ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCHOOL OF ENGINEERING

DEPARTMENT of
ELECTRICAL, ELECTRONIC AND INFORMATION ENGINEERING
“Guglielmo Marconi”
DEI

MASTER DEGREE IN
AUTOMATION ENGINEERING

MASTER’S DEGREE THESIS
in
Industrial Robotics

A Myo-Controlled Wearable Manipulation System with Tactile Sensing for Prosthetics Studies

CANDIDATE

Marco Perozzi

SUPERVISOR

Chiar.mo Prof. Gianluca Palli

CO-SUPERVISOR

Dr. Roberto Meattini

Academic Year
2020/2021

Session
IV

Contents

List of Figures

Abstract

The following thesis project aims to study and realize a wearable manipulation system composed by an AR10 robotic hand, controlled via myoelectric signals and tactile sensors for prosthetic studies. The project starts with the kinematic study of the hand via MATLAB and Simulink, in order to obtain a complete insight on the robotic grasping device. Thereafter, a wearable support has been designed and printed to fix the robotic hand around the user forearm. Surface electromyography is acquired using a gForce gesture armband. A Simulink system has been developed to acquire and filter the signals, then the myoelectric data are elaborated to derive the command for the robotic hand. Tactile sensors are added by means of custom 3D-printed support on the fingertips in order to get a force feedback to allow the user to perform the grasp of different objects. Finally, in order to test the whole solution, a subject wearing the whole manipulation system carried out a series of tasks to evaluate the system's usability during dynamic grasps of different objects. The results of the tests report the accuracy of the manipulation system. The main goal of the project is to test a wearable manipulation system made to be worn by intact subjects, in order to study prosthetic grasping scenarios that can provide results useful for future developments involving amputees.

Chapter 1

Introduction

The world of robotics is currently facing the problem of replicating the human ability and flexibility in performing motor tasks. One of the most successful approaches is the *human-in-the-loop* approach, where the human interaction is the key element of the system design, within the framework of human-robot interface (HRI). Science and engineering have tried to mimic the sensory and motor function of a human hand since at least the sixteenth century [1]. The hand is one of the most functional limbs of the human body which is able to perform a variety of daily tasks. Artificial hand can have a wide range of application, from industrial manipulation to robotic prosthetics for disabled subjects. This thesis projects is focused on the area of prosthetic robotics, with the objective of creating a manipulation system composed by a myo-controlled robotic hand with tactile sensors to be used in prosthetic studies. Prosthetic hands are prescribed to patients who have suffered an amputation of the upper limb due to an accident or a disease. This is done to allow patients to regain functionality of their lost hands. Myoelectric prosthetic hands were found to have the possibility of implementing intuitive controls based on operator's electromyogram (EMG) signals [2]. The use of electromyography allows the patient to utilize the prosthetic device as a part of their body without the necessity of additional controls appendages. The device created in this project has numerous characteristics in common with a prosthetic hand. The most important common aspect is the wearability, the device can be worn by amputated subject but also by healthy subject. Preliminary studies, as [3], are usually carried out on healthy subjects since several aspect of the prosthetic can be explored and developed without the involvement of an amputated subject. This thesis project focuses on some aspect in which the involvement of amputated subject is not necessary, as the acquisition and decoding of the sEMG signals and the testing of the tactile sensors. Afterwards, other aspects need to be tested for both healthy and amputated subject, such as the behaviour of the myo-control in different position of the arm and the body of the subject.

1.1 Prosthetic Hands

A prosthetic hand refers to an artificially made device that serves as a substitute for a partially or totally lost hand. A prosthetic may be cosmetic, functional or both. A cosmetic prosthetic is usually made only to make the limb look natural and provides little to no functionality. A functional prosthetic, which this projects focuses on, can help patients to regain partial or complete functionality of the lost limb. Robotic prosthetics must be able to interact with the environment in a safe and reliable way in all the daily task of the user. Prosthetics require, also, reduced weight and encumbrance, simple controls to accommodate the limited number of inputs available for amputees, high interaction capabilities with humans and the environment, and features that enable devices to operate in harsh and unstructured conditions [4]. Based on the research of [5], from the viewpoint of amputees, a suitable prosthetic hand has different features such as resemblance with the human hand anatomy, low weight, low cost and high functionality (capability of performing handy grasp patterns, particularly power and precision grasp). For these reasons in the project has been taken in consideration a simple and light robotic hand, AR10 produced by Active8 Robotics. The use of a pre-built simple hand allowed to focus on the control and the design part of the project, maintaining the cost reasonable.

1.2 Electromyography

In humans the control of the movement lies in the nervous system, and originates in the transmission of bioelectrical signals through the body. The nervous system transmits along a serial line, in which the value is encoded by means of electrical impulses, and in particular in terms of their frequency. This impulses are know as action potentials [6]. Electromyography is the expression used to indicate the studies and the techniques related to the electrogram, which is the electrical signal measured from the contraction of muscles. Both electromyography and eloctrogram are referred as EMG in the following paragraphs. Although there are many variables that affect the values of a electrogram, the most appreciable rise in the magnitude of the EMG signal is given by the voluntary increase in the muscle tension. There are two kind of electromyography techniques: intramuscular EMG and surface EMG. The first, as the name suggests, consists in an invasive procedures requiring needle electrodes inserted into the muscle. The surface electromyography (sEMG) assesses muscle function by recording muscle activity from the surface above the muscle on the skin. The use of surface EMG recording provides a safe, easy, and noninvasive method that allows objective quantification of the energy of the muscle. It is not necessary to penetrate the skin and record from single motor units to obtain useful and meaningful information regarding . The technique allows the observer to see the muscle energy at rest and changing continuously over the course of a movement[7].

Chapter 2

AR10 Robotic Hand

The manipulation system used in the project is an AR10 Humanoid Robot Hand made by Active8 Robots, shown in Fig. 2.1. The AR10 features 10 degrees of freedom (DOF) that are servo-actuated within the hand's envelope. The AR10 Robot Hand provides a complete solution for academia and can be used across a variety of systems and interfaces. Manufactured from a hybrid construction, it balances strength and weight. It is an ideal platform to carry out research in the field of robotics. Its capability can be expanded by adding sensors or combining the hand with a robot arm.

2.1 ROS Control

Robot Operating System (ROS) is an open-source robotics middleware suite. Although ROS is not an operating system but a collection of software frameworks for robot software development, it provides services designed for a het-

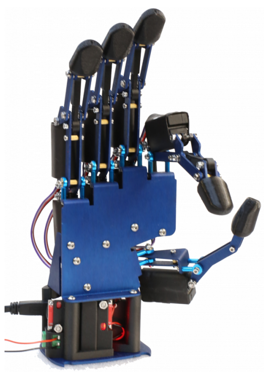


Figure 2.1: AR10 Hand

erogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. ROS processes are represented as nodes in a graph structure, connected by edges called topics. ROS nodes can pass messages to one another through topics, make service calls to other nodes, provide a service for other nodes, or set or retrieve shared data from a communal database called the parameter server. The AR10 the primary management is handled using ROS as a communication platform for the controls. The AR10 is also equipped with an assortment of Python scripts, which allows the user to have a basic control of the hand. In the ROS environment the hand is controlled by the hand node, which translates the commands given to the right values, for each servomotor on the fingers, to publish on the join_states node. Moreover, it is possible to control the hand using a Python script, which creates a simple command interface to publish basic commands such as: open, close and grasp. These commands can also be modified according to the user requirements.

2.2 Rviz Control

Rviz, abbreviation for ROS visualization, is a powerful 3D visualization tool for ROS. It allows the user to view the simulated robot model, log sensor information from the robot's sensors, and replay the logged sensor information. By visualizing what the robot is seeing, thinking, and doing, the user can debug a robot application from sensor inputs to planned (or unplanned) actions. By means of the Rviz platform is possible to control each servomotor with the sliders shown in Fig. 2.2. This allows the user to have a basic continuous control of the hand, both in simulation and with the real device. Unfortunately, Rviz does not allow the user to expand the control other than the basic sliders, which makes necessary the creation of custom programs to have a more powerful control.

2.3 Simulink Basic Control

The whole control part of the project has been developed on the MATLAB and Simulink paltforms. MATLAB (an abbreviation of "MATrix LABoratory") is a proprietary multi-paradigm programming language and numeric computing environment 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

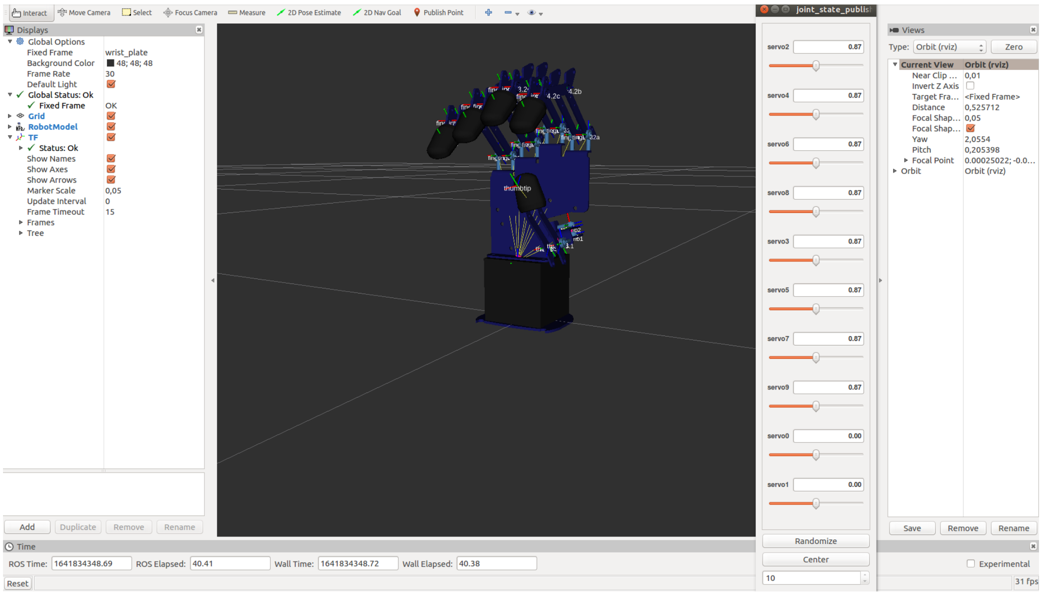


Figure 2.2: Rviz Hand Control

languages.

Simulink is a MATLAB-based graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and digital signal processing for multidomain simulation and model-based design.

In order to have a basic control of the hand in the Simulink environment a new model has been developed. This Simulink model in Fig. 2.3 allows the user to control the hand with its own direct commands, left part, or with the value of the servomotors published on the joints_state topic, right part. This scheme makes use of the Simulink ROS toolbox to create specific topics, or to publish and subscribe to topics in the ROS environment connected to the hand.

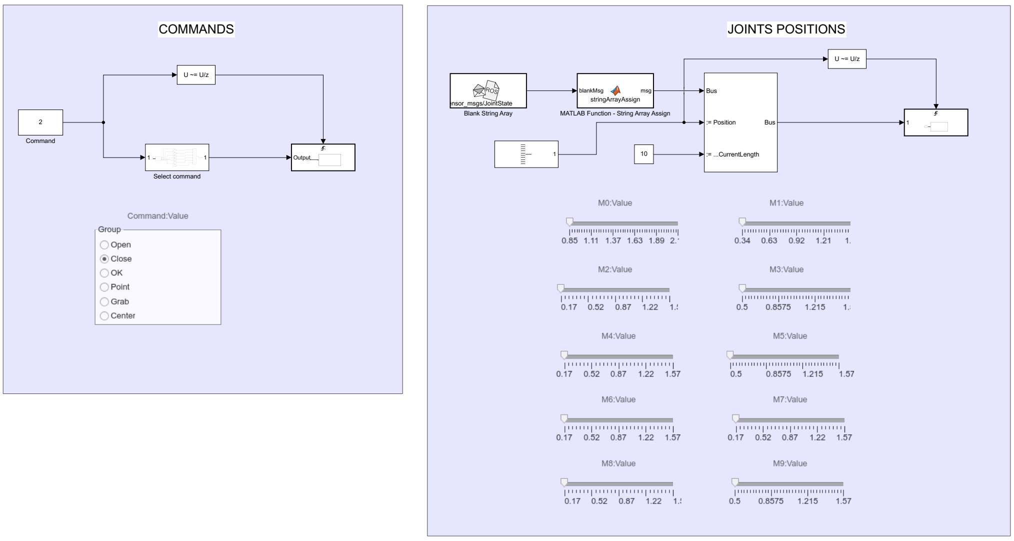


Figure 2.3: Simulink Hand Control

Chapter 3

AR10 Robotic Hand Kinematics

One of the most important part in the study of the AR10 is its kinematic model and the computation of related fingers' forward and inverse kinematics. A robotic finger can be schematically represented from a mechanical viewpoint as a kinematic chain of rigid bodies (links) connected by means of revolute or prismatic joints. One end of the chain is constrained to a base, while an end-effector is mounted to the other end. The resulting motion of the structure is obtained by composition of the elementary motions of each link with respect to the previous one. Therefore, in order to manipulate an object in space, it is necessary to describe the end-effector position and orientation [8]. A rigid body is completely described in space by its pose, meaning its position and orientation. In the case of a robotic hand the body has five different end effectors, the tip of the five digits, the combination of their poses creates the hand pose.

3.1 Direct Kinematics

The first step is to develop the direct kinematic for the hand. This consists in deriving the pose for each end effector from the knowledge of the joint variables. For the direct kinematics a MATLAB script has been developed which allows the user to obtain the poses of the fingertips of the hand for each possible joint configuration. Note that, in this case, the end-effectors can only have a certain orientation in space, given that it is enough to know the position of the finger tip. The script exploits the Interactive Rigid Body Tree object from MATLAB. The interactiveRigidBodyTree object, shown in Fig. 3.1, creates a figure that displays a robot model using a rigidBodyTree

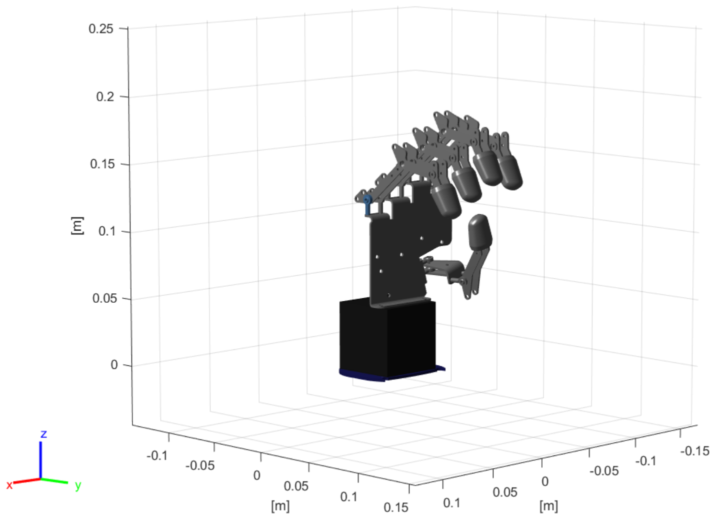


Figure 3.1: Interactive Rigid Body of the Hand

object and enables you to directly modify the robot configuration using an interactive marker [9]. The rigidBodyTree has been deducted automatically by MATLAB a function, from the urdf description file given by Active8 Robots. As a result, is possible to obtain the Cartesian coordinates of each finger tip for every configuration possible of the hand's servomotors automatically.

In Fig. 3.2 can be observed the trajectories of the fingertip in a 3D space, starting from a matrix containing the values of the joints during the closing motion.

3.2 Inverse Kinematics

The inverse kinematics problem consists in the determination of the joint variables corresponding to a given end-effector position and orientation. The solution to this problem is of fundamental importance in order to transform the motion specifications, assigned to the end-effector in the operational space, into the corresponding joint space motions that allow execution of the desired motion [8]. Note that, for the purpose of the study of a robotic hand, the end-effectors are, as already mention, five. In order to solve this problem, the hand can be considered as five different open chain robots with the same base. Is it, then, possible to solve the inverse kinematics problem for each open chain and, in the end, combine all the results to obtain the complete

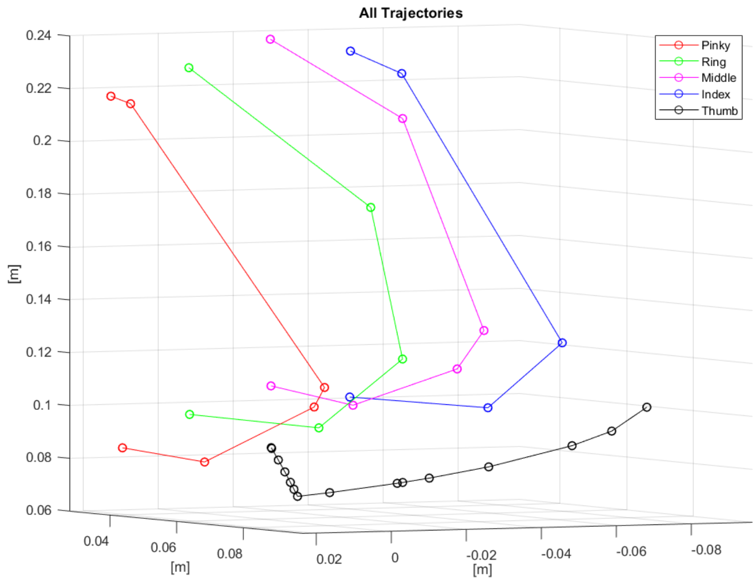


Figure 3.2: Fingertips Trajectories of a Closing Motion

inverse kinematics.

In Fig. 3.3 is shown the Simulink scheme developed in order to obtain the inverse kinematics of the hand for every possible combination of finger tip positions. The scheme uses the Inverse Kinematics block from the Robotic System Toolbox in Simulink, one block for each finger tip. Each block takes as input the position of its own finger and gives as output an array containing the values for all the servomotors, not only the ones directly connected to their own finger tip. Then, a MATLAB function elaborates all the different arrays to extrapolate the real vector of joint states to return. This scheme can be used to obtain the inverse kinematics for single poses, series of poses or complete motions.

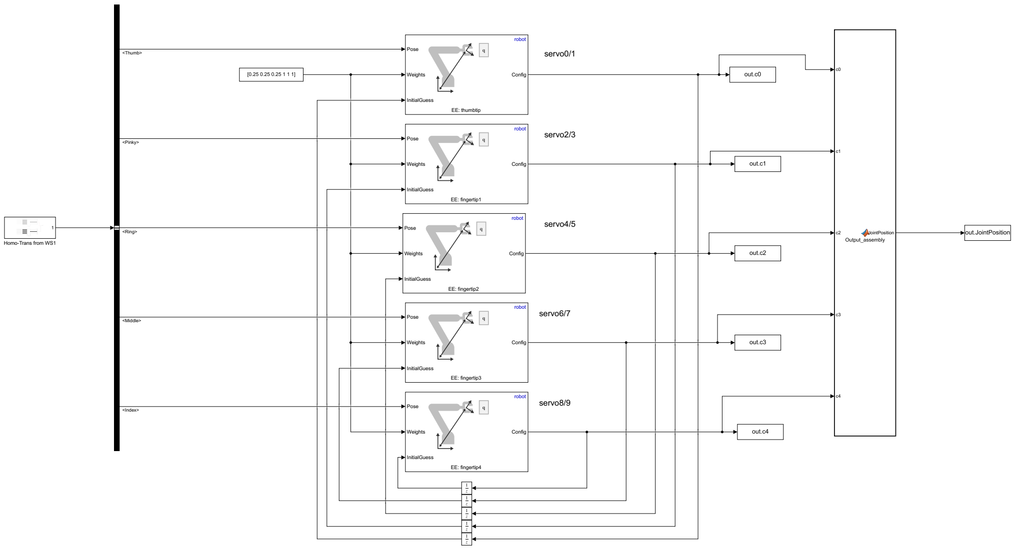


Figure 3.3: Inverse Kinematic Simulink Scheme

Chapter 4

Surface Electromyography for Robotic Hand Control

One of the main interests of the project is the ability to control the robotic hand using the signals acquired by the surface electromyography from the forearm of the subject. In order to do so, it is necessary to understand and process the data acquired from the sEMG.

4.1 Muscular Synergy

From a mechanical point of view the human hand is a very complex system, characterized by 21 DOFs controlled by 29 different muscles [10]. The results presented in [11] indicates that not all the DOFs are controlled independently, this implies a reduction of the DOFs from 21 to 2 or 3. This reduction allows to get the grasp synthesis for a large set of object by linearly combining only a few dominant hand postures (i.e. muscular synergies). In humans, on the basis of the muscular synergy concept, a unique neural drive is shared by different muscles, individually activated with a certain degree of excitation determined by spinal cord circuitries (i.e., the synergy weights) [12]. In the case of the forearm the motor control system generates a the superspinal neural drives to control the extensor and flexor groups of antagonists muscles. These neural drives are not directly measurable, and therefore the muscular synergy matrix S_M is exploited to online estimate such neural drives from EMG signals of the forearm muscles:

$$U(t) = S_M^{+} E(t) \tag{4.1}$$

where $U(t) = [u_e(t)\, u_f(t)]^T \in \mathbb{R}^2$ is the vector of the instantaneous values of the neural drives, S_M^{+} is the pseudo-inverse of the synergy matrix S_M and

Figure 4.1: gForcePro+ EMG Armband

$E(t) = [e_1(t) \cdots e_8(t)]^T \in \mathbb{R}^8$is the vector of the instantaneous values of the EMG channels. Then a control signal σ^{ref}, referred as *synergistic closure reference* of the hand, is derived as a linear combination of the neural drives:

$$\sigma^{ref} = \frac{U_1(t)}{k_1} - \frac{U_2(t)}{k_2} + k_{off} \tag{4.2}$$

where k_1, k_2 and k_{off} are three parameters for the normalization derived from the calibration phase in order to have $\sigma^{ref} \in [0, 1]$. This signal is used as a proportional controller of the closure of the robotic hand.

4.2 Acquisition and Neural Drive Derivation

In order to avoid conflicts with the data acquisition from EMG and force sensors implemented next, the whole control system has been divided in two parts running on two different machines: the EMG acquisition and the control part.

In the EMG acquisition part, for the electromyography, the gForcePro+ EMG Armband, produced by OYMotion, has been used. This device, shown in Fig. 4.1, has built-in 8-channel high-sensitive EMG and 9-Axis motion sensors.

The signal outputted by the armband is acquired with the acquisition tool of the armband itself via Bluetooth communication, then imported in the Simulink workspace using the serial reading function in the MATLAB Function block shown in Fig. 4.2.

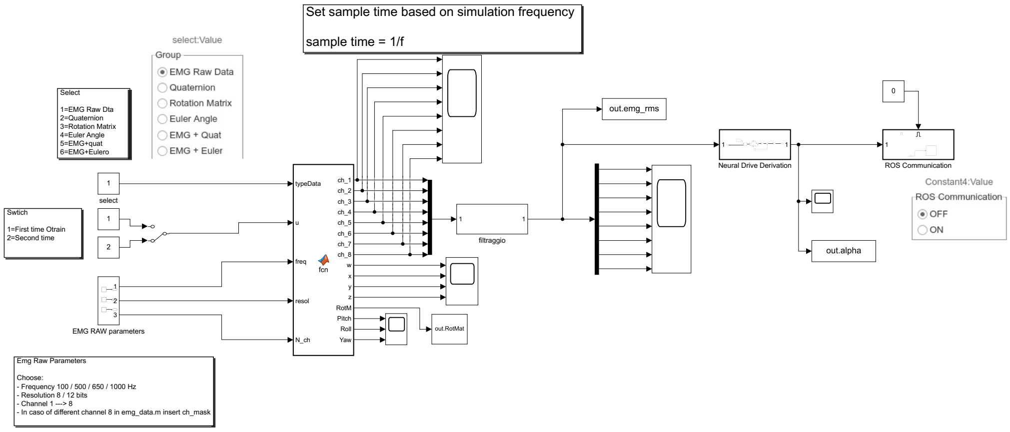


Figure 4.2: EMG Acquisition and Neural Drive Derivation

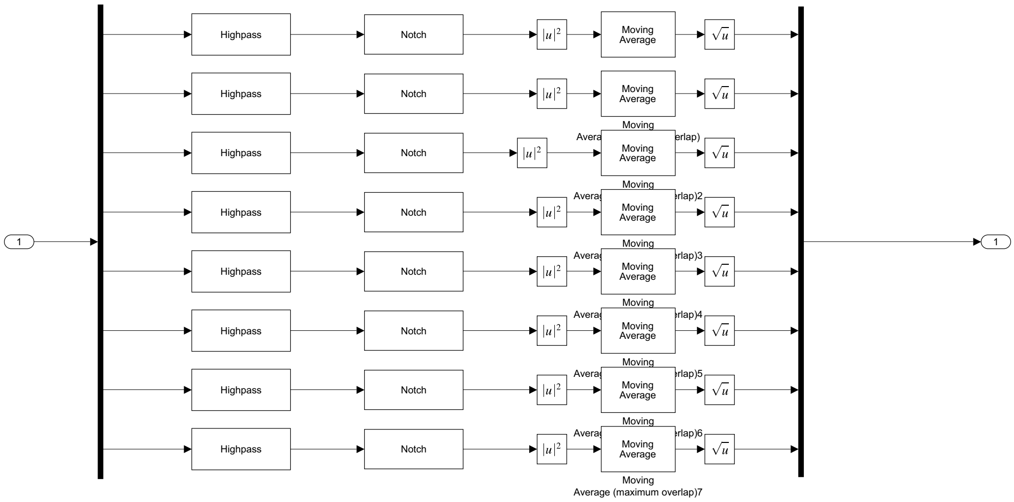


Figure 4.3: EMG Filtering Subsystem

For the filtering part, as can been seen in Fig. 4.3, each signal from the eight channels has been conditioned using a standard sequence of filters, composed by an high pass, a peak and notch and a moving average.

After filtering, the 8-dimensional EMG vector is used to derive σ^{ref}, as described in 4.1, then published on a ROS topic in order to be read by the control program running on another machine.

In Fig. 4.4 is shown an example of the signals acquired by the eight channels of the surface electromyography during a sequence of opening and closing of the hand by the user. It is important to point out the behaviour of all the channels in the firsts instants: all the signals have a spike due to the starting of the acquisition. This occurrence has to be taken into consideration in all the control scheme designed, as a matter of fact all of them will have a safe switch that will send a safe command to the hand. This switch will then disabled by the operator once the transient phase is concluded.

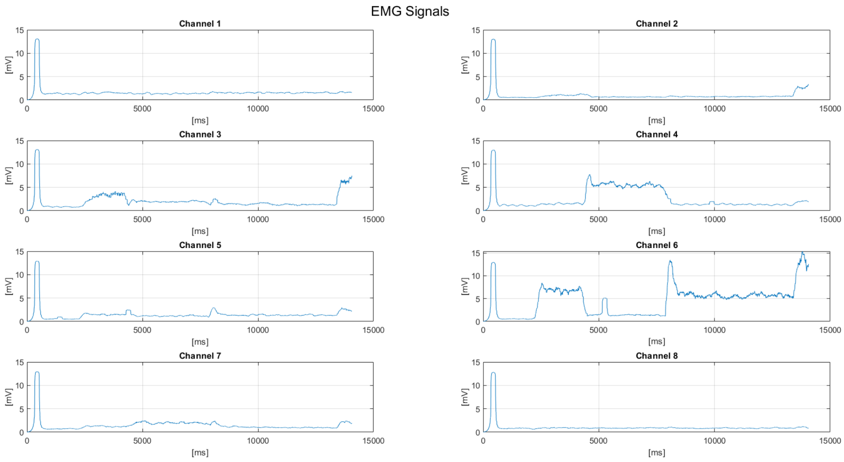


Figure 4.4: EMG Filtered Signals

4.3 Calibration

The most essential and delicate part in the myo-control field is the calibration phase. This phase is necessary in order to set some of the critical values in the transformation from EMG to control signal. The scheme used is the same shown in Fig. 4.2, with the precaution of blocking the ROS communication and initializing the parameters S_M^+, k_1, k_2 and k_{off} to a non-significant value.

As soon as the subject wear the armband on its forearm, a first training session is conducted. The calibration consists in registering the EMG data given by the a precise sequence of actions from the subject.

Starting from a resting position, shown in Fig. 4.5, the subject needs to slowly open and close their hand, maintaining the open and close pose for a couple of seconds, two consecutive times, then return to the resting position.

Note that, the execution of the training phase is crucial, since the data acquired are then used to determine the *muscular synergy matrix* S_M, the linear combination parameters k_1, k_2 and k_{off} and the thresholds for the detection of the commands.

To derive the *muscular synergy matrix* S_M related to the hand motion, is has been exploited the Non-negative Matrix Factorization algorithm (NFM)[12]. Using the acquired data the matrix $E \in \mathbb{R}^{8 \times n}$ is defined, containing n samples of the 8 channels. According to [13] E can be expressed as:

$$E = S_M U \tag{4.3}$$

where $S_M \in \mathbb{R}^{8 \times 2d}$ is the *muscular synergy matrix* and $U \in \mathbb{R}^{2d \times n}$ is the

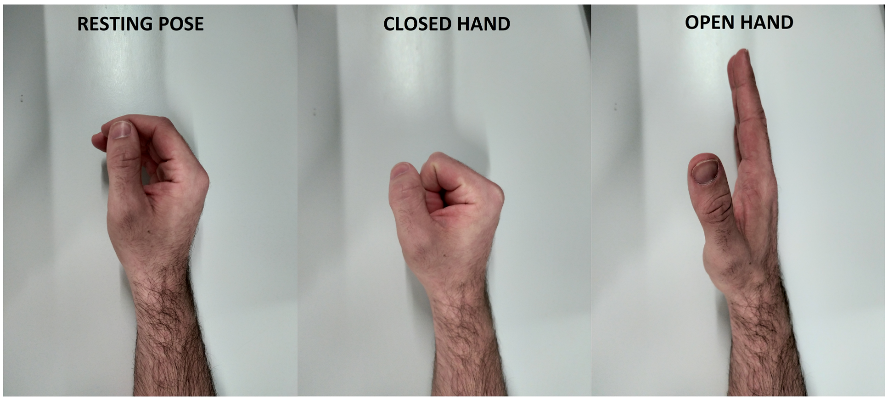


Figure 4.5: Calibration Hand Poses

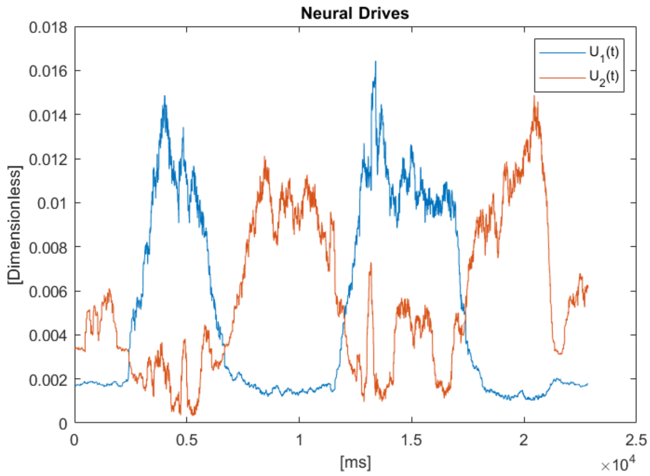


Figure 4.6: Calibration Neural Drives

neural drive matrix with d the number of DOFs controlled during the sEMG recording acquisition.

Since the proportional control approach presents instability when more than two DOFs are considered [13], the muscular synergies have been determined for only one postural synergy-based DOF: the coordinated opening and closing movements of the user's hand finger, meaning $d = 1$.

In the Fig. 4.6 is shown an example of the neural drive obtained from a calibration executed during one of the final tests. It can be observed that the two neural drives, referring to two antagonist group of muscles, have opposite behaviour, while one peaks the other has a trough.

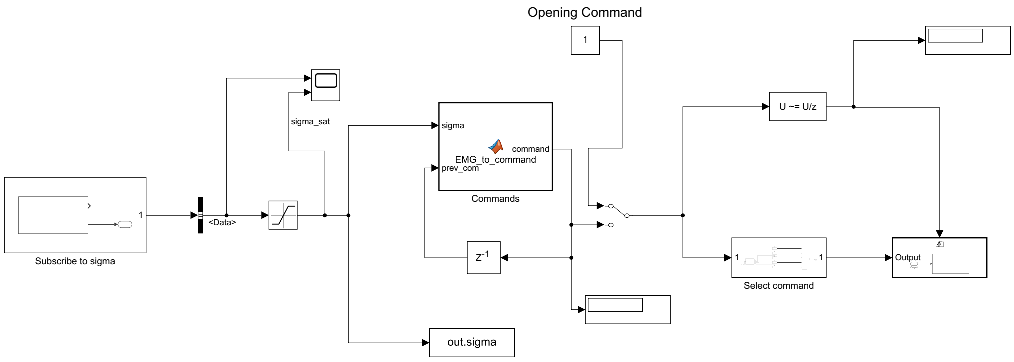


Figure 4.7: Control with Commands

4.4 Control with Basic Commands

The fist level of control exploits the commands given by the hand basic management itself. The scheme in Fig. 4.7 subscribes to the sigma ROS topic, reading on line the value of σ^{ref} extrapolated from the EMG acquisition, then select the appropriate command to send to the hand.

The scheme created allows to customize the thresholds that σ^{ref} has to surpass to send a certain command. Inside the MATLAB function has been implemented an hysteresis in order to counter the natural fluctuation of the signals around the thresholds. If the value of the σ^{ref} surpasses the high threshold, e.g. 0.8, the scheme publishes the closing command to the command ROS topic. When the signal falls below the lower limit, e.g. 0.5, the program publishes the opening command. In order to avoid the publishing too many commands the communication with ROS is executed only when the command changes.

In the Fig. 4.8 is shown an example of the values of σ^{ref} during a test

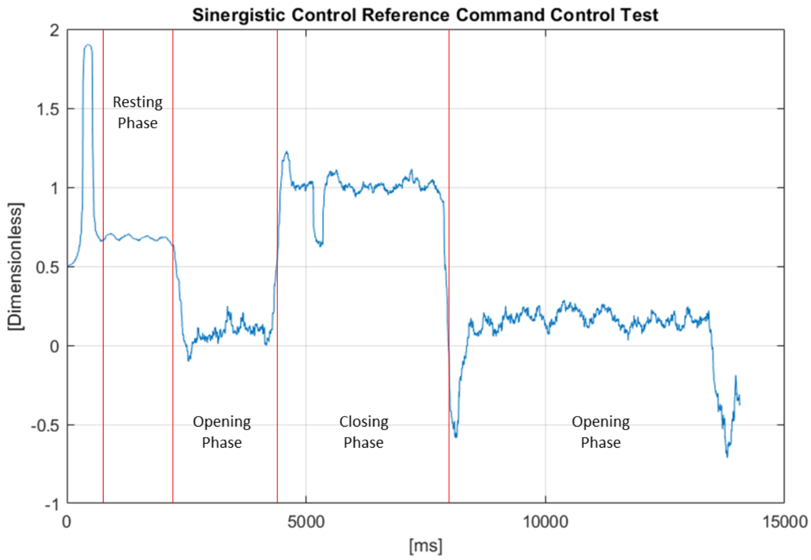


Figure 4.8: σ^{ref} in Command Control Test

with the command control scheme. After a brief spike, caused by the start of the acquisition, the subject opens, closes and then opens again their hand. It can be seen the behaviour of the σ^{ref} which reaches the value 1 during the closing phase and the values 0 during the opening phases. Moreover, it can be noticed that the values exceed the two threshold, this is and an intended behaviour needed to help the subject controlling the hand better. In this way the user can close the hand more vigorously to be certain to send the closing command or relax more the hand in case of the opening command. To avoid any problems due to this aspect, the signal received is saturated to 1, or 0, before any elaboration.

4.5 Continuous Control

In order to control the closure of the hand proportionally to the value of σ^{ref} the scheme shown in Fig. 4.9 has been developed. The goal of the scheme is to derive the matrix $M = [m_0 \cdots m_9]^T$, which contains the instantaneous values of the joints of the hand. This matrix then is published in the joints_values ROS topic to control the hand. The M matrix is derived as:

$$M = S_P \sigma^{ref} + S_{off} \tag{4.4}$$

where $S_P \in \mathbb{R}^{10}$ is called *postural synergy matrix*, which contains the

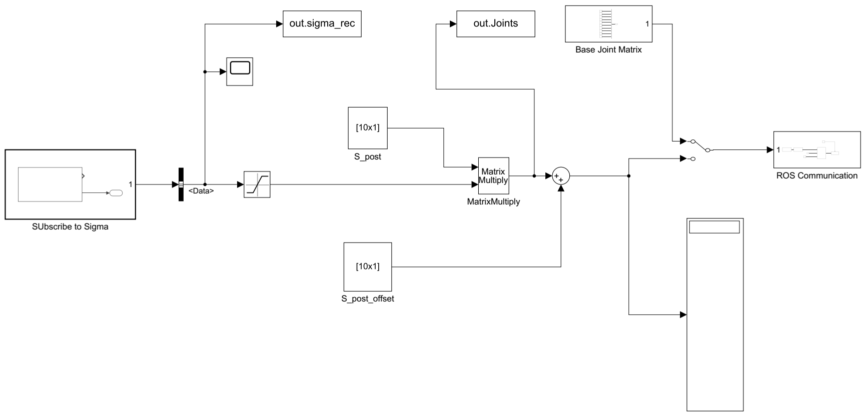


Figure 4.9: Continuous Control

postural synergy weights needed to coordinate the fingers positions during the opening and closure of the hand. The $S_{off} \in \mathbb{R}^{10}$ is a matrix containing the minimum values of the joints of the hand, used to assure that the values derived are inside the acceptable ranges of the joints.

Important to notice is the fact that, as pointed out before, there is a safe switch that allows the user to send the values for the open hand, this safe is needed for the initial instants of the acquisition because the EMG signals have an initial spike that would cause an abrupt closure of the apparatus. It is also needed in case any problem arises during the acquisition to block the hand in the opened configuration.

Chapter 5

Robotic Hand Sensing

The AR10 robotic hand does not provide any kind of feedback to the user. Because of that it is necessary to equip the manipulation system with a set of sensors that allows the user to have a measurement of the external environment. It has been chosen to add a series of tactile sensors that will allow the system to check the grasping force of the hand during the handling of different objects.

5.1 Tactile Sensors

In order to obtain a force feedback the hand has been equipped with three tactile sensors mounted on the thumb, index and middle finger, with custom support shown in Fig. 5.1.

This positioning has been chosen in order to perform a tripodal grasp to grab the objects. The sensors used are OptoForce 3D sensors (OMD-20-SE-40N), these sensors measure the magnitude and the direction of F_x, F_y, and F_z forces based purely on optical principles. Semi-spherical sensors

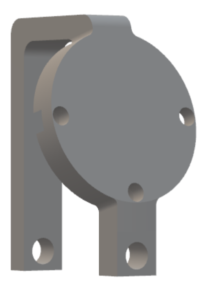


Figure 5.1: Sensor Custom Support

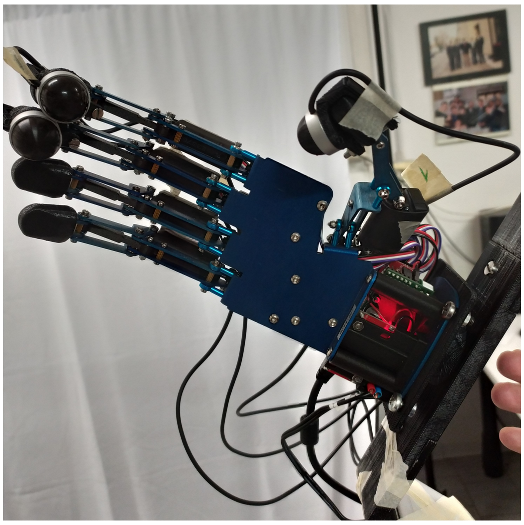


Figure 5.2: Hand with Sensors

are ideal as sensitive fingertips for humanoid robot hands, industrial grippers, harvesting robots, and due to its high durability there are various applications in the field of medical robotics (rehabilitation) and advanced robotics (e.g. exoskeletons) as well.

The tripodal grasp allows to grasp and rotate an object with the minimum number of finger involved, due to this choice the number of needed sensors has been reduced to three, as shown in Fig. 5.2.

5.2 Force Feedback

To implement the force feedback the scheme explained in section 4.5 has been expanded. The new control model shown in Fig. 5.3 acquires the data from the serial port of the force sensors and exploits them to manage the grasping of the objects.

After the acquisition the values of the force sensors are elaborated inside the Tripodal Grasp Subsystem, shown in Fig. 5.4, where, after a brief calibration phases needed to eliminate the constant values, are used to determine the locking of the fingers. When one of the values registered by the sensor exceeds a certain threshold, in one of the three direction, the function block the values for the joints of the implicated finger to the previous sent value. When all of the three fingers are locked the grasping phase is terminated, is then released when the user open their hand forcing the σ^{ref} to fall under a certain threshold, e.g. 0.15. The threshold can be customized to adapt to

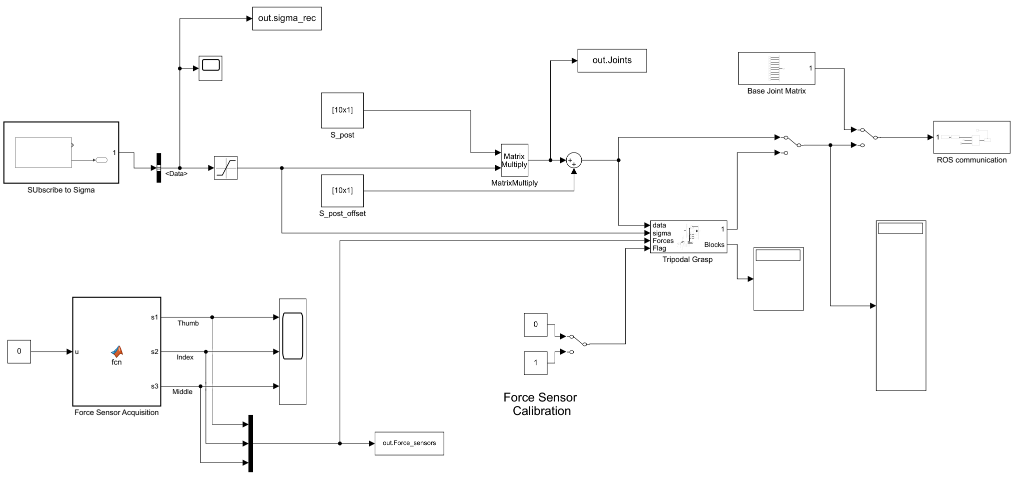


Figure 5.3: Continuous Control with Force Feedback

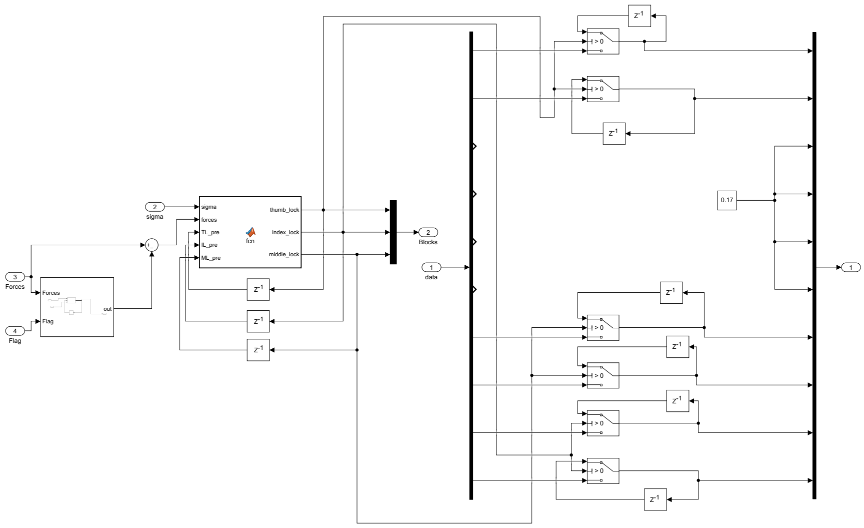


Figure 5.4: Tripodal Grasp Subsystem

the needed force to perform the grasp and the handling.

As can be seen in the upper part of Fig. 5.4, four out of the then component of J have been fixed to a constant minimum value, these components refer to the joints of the middle and pinky fingers which are blocked to the opened position.

Chapter 6

Experiment: Pilot User Study with Grasping Tasks

The last part of the project is the test of the control program for the hand in various circumstances. To make the hand wearable, a forearm support has been designed and printed. As it can be seen from Fig. 6.1 the support allows to put the robotic hand ahead of the subject real hand.

This arrangement lets the user control better the position of the manipulation device, considering it as an extension of their arm. This is also the desired position for the study because is the most user-friendly arrangement in case of a patient with an amputated or disabled hand.

The gForcePro+ EMG Armband was positioned slightly after the subject's elbow, to guarantee a better reading of the EMG signals.

6.1 Experiment Setup

The experiment has been divided in three parts. In each part the complexity of the setting was increased in order to let the subject get used to the system before performing complex tasks.

6.1.1 First Phase

In the first part of the experiment the user had to perform the calibration passages explained in 4.3, then they had to check all the functionalities of the hand, without wearing the forearm support.

Firstly, control the opening and closing of the hand via commands driven by the neural drive. In this instance the hand receive a closing commands when the value of the neural drive exceeds a certain value, here set as 0.8.

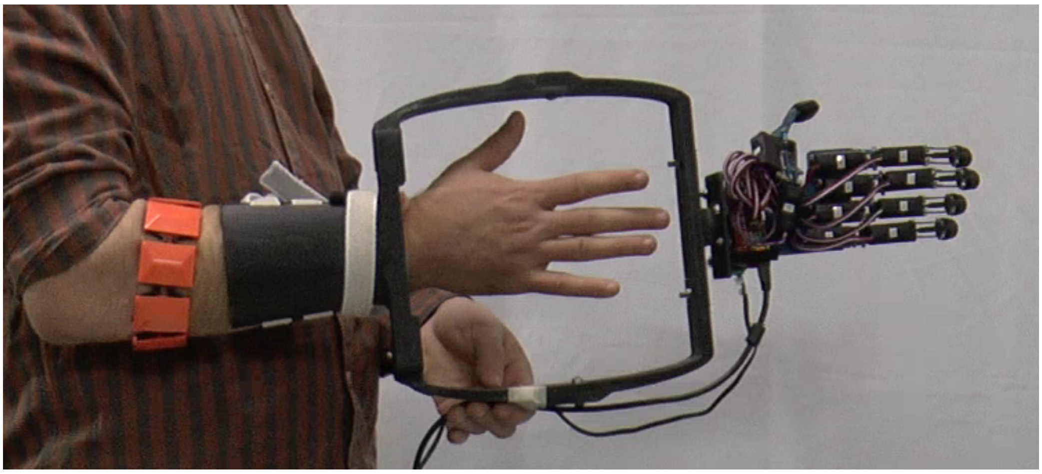
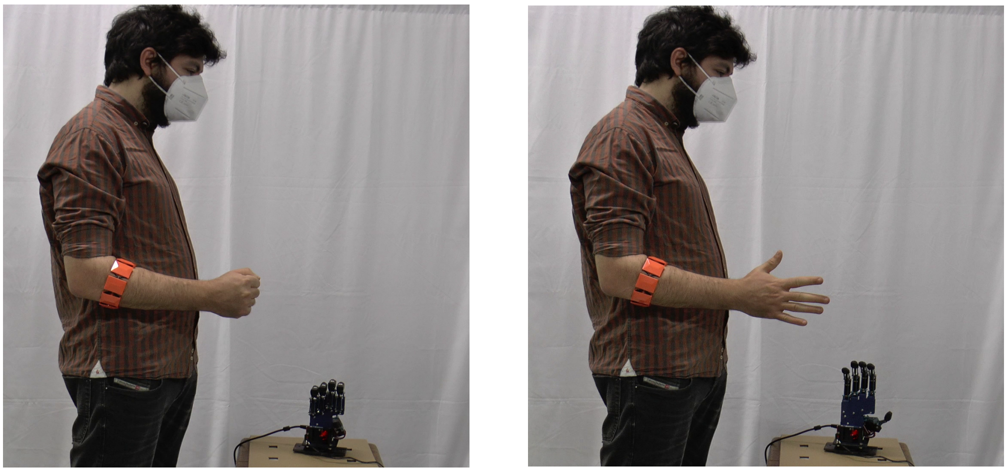


Figure 6.1: Hand Forearm Support

The opening command is sent when the neural drive comes back below the threshold set as 0.55.



(a) Closing Command (b) Opening Command

Figure 6.2: First Phase - Command Control

The user had to perform three closure and three opening of the hand, as shown in Fig. 6.2.

Subsequently, the user has to control the hand continuously, in this case, the hand closure is proportional to the neural drive and the joints of the hand are handled according to the postural synergy matrix, as outlined in section 4.5.

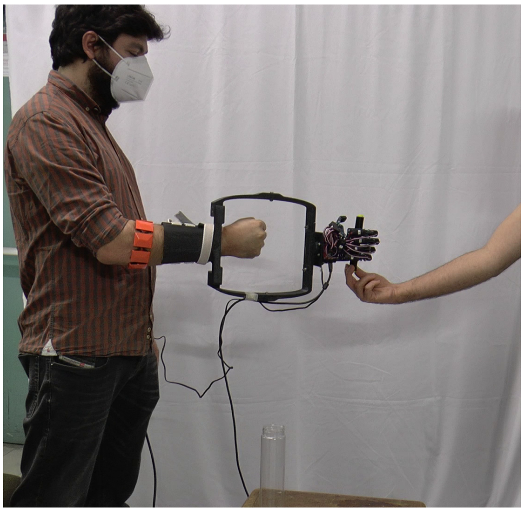
Figure 6.3: First Phase - Continuous Control

In this case, the user had to open and close the hand, as shown in Fig. 6.3, until they felt comfortable with the control.

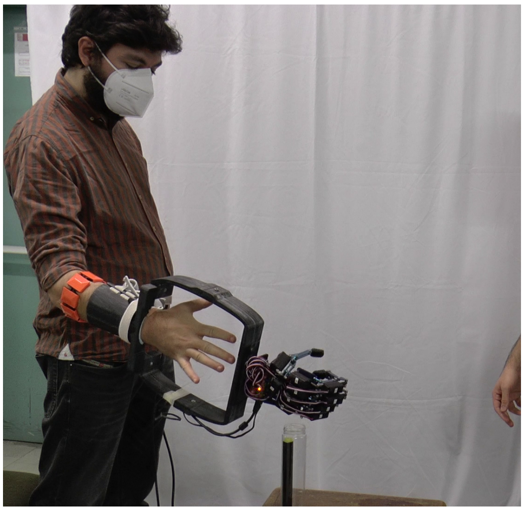
6.1.2 Second Phase

In the second phase, the hand was mounted on the support and then worn by the user.

In this stage, the subject is tasked to grasp an object, a sharpie, handed to him and then release it into a container positioned in two different spot. Firstly, the user grasp the sharpie to their left then release it in the container at their right at the same height, as shown in Fig. 6.4.



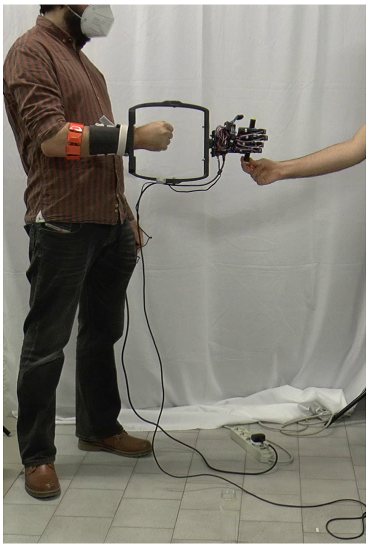
(a) Grasp Sharpie on the Left



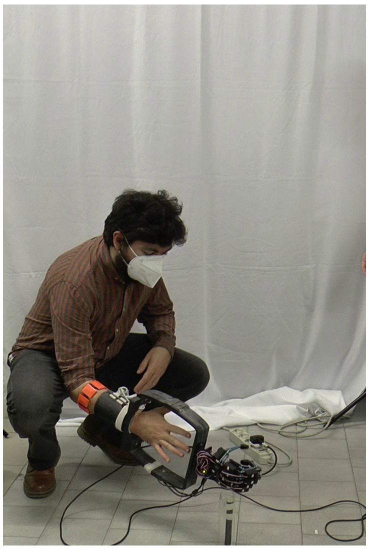
(b) Release Sharpie on the Right

Figure 6.4: Second Phase - Grasp Object from Left to Right

Therefore, the same operation is repeated with the container positioned on the ground, as shown in Fig. 6.5.



(a) Grasp Sharpie Standing Up



(b) Release Sharpie on the Ground

Figure 6.5: Second Phase - Grasp Object from Standing to the Ground

Both tasks have been repeated ten times in order to check the accuracy. This part of the experiment is needed to test the usefulness and the capability of the basic command control, exploiting the commands of the hand.

6.1.3 Third phase

For the third and final part of the test, the force sensors were installed on the hand as shown in Fig. 5.2.

In this part of the experiment the subject was required to grasp three different object with a tripodal grasp. The control program of the hand, as explained in section 5.2, is capable of detecting the contact with the object and lock the fingers in position. Then after the completion of the grasp the user needed to move the object in order to check the effectiveness of the grasp.

Figure 6.6: Third Phase - Test Objects

In Fig. 6.6 are displayed the three object considered in the experiment: a soft ball, a little cardboard box, and a paper cup; their sizes are exposed in the table below.

Object	Size
Ball	D 70 mm
Box	75×50×60 mm
Cup	d 55 mm D 70 mm h 58 mm

These three objects have been chosen to have an extensive range of different aspects with variations in shape, softness and deformability. The series of the three grasping was performed 10 times, each time with a random permutation of the three objects, in order to check the accuracy of the control program.

6.2 Results

In the first part of the experiment, the subject showed no problems in controlling the hand in both the command set up and in the continuous control

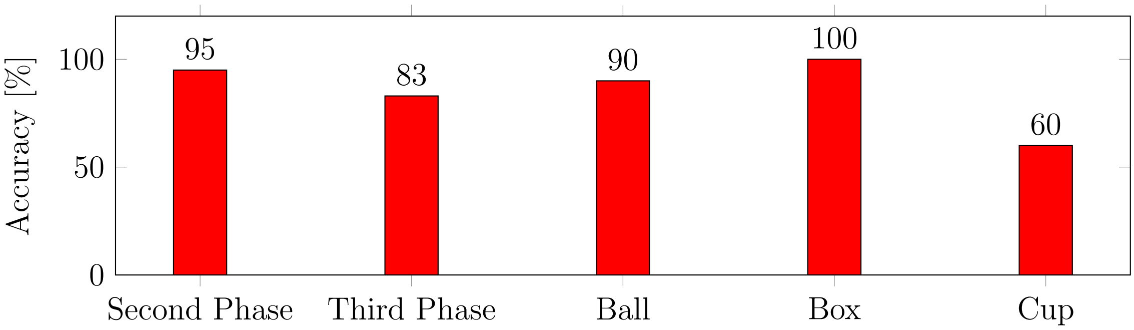


Figure 6.7: Tests Accuracy

set up. In this part the hand responded correctly to all of the commands given by the user. This part was necessary, other then for the calibration, also to let the subject get comfortable with the interactions with the whole system.

In the second part, the user performed successfully the grasp and release of the sharpie from left to right in all ten instances. In the case of the release on the ground the success rate was 90%. The total accuracy of the phase has been derived as 95% as shown in Fig. 6.7. In this setting, the user had minor problems with the positioning and the orientation of the robotic hand with respect to the container, causing to miss the the container during one of the releases.

For the third part, the total accuracy was of 83%, since 25 out of 30 grasps were considered a success. The most interesting results come out if the accuracy is calculated considering each object. In the case of the cardboard box the accuracy was 100%, for the ball 90%, but for the paper cup was 60%, as displayed in Fig. 6.7. It was noticed that the control program had no problem with a rigid box, since it did not deform during the grasp. While, for the paper cup, the failures were due to the deformation of the object caused by the tightness of the grasp. In the case of the ball the only failure was caused by the missing grasp which caused the drop of the ball. Important to notice that this was the only instance where the grasp was not assured, in all the other occasions the object remained in the grasp of the hand.

Chapter 7

Conclusions

In this thesis project a myo-controlled wearable manipulation system with tactile sensors has been successfully developed. The study of the system allowed to develop programs capable to derive direct and inverse kinematic of the robotic hand without the necessity of manual calculation. The designed wearable support system is capable of supporting the hand without graving excessively of the user. The control programs produced are capable of acquiring and translating correctly and reliably the sEMG signals of the user's forearm, exploiting both a basic command control and a more complex continuous control. The introduction of the force feedback, with the tactile sensor, enabled the development of a grasp control with an acceptable success rate, even with a different typologies of objects. The testing on a healthy subject introduced a interesting starting point for future prosthetic studies involving both intact and amputated subjects. Future development will involve the introduction of better and smaller force sensor in order to make the whole system lighter and more flexible. Moreover, the following step will be to include a vibrotactile stimulation feedback in order to allow the user to feel the force feedback and, eventually, other kinds of feedback. Another aspect to improve in the future will be the decrease of the delay in the control of the hand, which will be a feasible goal by adjusting some of minor features of the control programs.

Bibliography

[1] K. J. Zuo and J. L. Olson. “The evolution of functional hand replacement: From iron prostheses to hand transplantation”. In: *Plast Surg (Oakv)* 22 (2014).

[2] T. Iberall C. L. MacKenzie. *The Grasping Hand*. Elsevier Science B.V, 1994.

[3] S. A. Dalley, D. A. Bennett, and M. Goldfarb. “Preliminary Functional Assessment of a Multigrasp Myoelectric Prosthesis”. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Sept. 2012).

[4] C. Piazza et al. “A Century of Robotic Hands”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (May 2019).

[5] F. Cordella et al. “Literature Review on Needs of Upper Limb Prosthesis Users”. In: *Frontiers in Neuroscience* 10 (May 2016).

[6] E. R. Kandel, J. H. Schwartz, and T. M. Jessel. *Principals of Neural Science*. MCGraw-Hill, 2000.

[7] E. Criswell. *Cram’s Introduction To Surface Electromyography*. Jones and Bartlett Publishers, 2010.

[8] B. Siciliano et al. *Robotics*. Springer, 2009.

[9] *MATLAB Documentation*. MathWorks ®.

[10] L. A. Jones and S. J. Lederman. *Human Hand Function*. Oxford University Press, 2006.

[11] M. Santello, M. Flanders, and J. F. Soechting. “Postural Hand Synergies for Tool Use”. In: *The Journal of Neuroscience* 18 (Dec. 1998).

[12] R. Meattini et al. “An sEMG-Based HumanâRobot Interface for Robotic Hands Using Machine Learning and Synergies”. In: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 8 (July 2017).

[13] N. Jiang, K. B. Englehart, and P. A. Parker. “Extracting Simultaneous and Proportional Neural Control Information for Multiple-DOF Prostheses From the Surface Electromyographic Signal”. In: *IEEE Transactions on Biomedical Engineering* 56 (Apr. 2009).