UNIVERSITÀ
DEGLI STUDI
FIRENZE

sEMG-based control strategy for a Hand Exoskeleton System

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in

INDUSTRIAL ENGINEERING
Scientific Disciplinary Sector ING-IND/13

Candidate
Nicola Secciani

Nicola Secciani

Advisor
Prof. Benedetto Allotta

Co-advisor
Dr. Alessandro Ridolfi

Coordinator
Prof. Maurizio De Lucia

Department of Industrial Engineering
University of Florence
Florence, Italy

XXXII Ph.D. Cycle - Years 2016/2019

To Giuliano,
uncle and friend.

Tons of people have my gratitude for their contribution to my experience at the University of Florence during the last three years. Time has silently flown by as I grew up personally and academically. Knowing in advance that these few lines will never be enough, I will still try to show how much I am grateful for the opportunity, the teaching, the trust, and all the good received.

Here we go! First and foremost, I am indebted to my advisor, Professor Benedetto Allotta. It all started in Spring 2014 when he gave up one of his lunch breaks to explain Bode diagrams to a poor undergraduate student who was entering the world of Robotics and Automation for the first time. The interest and passion he inspired me in that context have been the driving force behind this journey. The possibility he gives me to continue to study and work in the field of Robotics for Medicine and Healthcare is what fuels my itinerary.

Additional thankfulness goes to my co-advisor, Doctor Alessandro Ridolfi, for he has been the one who welcomed me within his workgroup at the Mechatronics and Dynamic Modeling Laboratory of the Department of Industrial Engineering. His insightful comments, encouragement, and guidance through each of my steps have been crucial in defining the path of my research.

No one can do anything without backup from a great team! That is why a huge thanks goes to my lab mates, past and present. Thanks for having accepted me in my "exuberance", for the stimulating discussions, for each day or sleepless night we have worked together, and for all the fun we have had in these last three years. A special thanks to Matteo Bianchi, with whom I shared the passion for this project. I have felt like part of a beautiful extended family.

Key-role has been played by the exoskeleton "pilot", Massimo Bianchini. From the initial idea to the last tests, he has always been present with monumental patience. Thanks to his feedback and insights, he has genuinely accompanied my research activities, allowing them a remarkable quantum leap. Very little would have been possible without his priceless cooperation and help.

Staggeringly important have been, finally, my family, my girlfriend, and my lifetime friends. Mom and dad are the ones who made it possible to start this journey; sister, grandma, cousins, aunts, and uncles are the ones who embellished it with their presence. I could not be more grateful to them, for they have always supported me with warm love and endless backing. Thanks to Martina, my loving, supportive, encouraging, and patient girlfriend, for, in these last years, she has been becoming vital not only for my course of study but also for my life. Lastly, I would like to express my sincere gratitude to all of my dearest friends as growing up with them has shaped the person I am today.

Nicola Secciani. University of Florence, October 2019.

Data from the latest World Health Organization estimates paint a picture where one-seventh of the world population needs at least one assistive device. At the same time, clinical facilities are more and more overcrowded. On one side, due to the aging of the population, on the other side, thanks to the advances in medicine for which pathologies once certainly deadly, nowadays present a low mortality rate. However, only a small percentage of those who need an assistive or rehabilitative aid can get it properly because the healthcare system is globally under heavy pressure.

The early 2000s are also characterized by a marked technological drive which, starting in the middle of the twentieth century, took the name of the Fourth Industrial Revolution. Increasingly smaller processors deliver ever higher computing power, production processes are optimized, data transmission reaches impressive speeds, and technology becomes more and more accessible.

In this terrain, robotics is making its way through more and more aspects of everyday life, and robotics-based assistance or rehabilitation to physically impaired people are considered two of the most promising applications of this widely investigated technology. Providing high-intensity rehabilitative sessions or home assistance through low-cost robotic devices can be an effective solution to democratize some services otherwise not accessible. Simultaneously, it will also contribute to lower the burden over the healthcare system.

The work presented in this thesis has aimed to tackle the topic mentioned above by developing an innovative control strategy to be implemented on a low-cost hand exoskeleton system to support people suffering from hand disabilities during the activities of daily living. Most of the independence in everyday life is due in fact to the activities carried out using the hands; this is why restoring their dexterity when pathologically lost, is vitally important.

This work has been conducted starting from the solutions available within state of the art and following the main trends, heading to the development of an intuitive and easy to manage control strategy based on the intention recognition from surface electromyographic signals. Exploiting epidermal measurements of the myoelectric activity lends itself well to research activities as it is a non-invasive procedure and therefore is widely studied in the literature, including in the field of control of robotic devices. The application of these techniques to the real control of robotic devices is rarely addressed, however, and only a few cases are reported in the literature.

The main contribution of this activity is hence not only to propose a novel control strategy but also to provide a detailed explanation of its implementation into a real device. The performance of the resulting systems has been tested enrolling a patient suffering from spinal muscular atrophy in a pilot study.

Contents

List of Figures

Acronyms

ABS Acrylonitrile Butadiene Styrene

ADC Analog-to-Digital Converter

ADLs Activities of Daily Living

AI Artificial Intelligence

ANN Artificial Neural Network

AT Assistive Technology

CAD Computer Aided Design

DIEF Department of Industrial Engineering

DIP Distal InterPhalangeal

DNN Deep Neural Network

DOF Degrees Of Freedom

DT Decision Trees

EB Entropy-Based

FD Frequency Domain

FIC Family of International Classifications Network

FN False Negatives

FP False Positives

GUI Graphical User Interface

HDsEMG High Density Surface ElectroMyoGraphy

HES Hand Exoskeleton System

ICA Independent Components Analysis

ICF International Classification of Functioning, Disability and Health

IDE Integrated Development Environment

ISPO International Society for Prosthetics and Orthotics

IZ Innervation Zone

KNN K-Nearest Neighbors

MCP MetaCarpalPhalangeal

MDM Lab Mechatronics and Dynamic Modeling Laboratory

ML Machine Learning

MoCap Motion Capture

MU Motor Unit

MUAP Motor Unit Action Potential

PCA Principal Components Analysis

PCB Printed Circuit Board

PIP Proximal InterPhalangeal

RF Random Forest

RI Rehabilitation International

ROM Range Of Motion

RP Random Projection

sEMG surface ElectroMyoGraphy

SMA Spinal Muscular Atrophy

SVM Support Vector Machine

TD Time Domain

TN True Negatives

TP True Positives

UNECE United Nations Economic Commission for Europe

UNIFI University of Florence

USB Universal Serial Bus

WHO World Health Organization

Chapter 1

Introduction

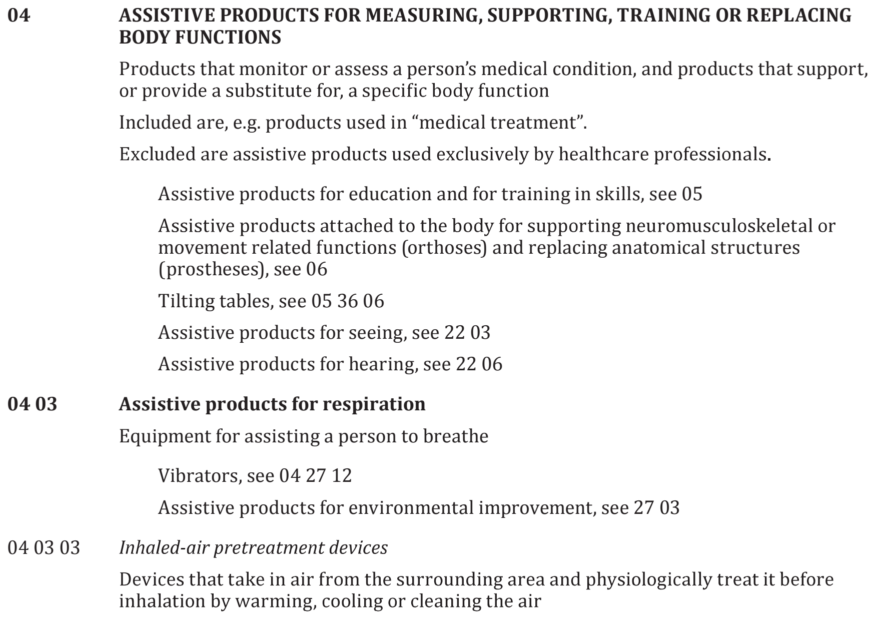
"We stand on the brink of a technological revolution that will fundamentally alter the way we live, work, and relate to one another."

K. Schwab. The Fourth Industrial Revolution. Foreign Affairs, 12TH Dec. 2015

Over the last years, this revolution — known as the "Fourth Industrial Revolution" or "Industry 4.0" and characterized by a range of new technologies that fuse the physical, digital and biological worlds together — has been increasingly impacting all disciplines, economies, and industries, and it has even been challenging the idea about what it means to be human. Driven by the tide of this change, robots have more and more permeated many aspects of human life to such an extent that even a delicate area as the Healthcare System has started consolidating the use of robotic devices as part of the so-called Assistive Technology (AT). Common examples of assistive technology are wheelchairs communication aids, prostheses, pill organizers, hearing, and memory aids.

Conventionally born at the dawn of the nineteenth century — when an Italian printer, Pellegrino Turri, designed a rudimentary typewriter for a blind friend — the AT has evolved over the years to obtain a first formal definition within the *Technology-Related Assistance for Individuals with Disabilities Act* which has become law under the US President Ronald Reagan in August 1988. Ten years later, the *United States Assistive Technology Act* has revisited what was written in the previous document to identify as AT, also called adaptive technology, any "product, device, or equipment, whether acquired commercially, modified or customized, that is used to maintain, increase, or improve the functional capabilities of individuals with disabilities."

Nowadays, the AT is a broad family of products adequately described within the International Standard ISO 9999, where its definition has been revised once again to match the terminology of the International Classification of Functioning, Disability and Health (ICF). As a part of the World Health Organization (WHO) - Family of International Classifications Network (FIC), the ISO 9999 defines as AT any "product (including devices, equipment, instruments, and software) especially produced or generally available, used by or for persons with disability to protect, support, train, measure or substitute for body functions, structures and activities, or to prevent impairments, activity limitations or participation restrictions". The same International Standard also offers a detailed three-level hierarchical classification — classes, subclasses, and divisions — of the AT products based on its function. Each class, subclass, or division consists of a three-pair digit code, a title, and notes and references to other parts of the classification if necessary (see Figure 1.1).



04 **ASSISTIVE PRODUCTS FOR MEASURING, SUPPORTING, TRAINING OR REPLACING BODY FUNCTIONS**

Products that monitor or assess a person's medical condition, and products that support, or provide a substitute for, a specific body function

Included are, e.g. products used in "medical treatment".

Excluded are assistive products used exclusively by healthcare professionals.

Assistive products for education and for training in skills, see 05

Assistive products attached to the body for supporting neuromusculoskeletal or movement related functions (orthoses) and replacing anatomical structures (prostheses), see 06

Tilting tables, see 05 36 06

Assistive products for seeing, see 22 03

Assistive products for hearing, see 22 06

04 03 **Assistive products for respiration**

Equipment for assisting a person to breathe

Vibrators, see 04 27 12

Assistive products for environmental improvement, see 27 03

04 03 03 *Inhaled-air pretreatment devices*

Devices that take in air from the surrounding area and physiologically treat it before inhalation by warming, cooling or cleaning the air

Figure 1.1: An excerpt from the ISO 9999 that shows how the classification is made: codes (on the left), titles, notes and references (on the right).

Six are the ISO 9999 editions that have been produced from 1992 to 2016 thanks to the collaboration with the International Society for Prosthetics and Orthotics (ISPO), the Rehabilitation International (RI), the United Nations Economic Commission for Europe (UNECE), and the WHO. Each version aimed to revise wrong titles or notes from the previous one, and add new ones in step with the latest technological trends. The last edition, the ISO 9999:2016, has 945 titles, of which about 44 are new, and 456 have been revised. It is worth noting that the standard reports that "all of the assistive products in this classification are primarily intended for use outside of health care settings; however, some of the products can be used in facilities such as rehabilitation centers". Furthermore, there is also a specific reference to the use of such devices in settings where employment-related activities are usually performed.

Recently, the WHO, within its report on disability [1] and its last action plan on disability [2], has pictured a detailed scenario of the current and future need of AT products all around the world. What emerges from these documents is a considerable demand for innovative solutions to tackle the issue of the disability. It is estimated that today more than one billion people need one or more assistive products — these numbers are even destined to double by 2030 due to the global population aging — however, only one out of ten people in need has access to assistive technology. The leading causes are high costs and a lack of awareness, availability, trained personnel, policy, and financing.

Such demographic, economic, social, technological, environmental, and political factors (DESTEP factors[1]) have paved the way to the advent of the Robotics for Medicine and Healthcare as a potent tool to overcome the availability limits of the standard AT [3, 4]. This branch of robotics is considered "the domain of systems able to perform coordinated mechatronic actions (force or movement exertions) on the basis of processing of information acquired through sensor technology, with the aim to support the functioning of impaired individuals, rehabilitation of patients, care and medical intervention of patients and also to support individuals in prevention programs", and can be divided in five different classes: (i) Robotics for medical interventions; (ii) Robotics supporting professional care; (iii) Robotics assisted preventative therapies and diagnosis; (iv) Robotic assistive technology; (v) Robotics for rehabilitation treatment.

[1] The DESTEP analysis is a model used for strategic decision-making over a process. It deals with external macroscopic factors. DESTEP stands for Demographic, Economic, Sociocultural, Technological, Ecological, and Politics. The purpose of such analysis is to cover all these six topics looking for strengths, weaknesses and possible correlations between them.

Examples of such systems are powered exoskeletons [5, 6], as the one shown in Figure 1.2, active prostheses [7, 8], or controlled robotic tools for surgery [9].

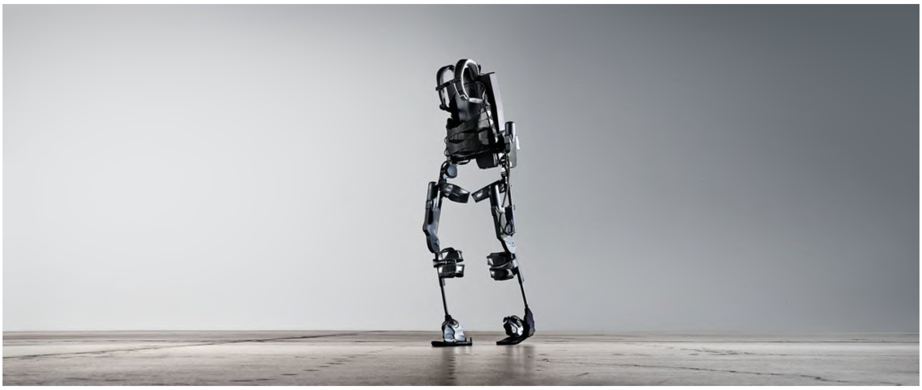


Figure 1.2: An example of an assistive robotic device. The figure shows a lower limb exoskeleton designed to assist people with motor disability during the gait.

Robotics for Medicine and Healthcare products offer many advantages over non-robotic ones. Having to "perform coordinated mechatronic actions", robotic devices generally incorporate an automatic control system to provide an intuitive and safe utilization. The control systems mentioned above are usually composed not only of hardware with computing power, such as processors but also of sensors, remote communication channels, and data storage capacity. The combination of these elements makes robotic devices capable of continuously monitoring the user's activity, making possible the fruition of remote services, as the telerehabilitation or the home assistance, otherwise difficult to achieve. Both telerehabilitation and home assistance would indeed positively impact on the Healthcare System, lightening the pressure that currently burdens on hospital facilities and rehabilitation centers, and, at the same time, it would also increase the time each patient can undergo their rehabilitation sessions. Besides, robotic devices are in general highly customizable and, thus, they can adapt well to different users or also to different needs of the same user — e.g., different rehabilitation exercises or different levels of assistance required. Precisely for the above-mentioned reasons, several studies in literature [10, 11] demonstrate that the correct use of such tools would remarkably improve the effectiveness of the rehabilitation therapies and the quality of life of people who need assistance.

However, as is natural, Robotics for Medicine and Healthcare also presents several flaws. Since it is still a relatively new discipline, the level of awareness in the population, the policies that manage and regulate its use, and the amount of investment and financing in this sector are so far meager. Consequently, the lack of a wholly developed dedicated industrial fabric for the production of these devices and their high level of manufacturing complexity cause their cost to remain very high and, thus, out of the reach of most of the people in need. Moreover, since such products are mainly designed to share the environment and to physically interact with people with disabilities, they have to meet strict requirements in terms of wearability, safety, and comfort. Therefore they represent also one of the most significant open points and one of the most challenging aspects in the human-robot interaction field.

The research activity presented within these pages stems from the desire to intervene in this delicate subject, aiming not only to develop innovative solutions in the field of robotic devices but also to democratize their use and dissemination thanks to the adoption of guidelines of simplicity and low-cost throughout the whole design process. In particular, the work summarized in this thesis focused on the challenging subject of assisting hand-impaired people employing a HESs and, going even further into detail, on the design, development, and testing of a dedicated control strategy which, exploiting sEMG signals, classifies and reproduces the user's motor intentions. The hand is, in fact, one of the most important providers of independence in everyday life, and, from the engineering point of view, it also represents a significant challenge both for the mechanical design and the control strategy because of its compound anatomy and the broad set of movements it can perform.

1.1 Overall framework

The work behind this thesis has been carried out at the MDM Lab of the DIEF of the UNIFI, operative in the field of Robotics for Medicine and Healthcare since 2013. In that year, with the development of the very first prototype of HES, the journey through the challenges and the open points of this subject began. Such exoskeleton was designed for a specific user: a person suffering from Spinal Muscular Atrophy (SMA) who proposed and voluntarily enrolled in the activity. The whole design process has been centered on his particular needs and requirements. That preliminary version represented, nonetheless, the first embodiment of a novel concept of a kinematic architecture which has been taken as a starting point for the improvements made in the following years.

Driven by the needs to enlarge the target group of the developed technology and to test it in a real case scenario, a collaboration with the Don Carlo Gnocchi Foundation[2] was born in 2016 — in particular with its rehabilitation center in Scandicci (Florence): the IRCCS Don Gnocchi. Since then, the Foundation has made the clinical expertise of its professionals, the use of equipment within the center, and the possibility to recruit patients for the joint research available to the MDM Lab researchers. The same collaboration, become part of the current research of the rehabilitation center, has continued over the years until today, providing fertile ground for the development of two other versions of the HES prototype and the birth of this thesis work. Despite the exploitation of sEMG signals, as a non-invasive control technique for wearable devices, is a well-known procedure, literature is still lacking in studies carried out on real patients outside the laboratory walls. At the time of writing, preliminary studies have been successfully concluded on a real patient, and other subjects are being recruited for testing campaigns, which will be the object of future projects.

As for the MDM Lab current research activity, two are the ongoing projects which involve the use of the HES: one, titled "HOLD: Hand exoskeleton system, for rehabilitation and activities Of daily Living, specifically Designed on the patient anatomy", funded by the University of Florence which will end in December 2019, the other, titled "BMIFOCUS: Brain Machine Interface in space manned missions: amplifying FOCUSed attention for error counterbalancing", funded by the Tuscany Region which will end in September 2020.

The two-year project HOLD aims to develop a HES that moves away from the laboratory to get as close as possible to the clinical environment in order to deal with real-case scenarios. Dysfunctions to the use of the hands can have pathological causes of a chronic (e.g., degenerative diseases) or incidental nature (e.g., strokes); in both cases, both rehabilitation and assistance are equally important since they are complementary in their purpose. While assistance aims to confer greater independence during the Activities of Daily Living (ADLs) and to increase the overall social interaction capabilities, on the other hand, the rehabilitative use of such device focuses on improving the outcome of the therapeutic approach. This project was born with a very practical spirit and, thus, the expertise of the researchers of the DIEF are merged with the expertise of the professional of the IRCCS Don Gnocchi to center the design process on the actual needs and requirements of real patients suffering from hand dis-

[2] The Don Gnocchi Foundation carries out rehabilitation activities and service for the elderly people in 28 residential structures and about thirty clinics spread in 9 Italian regions. https://www.dongnocchi.it

abilities. Started after the realization of the first exoskeleton prototype, the project focused in particular on improving its wearability, portability, and ease of use employing new mechanical design techniques — based on accurate ergonomic and structural studies to promote the prolonged use of the device — and through bio-inspired and flexible control strategies — to allow an intuitive and natural management of the device, both for rehabilitation and assistance. Finally, the fact that the resulting exoskeleton shall be much cheaper than other commercially available devices is another crucial point of the project.

The other two-year project BMIFOCUS aims instead to build up an integrated training platform to study the cognitive, sensory, and locomotor systems of pilots in extreme conditions such as piloting a spacecraft or a rover. Through the use of augmented reality, the end-users will be trained to manage stress and emotions in the face of possible contingencies. Such system will be composed of four subsystems: (i) a cerebral neurostimulation system which uses transcranial electrical stimulation techniques based on real-time analysis of biomedical signals to monitor and stimulate when needed the cognitive capabilities; (ii) an integrated acquisition system of electrophysiological and biomechanical signals; (iii) a simulation system of piloting tasks, that interfaces a virtual reality submodule with a haptic platform to reproduce one or more case scenarios through combined immersive experiences (motor, tactile, visual); (iv) a robotic rover to validate and test the whole system. In this framework, a bilateral HES shall be integrated with a training platform to reproduce the interaction forces with the virtual environments on the user's hand, e.g., during objects handling.

Figure 1.3: Logos of the ongoing projects at the MDM Lab of the UNIFI DIEF.

1.2 State of the art

Nowadays, HESs are proven solutions capable of speeding up the recovery process of the hand by giving the possibility to carry out high-intensity rehabilitative sessions, otherwise not achievable through the standard physiotherapy [12]. Such systems can also be used to monitor and record the patients' signs of progress by measuring suitable anatomic parameters, allowing for a more accurate evaluation of the rehabilitation status. Unfortunately, not always hand function is recovered even after the full treatment. There are cases [13, 14, 15, 16] in which post-stroke patients cannot regain the original dexterity of their affected arm. In such situations, the hand exoskeletons can still play a critical role by providing partial or complete assistance to the impaired limb. For both uses, the stringent requirements on weight and dimensions — which limit the electronics, the transmissible power, and the use of standard mechanical components — have meant that the currently available HESs do not show an outcome as positive as expected.

According to recent state of the art reviews [17, 18, 19, 20], four are the main ways to classify hand exoskeletons: (i) structure typology; (ii) number of Degrees Of Freedom (DOF); (iii) linkage system; (iv) type of actuation.

The first classification method separates HESs being based on whether they have a rigid or soft structure. Rigid exoskeletons — old and well-known typology — are generally made in plastic materials or metal, and they transmit the motion to the hand through rigid kinematic chains. The soft ones — new and emerging class — are instead made in inherently compliant materials, such as fabric and elastomers, and they typically act on the hand by directly changing their geometry and their shape [21, 22, 23, 24, 25]. The formers are preferable when high forces, accurate position, and fast dynamics are required, the latters when portability and comfort are the main requirements. For this reason, the use of the two different typologies is more complementary than substitutive [26].

The second classification is based on the number of DOF of the system. While the use of multi-DOF kinematic chains is largely reported [27, 28, 29], the use of rigid single-DOF ones is not large as well [30, 31]. The formers have complex kinematic architectures and well reproduce the natural hand movements, the latters have a simpler kinematics but struggle in following the hand motion. Due to their complicated functioning, multi-DOF solutions need complex setup and, thus, are mainly used for rehabilitative purposes since the portability requirement is not of crucial importance. This distinction applies only to rigid exoskeletons since the number of DOF in soft ones is not properly identifiable.

The third clustering parameter takes into account how the HES is connected to the fingers. This method groups the exoskeletons into the multi-phalanx ones, which independently act on the different phalanxes [32, 33], and the single-phalanx devices, which actuates only one phalanx [34]. In general, multi-phalanx and single-phalanx approaches are corresponding to a multi-DOF and single-DOF architecture and, therefore, they have the same pros and cons.

The fourth and final grouping relies on the type of actuators which can mainly be electric [35, 36] or pneumatic [37]. Electric actuators, unlike pneumatic ones, are usually more compact and transmit smaller forces to the hand.

The exoskeleton involved in this research activity is rigid, single-DOF, single-phalanx, and electrically actuated. Although it has not been the subject of this very thesis, the next chapter will go into detail about the mechanical evolution process that, over the years, has ensured that the HES in question can accurately reproduce the trajectory of the fingers.

An accurate study of the state of the art concerning possible control strategies has begun based right on such a result — which had already been partially achieved when the thesis started. In particular, since the device was already mechanically optimized to follow the trajectories of the fingers reliably, this first phase has focused on the fascinating field of recognizing and classifying the user's intention — regarded as the will to move a limb in a certain way — rather than searching for complete control strategies for robotic devices.

Latest trends in literature indicate the use of sEMG signals as one of the most promising ways to classify the user's motor intention [38, 39, 40, 41]. Many and different techniques have been investigated in recent times. Such methods are, more or less, based on the same procedure [42]: (i) collect time series of sEMG signals and split them into labeled time segments of an appropriate length; (ii) apply a filtering action to each segment and extract, if needed, time domain or frequency domain features (or a combination of them); (iii) use those features to train specific classifiers running pattern recognition algorithms, such as neural networks, support vector machines, hidden Markov models. Assuming that the existence of a two-way relationship between EMG patterns and the corresponding gestures is reasonable, the classifier output can then be translated into appropriate low-level commands for the actuation system of the HES in order to reproduce the classified intention.

While literature is well stocked with studies on sEMG classifiers [43, 44, 45, 46, 47, 48, 49, 50, 51, 52], not many are the available research works which, moving from the laboratory, are centered on the implementation of such methods to a real-use exoskeleton. This difference, which may seem small, is instead significant. The sEMG patterns modifications due to the use of an exoskeleton, or solely to its wearing, are indeed not negligible. Also, many refined techniques suggested by literature are not applicable in most cases: constraints on the complexity (e.g., number of DOF and number of sensors), wearability (e.g., setup for sensors and actuators, as well as the necessity to be as accessible as possible to a hand-impaired user), responsiveness (e.g., real-time requirements achievable on embedded electronics), and cost of the HES are also inherent limits to the classification strategy. As it will be observable in the detail of the other available works reported below, sophisticated strategies give way to more elementary approaches when it comes to the application to a real device. At the time of writing and to the best of the author's knowledge, few are the studies that can be found in literature that are strictly related to this thesis topic. The following selection will be analyzed based on three main characteristics: (i) sensors setup, in terms of number, positioning, and application on the user; (ii) classification method, in terms of classification criterion, classifiable gestures and exploited features; (iii) actuation protocol, discriminating between single-trigger and continuous-effort. Other aspects, such as the sensors typology and the assistive or rehabilitative use, will not be considered discriminant since the signals characterization and collection are not subject of this study.

Ochoa et al. (2011) - The control strategy presented in [53] exploits sEMG signals collected from two forearm muscular bands: the *extensor digitorum*[3] and from the *flexor digitorum*[4]. The number of sensors is not specified, and an external and trained supervisor performs their application on the patient. The control system, designed to classify only the opening and closing gesture of the hand, runs on a custom-made printed circuit board. It real-time compares the amplitude of each signal to a corresponding voltage threshold and classifies the relative intention every time one of the two limits is overcome. The device is then moved accordingly, and the entire Range Of Motion (ROM) is covered in percentile steps of 10%.

[3] The extensor digitorum is a muscular band of the posterior forearm, responsible for opening the four long fingers by acting principally on the metacarpophalangeal joints. Under continued action, it also extends the wrist, and then the elbow.

[4] The flexor digitorum muscular band, located in the anterior compartment of the forearm, can be split in two muscles: the flexor digitorum superficialis and the flexor digitorum profundus. It can be considered the antagonist of the extensor digitorum, and, therefore, it produces opposing joints torques and opposite movements.

Ho et al. (2011) - The method described in [54] uses again the muscular signal generated by the *extensor digitorum* to trigger the opening action of the HES. At the same time, unlike the previous case, the management of the closure is entrusted to the signals coming from the *abductor pollicis brevis*[5]. A trained person places one sensor on each muscle, and the classification is performed by comparing the amplitude of the signals with a threshold set at 20% of the user's maximum voluntary contraction of the corresponding muscle. Once the user's intention, namely, an opening or closing gesture, is classified, the actuation system is triggered to execute the entire movement at once. The whole control system runs on embedded hardware, fully wearable.

Leonardis et al. (2015) - Here in [55], an approach that involves the use of both hands is proposed: a bilateral HES for rehabilitation. It is meant to be used when the patient has only one impaired hand, since the driving sEMG signals are collected from the non-paretic limb. The sensor application still needs to be performed thanks to external intervention, and the involved muscular bands are those three mentioned above. However, one crucial difference concerning the two previous works is represented by the fact that the signals collection does not take place on the arm influenced by the exoskeleton and, thus, the exoskeleton itself does not modify the sEMG patterns on which the classifier works. In this work, the authors use a light neural network, implemented on a desktop PC, as the classifier, but no reference is given to the actuation method.

Meeker et al. (2017) - In [56], the authors present the application of a commercially available armband to the control strategy of a HES. The elastic armband includes eight sensors and, although it does not need to be precisely positioned by a trained person, it still requires an external intervention to be worn by the patient. A random forest algorithm, fed with the raw signals from the eight sensors, is in charge of the classification of the only two classifiable movements: opening and closing of all the fingers. Interesting that the classifier output is not directly translated into a command signal, but it is first processed through a median filter and then compared to two thresholds. In the paper, nor the hardware nor the actuation method are mentioned.

[5] The abductor pollicis brevis is a superficial muscle located on the palm, right under the base of the thumb. As the name suggests, it is responsible for the abduction of the thumb, defined as the movement towards the palm.

1.3 Contribution and thesis structure

Starting from the scenario mentioned above, the research activities presented within this thesis dealt with the topic of providing the low-cost assistive hand exoskeleton prototype previously developed within the MDM Lab of the UNIFI DIEF with an intuitive and reliable control strategy, which guaranteed the portability, wearability, and inexpensiveness of the device. As will be further discussed in Chapter 2, at the beginning of this work, the HES control system consisted of a two-button trigger, connected in open loop to the device. The necessity of using one hand to operate — by pressing the buttons — the exoskeleton placed on the other, and the complete lack of feedback on the user's hand state made the employment of the device unintuitive as well as unsafe.

From the analysis of the state of the art, it emerged that not only a feasible but also preferable and widely undertaken way to replace the trigger system could have been the acquisition and classification into appropriate motor intentions of the user's sEMG patterns. The main reason was the non-invasive and extraordinarily intuitive type of control that could be achieved; sEMG signals can be collected directly from the skin and are derivatives of the natural means by which the brain controls the body movement. However, even if literature presents many promising solutions about sEMG classifiers, only a small number of studies about the implementation of such decision-making algorithms on HES can be found. Focusing just on the few related works reported in the previous section, it can be seen that: (i) they all present control techniques which are based, when specified, on the classification of only three of the possible hand gestures — resting, and complete opening or closing; (ii) the independent control of the single finger is taken into account by none of them; (iii) the works reported by Ochoa et al. and Ho et al. are the only ones which explicitly indicate that the whole control system, including the classifier, runs on an embedded hardware — necessary condition for the HES to be portable and fully wearable; (iv) the number of total sensors varies in a range from two to eight and, in each case, they need to be positioned by an external person; (v) only the work of Ochoa et at. shows a continuous-effort actuation method, which, compared to the single-trigger ones, offers more control over the ROM and lends itself also to rehabilitation scenarios.

The main contribution of the work proposed in this thesis is to offer a solution that, embodying a suitable trade-off among the strategies mentioned above, overcomes some of the limits of the current state of the art, and it is ready to be part of a proper clinical trial after the approval of an ethics committee.

Such novel sEMG-based approach aims to merge the strengths of the already known techniques and introduce a new classification criterion never applied before to wearable robotics. In particular:

- it involves the use of sEMG sensors placed adequately in specific positions to maximize the selectivity of the measurements on the specific muscles and, thus, to reduce possible disturbances deriving from the activation of other muscle groups;
- it provides the possibility to classify only the three elementary hand motor actions (opening, closing and resting) to keep the complexity of the whole classification system low and to exploit a classifier computationally light enough to be run on embedded controller boards;
- it is based on a continuous-effort actuation method to offer a complete control over the entire ROM, to allow for intermediate poses of the hand, and to result as natural as possible;
- it is specifically designed for the clinical sector and, thus, it can be quickly and straightforwardly tuned on different patients' needs by non-technical staff through a dedicated and easy-to-use GUI developed in the framework of this research activity;
- it is integrated into a patient-friendly, low-cost and fully wearable HES.

Chapter 2 gives a detailed description of the state of the art of the HES developed by the MDM Lab of the UNIFI DIEF at the beginning of this thesis work. Since this research activity has been performed in a symbiotic and circular manner with the mechanical development of the device, this chapter also gives an overview of the mechanical modifications and improvements the exoskeleton has undergone until today.

Chapter 3, after a brief introduction of the physiology of the sEMG signals and the base principles of the classification theory, reports all the work done towards the implementation of the new control strategy: the changes made to the electronics of the system, the choice of the classification algorithm, and the development of two dedicated GUIs.

Chapter 4 the actual implementation and the testing of the new control strategy. The modifications to the old HES electronics also involved an onerous mechanical redesign of the device, and an overview of the final system will be given. The chapter will finally focus on the experimental tests that have been performed in a pilot study and their results.

Chapter 5 concludes the thesis, providing a critical analysis of the achievements and opening up scenarios for possible future developments.

Chapter 2

First-stage device

THE work described in this thesis started in 2016 from the need to implement a safe and effective control system on an assistive hand exoskeleton developed by the researchers of the MDM Lab of the DIEF over the previous three years [57, 58]. Starting from an in-depth kinematic study of the mechanisms that transmit motion to the fingers, the main features of such HES will be discussed to clarify its functioning and highlight the aspects on which the control system requirements (reported in Section 1.3) have been identified.

Section 2.1 deals with the closed-form forward kinematic study, conducted to mathematically characterized the mechanisms chosen to actuate the user's fingers. Such mathematical delineation has been the basis of an optimization procedure, implemented to geometrically optimize the same mechanism architecture to fit different hand sizes.

Section 2.2 describes the hinted optimization procedure, giving an overview of the idea behind and presenting in short the process itself. Details are left to the in-depth analysis presented in [59].

Section 2.3 finally describes the HES which embodies the aforementioned kinematic analysis and optimization procedure. The end of the section emphasizes the main flaws found according to the results of several intermediate clinical tests performed on a single patient at the IRCCS Don Gnocchi[1] under the supervision of health professionals.

[1] The Don Carlo Gnocchi Foundation has agreed to the execution of the tests upon the presentation of an informed consent form signed by the involved subject (available on request).

2.1 Kinematic synthesis

The first HES prototype was composed of five different parts: the housing for motors and electronics, and four independent planar mechanisms for the transmission of motion to the index, middle, ring, and little finger. This section is dedicated to the kinematics of the index mechanism, and to the scaling procedure implemented to adapt it to the other fingers. It will finally be discussed the possibility of using the scaling procedure not only to fit different finger dimensions but also directly to different hand sizes. The main purposes of this analysis were to mathematically characterized and, later on, numerically optimize a chosen single-DOF mechanism to make it reproduce, as accurately as possible, the complex trajectories of the fingers resulting from their multi-DOF structure. The choice of focusing on the optimization of a single-DOF mechanism — henceforth called "finger mechanism" — instead of opting for the exploitation of a multi-DOF one has been made according to the guidelines of wearability, simplicity, and cheapness imposed by the faced scenario. Such a decision has represented an effective solution in terms of functionality and manufacturability, leading to a smaller and lighter HES, easier to produce, actuate and control, and, thus, cheaper.

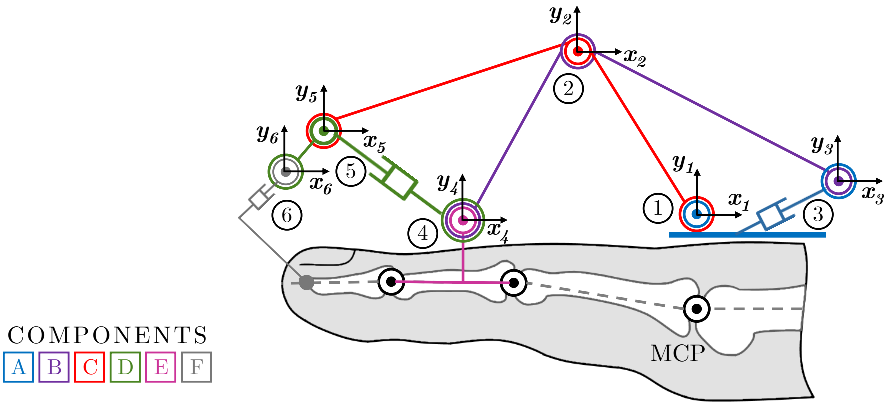


Figure 2.1: The graphical representation of the 2D kinematic chain of the finger mechanism. Joints are identified by circled numbers, while the different links are named according to the colors legend in the low left corner.

The study of the forward kinematics of the finger mechanism will be presented referring to what shown in Figure 2.1, where:

- ①, ② and ④ are revolute joints;
- ③, ⑤ and ⑥ are pin slot joints;
- ① is the only actuated joint and controlled DOF;
- reference systems x_1y_1 is integral with component A, x_2y_2 and x_5y_5 with component C, x_3y_3 with component B, x_4y_4 with component D, x_6y_6 with component F and x_ey_e with component E;
- x_1y_1, whose correct positioning is at a known distance on the vertical above the MetaCarpalPhalangeal (MCP) joint of the hand, is the only fixed frame against which the pose of all the others is calculated;
- component A, E and F are respectively integral with the back of the hand, with the intermediate phalanx (henceforth also called middle phalanx), and with the distal phalanx;
- $\mathbf{O}_4$, namely the origin of x_4y_4 and actual end effector of the mechanism, is the main interaction point between the finger and the exoskeleton[2].

The investigation will mainly focus on reconstructing the trajectory of the origin of frame x_4y_4 reported in frame x_1y_1 (${}^1\mathbf{O_4}$) as the joint 1 variable varies. Analytically knowing such a trajectory is indeed the first step to optimize the geometry of the whole finger mechanism, as it will be shown in the next section. However, since they will be required later, the calculation of the trajectory of the other remarkable points of the finger mechanism (${}^1\mathbf{O}_2$, ${}^1\mathbf{O}_3$, ${}^1\mathbf{O}_5$ and ${}^1\mathbf{O}_6$) will also be presented here below.

Analysing the kinematic chain of the finger mechanism the following eight independent relations can be identified:

$$\underline{{}^1\mathbf{O}_1} = {}^1\mathbf{O}_2 + \mathbf{R}_2^1\ \underline{{}^2\mathbf{O}_1} \tag{2.1}$$

$$ {}^1\mathbf{O}_2 = {}^1\mathbf{O}_3 + \mathbf{R}_3^1\ \underline{{}^3\mathbf{O}_2} \tag{2.2}$$

$$ {}^1\mathbf{O}_4 = {}^1\mathbf{O}_3 + \mathbf{R}_3^1\ \underline{{}^3\mathbf{O}_4} \tag{2.3}$$

$$ {}^1\mathbf{O}_5 = {}^1\mathbf{O}_2 + \mathbf{R}_2^1\ \underline{{}^2\mathbf{O}_5} \tag{2.4}$$

[2] Component F is an idle thimble which does not impact on the kinematic chain of the device. It is a passive element, added to accompany the motion of the distal phalanx.

$$ {}^{1}\mathbf{O}_5 = {}^{1}\mathbf{O}_4 + \mathbf{R}_4^1 \, {}^{4}\mathbf{O}_5 \tag{2.5}$$

$$ {}^{1}\mathbf{O}_6 = {}^{1}\mathbf{O}_4 + \mathbf{R}_4^1 \, \underline{{}^{4}\mathbf{O}_6} \tag{2.6}$$

$$ \underline{a_3}\,{}^{1}O_3^x + \underline{b_3}\,{}^{1}O_3^y + \underline{c_3} = 0 \tag{2.7}$$

$$ \underline{a_5}\,{}^{4}O_5^x + \underline{b_5}\,{}^{4}O_5^y + \underline{c_5} = 0 \tag{2.8}$$

where the generic vector ${}^{j}\mathbf{O}_i = \left({}^{j}O_i^x \; {}^{j}O_i^y \; {}^{j}O_i^z\right)^T \in \mathbb{R}^3$ denotes the position of O_i in $x_j y_j$, the generic matrix $\mathbf{R}_i^j$ represents the rotation of $x_i y_i$ with respect to $x_j y_j$, (a_3, b_3, c_3) and (a_5, b_5, c_5) are coefficients dependent on the linear constraints of the pin slot joints 3 and 5, and the terms referring to known geometrical quantities are underlined. Since the mechanism moves on a fixed plane, all the ${}^{j}O_i^z$ components are imposed equal to zero, and all the rotation matrices will result in elementary rotations about $\mathbf{z_i}$ axis through an angle α_i.

The kinematic state vector can now be defined as:

$$\mathbf{q} = \left[{}^{1}\mathbf{O}_2^T \; {}^{1}\mathbf{O}_3^T \; {}^{1}\mathbf{O}_4^T \; {}^{1}\mathbf{O}_5^T \; {}^{1}\mathbf{O}_6^T \; \alpha_2 \; \alpha_3 \; \alpha_4\right]^T \in \mathbb{R}^{18} \tag{2.9}$$

This vector contains the position of all the points of interest of the mechanism and three scalar angles of which one, α_2, represents the joint variable 1 and, thus, it is the control variable of the system. The elements inside the state vector can be calculated by solving the Equations 2.1 - 2.8 as function of $\alpha_2 \in \left[0 \; \frac{\pi}{2}\right]$ itself and of all the geometrical quantities of the generic mechanism enclosed in:

$$\mathbf{S} = [{}^{2}\mathbf{O}_1^T \; {}^{3}\mathbf{O}_2^T \; {}^{2}\mathbf{O}_5^T \; {}^{3}\mathbf{O}_4^T \; {}^{4}\mathbf{O}_6^T \; a_3 \; b_3 \; c_3 \; a_5 \; b_5 \; c_5]^T \in \mathbb{R}^{21} \tag{2.10}$$

In light of these premises, let $\tilde{\mathbf{q}} \in \mathbb{R}^{17}$ be the vector composed of all the unknown elements of $\mathbf{q}$ so that $\mathbf{q} = [\tilde{\mathbf{q}} \; \alpha_2]$, and let $\mathbf{f}(\alpha_2, \mathbf{S}) \in \mathbb{R}^{17}$ be the vector function composed by Equations 2.1 - 2.8, it is possible to solve the problem of the forward kinematics of the mechanism, $\tilde{\mathbf{q}} = f(\alpha_2, \mathbf{S})$, as follows. From henceforth, fixed geometrical quantities will no longer be underlined and every element of the vector $\tilde{\mathbf{q}}$, solved in terms of α_2, elements of $\mathbf{S}$ and already calculated terms, will be boxed to indicate that it is a known quantity and, as such, shall be considered in the resolution of the subsequent equations.

The vector Equation 2.1 can be split into two scalar equations[3]:

$$0 = {}^1O_2^x + c\alpha_2 {}^2O_1^x - s\alpha_2 {}^2O_1^y \tag{2.11}$$

$$0 = {}^1O_2^y + c\alpha_2 {}^2O_1^y + s\alpha_2 {}^2O_1^x \tag{2.12}$$

and thus:

$$\boxed{{}^1O_2^x = -\left(c\alpha_2 {}^2O_1^x - s\alpha_2 {}^2O_1^y\right)} \tag{2.13}$$

$$\boxed{{}^1O_2^y = -\left(c\alpha_2 {}^2O_1^y + s\alpha_2 {}^2O_1^x\right)} \tag{2.14}$$

Then, dividing Equation 2.2:

$${}^1O_2^x = {}^1O_3^x + c\alpha_3 {}^3O_2^x - s\alpha_3 {}^3O_2^y \tag{2.15}$$

$${}^1O_2^y = {}^1O_3^y + c\alpha_3 {}^3O_2^y + s\alpha_3 {}^3O_2^x \tag{2.16}$$

placed these two as a system with Equation 2.7, the followings are obtained:

$$\boxed{{}^1O_3^y = \frac{-A + \sqrt{(A)^2 - \left[\left(\frac{b_3}{a_3}\right)^2 + 1\right] H}}{\left(\frac{b_3}{a_3}\right)^2 + 1}} \tag{2.17}$$

$$\boxed{{}^1O_3^x = -\frac{1}{a_3}\left(b_3 {}^1O_3^y + c_1\right)} \tag{2.18}$$

$$\boxed{s\alpha_3 = \frac{-{}^1O_2^x + {}^1O_3^x + \frac{{}^3O_2^x}{{}^3O_2^y}\, {}^1O_2^y - \frac{{}^3O_2^x}{{}^3O_2^y}\, {}^1O_3^y}{{}^3O_2^y + \frac{\left({}^3O_2^x\right)^2}{{}^3O_2^y}}} \tag{2.19}$$

[3] As already mentioned, the z-component of all vectors is imposed equal to zero and, therefore, the calculation of such elements will not be taken into account.

$$\boxed{c\alpha_3 = \frac{{}^1O_2^y - s\alpha_3{}^3O_2^x - {}^1O_3^y}{{}^3O_2^y} \qquad (2.20)}$$

where:

$$A = \left(\frac{b_3}{a_3}{}^1O_2^x + \frac{b_3c_3}{{a_3}^2} - {}^1O_2^y\right) \tag{2.21}$$

and:

$$H = \left({}^1O_2^x\right)^2 + \left({}^1O_2^y\right)^2 - \left({}^3O_2^x\right)^2 - \left({}^3O_2^y\right)^2 + \frac{2b_3}{a_3}{}^1O_2^x + \left(\frac{c_3}{a_3}\right)^2 \tag{2.22}$$

Equations 2.3, 2.4 and 2.5 can now be split into:

$$\boxed{{}^1O_4^x = {}^1O_3^x + c\alpha_3{}^3O_4^x - s\alpha_3{}^3O_4^y \qquad (2.23)}$$

$$\boxed{{}^1O_4^y = {}^1O_3^y + c\alpha_3{}^3O_4^y + s\alpha_3{}^3O_4^x \qquad (2.24)}$$

$$\boxed{{}^1O_5^x = {}^1O_2^x + c\alpha_2{}^2O_5^x - s\alpha_2{}^2O_5^y \qquad (2.25)}$$

$$\boxed{{}^1O_5^y = {}^1O_2^y + c\alpha_2{}^2O_5^y + s\alpha_2{}^2O_5^x \qquad (2.26)}$$

$${}^1O_5^x = {}^1O_4^x + c\alpha_4{}^4O_5^x - s\alpha_4{}^4O_5^y \tag{2.27}$$

$${}^1O_5^y = {}^1O_4^y + c\alpha_4{}^4O_5^y + s\alpha_4{}^4O_5^x \tag{2.28}$$

In order to proceed in the calculation of $c\alpha_4$ and $s\alpha_4$, the last two equations have to be placed as a system with Equation 2.8 and, then, it can be solved similarly to the previous one, resulting in:

$${}^4O_5^y = \frac{-\left(b_5c_5\right) + \sqrt{\left(b_5c_5\right)^2 - \left({b_5}^2 + {a_5}^2\right)T}}{{b_5}^2 + {a_5}^2} \tag{2.29}$$

$${}^4O_5^x = -\frac{1}{a_5}\left(b_5{}^4O_5^y + c_5\right) \tag{2.30}$$

$$\boxed{c\alpha_4 = \frac{{}^1O_5^x + s\alpha_4{}^4O_5^y - {}^1O_4^x}{{}^4O_5^x} \qquad (2.31)}$$

$$\boxed{s\alpha_4 = \frac{1}{\frac{\left({}^4O_5^y\right)^2}{{}^4O_5^x} - {}^4O_5^x}\left({}^1O_5^y - {}^1O_4^y - \frac{{}^4O_5^y}{{}^4O_5^x}{}^1O_5^x + \frac{{}^4O_5^y}{{}^4O_5^x}{}^1O_4^x\right)} \tag{2.32}$$

where:

$$T = -{a_5}^2\left[\left({}^1O_5^x - {}^1O_4^x\right)^2 + \left({}^1O_5^y - {}^1O_4^y\right)^2\right] + {c_5}^2 \tag{2.33}$$

Finally, the two scalar equations coming from the breakdown of the remaining Equation 2.6 complete the kinematic analysis:

$$\boxed{{}^1O_6^x = {}^1O_4^x + c\alpha_4{}^4O_6^x - s\alpha_4{}^4O_6^y} \tag{2.34}$$

$$\boxed{{}^1O_6^y = {}^1O_4^y + c\alpha_4{}^4O_6^y + s\alpha_4{}^4O_6^x} \tag{2.35}$$

The kinematics of the mechanism is now completely described and the trajectory of ${}^1\mathbf{O}_4$ is reported in Figure 2.2.

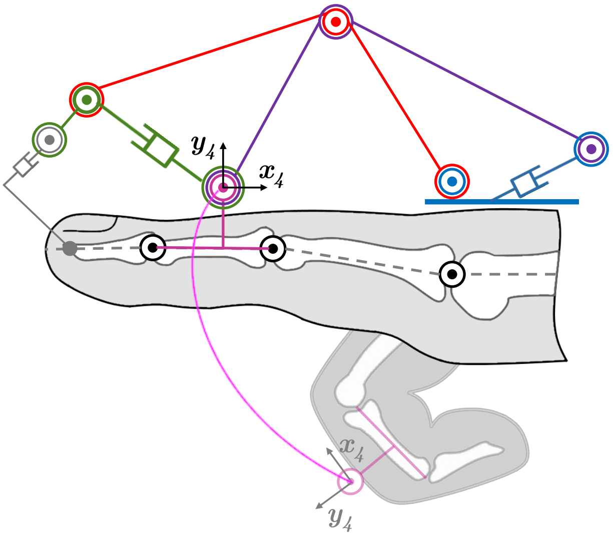


Figure 2.2: In pink, the trajectory ${}^1\mathbf{O}_4(\alpha_2, \mathbf{S})$ with $\alpha_2 \in \left[0\ \frac{\pi}{2}\right]$.

2.2 Optimization procedure

Once identified the trajectory ${}^{1}\mathbf{O}_4(\alpha_2, \mathbf{S})$ through the analysis of the forward kinematics of the mechanism, the following step is to reconstruct the trajectory of the same point starting from the kinematics of the finger. By comparing these two, it is possible to study the quality of the coupling between the exoskeleton and the hand, and optimize the mechanism by modifying the geometrical terms contained within the vector $\mathbf{S}$ to minimize the displacements.

As reported in Figure 2.2, the connection point between the hand and the exoskeleton (identified as the point $\mathbf{O}_4$) can be assumed placed approximately on the middle-upper part of the intermediate phalanx. The first step of the optimization procedure is to track the natural motion of such a point through a Motion Capture (MoCap) analysis of the user's index finger. The exploitation of the MoCap technique to define the objective function to optimize is a remarkable advantage since, downstream of a fully non-invasive procedure, the physiological finger motion can be tracked. Such tracking is possible by recording video images of markers placed on the lateral side of the right index finger in correspondence with the MCP joint, theProximal InterPhalangeal (PIP) joint, and the Distal InterPhalangeal (DIP) joint — the marker positioned on the MCP joint, assumed fixed in position during finger motion, constitutes the origin of the reference frame in which the trajectory of the other markers are calculated.

The acquired images are then processed through the open-source software Kinovea to extract the coordinate vectors of the trajectories of the markers as the finger moves. Henceforth, ${}^{M}\mathbf{O}_D$, ${}^{M}\mathbf{O}_P$, and ${}^{M}\mathbf{O}_4$ will indicate, respectively, the position of the DIP joint, the PIP joint, and the connection point calculated in $x_M y_M$. The position of the interaction point ${}^{M}\mathbf{O}_4$ cannot be directly extracted from the MoCap analysis but, assuming that component E and the intermediate phalanx constitute a single rigid body when the exoskeleton is coupled to the hand, it is possible to reconstruct the trajectory of a virtual marker placed on $\mathbf{O}_4$ starting from the trajectories of $\mathbf{O}_D$ and $\mathbf{O}_P$. In detail:

$$ {}^{M}\mathbf{O}_4^h = \frac{{}^{M}\mathbf{O}_D^h + {}^{M}\mathbf{O}_P^h}{2} + R_{z,\frac{\pi}{2}} \frac{m}{2} \frac{{}^{M}\mathbf{O}_D^h - {}^{M}\mathbf{O}_P^h}{||{}^{M}\mathbf{O}_D^h - {}^{M}\mathbf{O}_P^h||} \tag{2.36} $$

where the first addend represents the position of the midpoint of the second phalanx and the second addend is equivalent to a shift of half of the phalanx thickness (m) in the direction perpendicular ($R_{z,\frac{\pi}{2}}$) to the phalanx axis, namely the joining between DIP and PIP. The index h identifies a specific time instant because the trajectories of DIP and PIP joint are not analytical functions but time series of subsequent positions. However, in order to compare the trajectory just calculated with the one output by the forward kinematics other two more steps are required: (i) rewrite such trajectory in $x_1 y_1$; (ii) interpolate the new trajectory to get a continuous function to be then resampled at certain α_2 values. As shown in Figure 2.3, the trajectory of ${}^{M}\mathbf{O}_4$ can be easily reported to $x_1 y_1$ by adding the displacement vector ${}^{1}\mathbf{O}_M$, which is a fixed and know quantity:

$$ {}^{1}\mathbf{O}^{h}_{H} = {}^{M}\mathbf{O}^{h}_{4} + {}^{1}\mathbf{O}_{M} \tag{2.37} $$

where $\mathbf{O}_H$ is the position of the interaction point imposed by the hand kinematics. Knowing such trajectory, it is possible to interpolate this point series as a function of α_2 by imposing the initial and the final point of the trajectory with $\alpha_2 = 0$ and $\alpha_2 = \frac{\pi}{2}$ — respectively $(0, {}^{1}\mathbf{O}^{0}_{H})$ and $(\frac{\pi}{2}, {}^{1}\mathbf{O}^{f}_{H})$ — and accordingly remapping all the other points within this interval. What it is obtained is an approximation of the trajectory ${}^{1}\mathbf{O}_H(\alpha_2)$.

Section 2.1 proves that, given $\mathbf{S}$ (Equation 2.10), the vector that encloses the geometrical properties of the kinematic chain of the finger mechanism, it is possible to calculate in closed-form the trajectory of ${}^{1}\mathbf{O}_4$ (Equation 2.23 - 2.24) as a function of the joint variable α_2. The aim of the optimization procedure is instead to travel the other way around: the MoCap gives the trajectory of the contact point, ${}^{1}\mathbf{O}_H(\alpha_2)$, and the geometrical features in $\mathbf{S}$ become the

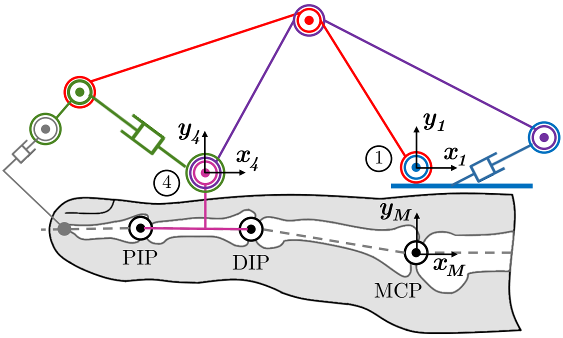


Figure 2.3: The finger kinematic chain can be simplified into three revolute joints which, starting from the *metacarpus*, direct the motion of the three phalanges on the flexion/extension plane.

free variables to be modified intending to make the finger mechanism track it as accurate as possible. Remembering ${}^{1}\mathbf{O}_1(\alpha_2)$ as the position of the contact point reproduced by the finger mechanism it is possible to define, firstly, the error between the two trajectories at a generic angular sample α_k (Equation 2.38), and, secondly, an objective function weighting not only the average but also the maximum error (Equation 2.39):

$$e_k = e\left(\alpha_k, \mathbf{S}\right) = \left| {}^{1}\mathbf{O}_H\left(\alpha_k\right) - {}^{1}\mathbf{O}_R\left(\alpha_k\right) \right| \tag{2.38}$$

$$f\left(\mathbf{S}\right) = \gamma \max_k |e_k| + (1-\gamma) \sum_{k=0}^{K} \frac{|e_k|}{K+1} \tag{2.39}$$

where γ, assumed as a scale coefficient to set the relative weight of the two errors has been set to 0.5 in this study to equally weigh maximum and average error. The optimization problem is hence defined as finding a particular vector of geometrical parameters $\mathbf{S}^{\star}$ which minimizes such objective function:

$$\mathbf{S}^{\star} = \min_{\mathbf{S}} f\left(\mathbf{S}\right) \tag{2.40}$$

Finally, because the nature of the problem itself was constrained — among the others, please remember the requirement of wearability of the HES — upper and lower boundaries have been imposed on each of the free variables. These constraints are chosen to rely on the user's anatomy and aim at defining a suitable geometry of the resulting mechanism which shall be safe for the user, practically manufacturable and not too bulky for the real use.

The constrained problem thus defined results to be nor convex nor linear because of the non-differentiability, non-continuity and the complexity of the objective function. The solution to such kind of problem is usually left to the recursive algorithm of numerical optimization. In this case, the applied techniques is the one proposed in [60] based on the Nelder-Mead simplex method [61]. The main idea behind this strategy is to perform a certain number of local searches for a sub-optimal solution, so that altogether, cover as much as possible the function domain. Each local search finishes when one or more conditions for the resulting simplex are met, and the starting point for the following search is probabilistically chosen to maximize the coverage of the domain. Once the number of iterations is over, the method examines all the convergence points to appoint the optimum.

2.3 The original prototype

Everything reported in the previous section has paved the way to the actual manufacturing of a prototype of HES designed explicitly for a patient voluntary enrolled in the study (Figure 2.4). Having suffered from SMA, a neuromuscular disorder resulting in progressive muscle impairment, since birth, he had lost, among other body functions, the ability to extend hand fingers autonomously. Such a thing, as he claimed, was more limiting to him than other disabilities as it prevented him from natural and comfortable objects handling. In this section, the details of the device developed to assist him will be discussed.

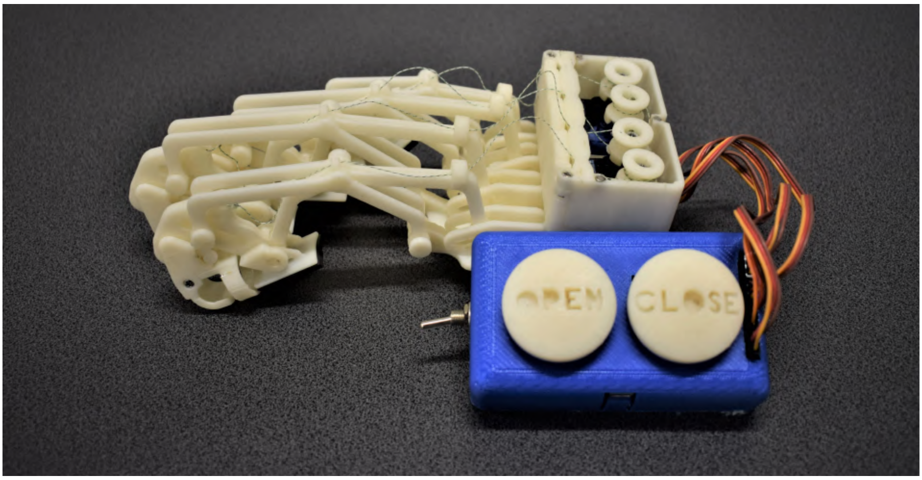


Figure 2.4: The original prototype of the assistive hand exoskeleton system developed at the UNIFI DIEF. The figure, in addition to the mechanical structure and the actuation system, also shows the original control system (the blue box) composed of a switch, a controller board, and two buttons, which triggered the movement of the four motors.

Before the beginning of the design process, a clinical assessment of the state of the patient's illness has been performed with the help of a physiotherapist. Some key points of the addressed scenario have thus been identified. Among them, the fact that the level of calcification of the finger joints was still low and that the patient still had a functional thumb mobility. Both of these points have proved to be of particular importance: the former, ascertained that the use of a finger-handling device would not have caused pain to the user, has given the green light to the mechanical design process; the latter has instead significantly lightened the process itself which could not have concerned, at least in the preliminary stages, a very complicated finger like the thumb.

According to the procedure presented in the previous section, the clinical assessment has been followed by the acquisition of the average two-dimensional trajectories of the user's index finger joints during the complete extension of the four long fingers. While the physiotherapist assisted the patient in performing several repetitions of the movement, video images of the patient's hand have been captured and processed using Kinovea. The acquired trajectories, extracted from the tracking of the markers placed on the index joints, have been then fed to the optimization algorithm leading to the definition of the first three-dimensional geometrical architecture of the index finger mechanism. Before moving on towards the manufacturing of such a mechanism, it has been virtually tested employing SOLIDWORKS® Motion. The mechanism has been paired with a 3D Computer Aided Design (CAD) model of the patient hand, and the kinematic coupling between the mechanism and the index finger has been qualitatively verified to avoid possible interpenetrations and assess the proper tracking of the finger trajectory. Once finished the preliminary investigations, the geometry of the index finger has been scaled to match the dimensions of the other fingers. Then, the docking for the four finger mechanisms, as well as the housing for the motors have been designed. Finally, just before the manufacturing phase, the components have undergone an additional reshaping: Figure 2.4 shows that the shapes of the components differ from the straight lines of the kinematic model. This final step, which has changed the shapes but not the kinematics, has been mandatory to avoid possible interpenetration with the user's finger. Moreover, integrating pins and shafts directly on the parts during this phase helped to reduce the lateral encumbrance of each mechanism remarkably. After that, all the components have been 3D-printed in ABS exploiting a Dimension Elite printer by Stratasys.

As visible in Figure 2.4, the mechanical structure is composed of five parts: four independent finger mechanisms, and a back frame designed as their docking station and as the housing for the actuation system. The choice of 3D-printing all the device parts in ABS has allowed not to worry too much about the complex geometries which some parts could take after the optimization process — the additive manufacturing technology is indeed free from some of the typical geometric limits of the standard machining based on subtracting processes. Besides, the use of such material has meant that the resulting device was already somehow inherently safe for the user. Indeed, the ABS has mechanical characteristics that make it suitable for transmitting rigid motion for this kind of application but, at the same time, it gives all the parts an intrinsic mechanical clutch function against high loads that could harm the user; in other words, ABS components and joints would have broken before the loads transferred to the hand had become dangerous.

Following the requirements of lightness — indispensable for achieving a highly wearable device — high power density servo motors, one per finger, have been chosen as the core of the actuation system. The motion transmission from each actuator, housed in the fixed frame on the back of the hand, to the finger mechanisms, positioned over each finger, has been designed to be carried out using a cable that was unwound or wrapped by custom pulleys integral with the motors shafts. It is worth noting that exoskeleton has been designed to intervene in the motion of hand unnaturally forced in a clenched fist deformity. Therefore, the exploitation of a cable transmission, capable of actively pulling the finger in extension and passively follow them during the flexion, fit perfectly into such a scenario and, also, it turned out to be one of the simplest and least cumbersome solutions. On the mechanisms themselves, holes have been reproduced for the passage and anchoring of the cables, while the pulleys have been 3D-printed after being designed of different sizes, aiming to confer the different fingers with the same MCP joint angular velocity. Moreover, since the length of cable to unwind to achieve a complete motion of the fingers resulted in being longer than the one unwound during a full rotation of the pulleys, these motors have been modified to allow for the continuous rotation of their shaft. Such an adjustment was preferred to the design of sufficiently large pulleys as it would have compromised the compactness and the wearability of the system.

An Arduino Nano control board, housed in a separate box placed on the forearm, managed the actuators' behavior according to the triggers coming from two buttons (the two white circles in Figure 2.4) designed to be pressed with the other hand even if impaired as well. A standard 6-Volt lithium battery pack — placed next to the shoulder using a Velcro band — powered the Arduino and the motors. The control strategy was straightforward: a very light code recursively checked for the state of the two buttons, if one of them happened to be pressed the control board reacted by sending the servo command to the actuators which, all together, started spinning at a fixed speed in the direction indicated by the specific pressed button. This prototype did not include sensors for automatically control the finger mechanisms position, and their ROM was manually managed by the user.

The whole mechatronic system presented above has been finally assembled and worn on the hand of the patient using a sport glove — to which it had previously been sewed — as a base. The user has been told to perform several complete opening and closing gestures while, once again, video images have been captured to be later processed with Kinovea. This time, the colored markers have not been placed on the index finger joints, which were not easily accessible once the HES was worn, but instead on the mechanism joints and their trajectories have been extracted from those images. The idea was to compare the acquired trajectories of the finger mechanism joints with the ones coming from the kinematic model optimized on the acquired trajectories of the patient's index finger joint as a preliminary assessment of the accuracy of the device in reproducing the correct motion. Even though this comparison, more qualitative than quantitative in the end, has given promising results, the tests conducted on this first version highlighted several flaws — some of those pointed out by the patient himself — which negatively impacted on its practical usability: (i) the exoskeleton, not allowing for the abduction-adduction movement of the fingers, forced the user to move the fingers only on the flexion-extension plane resulting in an extremely annoying feeling; (ii) having to use one hand to control the motion of the other because of the buttons-triggered control strategy resulted in being, as it could be expected, very unintuitive, unnatural and doubly clumsy; (iii) the absence of any type of position feedback over the flexion-extension angle of the MCP joint, and thus the lack of an automatic control over it did not make the user feel completely safe, causing additional mental stress and fatigue due to the necessary high amount of attention he had to pay continually.

In conclusion, the first-stage device ended up being compact (the HES did not exceed the hand size), fully wearable, light (it weighed just under 500 g) and cheap (it cost just over 500 €), nonetheless the user's feedback pointed out that it resulted still far from the real clinical application and the practical use. Starting from such a review, this thesis work and other parallel activities — aiming at revamping the mechanical structure and, in particular, the actuation system — have begun, and their results will be presented and discussed in the following chapters. As the heart of this work, the attention will be focused on the steps that have led to the development and implementation of an automatic control logic which overcame the limits arose testing this version of the HES.

Chapter 3

The new control system

BEYOND the just reported control issues — arisen during the HES tests — an important thing that has also emerged was the accurate tracking of the acquired trajectories of the fingers. This point was crucial for the choice of the new control strategy to implement. The availability of a device that already benefits from such mechanical optimization has indeed directed this thesis work towards the research for a suitable technique that would allow a reliable classification of the user's intentions to reproduce, leaving the motion management of the exoskeleton to a standard position and speed control.

When it comes to classifying the motor intentions of a person to control wearable robotic devices, the technical literature (Section 1.2) is evident in indicating the use of biological signals as the solution to be adopted to obtain a control experience as intuitive and natural as possible. In particular, in the last years, the use of sEMG signals, i.e., the result of the epidermal measurement of the muscular activity in the form of electric potential, is attracting the attention of more and more researchers. Surface electromyography is indeed an emerging discipline that boasts among its strengths the total non-invasiveness, the ease of setup, and, thanks to the technological progress of sensors and signal processing techniques, a remarkable characterization of the muscular state.

This chapter will deal with the topics mentioned above, giving the basis to understand the path taken, discussing the choices made in detail, and reporting every step of the work done. Finally, it will be essential to consider that the work phases described from now on are the result of a circular process: each phase has been carried out in parallel and has influenced the other ones.

Section 3.1 describes the fundamental principles of the physiology of sEMG signals and the theory of classification. They both will constitute the theoretical background to understand better the choices reported in the following sections.

Section 3.2 presents the electronics changes, which have been the basis for the implementation of the new control strategy. Each component is described in its functioning, and its main characteristics are stressed.

Section 3.3 tackles the main focus of this work: the choice and the development of the new control strategy. Particular emphasis will be given to the original requirements, the various issues that have come up during the early stages of testing, and the efforts taken for their resolution. Ample space is also left to the description of a custom GUI, designed to ease the process of sEMG signals collection and classifier initialization.

3.1 Theoretical background

This section aims to discuss the central theoretical notions on which the research work described in this thesis has been based. In particular, the principles of electromyography and classification will be addressed. Both are vast, complex, and above all, still in discovery subjects. What is written is not intended to be an exhaustive treatment of such topics but rather a summary of the information available in the literature, with particular attention given to those areas that will be referred to later along the thesis.

3.1.1 sEMG signal physiology

Electromyography is the discipline that, from the beginning of the 20th century, studies and interprets the bioelectric signals coming from the neuromuscular system, also called myoelectric activity [62, 63]. The neuromuscular system is the apparatus humans use to interact with the environment; it comprises nerve endings in the corticospinal tract, (α)-motoneurons in the spinal cords and skeletal muscles. The set of an (α)-motoneuron and the innervated muscle fibers is called Motor Unit (MU) and represents the elementary units responsible for the contraction of muscles [64, 65]. When a motoneuron is excited by a nervous stimulus, the action potential propagation activates all muscular fibers supplied by its axonal branches; the propagation starts from the Innervation Zone (IZ) and moves in the opposite direction toward the extremities of the muscle. The number of MUs per muscle in humans varies roughly in a range from 100 to 1000, moreover, their fibers are randomly distributed in the muscle and intertwined with fibers belonging to different MUs. For these reasons, their activation is observable only as an overlapping of the single contributes: this resulting signal goes under the name of Motor Unit Action Potential (MUAP). Collecting such signals is possible by placing in the proximity of the muscle one or more electrodes, which record the electrical potential generated by the electromagnetic field due to polarization and depolarization of muscle fibers.

In practice, there is much difference between the various electromyographic sensors commercially available, and it is precisely their typology and their functioning that discriminates the nature of the signal that can be measured. Intramuscular sensors allow for the collection of the classic EMG signals. Although this practice is decidedly invasive as needle-like electrodes are implanted directly into the muscle, the signals that are obtained are very selective and not very noisy. Epidermal sensors are instead designed to capture the electromyographic signal directly through the skin. This signal, which takes the name of sEMG, is characterized by a significantly higher noise level than the intramuscular one because it is filtered by layers of different tissue (e.g., fat, skin, other muscles) before reaching the electrode (see Figure 3.1). Nevertheless, surface electromyography is in active growth as, being a completely non-invasive technique, it lends itself very well to scientific experimentation. The third category of electromyography is the High Density Surface ElectroMyoGraphy (HDsEMG), which uses sensors made of dense matrices of individual electrodes. The use of such sensors allows for a better spatial representation of the muscular activity, paving the way to the EMG decomposition — i.e., decompose the signals into the single MUAPs to observe the behavior of the motoneurons.

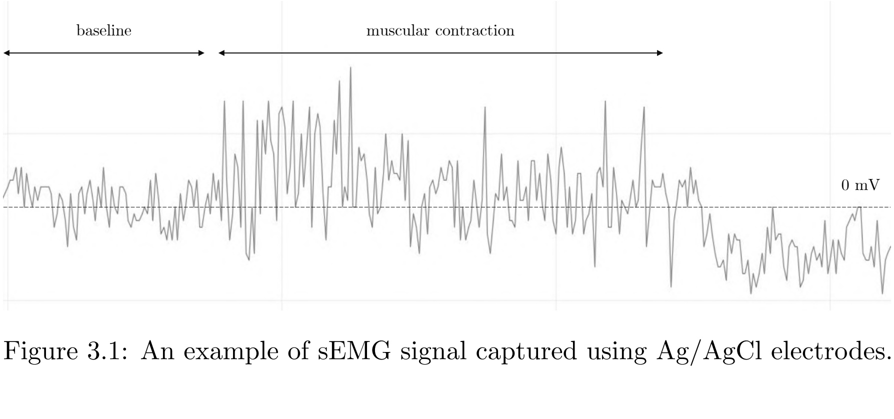


Figure 3.1: An example of sEMG signal captured using Ag/AgCl electrodes.

SURFACE electromyography, which is the technique exploited in this research work, will now be discussed in detail, presenting the main characteristics of a typical acquisition setup and the conditioning issues of sEMG signals.

The most used sEMG electrodes are made of a silver wire coated with a layer of solid silver chloride (Ag/AgCl). Unlike the other standard types of electrodes (silver chloride, silver, and gold), Ag/AgCl ones present lower sensibility to the rubbing on the skin and usually lead to a cleaner signal collection, especially when soaked with an electrolyte gel. Properties of such contacts are indeed higher signal to noise ratios and decent filtering of motion artifacts and power line disturbances.

Electrodes dimensions vary from a few millimeters to some centimeters, and they can be of circular or rectangular shape. While the shape does not influence much the acquired signal, dimensions do matter: the larger the electrode, the lower the selectivity of the measurement.

EMG signals can be, in general, acquired in monopolar or bipolar disposition. Monopolar acquisitions reflect the electrical status of the skin right below the electrode, to which, however, all the possible environmental (e.g., electromagnetic interference) and physiological (e.g., muscle crosstalk) disturbances also contribute. On the other hand, bipolar signals are acquired as the analogical difference between two monopolar signals. Using the difference of two signals, in particular when acquired from two sources very close to each other, means that the resulting measurement is less affected by the so-called common-mode disturbances, i.e., those that act in the same way on both monopolar signals. In light of the reasons mentioned above, not only the EMG collection is usually performed in the bipolar configuration, but in general, an additional electrode, common to all bipolar couples, is also added in a skin area free from muscular activity to serve as a shared reference of the overall electrical level of the body.

It is observable that, due to the bipolar acquisition methods and to a susceptible skin interface, electrode positioning and skin preparation are crucial for a good quality sEMG collection. For example, placing a bipolar sensor symmetrically with respect to the IZ and on the propagation direction of the action potential will result in a null signal detection since the two electrodes will record the same action potential but with opposite sign. Recommendations for the correct positioning [66] suggest thus to place the sensor in the area between the IZ and the tendon region to avoid such problems. As regards the improvement of the electrode-skin contact, the removal of body hair and layers of dead epidermal cells, together with the sprinkling with electrolyte liquid of the affected skin areas, are indicated as the steps to the correct skin preparation.

Whatever the type of surface electrodes used, the myoelectric signals appear on the skin in the form of an electric potential difference of limited amplitude (in the range of microvolts and millivolts) and bandwidth ($\sim$ 10 to 500 Hz). Such signals also have a stochastic behavior, present a low signal to noise ratio, and are affected by several disturbances (e.g., interference from the electric power line, electromagnetic pollution, motion, and electrocardiographic artifacts and crosstalk). For these reasons, they are particularly tricky to collect, and very accurate equipment is required for its measurements.

Commonly used Analog-to-Digital Converters (ADCs) have an operational range that varies from few up to 10 V, this means that sEMG signals have to be adequately amplified before their digital conversion. After the sampling, which according to the Nyquist-Shannon theorem, occurs at a frequency equal to or higher than 1000 Hz, the signals are processed by a differential amplifier, which is also responsible for the analogical filtering for rejecting the common-mode disturbances. Other undesired phenomena, as the aliasing, motion artifacts, and parasitic frequencies due to environmental disturbances, can be attenuated after the digitization using appropriate digital filters [67, 68].

However, even after all the processing mentioned above, the information contained in the raw sEMG signals remains challenging to interpret. It is a common practice to split the signals into epochs (or time windows), lasting from 250 ms to 500 ms [69], and extract representative indices (or features) from each time interval. These indices can be calculated mainly in the Time Domain (TD), and thus directly extracted from raw sEMG signals, or in the Frequency Domain (FD), hence derived from their spectrum analysis. Not requiring prior signal processing, TD features extraction is usually easier to implement. Nevertheless, sEMG signals are strongly non-stationary, and this introduces errors and uncertainty in the calculation of the TD features since it assumes the data to be stationary. Finally, in the few particular cases in which neither the TD nor the FD features prove representative for the problem in question, it is possible to use the so-called Entropy-Based (EB) features which are based on the concept of entropy of a stochastic signal. Since the latter features are only used in very uncommon cases, they will not be further explored in this thesis.

Well known examples of TD features are the Mean Absolute Value (MAV), also known as Average Rectified Value (AVR), and the Root Mean Square (RMS) defined as:

$$MAV_e = \frac{1}{N_e}\sum_{k=1}^{N_e}|s_k| \qquad RMS_e = \sqrt{\frac{1}{N_e}\sum_{k=1}^{N_e}(s_k - MAV_e)^2} \tag{3.1}$$

where the subscript e highlights the quantities that depend on the length of the epoch considered, N_e is the number of samples within the considered epoch, and s_k is the k-th signal sample of the epoch. Both these indices are very similar, and both are used to express the amplitude of the signal, but their physical meaning is different: the MAV can be interpreted as the myoelectric activity level; the RMS gives information about the power of the sEMG signal itself.

In the FD, common features are instead the Mean Frequency (MNF) and Median Frequency (MDF), which give information on the power distribution of the signal. Both are based on the assumption of having already calculated the power spectral density function, $P(f)$, that describes the power associated with the different signal frequencies. MNF and MDF are defined as follows:

$$MNF_e = \frac{\sum_{f=0}^{f_s/2} f P_e(f)}{\sum_{f=0}^{f_s/2} P_e(f)} \qquad MDF_e \ s.t. \sum_{f=0}^{MDF_e} P_e(f) = 0.5 \tag{3.2}$$

where f is the generic frequency and f_s is the sampling frequency.

The four listed above are just some typical examples of the many indices that can be extracted from the electromyographic signal to study its information content. An overview on other possible sEMG features, according to the most recent data available in literature [70, 71, 72], is given in Table 3.1.

3.1.2 Classification theory

Classification theory is a section of the model theory comprising all the doctrines regulating the grouping of objects based on their similarities, differences, or according to a given set of principles. Such a set of principles usually depends upon the nature of the objects themselves. The basic rules for organizing a domain of objects into classes require that such division should not leave two classes with any object in common and, as well, that all of the objects of the domain should be contained in one class. However, there are cases in which some objects can, with equal correctness, be accepted into two different classes. Classification theory, as it has just been described, can be applied in all the branches of knowledge, but, as it can be expected, it assumes different characteristics and rules depending on the context. In the fields of Computer Science and Robotics the term classification is strictly related to Artificial Intelligence (AI) and Machine Learning (ML) [73]. In particular, Classification is an actual division of supervised learning, which in turn is a subcategory of the ML together with the unsupervised learning. Nonetheless, classification prob-

Feature name	Acronym	Methodology
Integrated EMG	IEMG	TD
Mean Absolute Value	MAV	TD
Modified Mean Absolute Value (Type 1)	MMAV	TD
Modified Mean Absolute Value (Type 2)	MMAV2	TD
Mean Absolute Value Slope	MAVSLP	TD
Simple Square Integral	SSI	TD
Variance	VAR	TD
Root Mean Square	RMS	TD
Waveform Length	WL	TD
Zero Crossing	ZC	TD
Slope Sign Change	SSC	TD
Willison Amplitude	WAMP	TD
Autoregressive Coefficients	AR	TD
Mean Frequency	MNF	FD
Median Frequency	MDF	FD
Modified Mean Frequency	MMNF	FD
Modified Median Frequency	MMDF	FD
Shannon entropy	H_{sh}	EB
Fisher entropy	H_{fi}	EB

Table 3.1: Commonly-used features in surface electromyography.

lems can be found and addressed in each of the ML branches. Classifiers aim at labeling new instances in specific categories based on training performed on past observations.

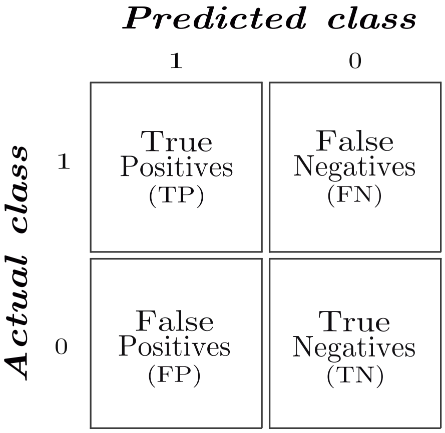
A conventional classification problem is tackled in three steps: (i) data collection and preprocessing; (ii) model training and validation; (iii) model testing. The first step plays a vital role since the quality and the phenomenon representativeness of the acquired dataset are crucial to obtain acceptable and consistent solutions to the problem. Moreover, data is rarely presented in the right format to be correctly processed by the model. There can be cases in which the classifier algorithm needs some features to be extracted from the raw data to improve its performance, others in which the required features are not independent, and the meaningful information has to be separated from the redundant one. In each of the above-mentioned cases, preprocessing is a mandatory activity. It can take the form of normalization, filtering action, features selection, and dimensionality reduction technique, e.g., Principal Components Analysis (PCA), Random Projection (RP), or Independent Components Analysis (ICA), or of a labeling procedure. This first phase ends by dividing the original dataset into three different subsets: a training and a validation set, which are used in the second phase and a test set, which is the protagonist in the third phase.

Before continuing with the explanation of the second and third stages, it is essential to remember that different classifiers are distinguished not only among the various classes but also within the same class: each classifier has its parameters and hyperparameters indeed. In general, it is possible to affirm that, once fixed the hyperparameters — which define the morphology of the classifier — the parameters — which strongly depend on the hyperparameter selection — establish the classification performance.

In the model training and validation phase, firstly, a metric for measuring the model performance is selected. However, it is usually recommended to have more than one metric, e.g., accuracy, i.e., the percentage of the correctly classified instances over the total number of them, is a commonly used one, but also the time required to perform a single classification is another critical value for real-time applications that might have to be taken into account. Then, different algorithm morphologies, obtained by exploiting distinct parameter and hyperparameter configurations, are trained over the training set until an acceptable level of the chosen metric(s) is reached and the best — the one with the highest value — is chosen. Finally, the parameters of the trained models are fine-tuned to fit also the validation dataset, which, being different from the training one, allows the learning process state to be monitored to avoid the “overfitting” issue. Overfitting is defined as a sort of excessive specialization of the classifier on the training data, which makes the classifier exceptionally performing only on the data with which it has trained. The training and validation strategy, i.e., the way the parameters and hyperparameters are recursively tuned for improving the model performance, is heavily dependable on the class of the model itself.

During the last phase, the focus shift to the level of generalization of the best model identified by the previous steps. Generalization stands for the ability of the classifier to adapt to unseen data; as a consequence, it also indicates the ability to generalize the prediction regardless of the type of input data. Qualitatively speaking, it is the opposite of overfitting and results as a key feature in providing an idea of the classifier performance within a real application task. This third step is carried out by feeding the classifier with the test set, the one subset which has been kept aside since right after the first passage. Typically, the generalization performance of the classifier is finally assessed using the same metric(s) used in the training and validation phase.

When it comes to ML classification metrics, it is noteworthy to introduce first the concept of Confusion Matrix. Confusion matrices are useful tools to evaluate the performance of ML classification models, represented by tables with different combinations of predicted, in the columns, and actual values, in the rows (see Figure 3.2). As a matter of fact, from the intersection of rows and columns it does arise the possibility to compute the desired metrics.



		Predicted class	
		1	0
Actual class	1	True Positives (TP)	False Negatives (FN)
	0	False Positives (FP)	True Negatives (TN)

Figure 3.2: A standard confusion matrix for a binary classification problem. The cells show the number (or the percentage) of samples that meet the characteristics indicated by the rows and columns.

The interpretation of the labels reported in the table is the following:

- True Positives (TP): the number of times the model predicts the condition when the condition is present;

- True Negatives (TN): the number of times the model does not predict the condition when the condition is absent;

- False Positives (FP): also known as "Type I error", the number of times the model predicts the condition when the condition is absent;

- False Negatives (FN): also known as "Type II error", the number of times the model does not predict the condition when the condition is present.

The following features can be defined as well: (i) the total number of actual negatives (TN + FP); (ii) the total number of actual positives (TP + FN); (iii) the total number of predicted negatives (TN + FN); (iv) the total number of predicted positives (TP + FP); (v) some reference metrics[1].

Accuracy - One of the most used metrics, it ranges from 0 to 1, and represents the ratio of correctly predicted observations to the total ones.

$$A = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3}$$

Instinctively assumed as a great measure of the model performance, it actually provides very little information about false positives and false negatives. In general, it is not recommended to base the assessment of the goodness of a classifier model solely on Accuracy, since it is a metric that can easily mislead. Indeed, other metrics, such as Precision and Recall, can be used to assess the impact of those wrong predictions on the model performance.

Precision - Also known as Positive Predictive Value (PPV), it ranges from 0 to 1, and is the ratio of true positives over all true positives and false positives:

$$P = \frac{TP}{TP + FP} \tag{3.4}$$

High precision means that the model total positive predictions are mostly composed of cases in which the condition is present (right guesses); it can be seen as percentage error on the total number of predicted positives.

Recall - Also known as True Positive Rate (TPR), it ranges from 0 to 1, and represents the ratio of true positives over the sum of true positives and false negatives:

$$R = \frac{TP}{TP + FN} \tag{3.5}$$

High recall means that the model predicts correctly the most of the cases in which the condition is present; it can be seen as percentage error on total number of actual positives.

[1] Many metrics can be used to measure the performance of a classifier: Accuracy, Precision, Recall, Specificity, F Measure (F1,F0.5,F2), Matthews Correlation Coefficient, ROC Area, Fallout. Only Accuracy, Precision, and Recall will be analyzed in detail as those mainly used in the continuation of this thesis.

The classification theory has developed over the years different classifier models, from the most straightforward and lightest — computationally speaking — to the most complex and most refined. Whether and when to apply one or the other is strictly dependent on the type of application. Some of the most common classification algorithms used in ML problems are reported below [74].

Dummy classifier - Also known as the "Baseline classifier", it gives a baseline accuracy (also called "null rate") that shall always be checked before selecting a more sophisticated classifier. A dummy one is a type of classifier whose behavior, rather than fitting a model by generating an insight into the training data, relies on simple rules in order to perform the classification task. It is also used as a comparison tool to verify that there are no errors in the implementation of other types of classifiers: i.e., any other classifier is expected to perform better on the given dataset since any analytic approach for a classification problem should be better than a random guessing approach. Among the possible classification criteria, the most used are: (i) "Most Frequent", which makes the dummy classifier always predicts the most frequent class label in the training data; (ii) "Stratified", which generates probabilistic predictions by respecting the class distribution of the training data; (iii) "Uniform", which makes predictions uniformly at random; (iv) "Constant", which always predicts a constant class label.

Logistic Regression (LR) - Logistic Regression is a simple but powerful classification algorithm and, for these reasons, one of the most exploited to tackle binary classification problems. Nonetheless, there are also extensions to deal with multi-class scenarios. In the One-vs-Rest (OvR) logistic regression, a separate model is trained for each class, predicting whether an observation corresponds to that class or not, and assuming that each classification problem is independent. Logistic regression is named after the function used at the core of the method: the logistic function also called the Sigmoid function. The output of the Sigmoid function is interpreted as the probability of a specific sample to belong to one of the two classes given its input features x_i linearly combined by the weights w_i. This probability is then processed by a quantizer that sets a minimum probability threshold of belonging to the given class, returns a zero (i.e., the sample does not belong to the class) or a one (i.e., the sample belongs to the class). The training phase of this classifier is achieved by modifying the weights w_i through a process that optimizes the likelihood function [75].

Support Vector Machine (SVM) - Support Vector Machines are classifiers primarily used to address supervised learning problems. Their popularity is due to their ease to be adapted to both linear and non-linear cases. In linear problems, they aim at defining the best "decision hyperplane" that separates the involved classes by maximizing the distance between this hyperplane and the closest sample points, which are called support vectors. Two are the ways used to handle non-linearly separable datasets: (i) a slack variable is introduced to relax the linear constraints and to allow for the optimization process convergence; (ii) in the kernel-based methods, samples are mapped onto a higher dimensional space to make them linearly separable, the decision hyperplane can thus be detected, then the inverse transformation is applied to project it back to the original space [76, 77].

Decision Trees (DT) - DT classifiers rely on dividing the dataset by asking if-questions until it is narrowed enough to make a prediction; as a consequence, their graphic representation looks like a tree ramification. This kind of classifier is still used in supervised learning cases. During the training and validation phase, these algorithms learn the most suitable series of questions to predict the class of the input data correctly. The maximum depth, which is the maximum number of branches from the roots to the leaves, the maximum and the minimum sample number for each branch, and the maximum feature number to check at every split represent the most relevant hyperparameters to be determined. When lots of features characterize the dataset, they usually tend to overfit, remarkably deteriorating their generalization performance. In such cases, the results of multiple decision trees, developed around the same dataset but taking into account less randomly-chosen features, are combined: they are called Random Forest (RF) algorithms [78, 79].

K-Nearest Neighbors (KNN) - In supervised learning, KNN classifiers are part of the family of the so-called lazy algorithms since the dataset is memorized rather than learned to be distinguished. Once defined an integer K and a distance metric (e.g., Euclidean distance, Manhattan distance), they classify each new sample by finding the other K closest samples and assigning to the new one the more numerous label in the neighborhood. The main strengths of the lazy algorithms, as KNN, are that they do not require a training phase and that they adapt significantly well to new data. Conversely, the major drawback is that the computational cost of the classification task increases every time a new data is labeled since more and more distances have to be calculated [80, 81].

3.2 Embedded hardware

As it was at the beginning of this thesis work, the exoskeleton prototype required substantial modifications to its electronic apparatus. The button control system and the absence of sensors for finger position feedback imposed strict limits on the usability of the device. Moreover, the simultaneous drive of four motors for the reproduction of the same movement on all the four long fingers was challenging to manage and also particularly subject to synchronization issues. The main components that replaced the previous electronics and formed the new embedded hardware are reported here below in Figure 3.3.

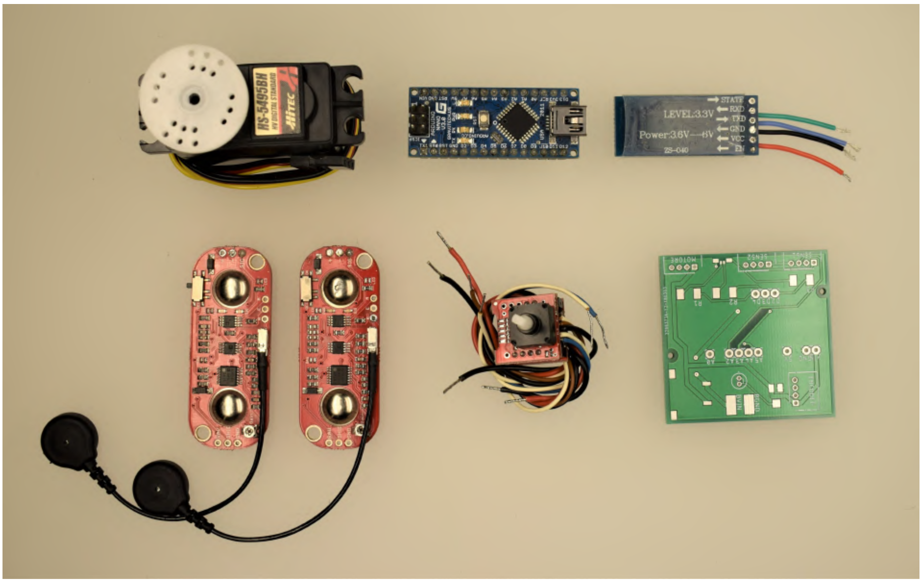


Figure 3.3: The main electronic components used during the revamping process of the device: from the top left corner, a servomotor, a micro-controller board, a Bluetooth module, two sEMG sensors, a motor driver and a custom PCB.

The choice of these components has been crucial for setting up the embedded hardware environment that would have host the control system. Having adequate performance was not the only constraint that guided the pick. Each part has been selected in complete agreement with the guidelines of a research activity aimed at developing a control strategy for a low-cost wearable device intended for clinical use: it should have been small, lightweight, low-powered (and thus safe), easy to set up, and as cheap as possible.

The design of the new hardware system has started with the choice of micro-controller. It is the component that has both computational power, having inside a micro-processor, storage capacity, thanks to the integrated memory, and it is capable of interacting with different input/output peripherals.

In light of these reasons, it is good practice to choose the micro-controller first because it is the nerve center of the control system: not only all the other components will have to interface with it, but also its performances will determine the general complexity of the manageable hardware architecture and the computational weight of the executable control logic.

Distinctive features of a microprocessor are usually its architecture (e.g., AVR, ARM), the clock frequency — i.e., the number of computational cycles that the processor carries out in a second — and the bus width — which determines the number of bits that can be processed per computational cycle. The integrated memory is instead evaluated by the number of bytes that can be stored inside the Flash memory — the one used to store the code — then, the SRAM memory — which is used for managing dynamic variables — and, finally, the EEPROM memory — which is the space that can be used to store long-term information. Then there is additional information that can be taken into consideration for the appropriate choice of the micro-controller: power supply voltage, energy consumption, logic level, number of input and output ports, embedded ADCs, communication protocols with other devices (e.g., USART, SPI, I2C, FTDI), and support for wireless connection (e.g., Bluetooth, WiFi).

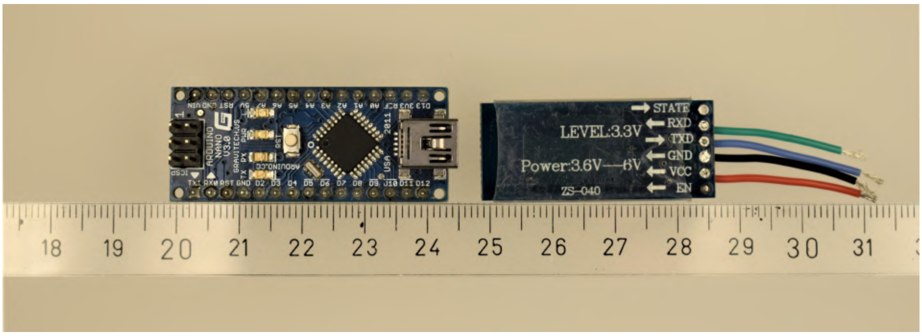


Figure 3.4: The Arduino Nano board and the HC-05 Bluetooth module.

The first-stage exoskeleton exploited an Arduino Nano controller board from the Arduino company (Figure 3.4). The characteristics of such micro-controller board — the main ones will be listed right below — have been considered sufficient for the application of the new control strategy and, therefore, it has been confirmed part of the project. It is based on the micro-controller ATmega328p from Atmel (part of Microchip Technology Inc. since 2016), which is its computational and storage core: 8-bit AVR architecture, 16 MHz clock frequency, 32kB, 2kB and 1kB of, respectively, Flash, SRAM, and EEPROM memory. The tiny Printed Circuit Board (PCB), in just 18 x 45 mm and 7 g, offers a Universal Serial Bus (USB) programming interface and ready-to-solder pins for connecting to the 32 ports of the ATmega328p. Of these 32 lines, 23 are general purpose input/output while the others are reserved for the serial, SPI, and I2C communications, and the power supply. Even if the Atmega328p has an operating voltage of 5 V, the board accepts from 7 to 12 V as input supply, thanks to the embedded voltage regulator, and, integrating some LEDs for monitoring the operational status, has a power consumption of about 20 mA. The presence of a 10-bit 6-channel ADC and the possibility of generating up to six PWM signals make this board perfect for a quick interfacing with different types of sensors and actuators. The cost of the Arduino Nano is about 20 €.

FIGURE 3.4 shows, alongside with the micro-controller board, the Bluetooth module HC-05 from Guangzhou HC Information Technology Company. It is one of the most popular and inexpensive class-2 Bluetooth modules used for radio frequency communications. Its cost is less than 10 €and, since it can be found already mounted on a breakout board, it is easy to implement in projects that involve micro-controller boards. This module has a range of 10 meters, its communication parameters can be changed through AT commands, and it is programmable both as a master and a slave device — the former is the one which looks for connections, the latter instead waits for an external connection to be received. The supply voltage can be between 3.6 and 6 V, the logic level is at 3.3 V, and the size and the weight of the component are respectively 28 x 15 mm and 5 g. The current required in the coupling phase is about 35 mA, while during the communication, it drops to about 8 mA. The HC-05 allows for the conversion of a serial port into a Bluetooth 2.0 port, generally with a Serial Port Profile (SPP), thus becoming a serial-over-Bluetooth communication port with a configurable Baud rate up to 1382400 bps.

This component has been added to the HES to provide it with a wireless connection that could be used both for managing the sEMG collection from a computer and for remotely monitoring the exoskeleton functioning.

Once the micro-controller has been confirmed, the choice of the other components have focused on those with a tested and reliable compatibility with it. This research becomes difficult in the field of surface electromyographic sensors, as many types are intended for exclusive use in clinical or laboratory environments rather than on embedded platforms. They are usually sold accompanied by a dedicated software that allows the management of the collection characteristics. The main problems with such software are that it runs on a desktop operating system which, in most cases, is Microsoft Windows, that is usually a proprietary software distributed under a licensing, and that this software, whose source code is not released, is the only way to intervene on data sampling. Finally, it is worth noting that the cost of these systems is around several thousand euros, which is not compatible with the order of costs that this research activity has as one of the main goals. In this complex scenario, a suitable solution for the sEMG signals collection has been found in the MyoWare™ Muscle Sensors from Advancer Technologies (Figure 3.5).

These sensors incorporate the housing for two monopolar snap electrodes into a small breakout board (20.8 x 52.3 mm) specifically designed to work with low-powered development boards as the Arduino and the Raspberry Pi. Even if there are two electrodes, the MyoWare™ behaves as a single bipolar

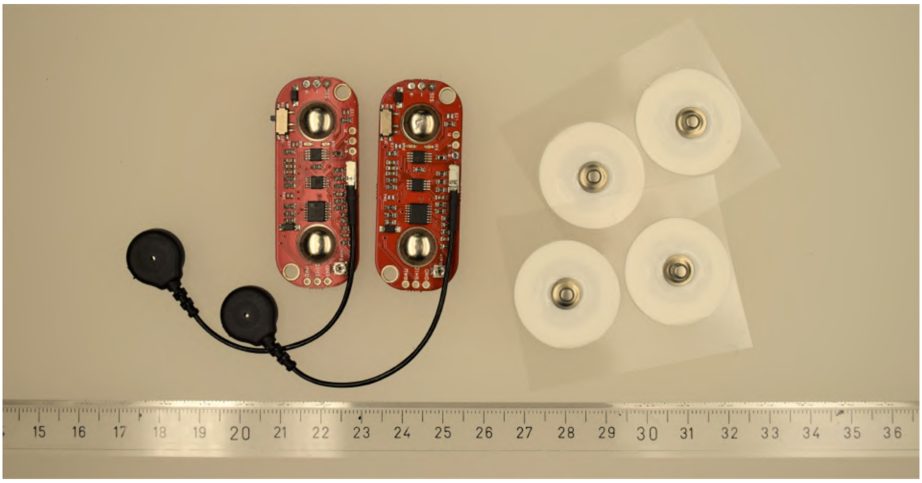


Figure 3.5: Two MyoWare™ Muscle Sensors and the Ag/AgCl snap electrodes used to capture the sEMG signal.

sensor capable of generating two different types of differential output. The third electrode housing is thought to host the reference electrode. Since the computational power available on micro-controller boards is notoriously low, these sensors have been equipped with an internal control logic that partly preprocesses the signal before returning it. In such a manner it is possible to have as output not only the raw sEMG signal but also the envelope of the rectified signal (see Figure 3.6). This feature is the equivalent to the IEMG — introduced in Table 3.1 — and gives already an idea of the force associated with the muscle contraction. The simple formula that allows the calculation of the IEMG is given below:

$$IEMG = \sum_{k=1}^{N_e} |x_k| \tag{3.6}$$

where x_k is the k-th sample of the time window on which this parameter is calculated, and N_e is the total number of samples of the same time window.

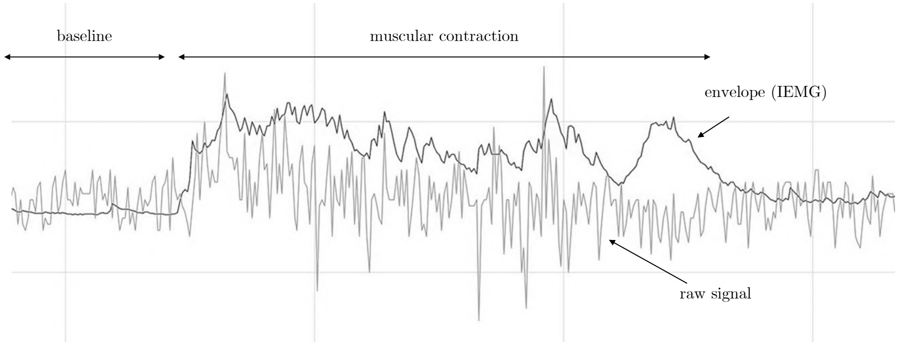


Figure 3.6: A qualitative representation of the difference between the two possible outputs of the MyoWare™ Muscle Sensors.

These measures are carried by a single wire and can be read by an analog input port. They can be powered with a voltage from 2.9 up to 5.7 V and have a maximum power consumption of 15 mA so that the necessary power can be supplied directly from the output port of any micro-controller board. With a straightforward design and a low price (∼ 35 € each), they are a remarkably good option to go for low-cost wearable applications. The choice of using only two of these sensors will be justified by the considerations that will be given in the next section.

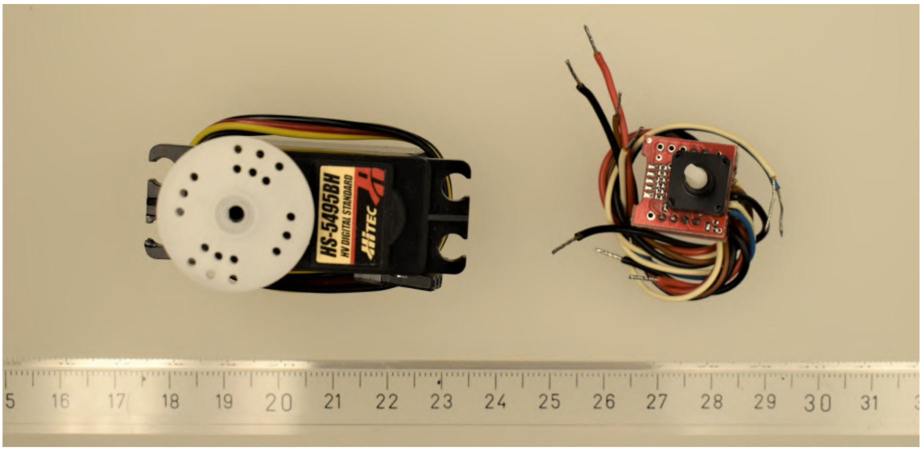


Figure 3.7: The HS-5495BH actuator and the 01^{TM} Supermodified V3.0 driver.

Likewise, in Section 3.3 the adoption of an actuation system equipped with a single servomotor will be discussed. The servomotor in question is the HS-5495BH High-Torque Servo from Hitec, visible on the left in Figure 3.7, next to the SuperModified V3.0 driver for RC-servos from 01^{TM} Mechatronics. The very possibility of integrating such a motor driver was discriminating in the choice of the actuator type and model.

The HS-5495BH can be powered with a 7.4 V power supply (the capability of a standard 2-cell LiPo battery) with a power consumption of 190 mA in non-load operations and a maximum stall current of 1400 mA. The maximum practicable torque is about 0.7 Nm, while the maximum achievable angular speed is 6.7 rad/s. Having to replace the use of four engines in the new electronic setup, these features have been heuristically and empirically considered sufficient for the movement of the four long fingers. Its dimensions, weight, and cost are 39.8 x 19.8 x 38 mm, 45 g, and 25 €.

The 01^{TM} Supermodified V3.0 is a fully enclosed motion control module composed of three stand-alone electronics layers: (i) a stage that manages the correct power supply (from 5 up to 24 V and up to 5 A) of the other two stages and to the motor; (ii) a controller stage, also equipped with the ATmega328p micro-controller, wholly dedicated to deliver a closed-loop PID control at 9.765 kHz on the angular position and speed of the crankshaft; (iii) a sensitized stage provided with a 15-bit absolute magnetic encoder which delivers angular feedback with 32768 discrete positions per revolution. The communication with the 01™ Supermodified V3.0 is straightforwardly achieved through standard protocols as the RS-485, UART, SPI or I2C. Another critical point is that, by following the guidelines released from 01™ Mechatronics itself, it is possible to fit this motor driver inside the right-size servomotors replacing their original electronics. In just 16 x 16 x 27.5 mm and for a price of 70 €, it turns a servomotor into a fully controlled — position and speed — continuous rotating motor directly interfaceable with the Arduino Nano board.

From the tests conducted on the previous version of the exoskeleton, as reported at the end of Section 2.3, it was found that a good part of the safety deficiencies in the use of the device was due to the total absence of an automatic control on the position of the fingers. To compensate for this lack, the same magnetic encoder inside the 01™ Supermodified V3.0 for 30 €, available as the 01™ MagEnc V3.0 Low Rev, has been chosen to be applied to the index finger mechanism. In particular, concerning Figure 2.1, the encoder was chosen to be mounted on the axis of the Joint 1 and integral with the component C, to measure the control variable α_2 and, indirectly, to provide a measurement also of the MCP joint status. This sensor can also measure the angular speed directly. The index finger mechanism has been chosen as it offers more lateral space available for the 01™ MagEnc V3.0 Low Rev installation and, since all the four long fingers are meant to be moved simultaneously, the measurements taken on it are assumed representative also for the status of the other ones.

The final step of this process was the design of a custom PCB to house all the components exploiting the Electronic Design Automation (EDA) software Autodesk EAGLE. The result is visible in Figure 3.8, which shows how the whole electronics has been hooked up together to the PCB. As already specified, the new electronic hardware still has at its core the micro-controller board Arduino Nano. Since the board is already tiny and dimensionally optimized, it has been opted to house the whole board on top of the PCB. In addition to the one reserved for the Arduino Nano, the space for four JST connectors (4 pins, 2 mm pitch), a bistable switch, an RGB LED, and a circuit both dedicated to monitoring the status of the battery have been set up on the new custom board. The use of JST connectors — reserved for the motor, the two sEMG sensors and the encoder — has been preferred to the direct welding of the components on the board to facilitate their eventual replacement. It is powered using a standard 2-cell LiPo battery (7.4 V), and the routing has been made using two different track measures. The smaller tracks connect the components with low current consumption — such as the encoder and the sEMG sensors — and have

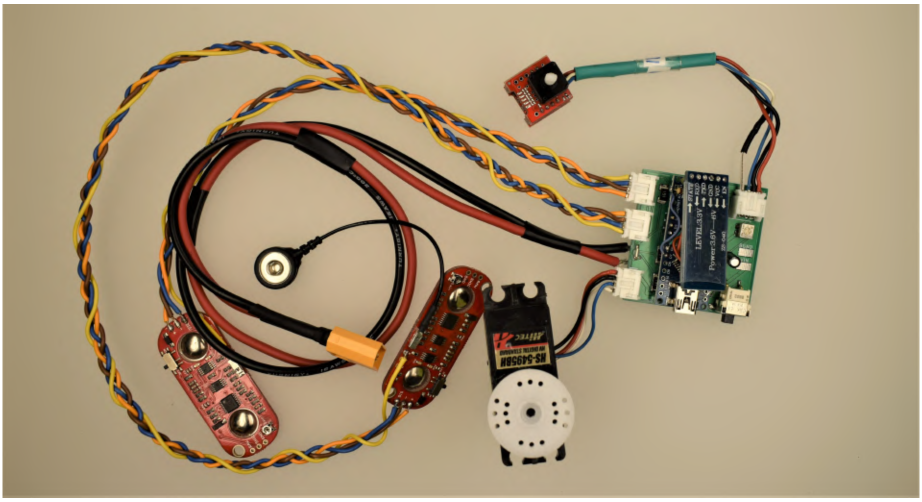


Figure 3.8: The new HES embedded electronics.

been oversized for safety with a thickness of 0.4 mm for current transport up to 1 A. The larger traces have instead been destined to the motor power supply and, therefore, dimensioned to withstand a current flow up to 2 A, which is 40% greater than the maximum current absorbed by the motor in stall conditions. The final dimensions of the board are 45 x 45 x 2 mm.

3.2.1 sEMG bracelet

During the preliminary assessment of the new integrated electronics — tests which took place on a test bench in laboratory and involved only healthy subjects — some issues regarding the wiring management and the dressing of the electromyographic sensors have emerged. The tests have been carried out by asking the subjects involved to wear two MyoWare™ sensors on the forearm and trigger the motion of an HS-5495BH servomotor with the muscle contraction. The sEMG sensors were worn on the *extensor digitorum* and the *flexor digitorum* and the control logic was that based on the signal amplitude thresholds presented by Ochoa et al., reported in Section 1.2. The subjects were sitting in front of a table on which the servomotor was fixed.

The motor motion was regularly activated according to the muscular contractions of the subjects. However, the wrist movements could sometimes establish excessive tensions in the cables that connected the sEMG sensors to the micro-controller. These traction or compression stresses quickly caused the detachment of the electrodes from the skin, compromising the signal acquisition. In an attempt to solve this problem a first-tentative forearm bracelet for the sEMG collection has been designed. The bracelet has been made by integrating some of the components already identified, such as the micro-controller, the Myoware™ sensors, and the Bluetooth module, with a 500 mAh Li-ion battery, a custom made 3D-printed case, and a Velcro band. This solution has not only proved to be valid for decoupling the acquisition system from the movements of the wrist by exploiting a Bluetooth bridge for data transmission, but it has also proved to be useful for lightening the workload on the original processor, taking on the sampling and the eventual preprocessing of the sEMG signals.

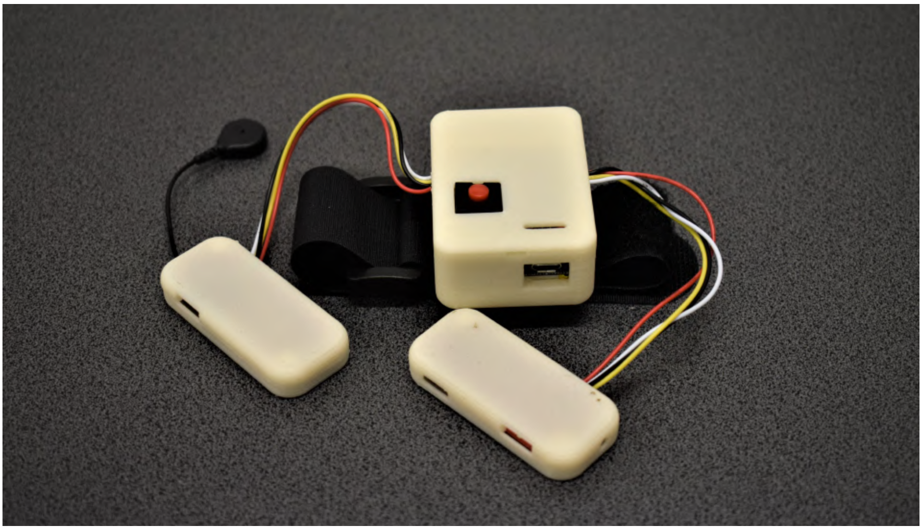


Figure 3.9: The sEMG bracelet developed to reduced the disturbances coming from the sensor wires. The central unit houses the micro-controller, the Bluetooth module and the battery. The two lateral units contain the two Myoware™ sensors. The total cost and weight of the bracelet are about 150 € and 80 g.

3.3 The novel control strategy

The lack of sufficient safety for the user, and the limited intuitiveness of the actuation of the original exoskeleton prototype have strongly highlighted the need to develop an alternative control strategy. Both issues have arisen in the light of the first clinical trials performed on a single patient and were mainly due to the absence of automatic control over the position of the fingers and the uncomfortable use of a two-button actuation strategy.

The first problem — the lack of security on the motion range control of the fingers — has been solved by inserting the feedback of an angular sensor in the new control strategy of the device. This sensor, the magnetic encoder presented in the previous section, has been integrated into the index finger mechanism to monitor the position and angular speed of the controlled joint. By knowing such measurements, it is indeed possible to uniquely reconstruct the position and angular speed of the MCP anatomical joint from the study of the 1-DOF kinematic chain presented in Section 2.1. The detail of the type of control that has been implemented thanks to the angular feedback from this encoder will be discussed in Subsection 3.3.1, where the overall architecture of the new controller algorithm will be examined and analyzed.

The second issue to be addressed concerned the difficulty in actuating the HES. The test user — a man, born with SMA — pointed out that, once the exoskeleton was worn on the one hand, having to use the other one to operate the device by pressing the buttons was not only very uncomfortable but also more cumbersome than not using the exoskeleton. Having had to deal with his illness from birth, the patient had developed alternative strategies of object grasping and handling for which the reduced residual mobility of his hands was sufficient. From this point of view, it was necessary to provide him with a device that would at least allows him to independently control the movement of the hands so that its effectiveness could then be compared to that of the replacement techniques he developed.

Finally, before going into depth in the analysis of the new control strategy, it is essential to remember the choice to focus on the development of a fast and robust technique for the classification of the user's motor intentions rather than on motion control techniques. The mechanical structure of the device, already optimized to accurately track the natural trajectory of the fingers, and based on a kinematic chain with a single degree of freedom did not require special attention indeed in controlling position and speed, except those just mentioned. Once the user's intention has been recognized, the device would be triggered and actuated, relying on the PID control, already integrated in the motor.

3.3.1 Control algorithm

The control algorithm proposed in this thesis [82, 83] is born for managing the movement of an assistive hand exoskeleton meant to help patients with clenched fist deformity to open their hands. This device consists of a motor block positioned on the back of the hand — which contains both the implementation and transmission systems and the control electronics — and four planar 1-DOF mechanisms that act on the flexion and extension movements of the four long fingers and are placed above them. The four finger mechanisms are optimized at the mechanical design stage to optimally track the finger trajectory of the specific patient for whom the device is designed. Their movement derives from the action of specific cables which, wound or unwound by servomotors, pull the mechanism in the extension movement or manage its flexion in a controlled manner — no forces are transferred to the fingers during flexion. In the original version of the control system, the actuation command was communicated to the motors using two buttons pressed by the user. Through the same buttons, it was also managed the ROM of the fingers; in other words, the user himself released the actuation button once he reached a limit of opening or closing deemed right. Ultimately, however, such a preliminary technique resulted in being remarkably uncomfortable and unsafe.

The basic idea for the new control strategy implementation relies on the introduction of new sensors in the electronic hardware to fill the gaps left by the previous one, and the fusion of known control algorithms to innovatively manage the information provided by the new sensor technology. Such information will be needed for the ROM management of the fingers, and to recognize and predict in some way the motor intentions of the user, without having him to perform a physical activity that involves the use of the hands. In addition to the above-reported features, the resulting overall system still requires to keep meeting the requirements introduced in Chapter 1 (i.e., to be fully wearable, lightweight, low-powered, stand-alone, low-cost). In light of these considerations, micro-controllers constituted the most suitable hardware technology for the development of the new control strategy by representing a feasible trade-off for the long list of design patterns and requirements. Hence the choice of all the devices detailed in Section 3.2. However, since this kind of hardware offers limited resources and computational power compared to desktop PCs, workstations or industrial controllers, the design of the new control system has begun with the awareness that a compromise would have to be found between the performance of the refined techniques tested on benchtop computers and the computational lightness required by the used hardware.

As already pointed out in Section 1.2 — the one dedicated to the analysis of the state of the art that characterized the first phase of this research activity — and reiterated at the beginning of this chapter, modern literature reports more and more convincing studies on the application of control techniques for wearable robotic devices based on the use of surface electromyographic signals. What these methods propose to do is, first of all, to classify the motor intention of the person by using the device within a set of possible classes, and, consequently, to control the robotic system by activating the state corresponding to the recognized class. Since sEMG signals are not straightforwardly decipherable, most of the strategies that compose the state of the art of such type of control make use of the high computational power distinctive of those bench equipment that is not suitable for an assistive application. There are also cases in the literature where naive variants of these methodologies have been implemented on wearable devices. The trade-off that is made in all these control strategies concerns the maximum number of classifiable movements. This design pattern is explained by taking into account that increasing the number of classes to be identified unavoidably increases, in turn, the complexity as well as the computational cost of the algorithm in charge of fulfilling the classification task. Algorithms with a high computational cost require a longer time to be executed — time which extends as the performances offered by the processors decrease — and, therefore, they risk resulting unresponsive and not functional: the literature affirms that a control latency higher than 300 ms is perceived as a considerable delay and produces an intense frustration in the user [84].

When it comes to exoskeletons for the hand, this aspect becomes even more complicated. First of all, the limited space available around the upper limb, as well as the low bearable weight, means that the exoskeletal system shall be considerably small. Consequently, by minimizing the size, the volume of the electronics that can be integrated into such a device is reduced, and, generally speaking, its computational power is also reduced. Finally, this limited technological scenario has to deal with the very high natural dexterity that a healthy hand would possess. The human hand can indeed perform lots of different movements, which are products of the contraction of several muscle bands located in a tiny space inside the forearm where they are very close to each other. In light of what is reported in Subsection 3.1.1 about the nature and the noise level of the sEMG signals, being able to accurately classify all the possible movements of the upper limb from these signals necessarily requires the use of high computational power machines running heavy ML algorithms — and yet, nowadays, it is still not possible to distinguish them robustly.

In this complex scenario, a suggestion for an alternative approach comes from a recent study, which demonstrates that the independent use of the fingers is not crucial in carrying out most of the common Activities of Daily Living (ADLs) [85]. From this point of view, it is possible to think of an assistive HES that could be of practical use even without allowing the movement of individual fingers. This assumption, which in itself dramatically reduces the number of hand gestures to be distinguished, also has a positive impact on the architecture of the mechanical and electronic hardware. The choice on the number of electronic components reported in the Section 3.2 have been made based on the simplification to consider classifiable just the three main motion of the whole hand, i.e., opening, closing, and resting. The use of only two MyoWare$^{\text{TM}}$ Muscle Sensors is the result of the combination of the fact that two are the sufficient muscle bands whose myoelectric activity has to be recorded to classify these three movements, and that in a low-cost perspective and aiming at high wearability, a low number of sensors is preferable. As it is possible to move all the fingers together, just a single motor is required for the actuation system — remembering that the actuation is cable-driven, a pulley with four different diameters, applied to the crankshaft, shall then allow the synchronous motion of the fingers despite their different size. Once it has been ascertained that the correct sizing of the multi-diameter pulley of the motor guarantees the same angular speed for all the fingers at any time, a single encoder mounted on a single finger mechanism is sufficient to characterize the position and angular speed of the all fingers.

Figure 3.10 and Algorithm 1 show the general scheme of the new control strategy, which can be split into two sequential subroutines: the "classification loop" and the "actuation loop". The first one is performed on the micro-controller integrated in the data collection bracelet (Section 3.2.1), whereas the second one runs on the micro-controller placed inside the motor block and is responsible for the management of the device actuation. Everything is designed to be run at a frequency of 50 Hz, which is a right control frequency for a system — the muscular one — interested by slow dynamics after all, even slower when it comes to patients suffering from motor disabilities. Although the second of the series, the control strategy will now be described by starting from the quick analysis of the subroutine that, downstream of the classification loop, handles the actuation of the exoskeleton, as it is conceptually easier and composed of only standard procedures that nothing new adds to what is already available in literature. More space will be left to the description of the other loop as it is the real innovative point compared to state of the art of control techniques for HES based on the classification of electromyographic patterns.

Algorithm 1 Pseudo code of the proposed control strategy

```
Input:
$F_p$ = current finger position
$F_v$ = current finger velocity
$M_a$ = current muscular activity

Classification loop:
if $M_a$ compatible with opening gesture then
    return open
end
else if $M_a$ compatible with closing gesture then
    return close
end
else
    return rest
end
Actuation loop:
if open then
    if $F_p$ compatible with range of motion then
        run motor to pull the cables to assist the hand opening
    end
    else
        max opening reached: stop motor and wait for a new close command
    end
end
if close then
    if $F_p$ compatible with range of motion then
        run motor to release the cables to follow the hand closing
        if $F_v$ lower than velocity threshold and motor is running then
            object grasped: stop motor and wait for a new open command
        end
    end
    else
        max closure reached: stop motor and wait for a new open command
    end
end
if rest then
    do nothing: wait for a new different command
end
```

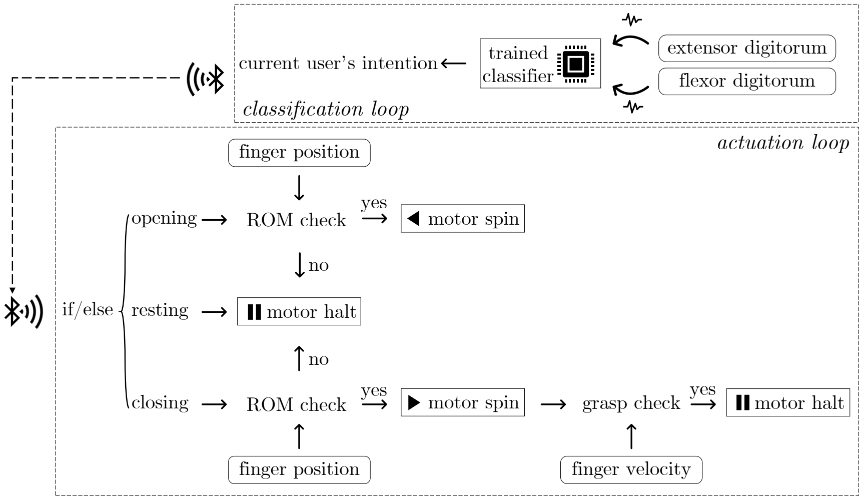


Figure 3.10: The block diagram of the overall control strategy. In the top box, the classification loop that, given the two signals sEMG, returns the user's intention. In the bottom box, the actuation loop that, given the user's intention, handles the actuation system of the exoskeleton. The motor commands (identified by ⏸, ◀ and ▶) are executed until the next one is received.

All that is required of the actuation loop can be summarized in the translation of the classification loop output (namely an opening, a closing or a resting intention) into a set of valid low-level commands to be sent to the actuator, while two inner loops take care of controlling the ROM of the fingers and the eventual grasping of an object.

Every time an opening intention is detected, the motor will be requested to spin in order to pull the cables which are connected to every finger mechanism. Therefore, the result will be an opening movement of the fingers, which shall take place as close in time as possible with the user's muscle contraction. Similarly, the motor shall spin in the opposite direction as soon as a closing intention is classified. Also in this case the motion of the exoskeleton should take place without excessive delay from the muscle contraction. When the classification loop recognizes neither of the previous two intentions, the actuator will be required to hold its position. Commands are given through the 01™ Mechatronics proprietary libraries provided with the motor driver.

To ensure that the movement of the exoskeleton is perceived as sufficiently reactive by the user, the actuation command is sent to the motor at a frequency of 50 Hz, thus allowing the exoskeleton to reproduce a different movement up to 50 times per second; subsequent intentions belonging to the same class are translated into a single prolonged actuation. This design choice presents a further benefit: the complete control over all the intermediate positions of the ROM. In fact, unlike the strategies which are based on motion triggers, the user can interrupt the motion of the exoskeleton, at every time once reached the desired degree of opening/closing, by simply relaxing the corresponding muscle contraction. However, the need to maintain a stable level of muscle contraction to achieve a smooth HES motion emerges as an inevitable price to pay; as a consequence, the device control will then be more fatiguing in the long run.

As previously mentioned, the translation of the user's motor intention into actual exoskeleton movement is not the only thing that is handled by this control loop. At each control cycle, the angular position of the fingers is measured before the actual actuation takes place. The position feedback of the MCP joint is indirectly real-time measured using the magnetic encoder mounted on the index finger mechanism; before issuing the command to the motor, this measure is checked to verify that has not already reached the permissible limit values. The maximum opening and the maximum closing position values are identified during a preliminary phase in which a physiotherapist, firstly, makes the user wear the device and, secondly, manually moves the patient's hand, making complete opening and closing gestures several times. During this process, the exoskeleton control system records the maximum and the minimum value measured by the encoder and set them as the admissible ROM upper and lower boundaries. The upper and lower bound to be set at the ROM for that user are then extracted from the average of the maximum and minimum values recorded by the encoder. Subsequently, during the regular functioning, if the encoder records that one of these limits has been reached, the motion is immediately stopped. The entire system waits then for the classification of the correct command to restart in the opposite direction. E.g., once the maximum closing limit has been reached, a halt is sent to the motor, any other closing command is rejected, and the motor will move again only after an opening command. The problem of the poor device safety resulting from the absence of an automatic control over the admissible ROM has been therefore solved.

While the outermost control loop supervises the ROM, another loop — nested inside the part of the algorithm that handles the closing motion — is in charge of detecting the grasping of rigid or semi-rigid objects. Since there are no force sensors on the HES, this detection is performed by using the angular velocity measurements of the MCP joint and the motor crankshaft. When the first one drops over a certain threshold and the motor is still running at full speed, continuing to unroll the cable, it is reasonable to think that the fingers have encountered an object or an obstacle during the motion. In these cases, the control system commands the actuator to stop before an excessive cable length is released. Preventing excessive cable unwinding by actively controlling the grip of objects has proved necessary since the first laboratory tests. Before this additional control level was introduced, the user had to relax the flexor muscles to stop the motor motion while grasping an object[2] Nevertheless, the transition from the closing to the resting intention has proved to be challenging when interfacing with an object and has often made the motor run after the actual grasping. As a result, the cable kept on being released even when not requested, and it risked giving rise to the problem of cable twisting and also introduced a deleterious delay in control actions after the grasping.

THE subroutine in charge of taking care of the sEMG signal classification is instead the first to be executed. The two Myoware^{MT} collects every 20 ms, the muscular activity of the *extensor digitorum* and the *flexor digitorum* Muscle Sensors. It is then processed by the Arduino Nano (the one on the bracelet), which classifies it in one of the three reproducible user intentions: hand opening, closing or resting. The EMG envelope has been preferred over the raw output for two different reasons: (i) to make most of the electronics integrated on the sensors and not to burden the micro-controller ADC with the sampling of the raw signal; (ii) because this feature in itself provides a qualitative idea of the state of the muscle activation and it is convenient for the discrimination of the three above-mentioned gestures. Once the current user's intention has been classified, the corresponding signal is passed to the actuation loop, which, as already described, translates it into the appropriate control commands for the actuation system.

The related works reported in Section 1.2, present similar techniques which independently monitors the myoelectric state of the same muscle bands, and then recognize the user's intention by comparing the amplitude of the collected signal with respective lower thresholds, corresponding to the minimum levels of muscle activation for some given movements. As a consequence, this strategy makes the classification process one-dimensional. However, even though this

[2] As much as having to relax the muscles to maintain the grip on an object may seem counter-intuitive; it is crucial to remember that the exoskeleton has been designed for people whose hands are forced to fist by a considerable tendon tension and, therefore, can exert passive closing forces even when their muscles are relaxed.

method is computationally very light and produces a decent control experience, it carries some inherent problems. Unwanted movements, little impacts with objects in proximity of the sensors, and skin-electrode sliding might indeed produce motion artifacts. From the applied use of a similar technique, however, it can be seen that these artifacts, although not corresponding to actual muscle activation, can easily be misinterpreted by the sensors to which they appear

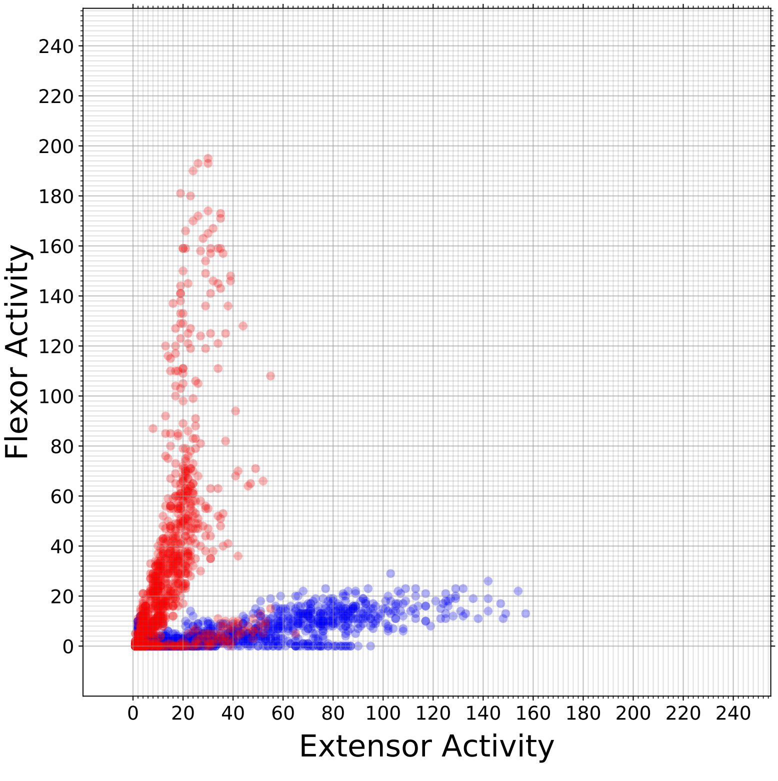
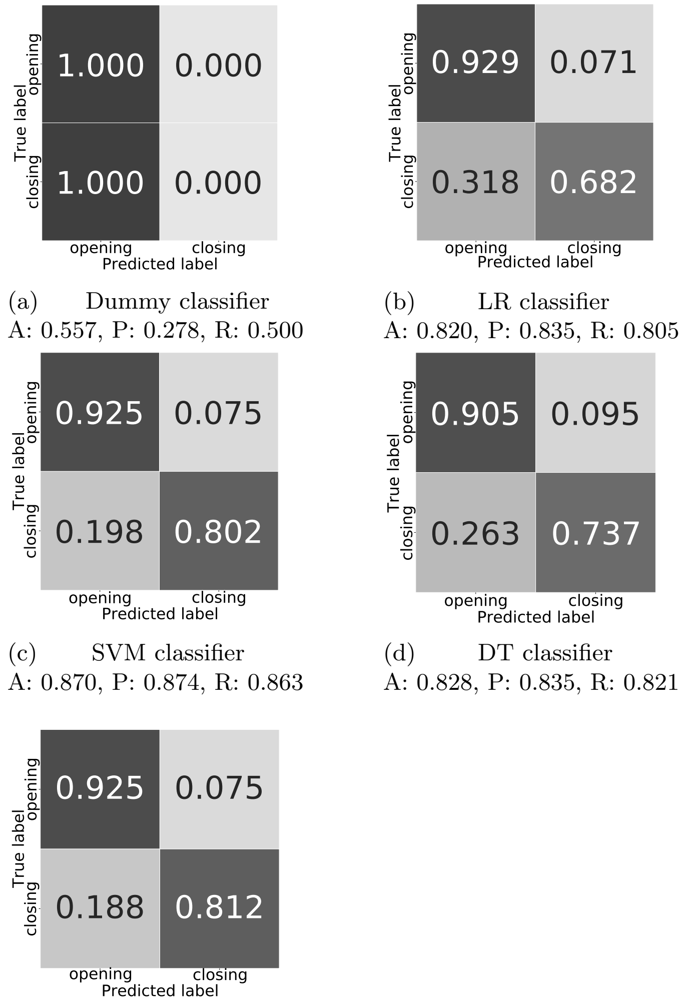


Figure 3.11: The graph shows a two-dimensional representation of the forearm muscular activity during hand opening (blue) and closing (red). On the abscissa axis the sEMG signal measured by the sensor placed on the *extensor digitorum* is shown, whereas on the ordinate axis the signal from the *flexor digitorum* is reported. The scale of the axes will be discussed later in Section 3.3.2.

as muscular activity, generally of high intensity. Besides, different movements might result in the same muscular activation because of the cross-talk effect. Paying particular attention to the movements of the wrist when handling objects is a typical example of how this problem can be harmful for such an application: the wrist extensor muscles are located under the extensor muscles of the fingers. An extension movement of the wrist would indeed most likely activate the opening motion of the exoskeleton, producing the hand opening and the consequent fall of the object. Being subject to these types of disturbance makes these methods considerably vulnerable and prone to misclassifications.

Aiming at reducing the risk that these unforeseen events may cause damage to the system or even to the user, the new control strategy has been based on a different classification criterion, which no longer relies on the value of the individual sensor, but on the combination of the measures provided by both. Rather than independently comparing the two signals to a lower threshold, every sample interval a point — whose coordinates are the envelopes of both the sEMG signals — is plotted on a 2D Cartesian plane (as visible in Figure 3.11) and is tested against the belonging to a specific area. The classification process thus becomes two-dimensional, and a disturbance acting on a single muscle has fewer chances to lead to misclassification.

The choice of the classifier required more than one attempt to find an optimal solution for the under-consideration case and resulted in an iterative procedure rather than a deterministic process. However, the procedure followed for each model remained unchanged and was characterized by the steps below: (i) a sufficient dataset has been collected; (ii) the original dataset has been split into the training and validation set, equivalent to the 80% of the original one, and the testing set, corresponding to the remaining 20%; (iii) accuracy, precision, and recall have been selected as the evaluation metrics; (iv) during the training and validation phase, the cross-validation and the grid search techniques have been exploited to tune the classifier parameters and hyperparameters; (v) the test dataset has been used to extract the real metrics of the trained model. The whole process has been carried out by using the open-source library Scikit-learn: the gold standard, in terms of ease to use, flexibility, and completeness for Machine Learning in Python [86]. All the different algorithms introduced in Subsection 3.1.2 have been trained on the randomly chosen 80% of the dataset shown in Figure 3.11 to distinguish the user's intentions on a set consisting of only two classes: hand opening and closing. The resulting metrics for the different classifiers are visible in Figure 3.12 and Figure 3.13, where the confusion matrices and the corresponding decision boundaries are reported.



(a) Dummy classifier
A: 0.557, P: 0.278, R: 0.500

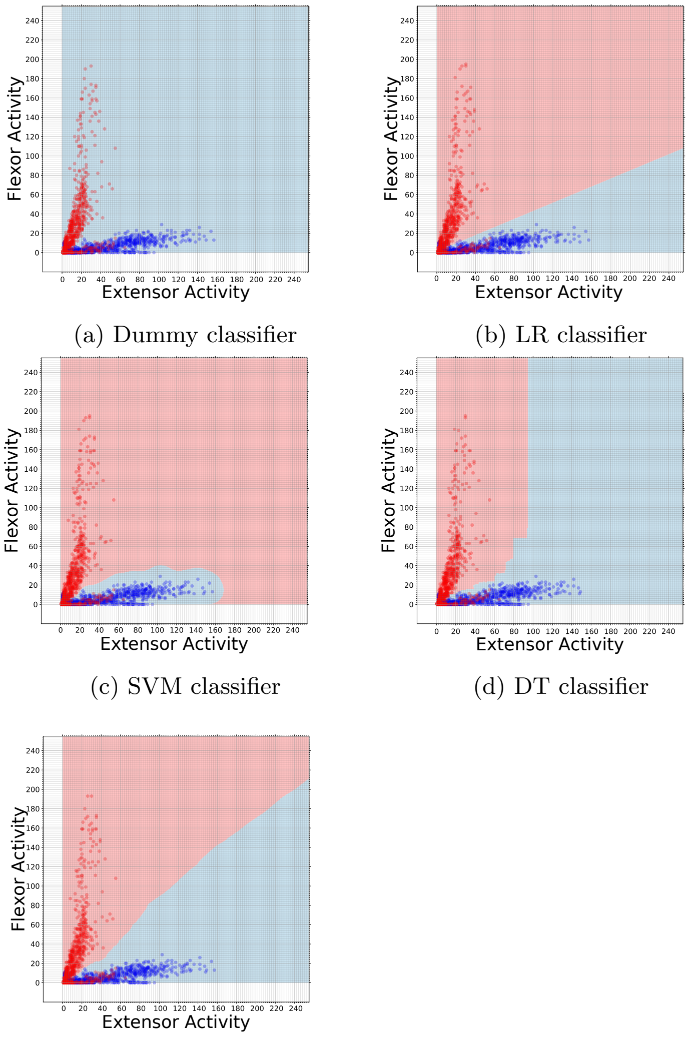
(b) LR classifier
A: 0.820, P: 0.835, R: 0.805

(c) SVM classifier
A: 0.870, P: 0.874, R: 0.863

(d) DT classifier
A: 0.828, P: 0.835, R: 0.821

(e) KNN classifier
A: 0.878, P: 0.880, R: 0.869

Figure 3.12: The confusion matrices from the testing of the five 2-class classifiers. A stands for Accuracy, P for Precision, R for Recall.



(a) Dummy classifier

(b) LR classifier

(c) SVM classifier

(d) DT classifier

(e) KNN classifier

Figure 3.13: The decision boundaries from the testing of the five 2-class classifiers. In blue and red the opening and the closing intention respectively.

Three are the principal considerations that can be done by analyzing the results reported above. Firstly, by looking at the different accuracy, precision, and recall values, it can be seen that the Dummy classifier arises by far as the worst-performing algorithm among the tested ones. It is relevant to notice that the verification of such a condition provides a preliminary (but essential) confirmation of the classification process goodness. Secondly, continuing to focus on the perspective of a quantitative analysis in terms of the judgment metrics, SVM — in the radial-basis-function-kernel-based version — and KNN highlighted outperforming results by best fitting this non-linearly separable dataset. Thirdly, the performance of all the algorithms were not utterly satisfying since none of them achieved an acceptable level of accuracy, precision, or recall.

This last datum can be interpreted by concentrating the attention on the decision boundaries shown in Figure 3.13. In qualitative terms, by merely looking at how the two classes are divided in the Cartesian plane, it can be noted that the classification was very accurate in those areas which were representative of a substantial muscle contraction — i.e., where the flexor or the extensor activity signal assumed large values. However, the same classification accuracy could not be achieved where the muscle contraction was weak, and the corresponding sample fell around the origin. As the graphs show, the sEMG signals collected during the opening and closing of the fingers are indeed well distinguishable in most of the two-dimensional space. Conversely, a not-negligible overlap among the samples of the two different classes is present just in the area near the graph origin, the one which collects all the states of weak muscle relaxation. As a result, it was almost impossible to distinguish one movement from the other during the initial moments of the muscle contraction.

Nevertheless, this is the state in which both movements always begin and, therefore, it was necessary to delimit this area by imposing a “resting” label on all the samples that fell into it. Since the rest state represented the safe mode of the system — it did not imply any HES motion — the choice of associating each point which fell in the above area to the resting class was made in an attempt to mitigate possible damages resulting from incorrect classifications keeping the exoskeleton stationary until a less ambiguous signal was recorded.

Aiming at confirming what had been qualitatively perceived from the decision boundaries of the four tested classifiers, the K-means clustering technique has been used. Widely exploited in unsupervised learning problems [87, 88, 89], this technique partitions a not labeled dataset into a K number of clusters. The base idea is that the closer the data points are, the higher the possibility they belong to the same class is. Chosen K as the number of classes to be found, K-Means, in particular, clusters the data points utilizing a recursive algorithm which, firstly, assigns K randomly-chosen centroids to the different classes and, then, optimizes their positions by minimizing the sum of squared distances from every point of the cluster, which can be seen as an index of the dispersion of the samples assigned to a specific cluster. The “Elbow method” [90, 91] has been utilized to determine the optimum number of classes to be searched within the given dataset, which has turned out to be exactly three. Once the number of classes has been established, the algorithm has been executed on the original dataset to identify the different clusters; the results are visible in Figure 3.14.

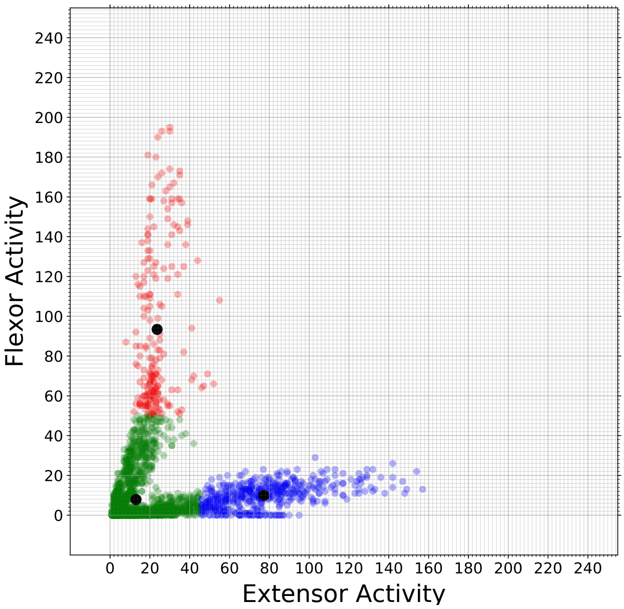
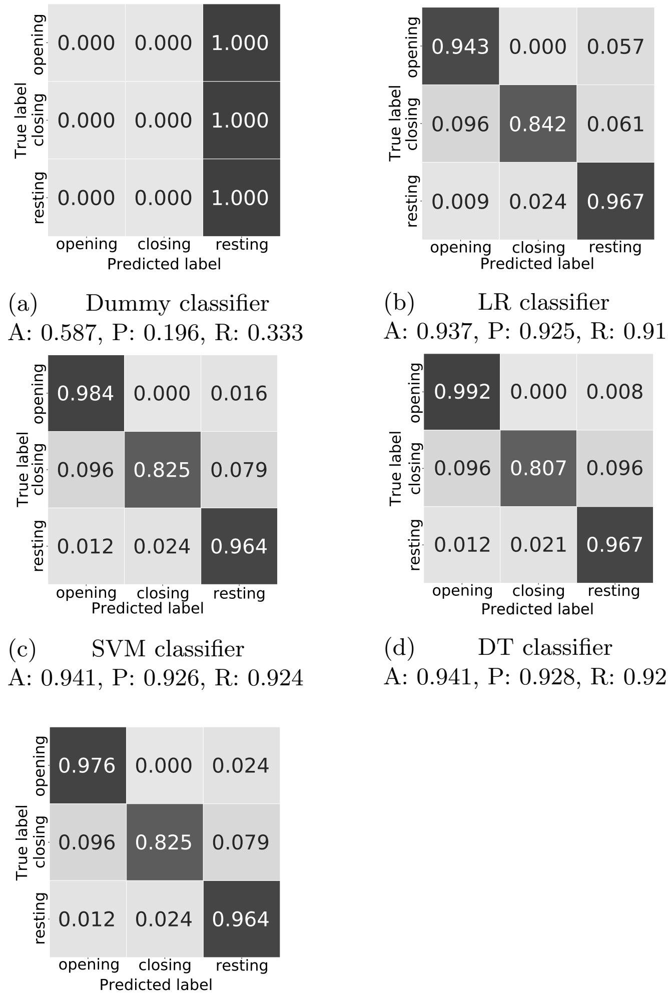


Figure 3.14: The graph shows the clustering performed by exploiting the K-means technique as the K value was set to three. In red and blue, the intention of closing and opening respectively; in green, the newly identified resting intention. Furthermore, the black circles represent the optimized centroids for each cluster.

In light of the considerations mentioned above, the classification tuning has been performed again over the same algorithms. This time, however, the different classifiers have been trained and tested on a modified dataset, consisting of the three classes that the clustering had identified. The results of this new attempt are reported in Figure 3.15 and Figure 3.16.



(a) Dummy classifier
A: 0.587, P: 0.196, R: 0.333

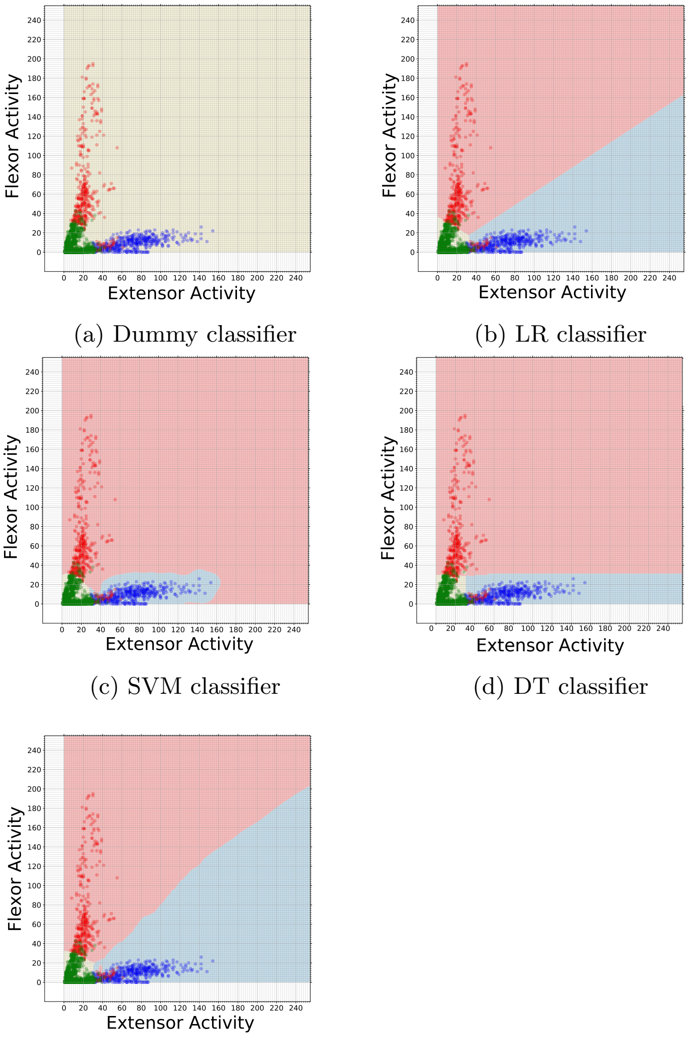
(b) LR classifier
A: 0.937, P: 0.925, R: 0.918

(c) SVM classifier
A: 0.941, P: 0.926, R: 0.924

(d) DT classifier
A: 0.941, P: 0.928, R: 0.922

(e) KNN classifier
A: 0.939, P: 0.925, R: 0.922

Figure 3.15: The confusion matrices from the testing of the five 3-class classifiers. A stands for Accuracy, P for Precision, R for Recall.



(a) Dummy classifier

(b) LR classifier

(c) SVM classifier

(d) DT classifier

(e) KNN classifier

Figure 3.16: The decision boundaries from the testing of the five 3-class classifiers. In blue, red and green the opening, the closing and the resting intention respectively.

The resulting metrics for the classifiers tested on the recognition of the three possible motor intentions report values that, on the whole, have improved up to 10% compared to those of the classifiers previously evaluated on the recognition of the only intentions of opening and closing: except for the Dummy classifier, all the other algorithms have metrics ranging from 92% to 94%. The apparent improvement in performance is mainly shown by the decrease in the number of false positives and negatives, which is represented by all the terms that are not on the main diagonal of the confusion matrices. This reduction is a direct consequence of the addition of the new resting class to the dataset, and the plots of the new decision boundaries confirm it. If these new plots are compared to the plots from the 2-class classifiers, no particular differences in the level of class separation can be spot where the sEMG signals are strongest; conversely, remarkable changes in the class separation can be easily noticed right where the new points labeled as “resting” have been added. Thus acting, the number of harmful misclassifications has been reduced at the expense of a worse sensitivity to small muscle contractions. As a matter of fact, before the myoelectric activity could be classified into an actual motor intention, i.e., opening or closing, the muscle contraction has to exceed the area of the Cartesian plane close to the origin which contains all the resting-labeled samples — indeed the resting intention does not involve any exoskeleton motion.

Although the classification had assumed a sufficiently reliable functioning to be able to undergo a real experimentation, intrinsically, the problem of the management and rejection of misclassifications due to peaks of the signal caused by environmental disturbances remained unchanged. Indeed, through the use of the standard classifiers presented so far, the problem of upper delimiting the classes to avoid misclassifications from the recordings of possible artifacts had not yet been solved — the only case in which a complete delimitation of a class is recorded is that of the SVM shown in Figure 3.16c but just for the opening intention. An appealing alternative that contained all the characteristics sought in the classifier has been offered by rummaging the state of the art for a particular two-dimensional algorithm used to address classification problems. Once defined the classifier characteristics to be suitable to the case study under consideration (i.e., that it shall not only separate at least three different classes in a two-dimensional plane, but also circumscribe them in closed and limited areas) and recalled the constraints imposed by the clinical scenario to which the HES has to be applied (i.e., the classification shall be done in real-time while exploiting limited computational power), the algorithm, known as “Point-in-Polygon”, has been identified and adopted as the most viable path among the others available in the scientific literature.

Algorithm 2 Point-in-Polygon algorithm (Ray casting version)

```
Input:
$N_v$ = number of the polygon vertices
$V_x[\ ]$, $V_y[\ ]$ = arrays of coordinate of the vertices
$T_x$, $T_y$ = coordinate of the test point

int $j = N_v - 1$
bool inside = false
for ( int i = 0; $i < N_v$; i ++) do
    if ( ( ( $V_y[\ ][\ i\ ] < T_y$ and $V_y[\ ][\ j\ ] \geq T_y$ ) or ( $V_y[\ ][\ j\ ] < T_y$ && $V_y[\ ][\ i\ ] \geq T_y$ ) ) and ( $V_x[\ ][\ i\ ] \leq T_x$ or $V_x[\ ][\ j\ ] \leq T_x$ ) ) then
        inside ^= ( $V_x[\ ][\ i\ ] + ( T_y - V_y[\ ][\ i\ ] ) / ( V_y[\ ][\ j\ ] - V_y[\ ][\ i\ ] ) * ( V_x[\ ][\ j\ ] - V_x[\ ][\ i\ ] ) < T_x$ )
    end
    j = i
end
return inside
```

This algorithm, which can be implemented in the few lines of code reported in Algorithm 2, has been known since the early 1960s [92]. Even though it is widely used in fields such as computational geometry and computer graphics, it had never found application in the field of assistive robotics before. Turning to the functioning strategy, the Point-in-Polygon algorithm checks whether a point in the plane lies inside or outside a given polygonal area. Moreover, two major versions of the base algorithm have been developed for this purpose by following two distinct approaches.

Ray casting version - Also known as the "Crossing number algorithm" or the "Even-odd rule algorithm" due to its operation logic, it is based on the Jordan curve theorem. First proposed in 1887 by the French mathematician Camille Jordan, it states that: "*any continuous simple closed curve in the plane, separates the plane into two disjoint regions, the inside and the outside*[3]." [93] A direct consequence of this theorem is the mathematical certainty that any path from a point in one region to a point in the other region shall pass through

[3] Fun fact: even if it sounds pretty evident as a theorem, it has been very hard to verify. The proof Jordan gave in his famous textbook "Cours d'Analyze de l'École Polytechnique" turned out to be flawed, and the first valid proof was given almost 20 years later by the American mathematician Oswald Veblen, in 1905.

the curve; it is on this observation that the logic of the algorithm is built. Given a geometrical plane and given a polygon and a point on such a plane, be the point chosen at will inside or outside the polygon, by drawing from that point a straight line in any direction the perimeter of the polygon will be intersected a certain number of times. By moving the point on this line, for each of these times it crosses the perimeter, it is known that it will pass from one region of the plane to the other, but above all, it is straightforward to notice that the point initial region of belonging can be traced back from the total number of intersections with the perimeter of the polygon (see Figure 3.17).

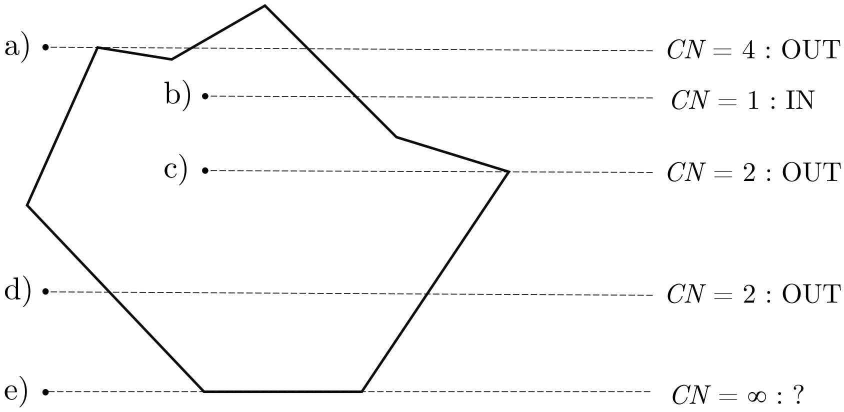


Figure 3.17: The graphic representation of the operation principle of the ray casting version of the Point-in-Polygon algorithm. CN is the Crossing Number, i.e., the number of times the ray crosses the polygon perimeter: if it is odd, the point lies inside the polygon; the point lies outside if it is even. Case a), b) and d) are correctly classified; c) and e) are particular failure cases of this method.

In practice, for simplicity, a ray casting to the right is performed, which means that, given the coordinates of the starting point, the ray is plotted keeping that ordinate constant. If the ray crosses the polygon edges an even number of times, the point belongs to the outside region, vice versa if the perimeter of the polygon is intersected an odd number of times. This technique has, however, singularities when the point is on the edge of the polygon and, when implemented on a computer, the result might be wrong even if the point lies in a small enough area around the perimeter due to finite precision arithmetics.

The most common implementation of such algorithm — the one presented in Algorithm 2 — recursively checks for intersections with all the edges of the polygon in turn. The main drawback of this method is that the result might be ambiguous when the ray passes precisely through one of the vertexes of the polygon. In this case, the ray will intersect two edges at their endpoints, the resulting number of crossing will be even, and the point will be counted as external. While the result is correct for some of the cases (e.g., case a in Figure 3.17), others require that the number of intersections is not counted more than once, as in case c in Figure 3.17. A similar problem arises when checking for intersections with horizontal segments (case e). Even if this is not generally a big issue — as the computational speed provided by the Point-in-Polygon algorithm brings much more value than the one lost in lack of accuracy in this very particular cases — the ambiguity can be solved by counting the intersection on a vertex only if the second vertex of the edge lies below the ray.

Overall, the Point-in-Polygon in its Ray casting version is a lightweight and high-performance algorithm when it comes to two-dimensional classification problems, whose level of complexity and calculation time are directly proportional to the number of polygon vertices, and whose robustness strongly depends on the finite arithmetics of the machine on which it is implemented.

Winding number version - The second version of the Point-in-Polygon algorithm still lays the foundation of its operation logic on Jordan's Theorem, but it is based instead on the calculation of the winding number of the given point for the given polygon. In mathematics, this number — which in automation forms the basis of the Nyquist criterion for the stability of unstable open-loop systems, with a possible single or double pole in the origin — is defined as the total number of times that a closed curve in the plane travels counterclockwise around a certain point. The winding number is, therefore, a characteristic of the curve-point pair rather than the individual geometric entities. The point is certainly inside the curve if the resulting winding number is different from zero and, for this reason, this version of the algorithm is also called "Nonzero-rule algorithm"; Figure 3.18 shows its base functioning.

Two are the main ways to compute the winding number. The first technique involves to sum up the angles subtended by each side of the polygon [94]. It is generally slower than the ray casting algorithm since it involves the use of inverse trigonometric functions, which are computationally heavy. Nonetheless, since the sum of all such angles in a polygon can add up to multiples of 2π only, the algorithm can be sped up by just keeping track of the number of quadrants

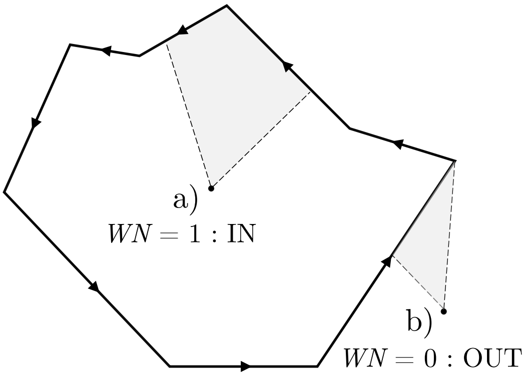


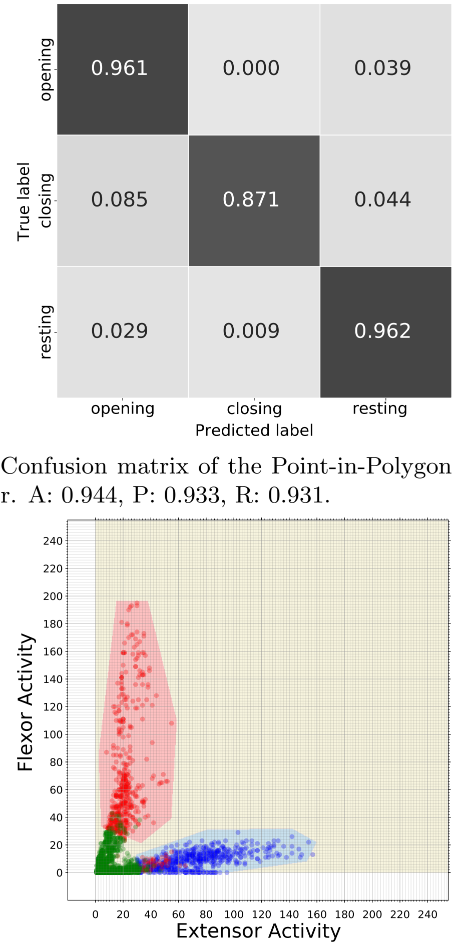
Figure 3.18: The graphic representation of the operation principle of the winding number version of the Point-in-Polygon algorithm. *WN* is indeed the Winding Number, i.e., the number of complete counterclockwise turns that the segment (dashed in the figure), which has as its extremes the point to be tested and any point of the polygon perimeter, makes around the point itself along the entire perimeter: if this number is different from zero, the point lies inside the polygon; conversely the point lies outside if it is zero.

that the polygon winds as it turns around the test point [95]. This second method, a direct improvement of the previous one, makes the speeds of the two versions of the Point-in-Polygon algorithm comparable. The winding number version, however, provides the correct answer also for non-simple polygons or particular cases, whereas the ray casting algorithm, as it has been presented, fails. Nevertheless, a more recent review of the ray casting [96] method has also smoothed out this difference, making the two versions practically equivalent.

Since there are no substantial differences in the execution speed or the results of the two codes, it has been arbitrarily chosen to use the Point-in-Polygon algorithm in its ray casting version.

THE use of this type of classifier results to be particularly interesting. Indeed, it is possible to enclose only the opening and closing classes within well-defined areas — as shown in Figure 3.19b — and implicitly assign the resting label to the remaining of the two-dimensional space. Not only the classification of the motor intentions becomes more selective and, consequently, more reliable in terms of the exoskeletal output but, at the same time, the rejection of disturbances and unwanted movements [97] is automatically improved and has no additional computational costs. While maintaining a performance comparable to the other classifiers — see Figure 3.19a — the Point-in-Polygon also offers this additional safety level: all spurious electromyographic patterns, which do not selectively fall within the polygonal areas traced to delimit the intentions of opening and closing, are cataloged as resting and will not produce undesired effects on the HES.

One of the disadvantages of using this algorithm, however, is the great difficulty of efficiently automating the tracing of the polygons. Although the definition of polygonal decision boundaries for three classes, working in a two-dimensional plane, represents a simple task for human intelligence, it is still challengingly accomplished with the modern AI algorithms. One of the possible solutions could be to develop a Deep Neural Network (DNN) — an Artificial Neural Network (ANN) prepared for the application of Deep Learning techniques — that was able to trace the polygons optimizing the separation of the classes. Not only would this have required extended time to train the network, but above all, it would not have adapted very much to an environment, such as the clinical one, characterized by a great diversity of patients — multiplicity that increases even more if the different types of pathologies and the different stages of the single disease are considered — which would have required frequent training of new networks on high-performance workstations, usually not included in clinical facilities. However, these steps require experience in the field of Computer Sciences to be carried out in the best possible way; instead, they will probably be carried out by personnel with non-technical expertise, perhaps by the physiotherapists themselves, already employed within the clinical environment. In light of this perspective, once abandoned the idea of automating the design of polygons, a custom GUI has been developed to provide a user-friendly tool for a quick and easy data collection and tuning of the classification parameters — namely the training phase of the classifier which will be carried out manually — that could be used on any desktop-PC. Such GUI, specifically designed to be used by clinical staff, also allowed the trained parameters of the classifiers to be straightforwardly uploaded directly to the micro-controller board. The next section will present how it works in detail.



(a) Confusion matrix of the Point-in-Polygon classifier. A: 0.944, P: 0.933, R: 0.931.

(b) Decision boundaries of the Point-in-Polygon classifier. In blue, red and green the opening, the closing and the resting intention respectively.

Figure 3.19: The results of the testing of the Point-in-Polygon classifier.

3.3.2 Graphical user interface

The research for the classifier model that best suited the scenario under consideration has concluded with the adoption of the Point-in-Polygon algorithm. In particular, it has been chosen in its ray casting version which — taking as inputs the number and the coordinates of a polygon vertices, and the coordinates of a given test point — counts the total number of times a ray from the point intersects the perimeter of the polygon and, based on this number, determines whether or not the point falls within the figure. As already mentioned, the choice of this classifier has implied the development of a phase of training and parameter tuning to be carried out through an external intervention. Since the HES has been designed for use in a clinical environment, it has been immediately mandatory to develop a software application — a Graphical User Interface — that would allow a straightforward collection of the training dataset of sEMG signals, a quick identification of the classifier parameters (i.e., number and coordinates of the polygon vertices), and a straightforward programming of the micro-controller[4] to non-technicians personnel.

Developed for the first time at the Xerox Palo Alto research laboratory in the late 1970s, a GUI is an interface through which a user interacts with electronic devices such as computers, handheld devices, processors. This interface uses interactive components such as icons, menus, and other graphical objects and visual indicators to display information and related user controls. GUI representations are manipulated by a pointing device such as a mouse, trackball, stylus, or a finger on a touch screen, but it can also be navigated using a keyboard. By using specially designed and labeled images, pictures, shapes, and color combinations, objects were depicted on the computer screen that either resembled the operation to be performed or were intuitively recognized by the user. Unlike using command-line instructions or dealing with an Integrated Development Environment (IDE), GUI-based applications are much easier to learn and use, mainly because commands do not need to be memorized and users do not need to have any programming skills. Nowadays, such visual language has become commonplace in both operating systems and software applications, and there are even IDEs — which are nothing more than graphical interfaces in turn — fully dedicated to the development of custom GUIs.

One of the most used development environments for creating graphical interfaces as well as cross-platform applications is Qt Creator IDE, which, in the

[4] The micro-controller board Arduino Nano is the one in charge of classifying the sEMG signals and, therefore, the tuned classifier parameters shall be uploaded to it from the PC where the training phase is performed.

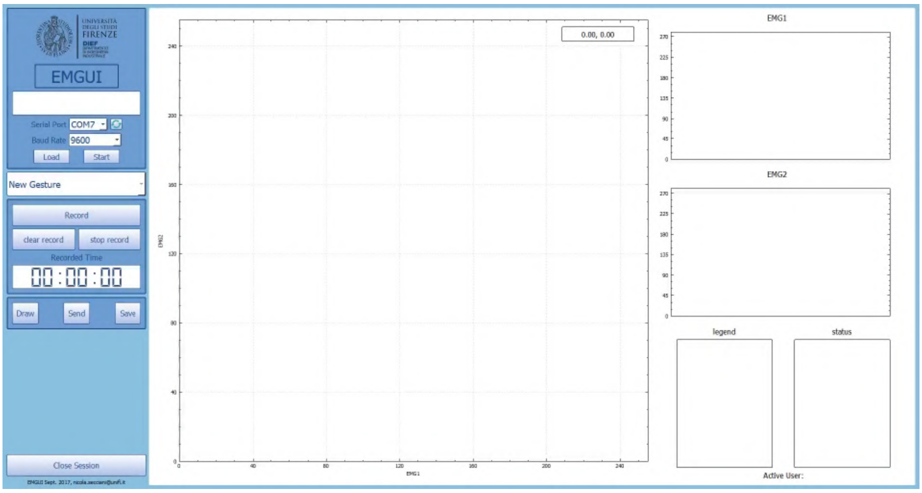


Figure 3.20: The main window of the GUI developed for manually training the classifier. The interface is developed almost entirely on this unique view and the interaction takes place via keyboard and mouse pointer.

framework of this research activity, has been chosen for its compatibility with all major desktop platforms and most mobile or embedded platforms. Qt Creator is a multi-platform IDE specifically designed for application development based on the C++ language. Released for the first time in 2009, over the years, it has evolved to become the hub of application development, integrating project management, source editing, graphics interface design and third-party debugging and versioning tools. Qt Creator IDE is distributed by the Qt Company in two versions: one with an LGPL license, which involves some restrictions on distribution methods, and one with a commercial license that also guarantees technical support for developers. The graphical interface for tuning the classifier has been designed using QT Creator IDE non-commercial distribution and targeting the PCs running Microsoft Windows operative system, which was, and still is, by far the first PC operating system for dissemination[5].

Figure 3.20 shows the main window of the GUI. The four main sections of the interface shown in the figure are: (i) on the top left corner, the University logo and the application name; (ii) below that, the command layout that

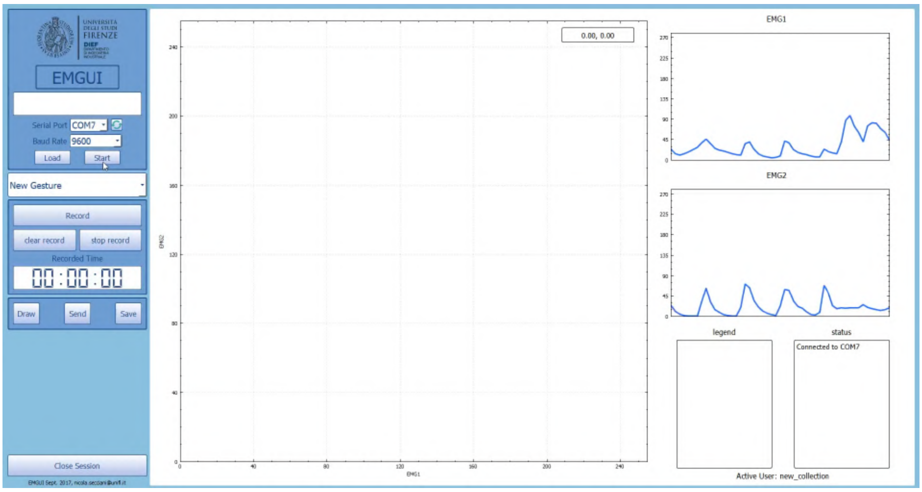
[5] Source: © StatCounter, https://gs.statcounter.com/os-market-share.

contains all the buttons through which commands are sent to the interface; (iii) in the center, an interactive two-dimensional Cartesian plane that is populated with the points sampled during the collection phase; (iv) on the right, from the top to bottom, two real-time graphics, the legend, and the status box.

Going into the detail of the command layout, starting from the top, there is a line of editable text in which to enter the title of the training session to be opened (e.g., patient_name_current_date). Immediately below are the commands for initialization of the serial port: the interface has a button for automatic search of all the serial devices connected to the platform on which it is executed, however, not wanting to limit the number of devices that can be used in conjunction with the interface, it was preferred to implement the manual selection of the port rather than the automatic connection. Nevertheless, in an attempt to simplify this step as much as possible and reduce the actions to be performed to the bone, the interface saves in memory the parameters of the last paired device and suggests them by default during the opening of the further session. Below, the interface initialization box ends with two buttons: "Load", which opens the standard Windows folder browser and allows for the selection and loading of data from an old session — so doing the name of the session is loaded from the previous one along with the data contained — and "Start", which instead uses all the data entered previously to activate the other areas of the interface, disabled until now, and initialize the serial communication with the device. The serial connection can be either wired or Bluetooth. In the second case, the initialization will take a slightly longer time to complete because of the pairing.

Once "Start" has been pressed and the session has begun, the data acquisition options can be managed with the tools that the interface provides under the initialization box. If the session has started for a new user — hence without using the "Load" button — the central graph and the legend will remain empty, while the two graphs in the top right corner will start to show the real-time envelope of the two sEMG signals and the confirmation of the connection with the serial device in use will appear in the "status" box (see Figure 3.21). Any connection errors will, in turn, be shown in red in the status box, and a new session shall be open to reset them. If the session is instead opened for an already-registered user, the behavior of the interface will be the same, but the user will also be able to see the previous acquisitions on the central graph.

The line of text, marked by default with a "New Gesture" label, is a drop-down menu that contains all the gestures acquired up to that moment — whether they have come from the current session or one of the previous sessions. The menu can be explored by clicking on the small arrow on the right of the panel, and scrolled with the mouse wheel. If already present in the list, it is possible



Serial Port COM7

Baud Rate 9600

New Gesture

Recorded Time 00:00:00

0.00, 0.00

Connected to COM7

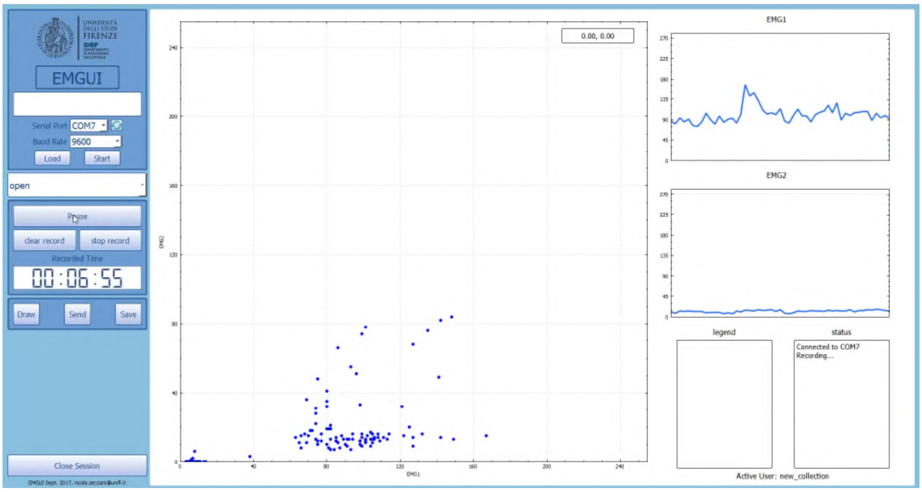
Active User: new_collection

EMGUI Sept. 2017, nicola.secciani@unifi.it

Figure 3.21: The figure shows what happens on the interface once the "Start" key is pressed. In the top right corner the two graphs show the real-time trend of the electromyographic data collected by the two MyoWare™ Muscle Sensors; in the bottom left corner, within the status box, the serial port and its connection status ared shown; below the title of the current session is reported.

to select a specific gesture and proceed with the acquisition of new data by pressing "Record" — this option overwrites the previously acquired dataset. If the deletion of the old data is not desired or until there are no gestures already recorded in memory, the acquisition of a new gesture can be made, once again, by pressing "Record" while leaving the default label on the gesture drop-down menu. Before the acquisition starts, the name of the gesture that will be recorded shall be entered in the text box that appears — the insertion of a name already within the gesture list will cause the creation of a copy and not the overwriting of the data. There are no limits to the number of gestures that can be acquired using the GUI, but the constraint of being able to classify and then reproduce only three hand gestures (opening, closing, and resting) remains.

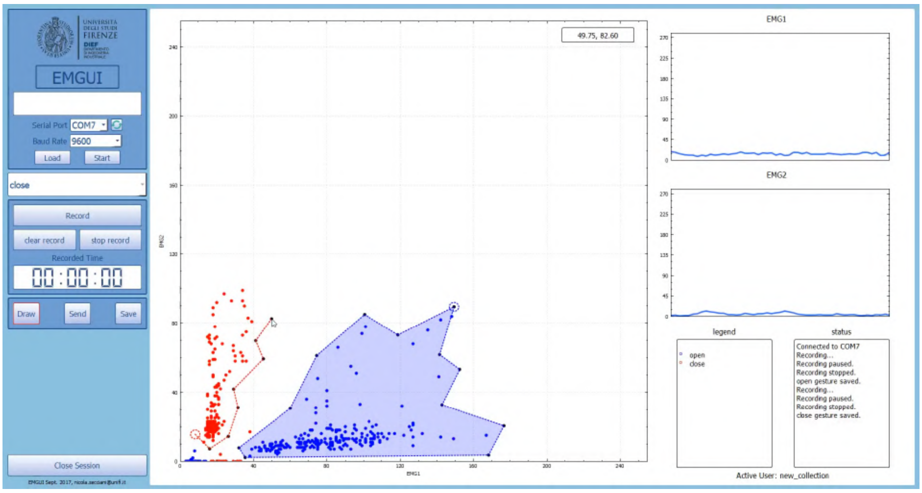
As reported in Figure 3.22, once in the data recording mode, the timer will start showing the recording time while the central graph will start populating itself with electromyographic data from the two real-time plots. The myoelectric signals are shown in the graph as points whose coordinates are the values of the envelopes recorded by the two different sEMG sensors — the reason why the



Serial Port COM7
Baud Rate 9600
open
Recorded Time 00:06:55
Connected to COM7
Recording...
Active User: new_collection

Figure 3.22: The custom GUI in recording mode. On the left, the label of the gesture can be read in the drop-menu, just below, the “Pause” button has replaced the “Record” one and, below again, the timer that marks the time from the beginning of the acquisition can be seen. In the graph in the middle, the points show in real-time the muscular state of the two muscle bands under examination. Finally, within the status box, on the bottom right corner, a new line says “Recording...”, confirming the current modality.

axes of the graph are limited to the interval of integers between 0 and 255 will be examined later on. On the graph ordinate axis, the muscle activation of the *flexor digitorum* is reported, while the abscissa shows the one from the *extensor digitorum*. Different gestures will be highlighted in different colors. As the recording proceeds, it is possible to pause the collection by pressing the “Record” button, on which the label will have been replaced by “Pause”. The collection will then resume from where it was interrupted by pressing the same button again. The “Clear record” button allows for the deletion of all the acquired data from the current gesture without interrupting the ongoing collection. The “Stop record” button has a double effect instead, depending on whether the current record contains data or not. If data is present, the button will proceed with interrupting the acquisition, followed by the saving of the data in a text file (.txt) within a specific folder, and, finally, with the insertion of the name of the gesture in the list of the acquired gestures. Inside the text file the



Serial Port COM7
Baud Rate 9600
close
Recorded Time 00:00:00
49.75, 82.60

Connected to COM7
Recording...
Recording paused.
Recording stopped.
open gesture saved.
Recording...
Recording paused.
Recording stopped.
close gesture saved.

Active User: new_collection

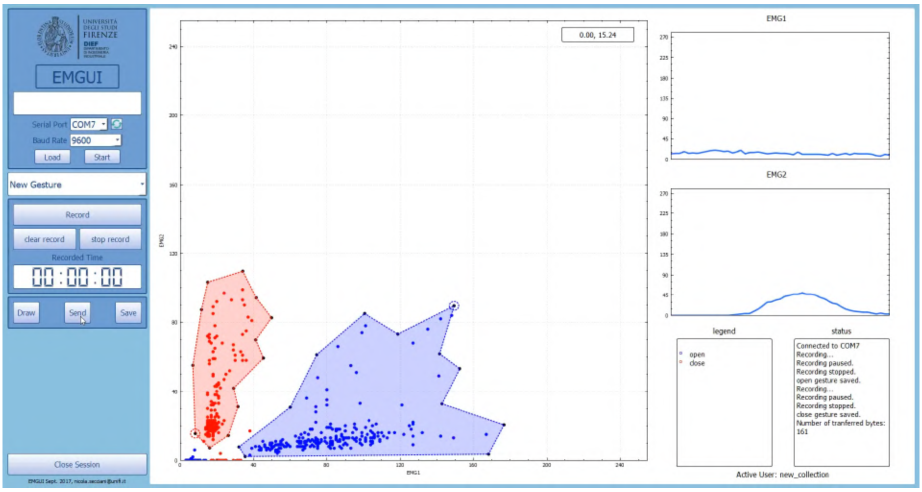
Figure 3.23: The drawing mode that allows to trace the polygons on the central graph. Vertices are drawn by clicking on the graph, polygon sides are previewed while moving the mouse and defined when drawing the second vertex, and, finally, on the top right corner of the Cartesian plane the coordinates are shown.

sEMG envelopes are saved trivially as pairs of integers separated by a horizontal tab; samples belonging to different sample times are reported on different rows. When there is no data recorded for the gesture being acquired (e.g., because it has just been cleaned up with the "Clear record" button), the stop button ends the recording and deletes all its tracks — this process can also be used to delete old recordings that are no longer useful. The points of the gestures correctly acquired and saved will remain on display on the Cartesian plane until the end of the session, or until they are deleted using the method described above. The status box will show the history of the actions carried out during the acquisition.

Once the acquisition phase of the (training) dataset has been completed — that is when an arbitrary number of gestures has been recorded — the next step will involve the manual drawing of two polygonal areas around the points that identify the closing and the opening gesture, as shown in Figure 3.23. In order to do so, the two gestures (one at a time) shall be selected from the drop-down menu and then it will be possible to interact with the central graph and draw the relative polygon by pressing the "Draw" button — a red border around the button will highlight the entering of the drawing mode. The first vertex to be drawn will be circled in the same color as the gesture points. The circle helps closing the polygon: it is sufficient to insert any other vertex within this circumference to end the drawing and close the polygon on the first drawn vertex. The area within the just drawn polygon will be colored with a light shade of the gesture color, and the "gesture_name gesture saved." line will be appended to the status box. The same steps can be repeated to draw the other polygon after having changed the selected gesture from the drop-down menu. It is also possible to delete a polygon already drawn by pressing the "Draw" button, having selected the gesture in question.

Next to the "Draw" button, the keys labeled "Send" and "Save" are in charge of respectively uploading the classifier parameters to the micro-controller board and saving the coordinates of the polygon vertices. The data saving part is done on text files, in a similar way to the saving of the electromyographic acquisitions. The reasoning behind the data transmission protocol — which also influenced the sampling of the sEMG signals itself — is insisted more articulated and engaging. It shall start from the awareness that the data sent to the Arduino Nano board have to be saved on the non-volatile memory so that, each time it is turned on, it can automatically reload them from the internal memory without having to connect again to the graphic interface. The only non-volatile storage that Arduino Nano can offer is called EEPROM memory, which stands for Electrically Erasable Programmable Read-Only Memory[6]. Contrary to what the name seems to suggest, this type of memory can be programmed byte to byte exploiting particular programming signals. This process is facilitated, however, by already implemented medium-high level functions; the Arduino IDE itself provides a ready-to-use library for such task. Unlike flash memory, which can handle multi-byte operations for many life cycles that often stands at ten thousand, EEPROM memory is designed to generally manage operations on a single byte for a number of life cycles that can reach up to one million. While this ensures a long duration of its integrity, it also makes it more challenging to manage the storage of information that shall take place in individual memory registers, each of which contains one byte, namely 8 bits — this means that in every register can be stored just one number from 0 to 255. It is precisely in light of this that it has been decided, from the very beginning, to map the 10-bit sEMG signal (a number between 0 and 1023) returned by Arduino Nano's ADC to 8 bits before plotting them on the Cartesian plane. The classifier parameters, namely the coordinates of the vertices of the polygons, will be hence already

[6] The ATMega328p micro-processor, the one at the core of the Arduino Nano board, offers 1 kB of EEPROM memory.



Serial Port COM7
Baud Rate 9600
New Gesture

```
Connected to COM7
Recording...
Recording paused.
Recording stopped.
open gesture saved.
Recording...
Recording paused.
Recording stopped.
close gesture saved.
Number of tranferred bytes:
161
```

Active User: new_collection

Figure 3.24: A new line in the status box indicates the total number of bytes sent to the micro-controller pressing the "Send" button. Immediately after, the serial communication is interrupted to avoid disturbances that could corrupt the data. The classifier training session is hence concluded.

in the correct format to be appropriately stored in the EEPROM memory of the micro-controller since they are generated from a Cartesian plan limited to abscissa and ordinate values lower than 255. In light of this, the data is then sent to the Arduino Nano in individual packets of one byte, each in a precise sequence. The first outgoing value represents the number of polygons — at the moment two is the only number that is possible to send. Then the number of vertices of the first polygon is sent. Assuming this number is K, the following $2K$ values to be sent are the coordinates of the vertices (first the abscissa and then the ordinate). The same is then repeated for the second polygon. All these values are transmitted interspersed with the separator character ";" to facilitate the micro-controller in receiving and parsing them. At the end of the data transmission, the confirmation of the correct execution of the command is added to the status box with the number of sent bytes attached (Figure 3.24).

During a regular use of the interface the last key to be pressed is the one labeled "Close Session", in the bottom left corner. Doing this proceeds with closing the session, emptying the serial buffer, and returning the interface to the original state shown in Figure 3.20.

THROUGH the procedure shown above, it is possible to perform the training phase of the Point-in-Polygon manually classifier using a custom made GUI running on Microsoft Windows. The classifier parameters are transferred from the desktop PC to the micro-controller via serial communication over a Bluetooth bridge, and are then stored on the EEPROM memory. Assigning these parameters on the specific patient's needs can improve classification accuracy and disturbances rejection, and, since these needs are very likely to vary over time thanks to therapy progress or due to the pathology evolution, it was essential to provide the clinical staff with an easy-to-use tool to quickly change them according to the situation. Their tuning is hence meant to be done manually by professionals who know the medical history of the future users and have followed them during their clinical pathway; it is also likely to be performed within the clinical facility before the patients get discharged.

Once the classifier has been prepared for the real-time recognition of the user's intentions, the code on the micro-controller is changed by pressing a hardware button provided by the Arduino Nano board. The new algorithm no longer sends sEMG signals outwards but, after having collected and correctly mapped them in the [0 255] interval, combines them as the coordinates of the test point whose position in the plane has to be tested. Every sample time, the position of this point is evaluated based on whether or not it belongs to a specific polygon. The code loads the polygon vertices into its flash memory from the EEPROM memory the first time this code is turned on. Classification is performed, therefore, in an exclusive way for precautionary purposes. This means that, although no constraints have been imposed on the reciprocal position of the two different polygons, each point shall belong to only one of them in order to be classified as an open or close intention. All points that simultaneously belong to both polygons or none of them are classified as rest intentions. The just achieved classification is translated into a numeric code — each class is assigned with an integer number — that is transmitted to the paired Bluetooth device as a single byte at 50 Hz frequency.

All this takes place on the micro-controller placed inside the bracelet module which can be interfaced, once again, with a desktop PC for monitoring the classification, or directly to the controller module responsible for implementing the exoskeleton. At this point, the user can control the exoskeleton behavior by reproducing as close as possible the signals which have been used to train the classifier. However, similarly to learning how to drive a car or to ride a bike, the correct, quick, and efficient use of such a wearable device is usually a process that takes time to be mastered and which needs training not only for the classifier but also for the user.

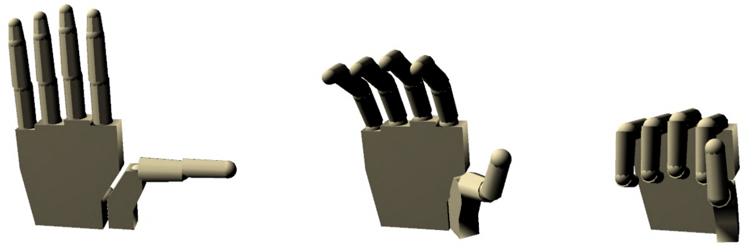


Figure 3.25: Three views of the 3D hand model developed for the training of the exoskeleton user.

For this reason, a further intermediate learning phase has been thought to be carried out before letting the patient wear the HES and testing the resulting control. Another GUI has been developed to ensure that this additional step would not have been too tedious. Instead of resort to the Qt Creator IDE again, Processing 3[7] has been used as it is open source too, it supports both 2D and 3D graphics, it is provided with OpenGL integration for accelerated 2D and 3D, it is well documented. Besides, it can be readapted to different platforms (GNU/Linux, Mac OS X, Windows, Android, ARM). In this graphical interface, has shown in Figure 3.25, a straightforward 3D hand model is displayed — the view can be dynamically changed by dragging the mouse cursor. The movement of this hand is directly related to the type of user's intention that is classified by the electromyographic bracelet, and, as it would happen by controlling the exoskeleton, it is possible to control not only the complete movements but also all the intermediate positions. The speed of the hand motion can be adjusted according to the user's taste, and it is then translated into the actual speed of the exoskeleton movements.

[7] Processing 3 is the third major version of a software sketchbook which takes the name from the homonym programming language.

Chapter 4

Tests and results

All the electronic changes presented in Section 3.2 have necessarily been followed by a profound mechanical redesign of the exoskeleton architecture and transmission system. Also, this process has been used to address some of the problems that emerged while testing the first device (Section 2.3). Although they have not been the subject of this thesis work, they will be briefly described in this chapter in order to make the discussion on the conducted tests more complete. Afterward — in Section 4.1 and Section 4.2 — the chapter will focus on the actual testing phase of the HES resulting from all the modifications introduced by this research activity and by the other works which have accompanied it. The experimental protocol will be described in detail, and the solutions introduced to tackle the significant issues that have arisen during these new tests will be emphasized. At the end of each section, the test results will be presented in terms of device usability aiming at handling objects from daily use.

From the study of the strengths and weaknesses of the mechanical design process that has led to the production of the first HES prototype, the ergonomic, kinematic and dynamic study of the coupling between the hand and the exoskeleton has undergone significant improvements. Over the years, these improvements have resulted in an evolutionary process that has not only radically changed the appearance and equipment of the device [98, 99], but has also made it considerably easy to adapt the exoskeleton to the most diverse hand sizes, including those of children [100, 101]. The wise choice of the production process of the various components — carried out by additive manufacturing in ABS — has also made the replacement of broken or old parts cheap and quick.

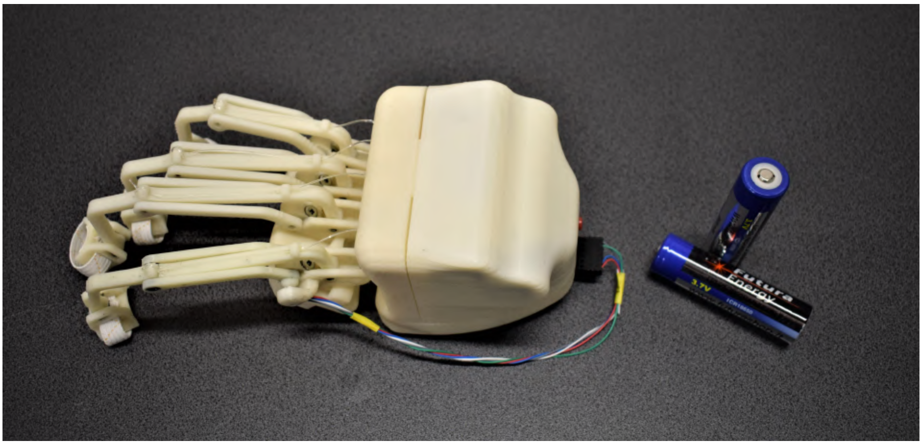


Figure 4.1: The final version of the assistive HES developed at the UNIFI DIEF. On the right, the two rechargeable lithium-ion batteries that power the device.

The new prototype, shown in Figure 4.1, is already at first glance very different from the one shown in Figure 2.4. However, among all the differences — which will be detailed below — it is also possible to see how the planar mechanisms of the fingers, although slimmer than the previous ones, keep the same kinematic structure at the base: a planar single-DOF kinematic chain which acts on the middle phalanx to make the finger perform a complete flexion/extension movement, that is an average ROM for the MCP rotation of about 80 degrees. The motion remains actuated through a cable-driven transmission.

The first notable difference with the previous prototype is that this final version has been designed to have a modular structure in which the motor and control box and the module that collects all the finger mechanisms are two separate and independent entities. The main idea behind this design choice is to minimize downtime for maintenance, whether it is programmed or not. The mechanisms of the fingers are the part that will be mainly subject to replacement as the weaker part from a mechanical point of view and hence easy to break. Each finger mechanism has been designed to bear a maximum load of about 15 N applied by the closing hand to the contact point — value that has been considered an appropriate force output for the manipulation of everyday objects during normal activities [102]. Moreover, since both the recovery from the pathology and the pathology itself may evolve, the trajectories of the fingers may change during the use of the device and, therefore, the geometry of the finger mechanisms might need to be revised. Finally, the same actuation box can be used for different but similar hand sizes, thus streamlining both the production and the design process.

The latter, as it can be seen in Figure 4.2, now incorporates not only all the electronics presented in Section 3.2 but also the batteries, which are therefore no longer located along the forearm. If on one side this means a worse disposition of the masses, which are now concentrated on the hand — the miniaturization process is left to future investigations — on the other it allows for the complete elimination of visible power wiring, making the system electrically safer. The power supply stage, consisting of two rechargeable 3.7 V lithium-ion batteries connected in series with a capacity of 2600 mAh each, has been designed to last from one to two hours depending on the usage. Also, the actuation system has had to undergo changes that have drastically changed its functioning. Reducing the number of motors from four to one has led to the necessary redesign of the transmission of motion from the actuator to the four finger mechanisms. While in the previous device each motor unwound and wrapped the actuation cable around a pulley integral with its shaft, concentrating on the left module in the figure, it can be seen that in the new actuating system the motor no longer acts directly on the cables, but instead sets in motion a secondary shaft through

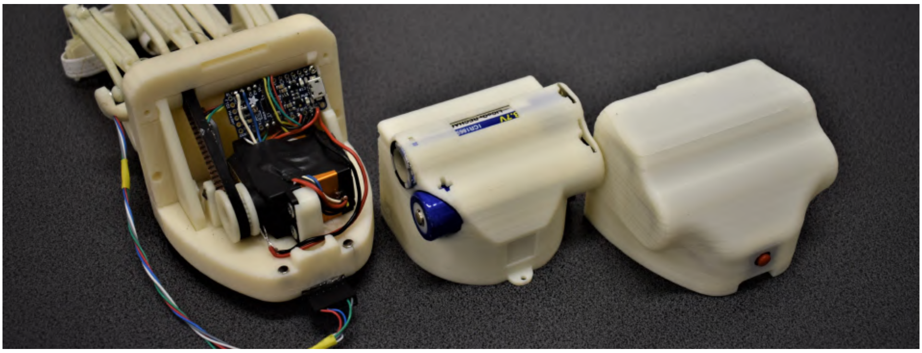


Figure 4.2: An exploded view of the new motor box: on the left, the innermost module houses the actuator, the transmission system, and the control electronics; in the centre, the power stage contains the batteries and it is the only stage accessible to the user once the system has been delivered; on the right, the case enclosing and protecting the other modules, equipped with a switch button.

a toothed belt drive. Four pulleys of different diameters are mounted on this shaft and distributed along its length which, rotating at the same angular speed, wind and unwind different quantities of cable. This adjustment allows the same extension and flexion time to be obtained for all the fingers, although they have different dimensions and, thus, their trajectories have different lengths.

Another meaningful change is that an additional passive DOF has been introduced in joint ① in the form of a second revolute joint — please continue to refer to Figure 2.1 for joint and component nomenclature. In particular, this revolute joint has the axis lying on the flexion/extension plane of the finger and has been added to pander its natural movement of adduction/abduction passively. This addition has resulted not only in improved ergonomics and user comfort but also allows for the alignment of the plane of action of the mechanism with the flexion/extension plane of the finger and helps them to be arranged more efficiently for the object grasping. To note that this inclusion did not affect the total number of DOF of the finger mechanism for it continues to be integral with the finger and to act in the same two-dimensional plane. As a result, the new HES has a total of eight independent DOFs, of which four active and controlled, and four passive, uniformly distributed on all the four long fingers. The average ROM for the MCP rotation is about 80 degrees on the flexion/extension plane while it is roughly 15 degrees on the adduction/abduction one.

The last significant variation from the original prototype concerns the ergonomics of the system dressing. As mentioned in Section 2.3, the previous device was placed in position, over the hand, using a sports glove to which it was sewn. The elasticity of the glove, however, had a double effect: on the one hand, it ensured greater safety of the coupling — as it was intrinsically compliant with any possible displacement between the hand and the exoskeleton — on the other hand, this same characteristic made the exoskeleton produce different trajectories from those for which it was designed — each displacement modified the coupling and thus the relative position between joint ① and the MCP joint. In addition to being uncomfortable and limiting for the user — who was forced to make unnatural movements — this excessive mobility of the exoskeleton base did not ensure a reliable repeatability of the motion. To overcome this problem, an anatomical wrist splint was used to provide a sufficient rigid interface base with the upper limb.

A splint is defined as a device that — when applied over or around an injured segment — increases, improves, or controls the impaired function. They are made by a suitably trained therapist, using thermoplastic material that can be modeled even at relatively low temperatures (about 75°). The splints are modeled directly on the anatomy of the patient and have significant advantages

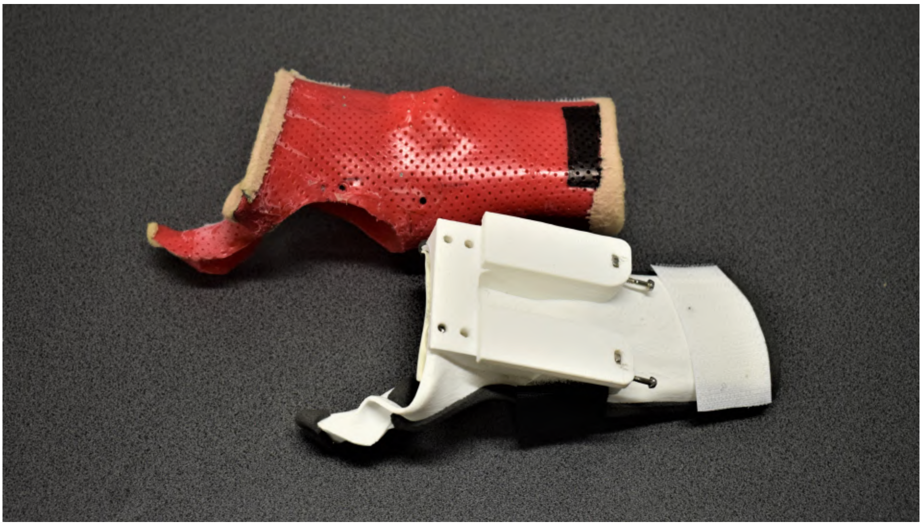


Figure 4.3: On top, the anatomical wrist splint thermoformed on the patient's anatomy; below, the 3D printed reproduction in ABS incorporating a slide and threaded holes for mounting the two HES modules.

for the therapy itself as their wearing is a rehabilitation treatment in itself. Moreover, they are light, resistant, easily removable, easily washable, and have the advantage of being able to be continuously reviewed and modified according to the new requirements of the treatment plan. There are different types of splints: (i) the immobilization ones, which aim to maintain a correct posture of the limb to limit the onset of deformity and to protect the repair of injured tissues; (ii) the ones for protected mobilization, designed to selectively neutralize the proximal or distal joints to facilitate specific muscle or motor activity; (iii) the integrative ones, designed to facilitate a function by compensating with their rigidity the presence of paralyzed muscles or even prevent the appearance of deformities caused by the imbalance between the agonist and antagonist muscles; (iv) the ones for joint recovery, which promote joint recovery in cases of rigidity. In this case, the splint that has been exploited (see Figure 4.3) can be considered as a combination of the typologies for protected mobilization and the integrative one. The main task of the splint was to stiffen the wrist articulation so that the interaction forces between the exoskeleton and the hand were discharged on throughout the forearm. Besides, it also resulted in concentrating

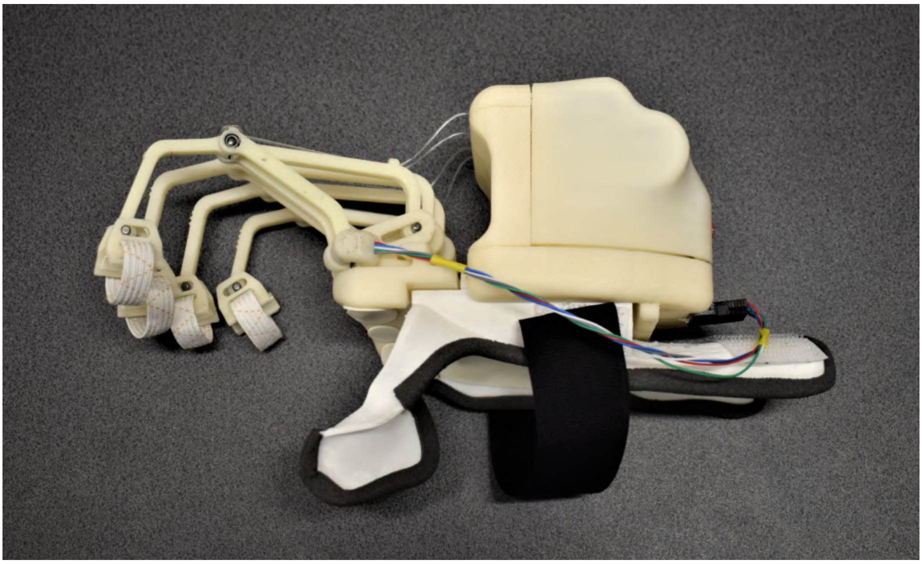


Figure 4.4: The final HES mounted on the tailor-made splint ready to be tested.

the HES action on the fingers and giving greater stability to the system (protected mobilization). Secondly, this splint also provides a rigid support that keeps the first phalanx of the thumb in position of semi-opposition to the palm to facilitate the object grasping, mitigating the tendon retraction (integrative).

Just as it has come out of the thermoforming process, however, the splint was still not suitable as an anchor base for the exoskeleton. A three-dimensional CAD model, which would allow for the processing of the upper surface while maintaining the anatomical correctness of the lower surface, has been created. This process has taken place by exploiting a measuring arm equipped with a 3D laser scanner. Finally, the housing for the module of the fingers — which provides for its attachment through some screws — and a magnetic slide for anchoring the motor box have been designed on the CAD model (both visible in Figure 4.3), and a 3D-printed version of the splint has been produced.

Figure 4.4 shows the complete and assembled version of the HES made within the MDM Lab at the UNIFI DIEF. The device — whose cost and weight have remained almost unchanged since the previous version (excluding the sEMG bracelet) — is fixed to the upper limb through two Velcro straps.

4.1 First test session

A first test session, in the form of a pilot study, has been carried out at the Don Gnocchi IRCCS Rehabilitation Centre once the development of the prototype described above had been completed. The purpose of this study was to verify the actual impact of the exoskeleton use in object handling when worn by a real patient. Given the de facto exploratory nature of these tests, the minute size of the test sample, and due to the low repetitiveness of the classifier training — which was manually performed — no statistical analysis on the classification accuracy has been performed. The test focused instead on measuring grabbing and handling times. One subject (male, aged 54, 1+ Modified Ashworth Scale [103]) has been enrolled for the study. The patient suffers from birth from SMA type II, which is a neuro-degenerative disease with several possible different outcomes. In this specific case, SMA produced selective damage to the extensor muscles of both hands, causing a clenched fist deformity and resulting in hand opening impairment due to tendons retraction. Before the beginning of the tests, the subject has been provided with a written informed consent form and an information sheet.

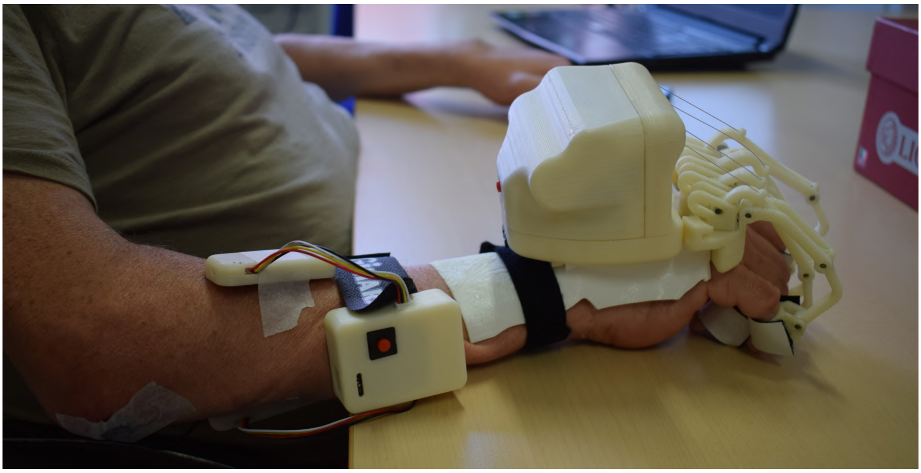


Figure 4.5: The HES and the sEMG bracelet mounted on the test subject's right arm.

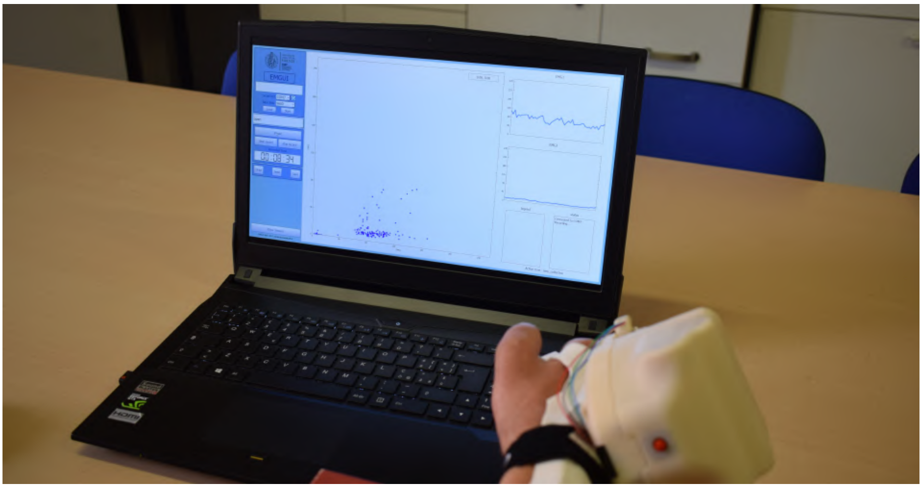


Figure 4.6: The figure shows the collection of the sEMG signal during the training phase which has preceded the test session.

Experimental setup - In order to provide a comfortable and high customized clamping system on the patient's limb, the ergonomics of the device has been improved by housing the HES on a 3D-printed splint, directly manufactured starting from the user's anatomy. Such tailor-made interface has been then fixed to the user's upper limb employing a Velcro fastening, as visible in Figure 4.5, so that it assures a stable kinematic coupling between the exoskeleton and the hand. The bracelet has been previously placed on the user's forearm, and the two EMG sensors have been placed, with the help of a physiotherapist, on the *extensor digitorum* and *flexor digitorum* muscle bands exploiting adhesive Ag/AgCl electrodes. However, since the selected subject was not able to provide reliable signals concerning finger extension without feeling annoyance, it has been decided to use the remarkable cross-talk components from the muscular activity coming from the wrist motions. Before starting the test, the subject has conducted a complete training session of the classifier exploiting the GUI described in Subsection 3.3.2 under the supervision of the same physiotherapist[1] who assisted the dressing. Firstly, the sEMG signals related to the flexion and extension of the wrist have been recorded and labeled respectively as hand

[1] The physiotherapist has been taught how to use the GUI to properly tune the classifier parameters in two practice sessions for a total of four hours the week before the tests.

closing and hand opening intention (the process is shown in Figure 4.6). This phase had necessarily to be performed while the subject was wearing the whole system since the dressing of the device itself also influences the overall muscular activity. Secondly, the physiotherapist has drawn the gesture polygons in the way he thought best. Thirdly, the patient has virtually tested the control experience by commanding the motion of the 3D hand model. A user-training approach has been followed to teach the patient to consistently produce EMG patterns corresponding to clear commands for the exoskeleton, and the transition to the actual test phase has taken place only when the subject expressed positive feedback on the reliability, reactivity, and sensitivity of the classification output. Until that moment, for each time the control was considered not suitable, the training process was repeated from the second step.

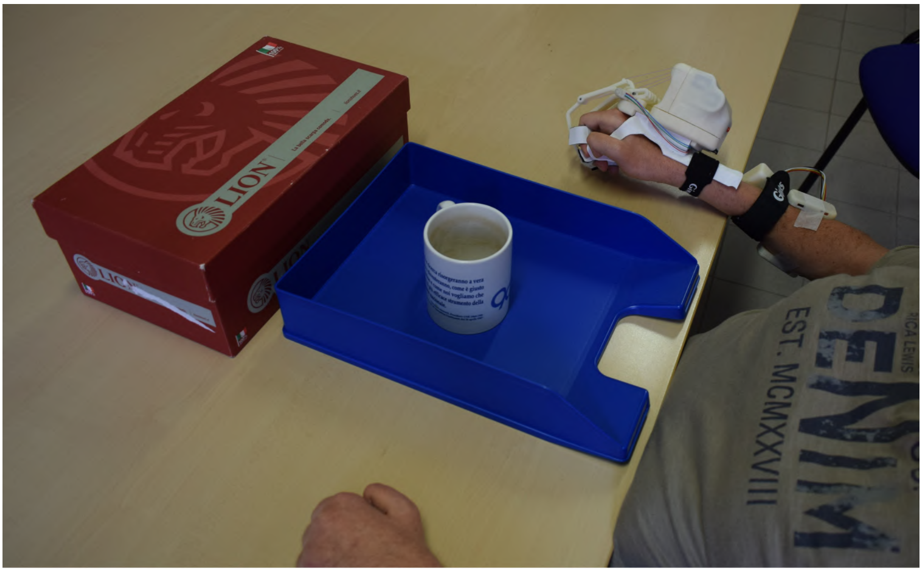


Figure 4.7: An example of the test scenario: the subject is seating in front of a blue dais and a shoe box, in the center of the blue dais one of the test objects (in this case an empty mug) is placed, such object that must be moved over the shoe box, and the action shall be carried out using only the upper right limb on which the HES has been previously mounted.

Experimental protocol - The Single-Case Experimental Design methodology [104] has been chosen for the study: 10 objects of different sizes and shapes from those of daily use (i.e., an apple, a tangerine, a tennis ball, a mug, a 0.5 l bottle of water, a door handle, a spray can, a pot, a felt-it pen, a smartphone) have been selected for a chronometric functional task test. The subject seated in front of a table on which the objects mentioned above have been previously placed within a fixed area (Figure 4.7). The subject, wearing the exoskeleton, has been asked to grasp each object and move it on a standard shoebox (12.5 x 28.5 x 10 cm) placed 10 cm away from the initial position. For each object, a total of five trials have been conducted, and the average grasping time was calculated. Figure 4.8 shows instead the subject wearing the HES while carrying out one of the chronometric tasks.

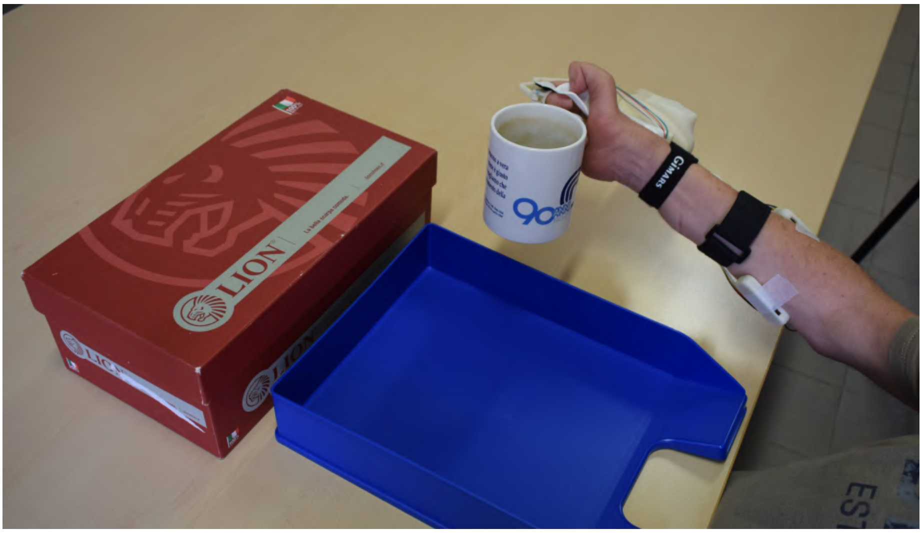


Figure 4.8: The HES user as he moves an empty mug over the shoe box.

Results - The results of the first test session are reported in Figure 4.9. More than the actual time taken to perform the complete tasks — which still results to be quite high compared to reasonable standards for able-bodies — it is interesting to note how these results give an overview of the behavior of the HES as the type of the handled object varies. They show that, in general, grasping time is longer as the object shape gets more complicated, e.g., grasping an apple is quicker than grasping a felt-it pen. This trend can be attributed to the fact that, when actuating four 1-DOF mechanisms at once, grasping an object with a spherical shape requires less time since the exoskeleton better adapts to such forms. It is also worth noting that lateral grasping arose to be more complicated than the vertical one. This difference can be seen comparing grasping of the bottle of water with the tangerine/tennis ball one: the latter required less time despite a similar diameter of the grip. The difficulties encountered by the subject to produce the correct sEMG pattern in the different positions might be attributed to the fact that the upper limb muscles differently compensate for the exoskeleton weight and such cross-talk effects can modify the sEMG signals.

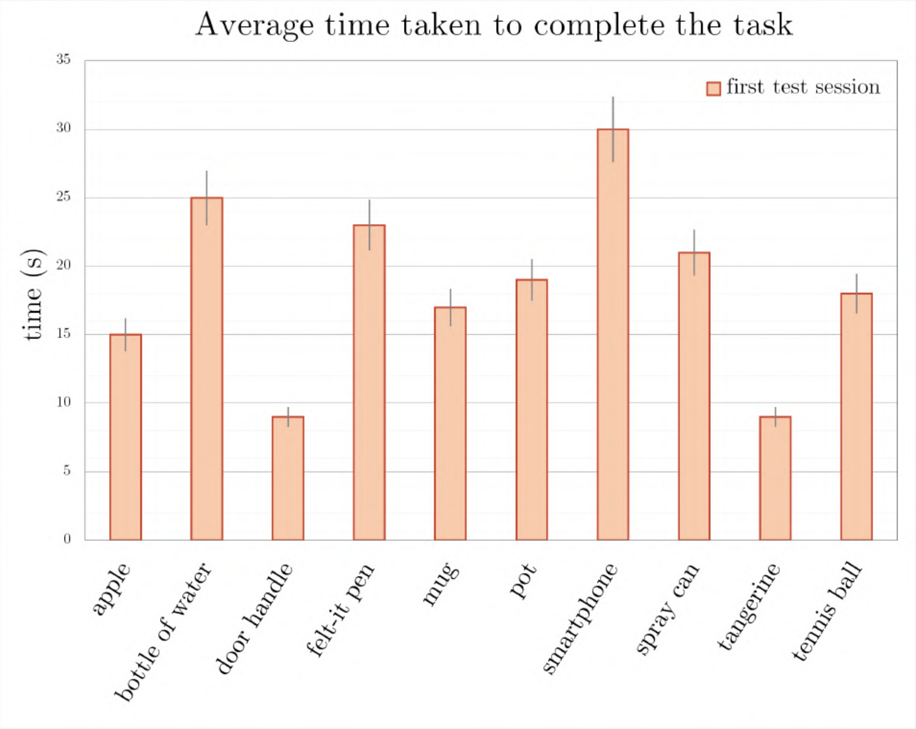
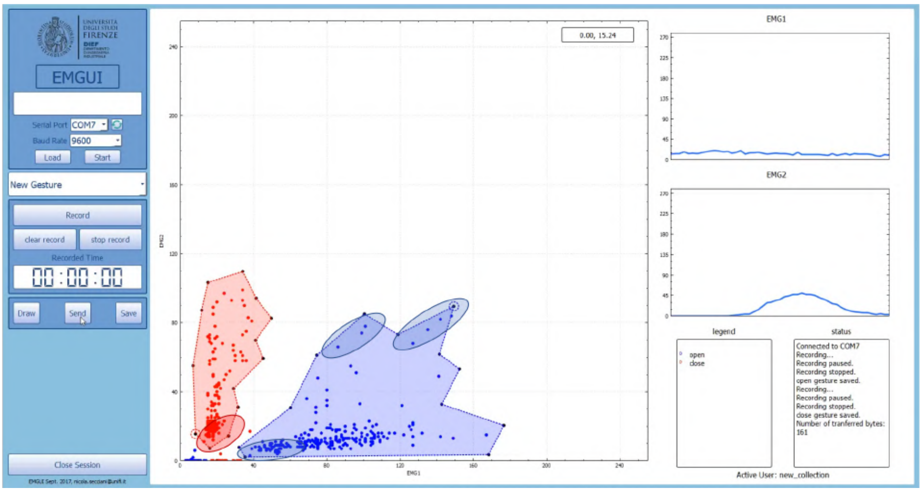


Figure 4.9: The graph shows the results of the second chronometric tasks test. The histogram reports on the x-axis the objects involved and on the y-axis the average time (rounded to the second) taken to complete the exercise.



Serial Port COM7

Baud Rate 9600

```
Connected to COM7
Recording...
Recording paused.
Recording stopped.
open gesture saved.
Recording...
Recording paused.
Recording stopped.
close gesture saved.
Number of tranferred bytes:
161
```

Active User: new_collection

Figure 4.10: In the central graph of the figure, some conditions in which the control becomes glitchy are highlighted by the circled areas. This happens because of the alternation between muscular states whose signals fall alternately in and out of the polygonal areas, producing a flickering transition from motion to stop of the actuator.

Discussion - These tests have also highlighted that executing a command action at 50 Hz frequency results in a stressful operative condition both for the motor and for the user himself. The control system forced the motor to execute the commands coming from the classification loop every time they were different from the previous one. This condition has proven to occur quite frequently when the user (voluntarily or not) has found himself operating the exoskeleton with muscular signals mapped very close to the perimeter of the gestures polygons. In such cases, the classified intention would likely jump from “closing” to “resting” and then back to “closing” very often, as visible in Figure 4.10, even if the actual intention did not change. The motor might have hence been told to move (when the current test point was inside the polygon) and to stop (when the test point was outside the polygon) up to 50 times per second. This working condition has ended up resulting in a glitchy actuation of the exoskeleton which caused: (i) a quick overheating of the motor; (ii) a consequent loss of performance of the whole system; (iii) the onset of a sense of frustration in the user, fatigued by the vibrations and the problematic use.

These tests have also confirmed the goodness of leaving the decision-making process of the classifier parameters to physiotherapists. During the training phase, it has been seen how choosing some key characteristics of the classifier — such as the size and position of the polygons — through a direct interaction with the patient, and basing on his needs and the current and past history of his disease, produces positive results not only on the quality of control but also on the involvement of the patient in the exercise. On one side, changing the size of the polygons affects the selectivity of the classifier: the larger the area underlying the polygons, the easier the user will be able to produce a muscular signal that falls within those figures. Even though, at the same time, this will increase the probability that such intention will be incorrectly classified when not desired. On the other side, the sensitivity of the classification depends on the position of the polygons: the closer the polygon is to the origin of the graph, the less muscular effort is required to trigger the corresponding movement. However, it will also be more challenging to maintain a state of muscle relaxation such as not to incur in erroneous classifications of movement intentions. Furthermore, by bringing both polygons closer to the origin and, therefore, even closer together, there is also a higher risk of particularly harmful incorrect classifications. E.g., a classification of an intention to open when handling a fragile or precious object, or a classification of an intention to close when wanting to release a pungent or hot object quickly. It is therefore essential that these parameters are chosen together with the patient so that he/she is not only more aware but also more in control of the strategy that will be implemented on the HES.

4.2 Second test session

During the second test phase, both the same experimental setup and protocol have been adopted. The same classification strategy has also been exploited. However, changes have been made to the actuation loop to mitigate the glitchy control issue, which has arisen in the previous test session. The chain of instructions that translated the classified intention into low-level commands for the motor has been modified by adding a filtering action provided with a sort of memory of the last executed command and the last classified intentions. Within this new control strategy, the classifier output was no longer translated directly into the corresponding motor command — which means that the motor could have received a new instruction up to 50 times in a second — but it was instead added to a vector containing the last five classified user's intentions; every time a new value is added to such vector the oldest one is overwritten. The vector size — five elements that correspond to a time window of 100 ms since a new element is added every loop — has been chosen not to introduce too much latency, providing a still responsive control experience for a real-time application. The last executed intention is instead stored in a separate variable.

A customized filtering action has been implemented to combine these six variables into one output to determine which command action has to be executed by the HES; the main steps of its operation logic are detailed below. Firstly, each classifiable intention has been assigned to an integer number, such as the rest state is the number in between the other two.

$$\textit{opening} = 1, \qquad \textit{resting} = 2, \qquad \textit{closing} = 3 \tag{4.1}$$

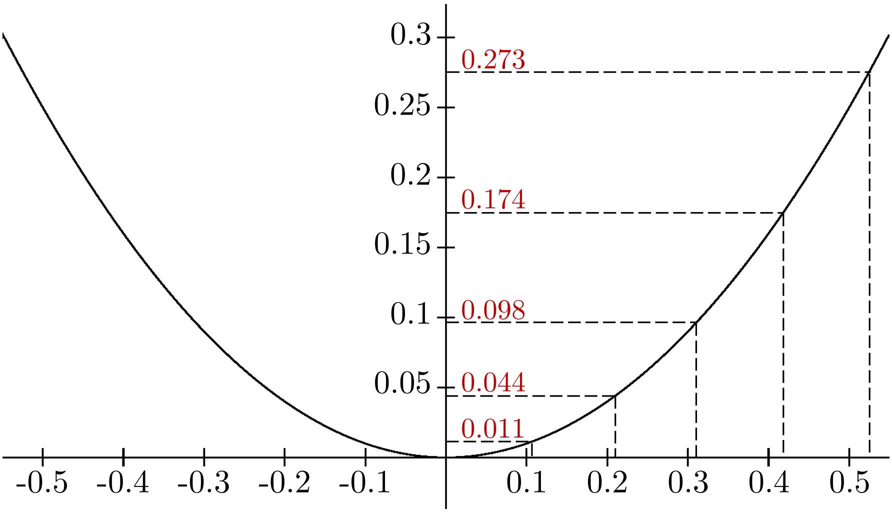


Figure 4.11: The selection process of the five weights to be matched to the different elements contained in the vector of the classified intentions. These values — which, in the figure, are distributed along the ordinate axis and highlighted in red — are extracted from the equation of a unitary parabola $\left(y = x^2\right)$ so that their sum is equal to 0.6 and their corresponding abscissa are evenly spaced on the x-axis. This last characteristic has been chosen since the time interval between the classification of the different intentions remains constant.

Secondly, a weight of 0.4 is assigned to the current executed intention, and the other five weights — heuristically gathered from a unitary parabola for a total sum of 0.6 — are assigned to the classified intentions. The smallest weight is matched to the oldest classified intention, while the most significant weights to the youngest one.

$$c_e \rightarrow w_e = 0.4 \tag{4.2}$$

$$i_{t-4} \rightarrow w_{t-4} = 0.011 \tag{4.3}$$

$$i_{t-3} \rightarrow w_{t-3} = 0.044 \tag{4.4}$$

$$i_{t-2} \rightarrow w_{t-2} = 0.098 \tag{4.5}$$

$$i_{t-1} \rightarrow w_{t-1} = 0.174 \tag{4.6}$$

$$i_t \rightarrow w_t = 0.273 \tag{4.7}$$

Where c_e and w_e are the last executed intention and its respective weight, $i_{t-4}, ..., i_t$ the classified intentions at different time step t, and $w_{t-4}, ..., w_t$ the corresponding weights, chosen as shown in Figure 4.11.

Finally, a weighted average is calculated and rounded to the closest integer corresponding to an intention. The result of these calculations is then assumed as the new intention (n_e) to be executed by the exoskeleton.

$$n_e = w_e c_e + \sum_{k=0}^{4} w_{t-k} in_{t-k} \tag{4.8}$$

Where in_{t-k} are the values assigned to the corresponding intentions i_{t-k}, accordingly to Equation 4.1.

This modification has been done in order to reproduce the effect of a windowed smoothing filter — window length = 100 ms, overlap = 20 ms — which has proved to reject single jumps from one intention to another actively. Such a filter also gives much more importance to the latest classified intentions than to the oldest ones, and it provides the control system with a remarkable but not excessive inertia to changes. While 40% of the decision on the new command to be executed depends in fact on the last intention reproduced by the exoskeleton — this inertial weight is represented by w_e — the remaining 60% of choice falls instead on the intentions classified in the last time window of 100 ms and, in particular, roughly half of this decision weight depends exclusively on the last of them (27.3%). The selected weights are the result of laboratory tests, which

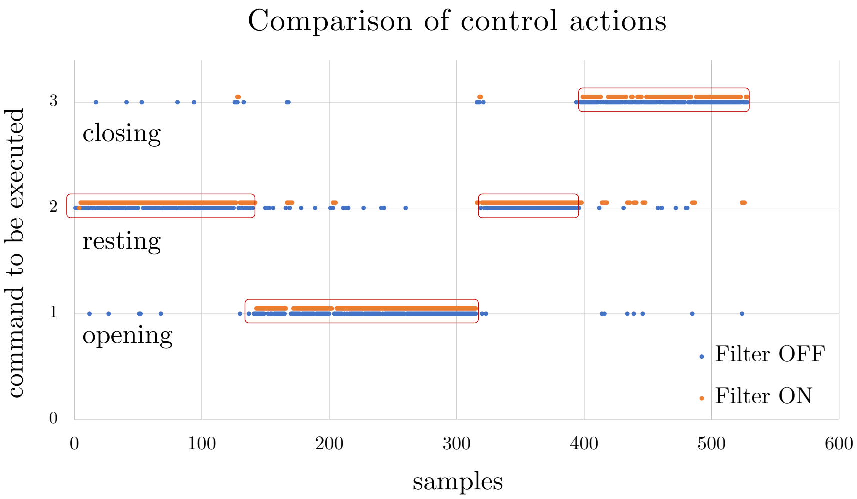


Figure 4.12: The graph shows the difference between the resulting command to be executed generated by the old control strategy (in orange) and the one generated by the new control strategy after the implementation of the memory filtering (in blue) at a sampling frequency of 50 Hz. The real actions to be reproduced are circumscribed in red. The intention labels — opening, closing, and resting — are assigned to different y values according to Equation 4.1.

highlighted how the modified control experience turns out to be much smoother, less power consuming, and less fatiguing for the user if compared to the previous one. The qualitative results of this comparison, in terms of classification accuracy, are shown in Figure 4.12.

The figure shows a manifest decrease in the number of misclassified intentions, that is, all the points that are not inside the red rectangles. The result that is even more interesting, though, is the almost complete elimination of those classifications that are mistakenly recognized as the movement opposite to the real one (i.e., opening intentions classified as closing ones and vice versa). These are particularly harmful because, by imposing a sudden inversion of the exoskeleton motion, subject both the actuation system and the mechanical structure to significant stresses. Moreover, reproducing an intention which is opposite to the one the user wants, not only the control experience risk becoming considerably annoying, but it is also possible to incur in painful or dangerous situations for the user; such as the release of a fragile object during the handling, or the prolonged tightening while touching a pungent or hot object.

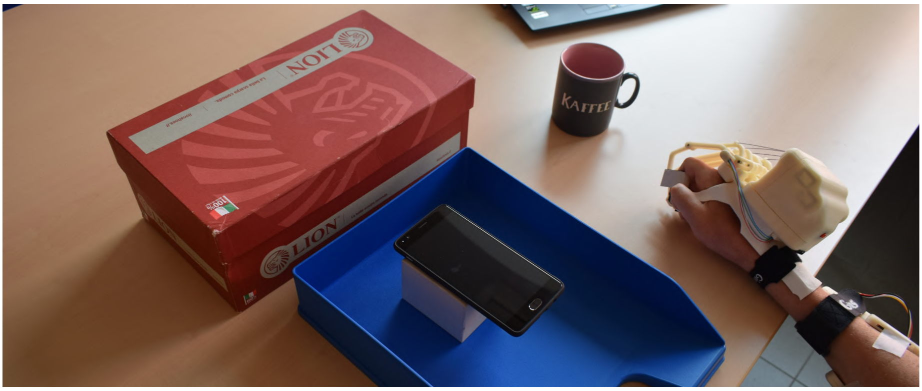


Figure 4.13: During the second test session the experimental setup remained unchanged. In this session, as well as in the previous one, a second small box (white in this figure) has been used to ease the grasping of small or thin objects.

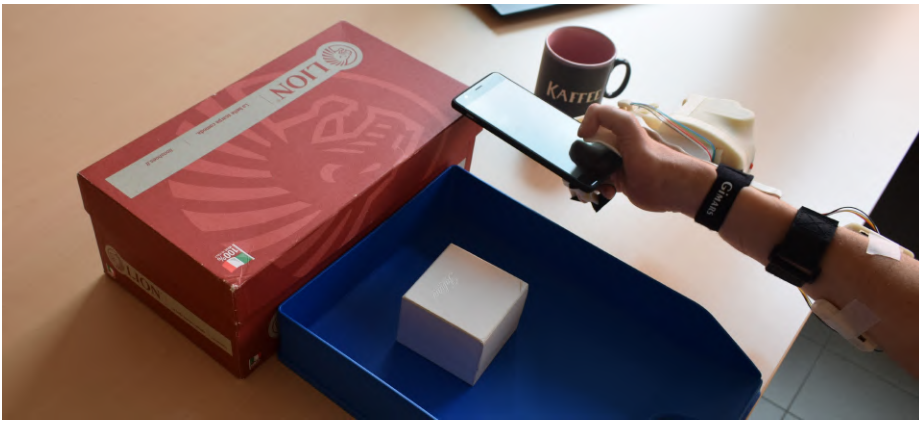


Figure 4.14: The experimental protocol has not been modified as well. The figure shows the HES user as he moves a smartphone over the shoe box.

Results - The second test session has been carried out following the exact procedure of the first one, and involving the same patient and physiotherapist: (i) the HES has been worn on the subject's right upper limb by the physiotherapist; (ii) the subject has undergone the training phase for the classifier, at the end of which the physiotherapist has chosen the polygons for the classification; (iii) the control has been then tested on the virtual 3D hand model; (iv) the subject has finally carried out the pick-and-place chronometric task tests five times per object and the average time to complete each task has been calculated (see Figure 4.13 and Figure 4.14 for another example). The results of this second test are shown below in Figure 4.15, placed side by side with the results of the previous test.

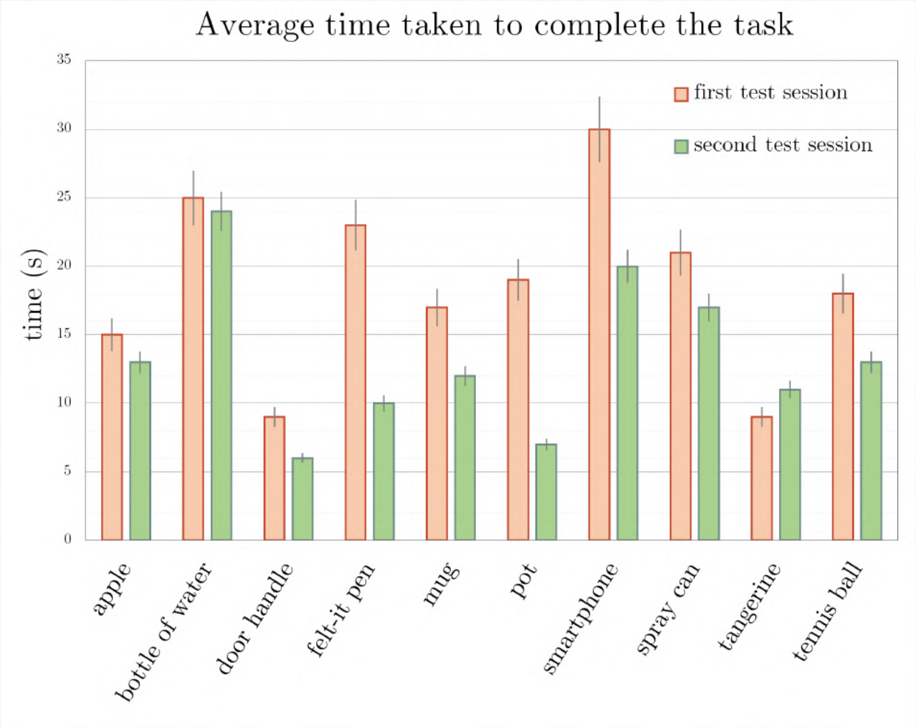


Figure 4.15: The graph shows the results of the first and second test sessions, in orange and green respectively.

Discussion - The evidence from the second test session shows that the introduction of the filtering action benefits not only the control experience but also the effectiveness of the device. It should be noted that in almost all cases — except for the test that involved the tangerine, where the average time required to perform the task has increased — the results indicate that the new control strategy has undoubtedly had a positive impact on reducing the time for the completion of the tasks. In some cases, such as the pot and felt-it pen, the time recorded has more than halved. It should be added that these improvements are not only due to the use of an improved control strategy. Although it has not yet been possible to carry out specific tests to verify the validity of such an assumption, it would, in fact, be unreasonable to exclude the sharing of other factors in improving the performance of the system. The factor that will certainly have to be taken into consideration is the human factor: it is likely that, having already carried out a previous test session, both the subject and the physiotherapist have learned from the experience to better use the tools with which they had to interact. Therefore, just as it is to be expected that the subject has become more familiar with the HES and that he has been able to manage it better, it is fair to assume that also the physiotherapist has been able to take advantage of the previous experience to better tune the classifier on the subject's needs and abilities through the graphical interface.

The last considerations precisely concern the GUI developed for the tuning of the classifier parameters. The second test session has confirmed some of the strengths the graphical interface had already shown in the previous one. Firstly, its use proved to be particularly simple and intuitive: the four hours of teaching, that the physiotherapist who has supervised the tests has undergone, has turned out to be more than sufficient for the complete mastery of the instrument. Secondly, the time required for the system to be ready for the correct classification of sEMG signals — operated through the GUI — turned out to be remarkably short: in both sessions, it was never more than half an hour of classifier training. Finally — although the previous observation has also opened up scenarios in which the training of the classifier takes place before any daily use — it is possible to use the HES successfully while keeping unchanged the parameters identified in a past training phase. This occurrence is due to the possibility of saving data from all the training sessions a patient is subjected to, and then load them all together in the same session in order to identify the polygons that fit the overall set of data; producing a robust classification to the various possible conditions of use of the device.

Chapter 5

Conclusions

This work collects part of the research activities on wearable robotics carried out at the Mechatronics and Dynamic Modelling Laboratory of the Department of Industrial Engineering of the University of Florence from 2016 to 2019. What is reported in these pages focused, in particular, on the development of a control strategy for a hand exoskeleton system based on the exploitation of surface electromyographic signals. Paving the way to innovative low-cost solutions in the field of robotic assistance for people with hand disabilities has constituted the main subject of this work, motivated by the fact that people who have lost — wholly or partially — hand dexterity are deeply compromised in the possibility of living an independent and healthy life. Under the pressure of substantial technological progress in the middle of the Fourth Industrial Revolution, with the World Health Organization indicating a clear positive trend in the number of people who will need assistance in the short term future, robotic devices nowadays represent a useful tool to democratize the regaining of a better quality of life, both for those who have lost it as a result of an accident or for those who see it undermined due to a chronic disease.

The overall research activity aimed not only to present a novel sEMG-based strategy to control a lightweight ($\simeq$ 500 g) and low cost ($\simeq$ 500) fully wearable assistive HES, but also to describe the patient-centered process that has led to its implementation on an embedded system. The focal point of the whole activity remains the choice of the classifier algorithm for the recognition of the user's motor intentions to be reproduced by the exoskeleton, the very same device which has later been tested on a real patient during simulated ADLs. The novelty of this work lies in the fact that, comparing it with the related works available in literature — reported in Chapter 1 — and with the previously implemented control strategy — described in Chapter 2 — the proposed solution shows a peculiar aspect which raises the bar of the current state of the art of wearable assistive technologies. Unlike other studies, the complexity of the classifier has been brought from a one-dimensional to a two-dimensional level exploiting a particular algorithm whose use in the field of wearable robotics is not reported in any other work in literature even though its performance has proved to be valid in comparison with standard machine learning techniques — as reported in Chapter 3. This lightweight decision-making algorithm, which is more robust to external disturbances and more flexible to the classification of different intentions compared to the one-dimensional strategies, has remained suitable for embedded applications and this is another crucial point of the work. Two test sessions have been performed on a real patient in a clinical environment to validate the discussed approach and qualitatively assess the usability of the proposed control strategy during everyday life object handling. Promising results — outlined in Chapter 4 — have been obtained at the end of this pilot study, highlighting the goodness of the proposed solution, which has proved to be an attractive approach in the development of control strategies for wearable robotic devices. Nonetheless, there is still room for improvement.

Through this research activity, a more mature state of the device has been reached, for which it is now possible to request approval from an ethics committee for the actual clinical trial. Enlarging the test sample will allow for a proper statistical analysis both for classification accuracy and for the average grasping time of different everyday objects. It will also honestly assess if the same strategy can fit different users with different pathologies. In parallel, a variant of the actuation strategy will be tested for accelerating the grasping process but also to provide a more natural control of the hand motion. The base idea will be to command the motor to move at a speed proportional to the level of muscle contraction. Finally, the classification phase will be further enhanced: the development of a new classification procedure based on the execution of more sophisticated algorithms has already started to be designed. Micro-controllers technologies are evolving fast nowadays, and some manufacturers already provide tools for the implementation of pre-trained artificial neural networks directly in their microchips. The exponential growth of the computational power of the micro-controllers is opening, in fact, the doors of deep learning to wearable applications based on low-power embedded electronics.

All these topics, whose investigation constitutes a natural continuation of the research activity carried out thus far, will be subjected to further developments.

References

[1] W. H. Organization, “World report on disability,” tech. rep., 2011.

[2] W. H. Organization, “WHO global disability action plan 2014-2021: better health for all people with disability,” tech. rep., 2014.

[3] M. Butter, A. Rensma, J. v. Boxsel, S. Kalisingh, M. Schoone, M. Leis, G. J. Gelderblom, G. Cremers, M. d. Wilt, W. Kortekaas, *et al.*, “Robotics for healthcare: Final report,” tech. rep., 2008.

[4] G. J. Gelderblom, M. De Wilt, G. Cremers, and A. Rensma, “Rehabilitation robotics in robotics for healthcare; a roadmap study for the european commission,” in *2009 IEEE International Conference on Rehabilitation Robotics*, pp. 834–838, June 2009.

[5] R. A. R. C. Gopura, K. Kiguchi, and D. S. V. Bandara, “A brief review on upper extremity robotic exoskeleton systems,” in *2011 6th International Conference on Industrial and Information Systems*, pp. 346–351, Aug 2011.

[6] Z. Lovrenovic and M. Doumit, “Review and analysis of recent development of lower extremity exoskeletons for walking assist,” in *2016 IEEE EMBS International Student Conference (ISC)*, pp. 1–4, May 2016.

[7] C. Toledo, L. Leija, R. Munoz, A. Vera, and A. Ramirez, “Upper limb prostheses for amputations above elbow: A review,” in *2009 Pan American Health Care Exchanges*, pp. 104–108, March 2009.

[8] J. D. Weiland and M. S. Humayun, “Retinal prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1412–1424, May 2014.

[9] G. P. Mylonas, V. Vitiello, T. P. Cundy, A. Darzi, and G. Yang, "Cyclops: A versatile robotic tool for bimanual single-access and natural-orifice endoscopic surgery," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2436–2442, May 2014.

[10] M. Zampolini, E. Todeschini, M. Bernabeu Guitart, H. Hermens, S. Ilsbroukx, V. Macellari, R. Magni, M. Rogante, S. Scattareggia Marchese, M. Vollenbroek, and C. Giacomozzi, "Tele-rehabilitation: present and future," *Annali dell'Istituto superiore di sanitá*, vol. 44, pp. 125–134, January 2008.

[11] C.-A. Smarr, C. B. Fausset, and W. A. Rogers, "Understanding the potential for robot assistance for older adults in the home environment," tech. rep., Georgia Institute of Technology, April 2011.

[12] L. Rosenstein, A. Ridgel, B. S. A. Thota, and J. Alberts, "Effects of combined robotic therapy and repetitive-task practice on upper-extremity function in a patient with chronic stroke," *The American journal of occupational therapy: official publication of the American Occupational Therapy Association*, vol. 1, pp. 28–35, January 2008.

[13] A. Heller, D. Wade, V. Wood, A. Sunderland, R. Hewer, and E. Ward, "Arm function after stroke: measurement and recovery over the first three months," *Journal of Neurology, Neurosurgery and Psychiatry*, vol. 131, pp. 714–719, July 1987.

[14] D. Wade, R. Langton-Hewer, V. Wood, C. Skilbeck, and H. Ismail, "The hemiplegic arm after stroke: measurement and recovery," *Journal of Neurology, Neurosurgery and Psychiatry*, vol. 46, pp. 521–524, July 1983.

[15] A. Sunderland, D. Tinson, L. Bradley, and R. Hewer, "Arm function after stroke. an evaluation of grip strength as a measure of recovery and a prognostic indicator," *Journal of Neurology, Neurosurgery and Psychiatry*, vol. 52, pp. 1267–1272, December 1989.

[16] H. Nakayama, H. Jrgensen, H. Raaschou, and T. Olsen, "Recovery of upper extremity function in stroke patients: the copenhagen stroke study," *Archives of Physical Medicine and Rehabilitation*, vol. 75, pp. 394–398, May 1994.

[17] P. Heo, G. M. Gu, and S. Lee, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *International Journal of Precision Engineering and Manufacturing*, vol. 13, pp. 807–824, May 2012.

[18] M. Troncossi, M. Mozaffari-Foumashi, and V. Parenti-Castelli, “An original classification of rehabilitation hand exoskeletons,” *Journal of Robotics and Mechanical Engineering Research*, vol. 1, pp. 17–29, December 2016.

[19] R. A. Bos, C. J. W. Haarman, T. Stortelder, K. Nizamis, J. L. Herder, A. H. A. Stienen, and D. H. Plettenburg, “A structured overview of trends and technologies used in dynamic hand orthoses,” *Journal of NeuroEngineering and Rehabilitation*, vol. 13, p. 62, June 2016.

[20] M. Sarac, M. Solazzi, and A. Frisoli, “Design requirements of generic hand exoskeletons and survey of hand exoskeletons for rehabilitation, assistive, or haptic use,” *IEEE Transactions on Haptics*, vol. 12, pp. 400–413, June 2019.

[21] M. Delph, S. Fischer, P. Gauthier, C. Luna, E. Clancy, and G. Fischer, “A soft robotic exomusculature glove with integrated semg sensing for hand rehabilitation,” in *International Conference on Rehabilitation Robotics (ICORR)*, June 2013.

[22] H. K. Yap, F. Nasrallah, J. H. Lim, F. Z. Low, J. C. H. Goh, and R. C. H. Yeow, “Mrc-glove: A fmri compatible soft robotic glove for hand rehabilitation application,” in *International Conference on Rehabilitation Robotics (ICORR)*, pp. 735–740, August 2015.

[23] R. Deimel and O. Brock, “A novel type of compliant and underactuated robotic hand for dexterous grasping,” *The International Journal of Robotics Research*, vol. 35, pp. 161–185, August 2015.

[24] P. Polygerinos, Z. Wang, K. Galloway, R. Wood, and W. C.J., “Soft robotic glove for combined assistance and at-home rehabilitation,” *Robotics and Autonomous Systems*, vol. 73, pp. 135–143, September 2015.

[25] A. Stilli, A. Cremoni, M. Bianchi, A. Ridolfi, F. Gerii, F. Vannetti, H. A. Wurdemann, B. Allotta, and K. Althoefer, “Airexglove a novel pneumatic exoskeleton glove for adaptive hand rehabilitation in post-stroke patients,” in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 579–584, April 2018.

[26] C. Walsh, “Human-in-the-loop development of soft wearable robots,” *Nature Reviews Materials*, vol. 3, p. 78, May 2018.

[27] A. Chiri, N. Vitiello, F. Giovacchini, S. Roccella, F. Vecchi, and M. Carrozza, "Mechatronic design and characterization of the index finger module of a hand exoskeleton for post-stroke rehabilitation," *IEEE/ASME Transactions on Mechatronics*, vol. 17, pp. 884–894, October 2012.

[28] A. Wege and G. Hommel, "Development and control of a hand exoskeleton for rehabilitation of hand injuries," in *International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3046–3051, August 2005.

[29] C. Takahashi, L. Der-Yeghiaian, V. Le, R. Motiwala, and S. Cramer, "Robot-based hand motor therapy after stroke," *Brain: a journal of neurology*, vol. 131, pp. 425–37, March 2008.

[30] M. Mulas, M. Folgheraiter, and G. Gini, "An emg-controlled exoskeleton for hand rehabilitation," in *International Conference on Rehabilitation Robotics (ICORR)*, pp. 371 – 374, June 2013.

[31] K. Rotella, M.F.and Reuther, C. Hofmann, E. Hage, and B. BuSha, "An orthotic hand-assistive exoskeleton for actuated pinch and grasp," in *Northeast Bioengineering Conference*, pp. 1–2, April 2009.

[32] L. Martinez, O. Olaloye, M. Talarico, S. Shah, R. Arends, and B. BuSha, "A power-assisted exoskeleton optimized for pinching and grasping motions," in *Northeast Bioengineering Conference*, pp. 1–2, March 2010.

[33] S. Moromugi, Y. Koujina, S. Ariki, T. Okamoto, A.and Tanaka, M. Q. Feng, and T. Ishimatsu, "Muscle stiffness sensor to control an assistance device for the disabled," *Artificial Life and Robotics*, vol. 8, pp. 42–45, January 2004.

[34] J. Iqbal, N. Tsagarakis, A. Fiorilla, and D. Caldwell, "A portable rehabilitation device for the hand," in *International Conference of the IEEE Engineering in Medicine and Biology*, pp. 3694 – 3697, August 2010.

[35] K. Tong, S. Ho, P. Pang, X. Hu, W. Tam, K. Fung, X. Wei, P. Chen, and M. Chen, "An intention driven hand functions task training robotic system," in *International Conference of the IEEE Engineering in Medicine and Biology*, August 2010.

[36] J. Iqbal and B. Khelifa, "Stroke rehabilitation using exoskeleton-based robotic exercisers: Mini review," *Biomedical Research*, vol. 26, pp. 197–201, December 2014.

[37] L. Lucas, M. DiCicco, and Y. Matsuoka, “An emg-controlled hand exoskeleton for natural pinching,” *Journal of Robotics and Mechatronics*, vol. 16, pp. 1–7, January 2004.

[38] R. M. Singh, S. Chatterji, and A. Kumar, “Trends and challenges in emg based control scheme of exoskeleton robots-a review,” *International Journal of Scientific and Engineering Research*, vol. 3, pp. 933–40, August 2012.

[39] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan, and T. Chang, “Surface electromyography signal processing and classification techniques,” *Sensors*, vol. 13, pp. 12431–12466, September 2013.

[40] N. Nazmi, M. Abdul Rahman, S.-I. Yamamoto, S. Ahmad, H. Zamzuri, and S. Mazlan, “A review of classification techniques of emg signals during isotonic and isometric contractions,” *Sensors*, vol. 16, p. 1304, August 2016.

[41] Q. Meng, Q. Meng, H. Yu, and X. Wei, “A survey on semg control strategies of wearable hand exoskeleton for rehabilitation,” in *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 165–169, June 2017.

[42] E. J. Rechy-Ramirez and H. Hu, “Bio-signal based control in assistive robots: a survey,” *Digital Communications and networks*, vol. 1, pp. 85–101, March 2015.

[43] B. Hudgins, P. Parker, and R. N. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Transactions on Biomedical Engineering*, vol. 40, pp. 82–94, February 1993.

[44] C. Sapsanis, G. Georgoulas, and A. Tzes, “Emg based classification of basic hand movements based on time-frequency features,” in *21st Mediterranean Conference on Control and Automation*, pp. 716–722, June 2013.

[45] E. Scheme, B. Lock, L. Hargrove, W. Hill, U. Kuruganti, and K. Englehart, “Motion normalized proportional control for improved pattern recognition-based myoelectric control,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 149–157, March 2013.

[46] A. A. B. A. Nadzri, S. A. Ahmad, M. H. Marhaban, and H. Jaafar, “Characterization of surface electromyography using time domain features for

determining hand motion and stages of contraction," *Australasian physical & engineering sciences in medicine*, vol. 37, pp. 133–137, January 2014.

[47] K. Veer and T. Sharma, "A novel feature extraction for robust emg pattern recognition," *Journal of medical engineering & technology*, vol. 40, pp. 149–154, February 2016.

[48] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers in neurorobotics*, vol. 10, p. 9, September 2016.

[49] N. Jaquier and S. Calinon, "Gaussian mixture regression on symmetric positive definite matrices manifolds: Application to wrist motion estimation with semg," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 59–64, September 2017.

[50] F. Ryser, T. Bützer, J. P. Held, O. Lambercy, and R. Gassert, "Fully embedded myoelectric control for a wearable robotic hand orthosis," in *2017 International Conference on Rehabilitation Robotics (ICORR)*, pp. 615–621, July 2017.

[51] G. Nakamura, T. Shibanoki, Y. Kurita, Y. Honda, A. Masuda, F. Mizobe, T. Chin, and T. Tsuji, "A virtual myoelectric prosthesis training system capable of providing instructions on hand operations," *International Journal of Advanced Robotic Systems*, vol. 14, p. 172988141772845, September 2017.

[52] Y. Wan, Z. Han, J. Zhong, and G. Chen, "Pattern recognition and bionic manipulator driving by surface electromyography signals using convolutional neural network," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, p. 1729881418802138, 2018.

[53] J. M. Ochoa, D. G. Kamper, M. Listenberger, and S. W. Lee, "Use of an electromyographically driven hand orthosis for training after stroke," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–5, June 2011.

[54] N. Ho, K. Tong, X. Hu, K. Fung, X. Wei, W. Rong, and E. Susanto, "An emg-driven exoskeleton hand robotic training device on chronic stroke

subjects: task training system for stroke rehabilitation," in *2011 IEEE international conference on rehabilitation robotics*, pp. 1–5, June 2011.

[55] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. P. Castelli, C. Procopio, G. Lamola, C. Chisari, *et al.*, "An emg-controlled robotic hand exoskeleton for bilateral rehabilitation," *IEEE transactions on haptics*, vol. 8, pp. 140–151, March 2015.

[56] C. Meeker, S. Park, L. Bishop, J. Stein, and M. Ciocarlie, "Emg pattern classification to control a hand orthosis for functional grasp assistance after stroke," in *2017 international conference on rehabilitation robotics (ICORR)*, pp. 1203–1210, February 2018.

[57] R. Conti, E. Meli, and A. Ridolfi, "A novel kinematic architecture for portable hand exoskeletons," *Mechatronics*, vol. 35, pp. 192–207, April 2016.

[58] R. Conti, E. Meli, A. Ridolfi, M. Bianchi, L. Governi, Y. Volpe, and B. Allotta, "Kinematic synthesis and testing of a new portable hand exoskeleton," *Meccanica*, vol. 52, pp. 2873–2897, January 2017.

[59] M. Bianchi, F. Fanelli, E. Meli, A. Ridolfi, F. Vannetti, M. Bianchini, and B. Allotta, "Optimization-based scaling procedure for the design of fully portable hand exoskeletons," *Meccanica*, vol. 53, pp. 3157–3175, May 2018.

[60] M. Luersen, R. Le Riche, and F. Guyon, "A constrained, globalized and bounded nelder-mead method for engineering optimization," *Structural and Multidisciplinary Optimization*, vol. 27, pp. 43–54, May 2004.

[61] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, January 1965.

[62] R. Merletti, P. A. Parker, and P. J. Parker, *Electromyography: physiology, engineering, and non-invasive applications*, vol. 11. John Wiley & Sons, August 2004.

[63] K. Mills, "The basics of electromyography," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, pp. 1132–1135, July 2005.

[64] C. J. De Luca, "Physiology and mathematics of myoelectric signals," *IEEE Transactions on Biomedical Engineering*, pp. 313–325, July 1979.

[65] M. C. Garcia and T. Vieira, “Surface electromyography: Why, when and how to use it,” *Revista andaluza de medicina del deporte*, vol. 4, pp. 17–28, April 2011.

[66] D. Stegeman and H. Hermens, “Standards for surface electromyography: The european project surface emg for non-invasive assessment of muscles (seniam),” *Enschede: Roessingh Research and Development*, pp. 108–12, January 2007.

[67] G. Li, Y. Li, L. Yu, and Y. Geng, “Conditioning and sampling issues of emg signals in motion recognition of multifunctional myoelectric prostheses,” *Annals of biomedical engineering*, vol. 39, pp. 1779–1787, February 2011.

[68] X. Li, S. Chen, H. Zhang, O. W. Samuel, H. Wang, P. Fang, X. Zhang, and G. Li, “Towards reducing the impacts of unwanted movements on identification of motion intentions,” *Journal of Electromyography and Kinesiology*, vol. 28, pp. 90–98, April 2016.

[69] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, “Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, pp. 186–192, December 2010.

[70] A. Phinyomark, R. N. Khushaba, E. Ibánez-Marcelo, A. Patania, E. Scheme, and G. Petri, “Navigating features: a topologically informed chart of electromyographic features space,” *Journal of The Royal Society Interface*, vol. 14, p. 20170734, December 2017.

[71] A. Phinyomark, R. N Khushaba, and E. Scheme, “Feature extraction and selection for myoelectric control based on wearable emg sensors,” *Sensors*, vol. 18, p. 1615, May 2018.

[72] C. Spiewak, M. Islam, M. A.-U. Zaman, M. H. Rahman, *et al.*, “A comprehensive study on emg feature extraction and classifiers,” *Open Access Journal Of Biomedical Engineering And Biosciences*, vol. 1, pp. 17–26, February 2018.

[73] C. M. Bishop, *Pattern recognition and machine learning.* Springer, January 2006.

[74] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, October 2007.

[75] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression.* John Wiley & Sons, January 2013.

[76] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, September 1995.

[77] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, pp. 199–222, January 1998.

[78] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, pp. 660–674, June 1991.

[79] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, September 1995.

[80] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, pp. 21–27, January 1967.

[81] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?," in *International conference on database theory*, pp. 217–235, January 1999.

[82] N. Secciani, M. Bianchi, A. Ridolfi, F. Vannetti, and B. Allotta, "Assessment of a hand exoskeleton control strategy based on users intentions classification starting from surface emg signals," in *International Symposium on Wearable Robotics*, pp. 440–444, January 2019.

[83] N. Secciani, M. Bianchi, E. Meli, Y. Volpe, and A. Ridolfi, "A novel application of a surface electromyography-based control strategy for a hand exoskeleton system: A single-case study," *International Journal of Advanced Robotic Systems*, vol. 16, p. 1729881419828197, February 2019.

[84] K. Englehart, B. Hudgins, *et al.*, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 50, pp. 848–854, August 2003.

[85] F. Montagnani, M. Controzzi, and C. Cipriani, “Independent long fingers are not essential for a grasping hand,” *Scientific reports*, vol. 6, p. 35545, October 2016.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, pp. 2825–2830, January 2012.

[87] E. W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *biometrics*, vol. 21, pp. 768–769, January 1965.

[88] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281–297, January 1967.

[89] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, pp. 651–666, June 2010.

[90] R. L. Thorndike, “Who belongs in the family?,” *Psychometrika*, vol. 18, pp. 267–276, December 1953.

[91] A. Kane, “Determining the number of clusters for a k-means clustering algorithm,” *Indian Journal of Computer Science and Engineering*, vol. 3, pp. 670–672, October 2012.

[92] M. Shimrat, “Algorithm 112: position of point relative to polygon,” *Communications of the ACM*, vol. 5, p. 434, August 1962.

[93] A. Korniłowicz, “A proof of the jordan curve theorem via the brouwer fixed point theorem,” *Mechanized Mathematics and its Applications*, vol. 6, pp. 33–40, November 2007.

[94] K. Weiler, “An incremental angle point in polygon test,” *Graphics gems IV*, vol. 4, p. 16, August 1994.

[95] K. Hormann and A. Agathos, “The point in polygon problem for arbitrary polygons,” *Computational Geometry*, vol. 20, pp. 131–144, December 2001.

[96] M. Galetzka and P. Glauner, "A simple and correct even-odd algorithm for the point-in-polygon problem for complex polygons," in *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, February 2017.

[97] X. Li, S. Chen, H. Zhang, O. W. Samuel, H. Wang, P. Fang, X. Zhang, and G. Li, "Towards reducing the impacts of unwanted movements on identification of motion intentions," *Journal of Electromyography and Kinesiology*, vol. 28, pp. 90–98, April 2016.

[98] N. Secciani, M. Bianchi, A. Meschini, A. Ridolfi, Y. Volpe, L. Governi, and B. Allotta, "Assistive hand exoskeletons: the prototypes evolution at the university of florence," in *The International Conference of IFToMM ITALY*, pp. 307–315, January 2019.

[99] N. Secciani, M. Bianchi, A. Ridolfi, F. Vannetti Yary Volpe, L. Governi, M. Bianchini, and B. Allotta, "Tailor-made hand exoskeletons at the university of florence: From kinematics to mechatronic design," *Machines*, vol. 7, p. 22, April 2019.

[100] M. Bianchi, N. Secciani, A. Ridolfi, F. Vannetti, and G. Pasquini, "Kinematics-based strategy for the design of a pediatric hand exoskeleton prototype," in *The International Conference of IFToMM ITALY*, pp. 501–508, January 2019.

[101] M. Bianchi, N. Secciani, A. Ridolfi, F. Vannetti, and G. Pasquini, "Model-based approach in developing a hand exoskeleton for children: A preliminary study," in *International Symposium on Wearable Robotics*, pp. 490–494, January 2019.

[102] K. Matheus and A. M. Dollar, "Benchmarking grasping and manipulation: Properties of the objects of daily living," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5020–5027, November 2010.

[103] C. P. Charalambous, "Interrater reliability of a modified ashworth scale of muscle spasticity," in *Classic papers in orthopaedics*, pp. 415–417, December 2014.

[104] J. D. Smith, "Single-case experimental designs: a systematic review of published research and current standards," *Psychological methods*, vol. 17, p. 510, December 2012.