic hands. In

- 2014 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5352–5357, May 2014. doi: 10.1109/ICRA.2014.6907646.
- [23] J. Ponce, S. Sullivan, J. . Boissonnat, and J. . Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 821–827 vol.2, May 1993. doi: 10.1109/ROBOT.1993.291933.
- [24] Dan Ding, Yun-Hui Lee, and Shuguo Wang. Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics and Automation*, 17(4):515–522, Aug 2001. ISSN 2374-958X. doi: 10.1109/70.954765.
- [25] wg-perception. <https://github.com/wg-perception>. Accessed: 2019-11-30.
- [26] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [27] Realtime urdf filter. [http://github.com/blodow/realtime\\_urdf\\_filter](http://github.com/blodow/realtime_urdf_filter). Accessed: 2019-11-30.
- [28] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study, 2014.
- [29] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. doi: 10.1109/MRA.2012.2205651. <http://ompl.kavrakilab.org>.
- [30] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001 vol.2, April 2000. doi: 10.1109/ROBOT.2000.844730.
- [31] rosserial. <https://github.com/ros-drivers/rosserial>. Accessed: 2019-11-30.
- [32] Thiemo Wiedemeyer. IAI Kinect2. [https://github.com/code-iai/iai\\_kinect2](https://github.com/code-iai/iai_kinect2), 2014 – 2015. Accessed June 12, 2015.

## REFERENCES

- [33] Lingzhu Xiang, Florian Echtler, Christian Kerl, Thiemo Wiedemeyer, Lars, hanyazou, Ryan Gordon, Francisco Facioni, laborer2008, Rich Wareham, and et al. libfreenect2: Release 0.2. Apr 2016. doi: 10.5281/zenodo.50641.
- [34] Christoph Hennemersperger, Bernhard Fuerst, Salvatore Virga, Oliver Zettinig, Benjamin Frisch, Thomas Neff, and Nassir Navab. Towards mri-based autonomous robotic us acquisitions: a first feasibility study. *IEEE transactions on medical imaging*, 36(2):538–548, 2017.
- [35] netft. <https://github.com/CloPeMa/netft>. Accessed: 2019-11-30.
- [36] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL <https://www.ros.org>.