

POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Department of Control and Computer Engineering

Master Thesis

Hannes Prosthesis Control Based on Regression Machine Learning Algorithms



Academic Supervisor:

Prof. Marcello Chiaberge

Candidate:

Dario Di Domenico
s263409

Company Supervisors:

Eng. Nicolò Boccardo

Eng. Andrea Marinelli

October 2020

Table of Contents

List of Figures	5
List of Tables	9
Abbreviations	11
Abstract	13
Introduction.....	17
I. Background	19
1. A Brief History	21
2. Muscle activation.....	23
2.1 Types of muscle fibers.....	23
2.2 Skeletal muscle	23
2.3 Mechanism of muscle contraction	25
2.4 Neuromuscular junction and biochemical changes	26
3. State of the Art.....	29
4. Control Strategies	31
II. Materials and Methods	37
5. Hannes	39
5.1 Hardware.....	41
5.1.1 Electrodes	42

5.1.1.1	Ottobock sensors	43
5.1.1.2	IIT sensors.....	44
5.2	Firmware.....	45
5.3	Software.....	49
5.3.1	E-DATS (EMG – Data Acquisition & Training Software).....	49
5.3.1.1	General settings.....	49
5.3.1.2	Data acquisition & Training phase.....	50
5.3.1.3	Real-time simulation	58
5.3.1.4	Model uploading & Simulation in Virtual Reality.....	59
5.3.2	Virtual Reality	60
6.	Study protocol	63
7.	Pattern-Recognition algorithms.....	67
7.1	Non-Linear Logistic Regression (NLR)	67
7.2	Regularized Least Squares (RLS).....	69
7.3	Artificial Neural Network (ANN)	70
7.4	Support Vector Machine (SVM)	70
7.5	Linear Discriminant Analysis (LDA)	71
8.	Algorithms optimization.....	73
III.	Results	77
9.	Results with MyoBock sensors	79
9.1	Optimization of the number of MyoBock electrodes	79

9.2 Optimization of classifiers complexity	84
9.2.1 NLR: minimization of the D-value.....	85
9.2.2 ANN: minimization of the number of layers and neurons	85
9.3 Comparative analysis between algorithms	87
9.4 Results on amputees whit MyoBock electrodes	90
10. Results with IIT sensors	93
10.1 Optimization of the number of electrodes	93
10.2 Optimization of classifiers complexity	98
10.2.1 NLR: minimization of the D-value.....	99
10.2.2 ANN: minimization of the number of layers and neurons	100
10.3 Comparative analysis between algorithms.....	101
10.4 Results on amputees whit IIT electrodes	104
IV. Discussion	107
V. Conclusion.....	111
VI. Bibliography.....	115

List of Figures

Figure 1.1: Wooden toe prosthetic, Egypt 1000-600 B.C.....	21
Figure 1.2: Levels of upper limb amputation [13].....	22
Figure 2.1: Organization of the skeletal muscle [15].....	24
Figure 2.2: Sarcomere structure [15]	25
Figure 2.3: Sliding filaments behavior in the contraction cycle	26
Figure 3.1: Some of the available upper limb prostheses	29
Figure 3.2: Multi-articulating prosthetic hand: i-Limb Ultra.....	30
Figure 3.3: Bebionic hand - Ottobock.....	30
Figure 4.1: Block diagram of a generic pattern recognition system [18]	34
Figure 5.1: History of <i>Hannes</i> hand (DDPSTUDIO Architecture & Design)	39
Figure 5.2: Transmission mechanism based on DAG system [5].....	40
Figure 5.3: <i>Hannes</i> system with electronic parts	42
Figure 5.4: Electric circuit of the instrumental amplifier	43
Figure 5.5: MyoBock electrode 13E200	44
Figure 5.6: IIT sEMG electrodes	44
Figure 5.7: Nested feedback control system	47
Figure 5.8: Sliding window mechanism	48
Figure 5.9: Example of conversion logic	49
Figure 5.10: GUI for choosing the DoF	50
Figure 5.11: Training panel and generation of the classifier model	51
Figure 5.12: Gesture measurement (G) with the addition of resting state (R)	52
Figure 5.13: Combination of four classes and the common resting state	52
Figure 5.14: One resting state acquisition.....	53
Figure 5.15: ' N ' different resting state acquisitions.....	53
Figure 5.16: Resting measurements distribution for 2-DoF systems.....	54
Figure 5.17: Resting measurements distribution for 3-DoF systems.....	54
Figure 5.18: 'N' random resting clusters from 4 resting measurements.....	55

Figure 5.19: Training measurements for a single gesture.....	56
Figure 5.20: Block diagram for the model generation.....	58
Figure 5.21: Block diagram for the classification phase.....	58
Figure 5.22: Block diagram for the classification phase.....	59
Figure 5.23: Uploading model interface	60
Figure 5.24: <i>Hannes</i> system in Blender environment.....	61
Figure 5.25: <i>Hannes</i> system in Unity environment	62
Figure 6.1: Electrodes placement over the corresponding muscles on the right arm	63
Figure 6.2: <i>Hannes</i> gestures.....	64
Figure 6.3: Setup pf the experiment.....	65
Figure 8.1: Scaling technique applied to EMG signals.....	73
Figure 9.1: F ₁ -score of the NLR as the number of Ottobock electrodes varies	80
Figure 9.2: F ₁ -score of the RLS as the number of Ottobock electrodes varies.....	81
Figure 9.3: F ₁ -score of the ANN as the number of Ottobock electrodes varies	82
Figure 9.4: F ₁ -score of the SVM as the number of Ottobock electrodes varies	83
Figure 9.5: F ₁ -score of the LDA as the number of Ottobock electrodes varies.....	84
Figure 9.6: Effect of the variation of polynomial degree on the EOF index	85
Figure 9.7: Effect of the variation in the number of layers on the EOF index	86
Figure 9.8: Effect of the variation in the number of neurons on the EOF index	87
Figure 9.9: Algorithms comparison in terms of F ₁ -score index.....	88
Figure 9.10: Algorithms comparison in terms of EOF index	89
Figure 9.11: Algorithms comparison in terms of classification index.....	89
Figure 9.12: Algorithms comparison in terms of abstention index	90
Figure 10.1: F ₁ -score of the NLR as the number of IIT electrodes varies.....	94
Figure 10.2: F ₁ -score of the RLS as the number of IIT electrodes varies	95
Figure 10.3: F ₁ -score of the ANN as the number of IIT electrodes varies	96
Figure 10.4: F ₁ -score of the SVM as the number of IIT electrodes varies	97
Figure 10.5: F ₁ -score of the LDA as the number of IIT electrodes varies.....	98
Figure 10.6: Effect of the variation of polynomial degree on the EOF index	99
Figure 10.7: Effect of the variation in the number of layers on the EOF index	100
Figure 10.8: Effect of the variation in the number of neurons on the EOF index	101

Figure 10.9: Algorithms comparison in terms of F ₁ -score index.....	102
Figure 10.10: Algorithms comparison in terms of EOF index	103
Figure 10.11: Algorithms comparison in terms of classification index.....	103
Figure 10.12: Algorithms comparison in terms of abstention index	104

List of Tables

Table 5.1: Number of samples and acquisitions for each gesture in 3-DoF	55
Table 5.2: Splitting of the dataset	57
Table 7.1: Encoding the variable D for NLR algorithm	68
Table 8.1: Confusion matrix	74
Table 9.1: F_1 -score of the NLR as the number of Ottobock electrodes varies	80
Table 9.2: F_1 -score of the RLS as the number of Ottobock electrodes varies	80
Table 9.3: F_1 -score of the ANN as the number of Ottobock electrodes varies.....	81
Table 9.4: F_1 -score of the SVM as the number of Ottobock electrodes varies.....	82
Table 9.5: F_1 -score of the LDA as the number of Ottobock electrodes varies	83
Table 9.6: Optimized number of MyoBock electrodes for different classifiers	84
Table 9.7: Effect of the variation of polynomial degree on the EOF index.....	85
Table 9.8: Effect of the variation in the number of layers on the EOF index.....	86
Table 9.9: Effect of the variation in the number of neurons on the EOF index.....	87
Table 9.10: Classifiers performance scores obtained with MyoBock sensors.....	88
Table 9.11: Classifiers performance scores obtained with MyoBock sensors on amputees	91
Table 10.1: F_1 -score of the NLR as the number of IIT electrodes varies	94
Table 10.2: F_1 -score of the RLS as the number of IIT electrodes varies	94
Table 10.3: F_1 -score of the ANN as the number of IIT electrodes varies	95
Table 10.4: F_1 -score of the SVM as the number of IIT electrodes varies	96
Table 10.5: F_1 -score of the LDA as the number of IIT electrodes varies	97
Table 10.6: Optimized number of IIT electrodes for different classifiers	98
Table 10.7: Effect of the variation of polynomial degree on the EOF index.....	99
Table 10.8: Effect of the variation in the number of layers on the EOF index.....	100
Table 10.9: Effect of the variation in the number of neurons on the EOF index.....	101
Table 10.10: Classifiers performance scores obtained with IIT sensors	102
Table 10.11: Classifiers performance scores obtained with IIT sensors on amputees	105

Abbreviations

ACh:	Acetylcholine	NS:	Not-Significant difference
ADLs:	Activities of Daily Living	PID:	Proportional Integral Derivative
ATP:	Adenosine Triphosphate	PR:	Precision
ANN:	Artificial Neural Network	PWM:	Pulse Width Modulation
BMS:	Battery Management System	RE:	Recall
DAG:	Dynamic Adaptive Grasp	RLS:	Regularized Least Squares
DoF:	Degree/Degrees of Freedom	RMS:	Root Mean Square
E-DATS:	EMG-Data Acquisition & Training Software	SCMM:	Scheda Controllo Motore Mano
EMG:	Electromyography	sEMG:	surface Electromyography
EOF:	Embedding Optimization Factor	SSC:	Slope Sign Change
HC:	Hand Closing	STD:	Standard Deviation
HO:	Hand Opening	SVM:	Support Vector Machine
IIT:	Istituto Italiano di Tecnologia	TCP:	Transmission Control Protocol
INAIL:	Istituto Nazionale per l'Assicurazione contro gli Infortuni sul Lavoro	TH:	Threshold
IP:	Internet Protocol	TR:	Training Set
LDA:	Linear Discriminant Analysis	TS:	Thresholds-optimization Set
MAV:	Mean Absolute Value	VR:	Virtual Reality
MUAP:	Motor Unit Action Potential	VS:	Validation Set
nFN:	number of False Negative	WE:	Wrist Extension
nFP:	number of False Positive	WF:	Wrist Flexion
NLR:	Non-Linear Logistic Regression	WL:	Waveform Length
nN:	number of true Negative	WP:	Wrist Pronation
nP:	number of true Positive	WS:	Wrist Supination

Abstract

Background

Nowadays the most widespread control strategy for the upper limb prosthesis relies on two muscles contraction which leads the amputees to an unnatural feeling of control. Due to low usage intuitiveness, impaired people must learn a new way of thinking to make the prosthesis perform the desired movements [1]. Since the clinically available myoelectric devices provide limited ability in the control of multiple joints at the same time, the movements of the prosthesis appear to be clumsy [1]. Therefore, the control is narrowed to the single DoF and the switching from one joint movement to another is usually implemented through simultaneous contraction of the flexor and extensor muscles of the wrist (co-contraction) [2]. Since the vast majority of amputees can still feel the lost limb in lifelike details, the basic idea is to exploit the residual muscular activity, related to the movements of the phantom limb, to control the prosthesis in more spontaneous manner [3]. Accordingly, recent studies on machine learning algorithms, based on pattern recognition, inspired the research work presented in this thesis. These studies show that prosthetic device with 2-DoF can be controlled through machine learning algorithms with high scores of accuracy, however, they currently fail when considering a higher number of DoF. Nevertheless, this work will contribute to demonstrate the opportunity of making more natural the control of *Hannes* system (with 3-DoF) through pattern recognition by exploiting several electromyographic signals (EMG) of the residual limb [4].

Materials and Methods

The thesis is carried out at the Rehab Technologies Laboratory of Istituto Italiano di Tecnologia (IIT) in Genova in collaboration with Centro Protesi INAIL of Vigorso di Budrio. The thesis aims to develop and test the pattern recognition algorithms for simultaneously controlling multiple-joints of the *Hannes* prosthesis by using up to six EMG sensors fastened at the forearm level. *Hannes* is the poly-articulated hand, developed by IIT and INAIL, able to restore over 90% of lost functionality in people with transradial amputation [5]. The three DoF implemented in the *Hannes* prosthesis are the opening and closing of the hand, and the pronation, supination, flexion and extension of the wrist. Five different supervised machine learning algorithms, used for prosthesis controllability, are compared: Artificial Neural Network (ANN), Non-linear Logistic Regression (NLR), Regularized Least-Squares (RLS), Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA), which is considered as the ground truth. Ten healthy subjects and four amputees are involved in the study in which six-EMG sensors are exploited to collect the electrical activity of the forearm muscles. The dataset acquired during the training phase is then used to build mathematical models of different algorithms. Afterward, the comparative analysis between the five machine learning algorithms is based on both performances and computational burden. The statistical analysis is evaluated through the Wilcoxon-Signed-Rank test (p -value lower than 0.05) with Bonferroni correction for multiple comparisons. Once the classifier is built, the subject is able to freely perform any combination of the aforementioned gestures to control *Hannes* in both real environment and virtual reality (VR). The optimization process aims at minimizing both the number of electrodes

as well as the complexity of the algorithms, on the basis of two indexes: the F_1 -score and the Embedding Optimization Factor (EOF), respectively.

Results

The comparative analysis, carried out on NLR, ANN, SVM, RLS and LDA, highlighted the predominance of NLR and LDA in terms of F_1 -score, EOF and classification indexes. Small differences appear between the results obtained with MyoBock sensors and those obtained through IIT ones. The general behavior of the system is almost the same, in fact, the NLR and LDA are the most suitable classifiers among the five studied. The NLR has obtained higher percentage of abstention than LDA and RLS, but this alleged drawback is counterbalanced by the high classification frequency. Although the analysis is performed on data from able-bodied subjects, the comparison has demonstrated that the algorithm with the best compromise is NLR, as it has no significant differences in performance with the LDA (ground truth).

Conclusions

As remarked by results obtained with both MyoBock and IIT electrodes, the NLR algorithm is the best choice since it guarantees high accuracy and, at the same time, low complexity. Even if the statistical comparison is carried out on healthy people, the thesis provides useful insights into the suitability of control strategies for upper limb prosthesis used by amputees. Although the control is still not smooth and fast, during the experimental tests, the amputees were astonished by this innovative method of translating human intentions into prosthesis movements.

Keywords

Hannes Hand, Upper Limb Prosthesis, Embedded system, Supervised Machine Learning, Non-Linear Logistic Regression, Pattern Recognition, Linear Discriminant Analysis, Support Vector Machine, Artificial Neural Network, Regularized Least-Square

Introduction

Hands and arms allow to operate the world around us: our houses, our offices, the means of everyday transportation [6]. Not only hands allow us to interact with the physical world, but they also help in greeting and talking. The loss of an upper limb incredibly lowers the quality of life, leading to severe impairment in daily-living operational functionality as well as to psychological damages. Since the amputation is a definitive impairment, from the beginning of civilization, humans started searching for improvements in amputees life. Initially, the adopted solutions were based on replacing the part of the body missing by means of mechanical and passive devices. Along with the technology evolution and electronic improvements, prostheses became mechatronic systems controlled by means of human intentions. The remaining neurons inside the stump are still able to carry information through little electrical signals that travel from the brain to the residual limb muscles in a fraction of a second. When the signal reaches the stump, it is withdrawn by artificial electrodes and it is used to move the active prosthesis. Nevertheless, the electromyographic (EMG) signals conversion into prosthesis movement is still a challenging problem. Since the first myoelectric prosthesis, created in the period 1944-1948 by Reinhold Reiter [7], several progresses have been made. The initial idea was to identify two loci of maximum independent muscular activity on the stump and using the two sEMG signals to operate the motor of one-degree-of-activation hand prosthesis. The antagonistic pair of muscles is used to control two movements independently: hand opening/closing (HO/HC), wrist pronation/supination (WP/WS), wrist flexion/extension (WF/WE), elbow flexion/extension, etc. The switching between configurations is usually implemented through muscular co-contraction [2]. Although the not natural control fashion, the antagonistic muscles are still used in upper limb prosthesis control. The researchers are studying new techniques based on pattern recognition and machine learning algorithms to improve the control strategies and making the devices more reliable and naturally activated [4].

The thesis aims to develop and test the pattern recognition algorithm to concurrently control a 3-DoF active prosthesis by means of six sEMG signals withdrawn at the forearm level. The available prosthesis movements are hand opening/closing (HO/HC), wrist pronation/supination (WP/WS), wrist flexion/extension (WF/WE) and they are simultaneously detected and activated. By exploiting the sequence of events included in the raw EMG data, the pattern recognition algorithms extract and classify useful features for the prosthesis control.

The master thesis is conducted at the Rehab-Technologies Laboratory of Istituto Italiano di Tecnologia (IIT) of Genova in collaboration with Centro Protesi INAIL of Vigorso di Budrio (BO).

The thesis is organized as follows:

- The part I summarizes the state of the art of upper arm prosthesis and it outlines the most widespread control strategies. There is also an additional chapter related to the muscle contraction, from the origin of the nervous signal to the desired movement;

- The part II defines the materials and methods used for the experimental study. A broad description of the *Hannes* system is included along with a discussion of the aforementioned machine learning algorithms;
- The part III reports the results of the experimental tests on able-bodied subjects and amputees, by considering different types of classifiers. This part is split into two chapters, the first highlights the results obtained with MyoBock electrodes, while the second one is related to the results with the IIT sensors;
- The part IV discusses the study results and the optimization techniques implemented to have high performances and, at the same time, low computational burden;
- The part V outlines the conclusions on the research work of this thesis and it introduces possible future developments.

I. Background

1. A Brief History

The loss of a limb is a severe impairment which could lead to feeling phantom pain in the missing part of the body and also permanent psychological suffering [8]. For multiple centuries, humans have tried to overcome the problem of amputations by replacing the lost limb with a static cosmetic prosthesis [9]. The prosthetic device does not only provide renewed functionality and aesthetic pleasure but it also serves to reestablish the sense of wholeness. Therefore, the earliest example of prosthesis was a wooden toe belonging to a noblewoman (Figure 1.1) found in an Egyptian gravesite. It is not weird that the first prosthetic device was something that might seem meaningless. However, the big toe was allegedly necessary for an Egyptian due to its crucial importance in wearing traditional Egyptian sandals. Although fashioning a new kind of shoes could have been much easier and cheaper, wearing the traditional thong sandals of Egypt was fundamental for the noblewoman. This underlines how amputees are considerably focused on the cosmetic and aesthetic aspects of prostheses.



Figure 1.1: Wooden toe prosthetic, Egypt 1000-600 B.C.

Over the years, the amputees required the addition of enhanced functionalities in order to exploit the prosthesis for performing different tasks. The mechanical part has been improved and the addition of harnesses system allowed the external control of the prosthesis through other parts of the body (residual limb, shoulder, chest, ...). In the USA an estimate of 100,000 people per year undergo an upper-limb loss, about half of which use mechanical prosthetics based on cable-driven systems [10]. The body-powered prosthesis had a huge impact on amputees' lives cause it played an important role in their independence restoration. Although the body-powered prostheses do not permit the recovery of full functionalities of the missing limb, they are still widely used due to their affordability mainly since they do not rely on a limitless power source [11]. A further feature of the body-powered prosthesis is the modularity which allows patients to wear both hands or hooks according to their needs. Another non-trivial aspect is that body-powered prostheses can last longer than the myoelectric ones and they are also easier to be controlled. Owing to the burgeoning culture of technology the mechanical development of the prosthesis was supported by the advances in the electronics

field. “It is an interesting coincidence that the results of the first experiments with myoelectric control were published in 1948, the same year in which the development of the transistor was announced” [12]. Hereafter the passive artificial upper limb was joined by the myoelectric controlled prosthesis. The exploitation of the myoelectric arm is strictly dependent on the remaining muscle activity of the stump, therefore it is relevant to carry out a prior analysis for attaining an ad hoc solution for the considered patient. A well-suited prosthetic design is figured out on the basis of different aspects such as the level of amputation, conditions of the residual limb and also patient’s professional goals. “The levels of upper limb loss can be classified as transcarpal, wrist disarticulation, transradial, elbow disarticulation, transhumeral, shoulder disarticulation and forequarter” [13] (Figure 1.2).

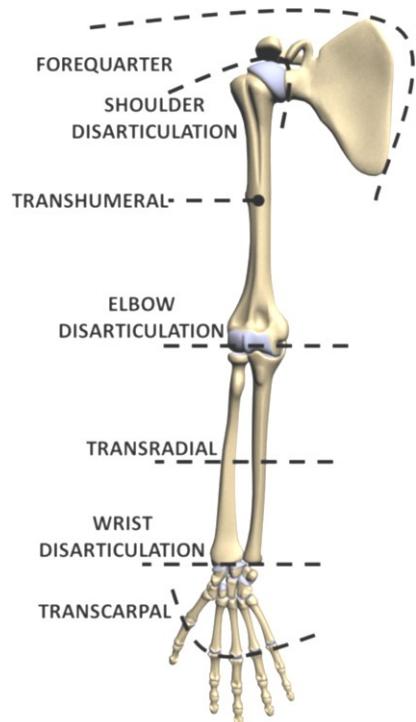


Figure 1.2: Levels of upper limb amputation [13]

The electrical activity generated at the muscle level is firstly gathered by sensors, then it is processed and used to regulate the current flow from the battery to the electric-motors. The main problems were the large size of electronic components such as a battery (held in an external bag) but also the outer wirings that were exposed to recurrent damages. Compared with body-powered prostheses the electric ones require higher maintenance as battery charge and plastic gloves replacement. The advent of microelectronics had a huge impact on several disciplines as automotive, aerospace and also biomedical field. The enhancements in microelectronics led to slimmer prosthetic devices therefore it establishes an important spread of myoelectric artificial upper limb.

The following chapter describes the process of muscle contraction, from the brain intention to the final body movement. In an amputee, despite the missing portion of a limb, the nerve impulses are still available and they are responsible for the prosthesis control.

2. Muscle activation

The muscular system is responsible for the movement of the human body, the trunk, the limbs, the respiratory system. Not only muscles act to produce movement and force, but they also stop the body movement, such as resisting gravity to maintain posture [14]. The three main types of muscle are: the skeletal muscle (voluntary muscle which is consciously controlled), the smooth muscle (involuntary muscle) and the cardiac muscle. The skeletal muscle, by shortening their length, serves to move bones closer to each other. Muscles usually work in opposite pairs: one muscle contracts, whereas the other relaxes to produce body movements. Since amputees voluntary control the prostheses, the input signals are withdrawn from the skeletal muscles. All contractions of the voluntary muscle (except for reflexes) occur as a result of conscious effort originating into the brain. The brain sends chemical and electrical signals (action potentials) through the nervous system to the motor neuron which innervates several muscle fibers.

2.1 Types of muscle fibers

Inside the skeletal muscle there are three different types of fibers which may be distinguished on a metabolic, mechanical and histological basis. The muscle fiber could be:

- Slow oxidative (type 1): they are characterized by slow twitch and a low myosin ATPase activity. These cells have important blood supply (“red fibers”) hence they are specialized for endurance activity and they are highly resistant to fatigue.
- Fast oxidative-glycolytic (type 2A): they have a fast twitch and a high myosin ATPase activity. These fibers can exploit aerobic energy sources based on oxygen supplying as well as anaerobic energy sources. The type 2A fibers are known also as “white fibers” due to their low reliance on oxygen. They are progressively recruited when additional effort is required and they are less resistant to fatigue than slow twitch fibers.
- Fast glycolytic (type 2B): they show forceful fast contraction and a high myosin ATPase activity. These fibers are only recruited for brief maximal efforts and they quickly tire.

2.2 Skeletal muscle

The skeletal muscle is composed by several integrated tissues which are muscle fibers, nerve fibers, blood vessels and connective tissue. The skeletal muscle is enclosed by three different layers of connective tissue. The epimysium is the most external one which allows the muscle to contract intensely while keeping its structural integrity. The second layer of connective tissue (perimysium) divides the muscle into bundles which are called fascicles. The fascicular organization allows the nervous system to trigger specific movement of a muscle by activating a subset fibers within a bundle. The blood vessels are intended for nourishment, waste removal and oxygen delivery to the muscle. The third connective tissue (endomysium) of collagen includes nutrients and extracellular fluid. Each muscle fiber is supplied by the axon of motor neuron for handling fibers contractions through electrical signals originated into the brain.

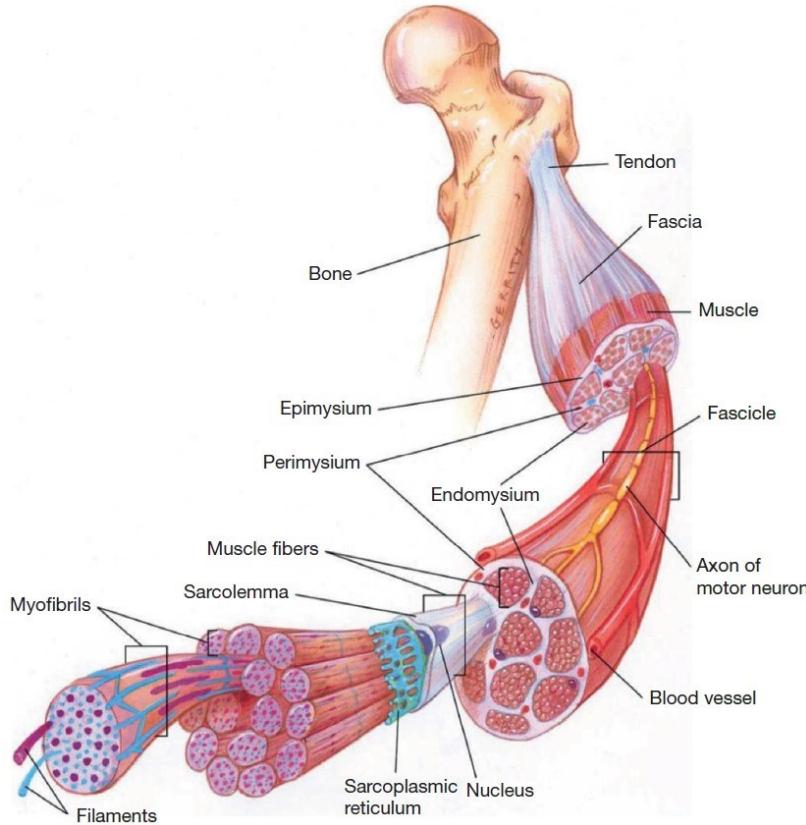


Figure 2.1: Organization of the skeletal muscle [15]

The muscle fibers are composed by long cylindrical multinucleated cells generated from the chain-fusion of embryonic cells, called myoblasts. The muscle fibers is wrapped by a plasma membrane which is called sarcolemma with a cytoplasm referred to as sarcoplasm. The sarcoplasmic reticulum acts as a conductor, storage and retrieval of calcium ions (Ca^{2+}) which are essential for the muscle contraction. The muscle fiber is made up by many myofibrils which are bundles of thin filaments (actin) and thick filaments (myosin). The actin is a family of globular proteins which form a helical-structure of long microfilaments. It contains myosin binding sites for creating cross-bridges with myosin heads. The myosin is a protein that converts chemical energy in the form of ATP into mechanical energy, therefore it acts as force and movement generation. The alternating thick and thin filaments gives myofibrils the striped appearance. The functional unit of the skeletal muscle is the sarcomere, an organized arrangement of contractile myofilaments and support proteins. Down the length of the longitudinal myofibril many sarcomeres run adjacent to one another. Each sarcomere is bordered by two Z-discs to which the actin filaments are attached while the myosin filaments are anchored to the center of the sarcomere, called M-line. Since the actin filaments are tied to the Z-lines the sarcomere shortens from both sides during the sliding motion. The H-zone is defined as the non-overlapping part between actin and myosin filaments.

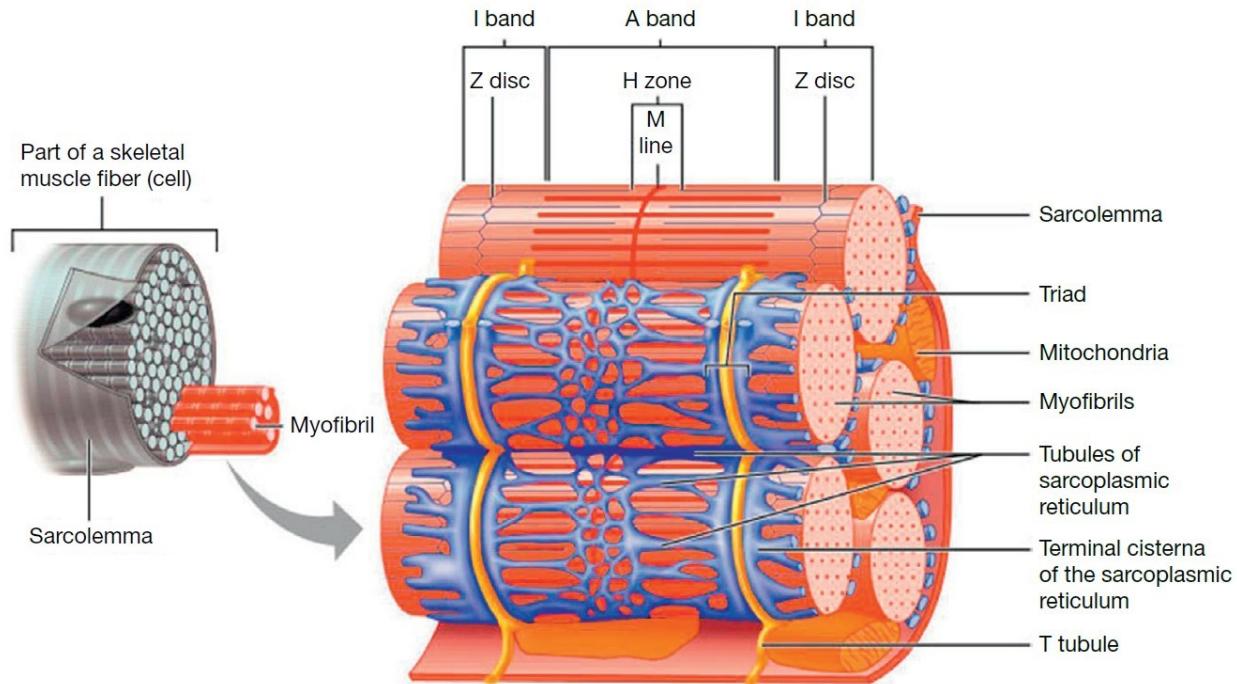


Figure 2.2: Sarcomere structure [15]

The sliding process is triggered by two regulatory proteins called troponin and tropomyosin. When the skeletal muscle is relaxed the thin fibrous protein (tropomyosin) blocks the cross-bridge binding sites of the actin filaments. In the presence of ATP and the calcium ions (Ca^{2+}) level is high enough, the Ca^{2+} bind to the troponin structure. This process first displaces the tropomyosin and then it exposes the myosin binding sites on actin, allowing the myosin to form cross-bridges.

2.3 Mechanism of muscle contraction

The muscle contraction is determined by the simultaneous shortening of the sarcomeres in the activated muscle fibers. This phenomenon is caused by the intensification of actin-myosin overlapping in each sarcomere. The increase of the overlapping is realized through a successive forming and breaking of cross-bridges between thin and thick filaments. The sliding process is triggered by the release of calcium ions (Ca^{2+}) which enhances the interaction of myosin and actin at the molecular level. The hydrolysis of ATP into adenosine diphosphate (ADP) and inorganic phosphate gives the needed energy for the muscular contraction. The reaction is catalyzed by the enzyme myosin's ATPase in the presence of ions Ca^{2+} and ions Mg^{2+} :



The energy obtained from the hydrolysis of ATP changes the angle of myosin head and it leads to the pulling movement, known as power stroke. Afterward, in the presence of other molecules of ATP, the myosin head will detach from the actin filament. Therefore a sufficient amount of ATP is important for completing muscular contraction. The

ATP generation happens through three different mechanisms: aerobic respiration, anaerobic glycolysis and phosphocreatine metabolism.

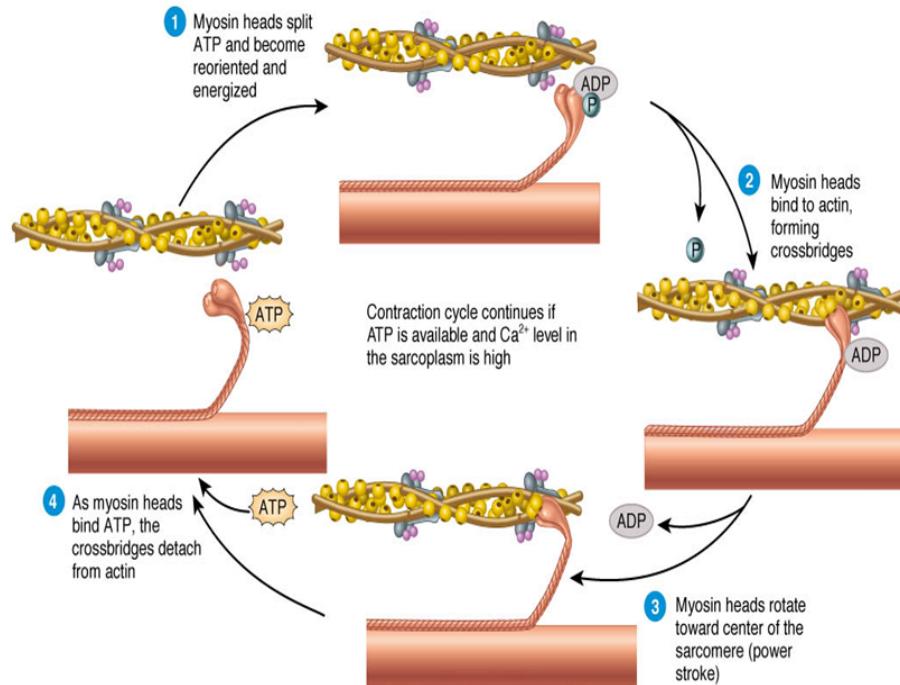


Figure 2.3: Sliding filaments behavior in the contraction cycle

2.4 Neuromuscular junction and biochemical changes

As previously mentioned, the calcium ions Ca^{2+} are stored in the sarcoplasmic reticulum and they are crucial for the activation of the muscle contraction since they displace the myosin binding sites on the actin filaments. The release mechanism of Ca^{2+} is triggered by the nervous system through the neuron's terminal, which is called neuromuscular junction. The combination of motor neuron and muscle fiber defines the motor unit. Each motor neuron innervates several muscle fibers, therefore, fewer fibers are innervated by one motor neuron more precise are the movements. The electrical signal originated into the brain travels along the axon of the motor neuron as an action potential. Once it reaches the presynaptic neuromuscular junction, the acetylcholine (ACh) neurotransmitter is released through small membrane-bound spheres called synaptic vesicles. The diffusion of the ACh within the synaptic cleft determines the activation of ACh-receptors located on sarcolemma of the muscular fiber. In resting conditions the potential difference across the plasma membrane is between $-60mV$ and $-90mV$ (Na^+ , Cl^- : outside, K^+ inside) and it is said to be polarized. The ACh-receptors act as channel for ions which are now allowed to move across the plasma membrane (Na^+ : outside \rightarrow inside, K^+ : inside \rightarrow outside), causing the sarcolemma depolarization. As the depolarization process occurs, further ions channels, which are called voltage-gated sodium channels, are triggered to open. At this point the action potential conducts rapidly along the sarcolemma in a wavelike fashion away from the neuromuscular junction.

The Motor Unit Action Potential (MUAP) is the sum of all signals generated at muscular fibers level when a depolarization occurs within a single motor unit. The sEMG sensor gathers all these signals therefore the non-invasive electromyography is less accurate than the invasive one. However, the invasive electromyography is able to withdraw a less noisy signal in the neighborhood of a specific muscular fiber through a conductive needle. In general, the non-invasive approach is preferred and the amputee is able to control the prosthetic device on the basis of sEMG signals.

The resting potential across the plasma membrane is restored by the sodium-potassium pumps which transport 3 Na^+ outside and 2 K^+ inside. The action potential propagates along the plasma membrane and, thanks to the periodic invaginations called T-tubules, it reaches the sarcoplasmic reticulum where it stimulates the release of calcium ions Ca^{2+} .

An overview of the currently available types of prosthesis is depicted in the following chapter. Whereas the paragraph 5.1.1 defines the structure of the EMG sensors and how they work.

3. State of the Art

As previously mentioned, the upper arm prosthesis has undergone considerable improvements over the years. The wide market of prosthetic devices allows amputees to choose their best solution accordingly to aesthetic and functional requests. The development of prostheses is based on two principal engineering challenges, on one hand, embedding the electronic components into a volume comparable to the replaced portion of the limb and, on the other hand, attaining an intuitive control. All the devices represented in Figure 3.1 are made by Ottobock, which, together with Ossür, is one of the leading players in the market of wearable human bionics. On the left in Figure 3.1 is reported the cosmetic passive prosthesis used by patients just for physically replacing the missing portion of the limb. The active hook, represented in the middle in Figure 3.1, is the MyoHand VariPlus Speed®, which is able to grasp objects with high gripping force and quickness. Although the lack of anatomical nature of hands, this device is widely used thanks to its high grasping force.



Figure 3.1: Some of the available upper limb prostheses

Over the years the research studies have developed advanced poly-articulated prosthesis characterized by both aesthetics and functionality as the Ottobock's Michelangelo (Figure 3.1 on the right) or the latest myoelectric hand i-Limb Ultra designed by Touch Bionics, now acquired by Ossür (Figure 3.2). The Michelangelo hand is suited for naturally operating the tools and technology of everyday life such as PCs, phones, etc. Oppositely, the i-Limb Ultra is an example of a multi-articulating prosthetic hand with five individually powered fingers. It has several functionalities i.e. locking the index finger for use in such activities as one-fingered computer typing or using ATM.



Figure 3.2: Multi-articulating prosthetic hand: i-Limb Ultra

A further prosthetic device is the Bebionic hand designed by RSL Steeper and acquired by Ottobock in these last years (Figure 3.3). It is a poly-articulated prosthesis with individual motors for each finger, allowing to manipulate objects of different shapes, textures and sizes together with an enhanced level of dexterity. The 14 different grips and hand positions are useful for carrying out everyday activities like holding credit cards, pointing, writing, typing and more. Although the previous models use similar technology for opening or closing the fingers, its range of movements is restricted.



Figure 3.3: Bebionic hand - Ottobock

The aim of the next chapter is to describe the currently available control strategies for the myoelectric prosthesis.

4. Control Strategies

As soon as the electronic field has achieved the goal of actuating myoelectric prostheses, researchers and manufacturers have faced the problem of control. The exploitation of dual-site (agonist-antagonist) muscles was the first idea for the prosthesis control and it is still the most widespread strategy available on the market. The aim was to identify two loci on the stump in order to recognize two opposite movements like hand opening-closing, wrist pronation/supination, etc. Although myoelectric devices use muscular signals for controlling the movements, the patient is required to devote a demanding mental effort. The main problem with the myoelectric standard control is the unnatural way of thinking when patients want move the device. They should not consider anymore the motion of their phantom limb but they must learn a new way of thinking [1]. The control of several degrees of motion is implemented by using only the extensor carpi radialis longus and the flexor carpi radialis (muscles with the highest muscular activity at the forearm level). With the sequential control the patients can handle only one DoF at a time and if they want switch from one joint to another they can exploit the muscular co-contraction or, if available, a button on top of the prosthesis [2]. In this way, amputees can change the specific movement under control but the motion of a joint must be preceded by its selection through the co-contraction. It is well evident how this control technique is far from the natural situation due to the time delay between the brain activation and the force generation. The standard control has experienced issues with the increase of available DoF in the prosthetic device owing to the difficulty in simultaneous joints control. The control approach with implantable myoelectric sensors ([16]) has the same characteristic of the sEMG control but it results more reliable since the signals, gathered through the needles, have a low noise corruption. The limited diffusion of this method is caused by the important health problems that may arise like infection, bruising or hematoma.

Another interesting control strategy is the Targeted Muscle Reinnervation (TMR), which is based on a surgical nerve-transfer procedure. Nerves that usually transmit signals to the natural arm are surgically connected to other muscles. Even after the amputation, nerves remain active and they are still able to carry information from the brain to the missing limb. “The innervated muscles then serve as biological amplifiers of the amputated nerve motor signals. By transferring multiple nerves, TMR myoelectric signals allow intuitive, simultaneous control of multiple joints in an advanced prosthesis” [17]. After the reinnervation the nerve grows inside the new target muscle and, consequently, it can generate the contraction. The amputee thinks to move the phantom limb but another muscle contracts (i.e. unused musculature in the residual limb or chest muscles). At this point the EMG signal is gathered by sEMG sensors and it is used for prosthesis movements. This technique is aimed at improving the control of the upper limb prosthesis because the neural control signals intended for the missing limb are used to control analogous functions in the prosthesis. Furthermore, the TMR technique generates additional control sites in such a way that patients do not have to use the same muscle for several prosthetic functions.

Since the vast majority of amputees can still feel the lost limb in lifelike details, the basic idea is to exploit the residual muscular activity, related to the movements of the phantom limb, to control the prosthetic device in a more natural manner. The researchers are studying an innovative control strategy and the most followed approach is based on

machine learning and pattern recognition [4]. The addition of more information, by gathering more signals from the forearm muscles, helps to make the control strategy more intuitive than the dual-site one. The application of pattern recognition algorithm requires the increase in the number of electrodes, which is ranging form 6 to 10 for this type of analysis. In the advanced control strategies, the group of sEMG patterns is mapped to a specific movement. Therefore, the pattern recognition classifier links a specific set of muscular signals to the single movement. The main idea behind this approach is to study these patterns and build a model capable to understand, based on a likelihood function, whether the reproduced signals are similar to the ones gathered during the training phase. The training phase plays an important role since the model accuracy is strictly related to the input data used to train the algorithm. It is crucial to provide useful signals with distinct patterns for each desired movements in order to let the classifier distinguish between different gestures. Although the standard approach (Data Abundance [1]) generally states that “the more samples are considered the more accurate is the classifier”, too detailed dataset could lead to a long training phase. It is important to choose a suitable trade-off between the time needed for training the classifier and its accuracy.

Myoelectric control systems based on pattern recognition techniques (Figure 4.1) rely on supervised machine learning classification algorithms [1]. The supervised learning exploits the labeled data, which is a dataset that has been classified, to infer learning algorithm. The dataset is used as the basis for predicting the classification of other unlabeled data through the use of machine learning algorithms. “Linear classifiers, such as Linear Discriminant Analysis (LDA), Logistic Regression (LR) or Support Vector Machine (SVM) with linear kernel, and nonlinear classifiers, such as Nonlinear Logistic regression (NLR) and SVM with nonlinear kernel, represent the state-of-the-art about pattern recognition classifiers” [1]. The main difference between linear and nonlinear classifiers is the shape of the decision boundary, in the former the boundary is a straight line or a plane, while in the latter the boundary is a curved line or a surface. Each part of the block diagram (Figure 4.1) is analyzed in order to achieve a final optimized solution. Since the system should run on an embedded system, the best choice of the algorithm should rely upon both performance and computational burden. The initial training phase (dotted path in the block diagram) is performed once, during which the classifier learns how to link the set of EMG patterns to a specific gesture. Afterward, on the basis of the new recorded EMG patterns (input signals), the trained classifier is used to distinguish between the available movements (output classes). The second block, called features extraction, has been introduced for enhancing the classification performances. The features are the most important components of the myoelectric signals, such as Mean Absolute Value, Mean Absolute Value Slope, Zero Crossing, Slope Sign Change, Waveform Length, etc. [1]. The EMG signal is analyzed by considering different time intervals of the same length, called segments, in order to reduce the computational burden and make the system response faster.

Some features that can be extracted from the myoelectric signal are:

- *Mean Absolute Value:*

$$M = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (1)$$

- *Mean Absolute Value Slope*: is the difference between mean absolute value of adjacent segments.

$$\Delta M = M_{k+1} - M_k \quad (2)$$

- *Root Mean Square*: is defined as the square root of the arithmetic mean of the squares of the values.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2} \quad (3)$$

- *Zero Crossing*: can be obtained by counting the number of times the waveform crosses the zero. A threshold must be included in the calculation to reduce the noise induced zero crossings.

$$x_i > 0 \text{ and } x_{i+1} < 0 \quad \text{or} \quad x_i < 0 \text{ and } x_{i+1} > 0 \quad (4)$$

- *Slope Sign Change*: represents the number of times the slope of the EMG signal changes sign. A suitable threshold, leading to high performances, should be chosen after a trial and error procedure.

$$SSC = \frac{1}{N} \sum_{i=1}^{N-1} f[(x_i - x_{i-1}) \cdot (x_i - x_{i+1})] \quad (5)$$

$$f(x) = \begin{cases} 1, & x \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases}$$

- *Waveform Length*: is a measure of the EMG signal complexity and it represents the cumulative length of the EMG signal waveform.

$$WL = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (6)$$

- *Mean*: is the central value of a discrete set of data.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (7)$$

- *Variance*: is defined as the average of the squared deviations about the mean for a set of values.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (8)$$

- *Standard Deviation*: is the measure of the amount of dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

$$\sigma = \sqrt{\sigma^2} \quad (9)$$

Once chosen the output class, the motor actuation can be modulated in position or velocity by considering the RMS

value of the EMG signals in order to make the control strategy even more natural.

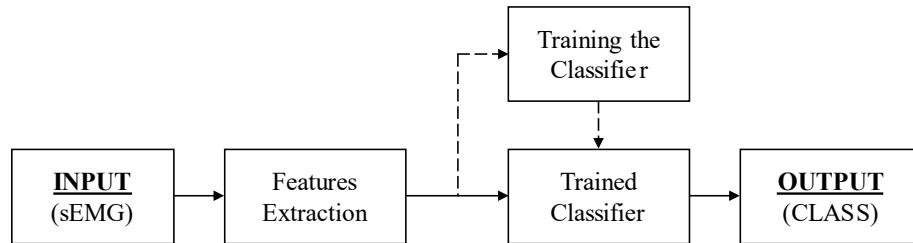


Figure 4.1: Block diagram of a generic pattern recognition system [18]

In this thesis, the pattern recognition algorithms are applied to the *Hannes* prosthesis, which includes 3 DoF: hand grasping, wrist pronation-supination and wrist flexion-extension. The comparative analysis among LDA, NLR, SVM, RLS and artificial neural network is carried out on sEMG data from 10 healthy people and four transradial amputees.

II. Materials and Methods

The whole materials and methods, exploited for the development of the study, are analyzed in the part II of this thesis. The focus is on the description of the carried out activities and the employed hardware. In compliance with any commercial and property secret, a detailed description of the *Hannes* system is made. It is important to state the difference between *Hannes* hand and the whole *Hannes* system. The former is the prosthetic device which can perform fingers movements, while the latter includes either the *Hannes* hand and the two DoF of the wrist (WPS – WFE). The upper limb prosthesis includes two separate DoF in the wrist, which are operated by two different electric motors, and the *Hannes* hand, which can only perform the grasping gesture. The *Hannes* hand is a prosthetic device realized by the cooperation between the Rehab Technologies department of IIT and Centro Protesi INAIL of Vigorso di Budrio.

In order to allow a better reading and understanding the part II is divided into several paragraphs. Chapter 5 describes the *Hannes* prosthesis concerning the mechanical design, the firmware and the software. The study protocol, exploited for the patient trials, is outlined in chapter 6. All the operations which precede the data collection are described in order to allow the study replication. In chapter 7 the machine learning algorithms used for the study are described. The comparison between the algorithms is based on variation of the parameters (i.e. number of electrodes, internal parameters of each machine learning algorithm), in order to achieve an optimized solution with low complexity and high performances. Chapter 8 reports the performed process for optimizing the functionality of the pattern recognition system.

5. Hannes

The *Hannes* hand prosthesis was developed by means of a three years joint collaboration (January 2014 – December 2017) between IIT and INAIL. It is a poly-articulated hand able to restore over 90% of lost functionality in people with transradial amputation. One of the principal aspects of innovations is the achievement of naturalness of forms, movements and orientation of the rotation axes and hand posture. *Hannes* incorporates key biometric properties that make this prosthesis uniquely similar to a human hand [5]. The study of the anthropometry of the real hand has guided the prosthetic design, from aesthetic to structural features, allowing the user to perceive the prosthesis not as an external tool but rather as an integral part of the body. An iterative design process has been developed to integrate the mechanical and electrical parts inside the forms and proportions of the hand. The crucial aspect of the *Hannes* hand is the differential mechanism at the base of the Dynamic Adaptive Grasp (DAG system, Figure 5.2), which gives the peculiar ability of prosthesis adaptation to the object's form and to any kind of external stresses.

The background in Figure 5.1 shows the preceding prototypes of *Hannes* while the device at the forefront is the latest model characterized by a considerable anthropometric shape. From left to right in the background of Figure 5.1 is possible to notice how the physical encumbrance decreases for making the prosthesis as similar as possible to the real hand.



Figure 5.1: History of *Hannes* hand (DDPSTUDIO Architecture & Design)

The prosthetic hand, called *Hannes*, incorporates high levels of biomimicry through the concurrence of anthropomorphism, performance, and functionality, which lead to better performance compared with other existing research and commercial prosthetic devices [5]. As represented in Figure 5.3, Hannes systems consists of three main

interacting physical components:

- *Hannes* hand;
- an active flexion-extension (F/E) wrist module and an active pronation-supination (P/S) wrist module;
- a socket that houses six surface electromyographic sensors, battery pack, and control electronics.

The DC motor (Faulhaber CR 2642 coupled with epicyclic planetary reducer 19:1) actuates the leader wire, thus transmitting the force to the cable-based mechanism accommodated in the palm. The leader wire originates from the shaft of motor and ends in the thumb, passing through two differential elements mounted onto linear guides that are each made of a custom-made bush bearing and two rails along which the bushing moves. Each bushing houses two idle pulleys: one pulley supports the leader wire, whereas the second pulley is used by the follower wire to actuate two adjacent fingers. There are two follower wires: the first wire moves the index and middle finger, while the second follower wire is used to actuate the ring and little finger [5]. The thumb can assume three different positions to reproduce different grasping movements: pinch, power and lateral configurations. The thumb is manually displaced in one of the above mentioned configuration. The lateral position is exploited to handle thin objects like credit cards, the power position is used for grabbing and carrying heavy items, while the pinch attitude is suitable for gripping small objects with the thumb and forefinger.

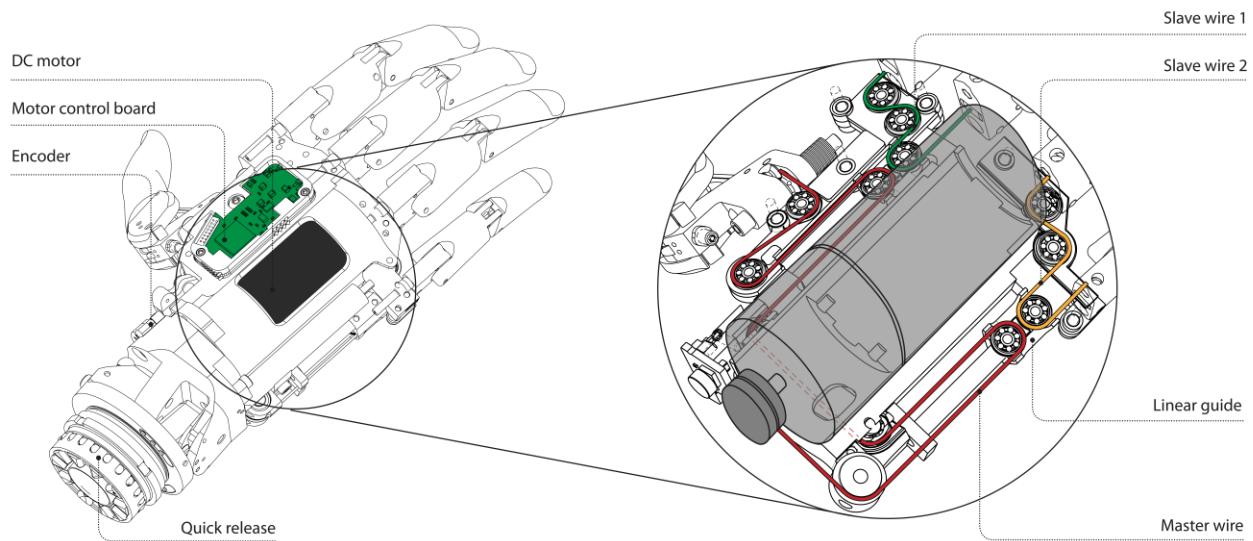


Figure 5.2: Transmission mechanism based on DAG system [5]

The socket is customized on the single patient since each stump has different size and shape. It contains a watertight environment in which all the electronic components are housed. Furthermore, on the socket surface there is a magnetic plug connector which is used for battery charging when the prosthesis is not in use. The quick release system (Figure 5.2) is implemented to allow an easy connection of the *Hannes* hand to the socket. The electric motor aimed at the movement in pronation-supination is located in the socket near the connection interface.

Hannes hand is available in two different sizes (small and large) to realize a prosthetic device with anthropometric dimensions as similar as those of the amputee [19]. For both sizes is possible to manufacture an aesthetic silicon glove

with special details like wrinkles, hair or veins. In *Hannes* hand all the fingers DoF are implemented with the exception of the distal interphalangeal joint. The thumb is characterized by an active joint responsible for abduction/adduction of the single segment and a passive hinge joint for the rotation. Whereas, the other four fingers have two actuated joints related to the flexion/extension movements, which are the proximal interphalangeal joint and the metacarpophalangeal joint.

5.1 Hardware

The interaction between the three main components of *Hannes* system is implemented through CAN bus and Slip Rings. As represented in Figure 5.3, the six sEMG sensors send information through wires-connection to the EMG-Master. This electronic board contains the firmware (described in paragraph 5.2) responsible for translating the electromyographic signals into the class related to the movement that the amputee wants to perform. The DC motor housed in the palm is controlled by the SCMM board (Figure 5.3) which receives information from the encoder sensors to know the actual position of the motor shaft. The EMG-Master provides the synthetized position reference to the SCMM board, which, subsequently, generates the needed PWM to drive the electric motor. The encoder provides the feedback information (θ_{out} in Figure 5.7) used by the PID controller to avoid reaching shaft positions that are outside from the physiological range (i.e. fingers cannot have unconstrained motion). The Battery Management System (BMS) is a dedicated electronic board that aims at monitoring battery state and ensuring safety operations. The rechargeable battery pack is able to supply the entire *Hannes* system with a voltage of 12V through two different wires (VSYS and GROUND). The three lithium prismatic cells, used inside the battery, are in 3s1p configuration. The interface between EMG-Master and SCMM is established by 4 wires: CAN-H, CAN-L, VSYS and GROUND. Since the *Hannes* hand can perform rotations of 360° in pronation/supination, a sliding connector (Slip Ring) is used to guarantee a continuous transmission of electric signals from the EMG-Master to the *Hannes* hand. The SCMPS (Scheda Controllo Motore in Prono/Supinazione) is the electronic board that controls the wrist in pronation/supination and it is not equipped with any encoder. In this case, no feedback information is needed, since the electric motor for the wrist in pronation/supination is designed for moving at the maximum velocity when activated. Conversely, the SCMFE (Scheda Controllo Motore in Flesso/Estensione) electronic board designed to control the wrist in flexion/extension is equipped with the encoder sensor to provide feedback to the motor shaft position time by time. The EMG-Master includes a TiVa-microcontroller (Texas Instruments) used for data processing, an Electrically Erasable Programmable Read-Only Memory (E²PROM of 32MB) which is a non-volatile memory for storing relevant data and a Flash Memory of 256MB. The Bluetooth® module is also embedded on board to send data to the E-DATS software (paragraph 5.3.1) during the data acquisitions and to set control parameters through a dedicated GUI (Graphical User Interface) developed in QT software. The EMG-Master board hosts 6 plug-ins for connecting the sEMG sensors and also a plug-in for the electric motor of the wrist in pronation/supination.

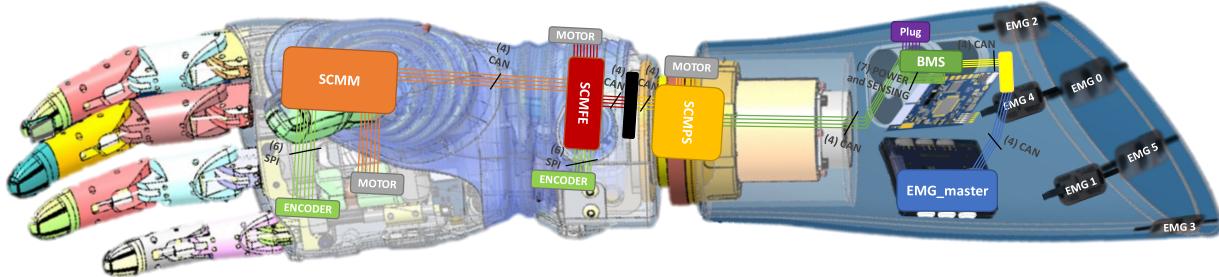


Figure 5.3: *Hannes* system with electronic parts

The nested feedback control system implemented on the SCMM board is shown in Figure 5.7. Once the SCMM receives the reference position of the electric motor (θ_{ref}), the difference between this value and the encoder measurement (θ_{out}) is computed to supply the first PID controller. The second PID controller exploits the tracking error (ε_i) between the reference current (i_{ref}) and the measured motor current (i_{out}) to have a more robust control system. The current tracking error (ε_i) is converted into a PWM signal used by the driver to actuate the electric motor. The SCMM board has a fault management system that blocks the entire process when unintended failures occur.

The EMG-Master is the core board of the whole *Hannes* system and it is intended for multiple functionalities. From the QT software is possible to set the board configuration through Bluetooth® connection. Several settings can be handled through this software. Firstly, the control strategy must be chosen among the standard dual-site or the Pattern Recognition one, since the *Hannes* system can be controlled by both. In addition, the active joints should be outlined on the basis of the available DoF. Furthermore, it is set the control method through which the joint is intended to move. The available methods are the one-speed control, which actuates the motor always with the same rotational speed, and the proportional control, which relates the motor speed to the RMS value of the EMG signals. The firmware available on the EMG-Master board is responsible for the generation of the position reference (θ_{ref} in Figure 5.7) as widely described in the paragraph 5.2.

5.1.1 Electrodes

The electromyography is the technique for studying the activation of the skeletal muscles through the recording of electrical potentials produced by muscle contraction. As already mentioned in chapter 2, the action potentials, which are generated at the neuromuscular junction, are the useful signals detected by the sensors. “The theory behind the surface EMG electrodes is that they form a chemical equilibrium between the detecting surface and the skin of the body through electrolytic conduction, so that the current can flow into the electrode” [20]. Due to its non-invasiveness, the surface electrodes are preferred over intramuscular ones at the expense of higher noise level. The surface myoelectric activity has a very small peak-to-peak amplitude (from some μV to few mV) and a limited bandwidth (15Hz – 400Hz) [21]. As a matter of fact, very sensitive instrumentation is required for the detection and amplification of such small signals. The EMG signal is acquired through differential amplification technique. The amplification stage should have a high input impedance and low output impedance, hence the instrumental

amplifier is used. The first stage (red dotted box in Figure 5.4), which is composed by two voltage-follower amplifiers, is used to isolate the electric circuit from the external environment. The isolation is carried out through a high input impedance since the measurement aims to quantify the current I that flows inside the sensing resistance R_S (patient body) without asking any current as input of the instrumental amplifier. Therefore the most commonly used sEMG sensor has a bipolar configuration with two detecting surfaces and, between them, a reference electrode (Figure 5.5 and Figure 5.6). The signals coming from the two detecting surfaces are the input of the instrumental amplifier and the common noise signals are cancelled.

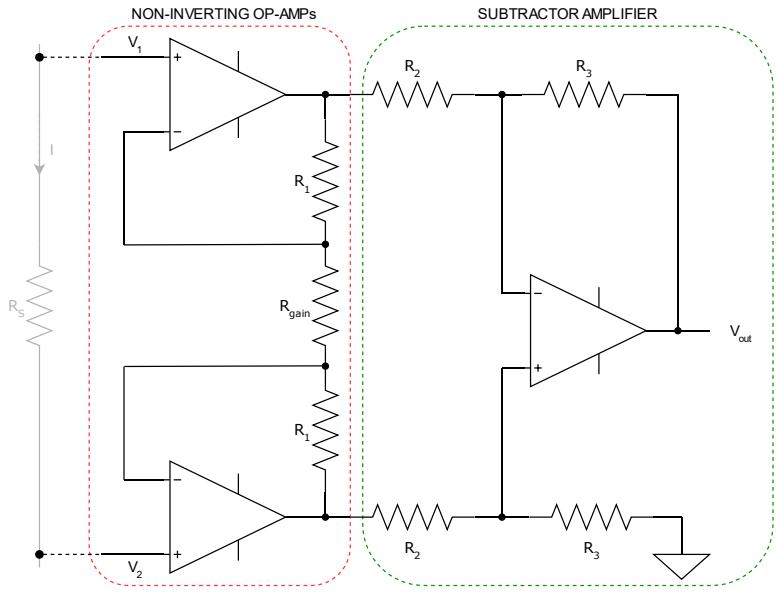


Figure 5.4: Electric circuit of the instrumental amplifier

The gain of the instrumental amplifier circuit is:

$$A_v = \frac{V_{out}}{V_2 - V_1} = \left(1 + \frac{2 \cdot R_1}{R_{gain}}\right) \cdot \frac{R_3}{R_2} \quad (10)$$

The potentiometer of the electromyographic electrodes is varied in position to increase/decrease the amplification of the muscular signals. By rotating the potentiometer knob, the amount of resistance (R_{gain} in Figure 5.4) can be increased or decreased thereby controlling the amount of current. The variation of the resistance through the potentiometer aims at changing the gain of the instrumental amplifier circuit, as described in Eq.(10). The paragraphs 5.1.1.1 and 5.1.1.2 outline the main characteristics of two different electrodes: the MyoBock sensor and the IIT sensor. Both of them are exploited in the thesis framework to establish how the control algorithm behaves when one or the other is used.

5.1.1.1 Ottobock sensors

The MyoBock electrode (Figure 5.5) 13E200 = 50 is made by Ottobock and it implements the bipolar configuration, where the middle plate serves as a voltage reference, whilst the two smaller plates act as detecting surfaces of the action potentials. The sensor dimensions are $27 \times 18 \times 9.5 \text{ mm}$ and its weight is 4.5 g . The MyoBock electrodes are

directly embedded in the socket of the prosthesis to allow a faster dressing. They ensure small sensitivity to high and low frequency interference allowing uninterrupted prosthesis usability. The inter-electrode distance , which is about $20mm$, is strictly related to the arm volume from which the signals are detected. The higher is the inter-electrode distance, the higher is the acquisition body volume and the more are the withdrawn signals. The MyoBock electrode has an electronic circuit responsible for rising the signal quality. Firstly the amplification stage is implemented, then, on one hand, the high-pass filter is applied for canceling the motion artifacts or temperature fluctuations, on the other hand, the low-pass filter is intended for removing high frequency interference from radio broadcasts, phones, computers, etc. The MyoBock electrode has a frequency bandwidth of $90 - 450 Hz$, a common mode rejection ratio (CMRR) higher than $100dB$ and an operating voltage of $4.8 - 7.2 V$.



Figure 5.5: MyoBock electrode 13E200

5.1.1.2 IIT sensors

The Rehab Technologies department of IIT has developed its own electrodes (Figure 5.6), which are based on bipolar configuration. The dimensions ($18mm$ of diameter) are slightly smaller than the MyoBock ones, with the purpose of reducing the encumbrance in the prosthesis socket. The size reduction of the electrode has a huge impact also on the mechanical aspect of the prosthesis. Since the Pattern Recognition control requires the exploitation of at least 6 sEMG electrodes, the smaller the holes in the prosthesis socket, the lower the mechanical stress. The IIT sensor has a reduced inter-electrode distance (approximately $12mm$) at the expense of information detection. The lower is the inter-electrode distance, the lower is the arm volume from which the signals are detected. In the same way as MyoBock electrode, the IIT sensor has an electronic circuit responsible for rising the signal quality. Firstly the amplification stage is implemented, then, on one hand, the high-pass filter is applied for canceling the motion artifacts or temperature fluctuations, on the other hand, the low-pass filter is intended for removing high frequency interference from radio broadcasts, phones, computers, etc. The IIT sensor has a frequency bandwidth of $90 - 450 Hz$, a common mode rejection ratio (CMRR) higher than $100dB$ and an operating voltage of $4.8 - 7.2 V$.

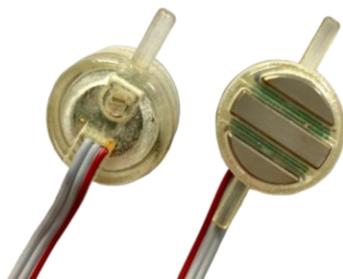


Figure 5.6: IIT sEMG electrodes

5.2 Firmware

Code Composer Studio (CCS) is an integrated development environment (IDE) that comprises a suite of tools exploited to develop and debug embedded applications, as the one used for the *Hannes* system. The firmware, which is written in C language and burned onto the EMG-Master (described in paragraph 5.1), has been realized by Rehab Technologies Lab of IIT and Centro Protesi INAIL of Vigoroso di Budrio. The main purpose of this thesis was to adjust the algorithm to identify 7 different gestures starting from the sEMG signals as input. Before the thesis work, the algorithm was able to separately predict only 4 gestures, which were HO, HC, WP and WS, since the active wrist in flexion/extension was not physically implemented on the device.

The matching between the prosthesis movement and the numerical value of each class is:

- Abstention: $ABS = 0$;
- Hand Closing: $HC = 2^0 = 1$;
- Hand Opening: $HO = 2^1 = 2$;
- Wrist Pronation: $WP = 2^2 = 4$;
- Wrist Supination: $WS = 2^3 = 8$;
- Wrist Flexion: $WF = 2^4 = 16$;
- Wrist Extension: $WE = 2^5 = 32$;

A first adaptation of the firmware allowed the machine learning algorithm to individually identify each class and separately actuate each electric motor. To make the control algorithm even more natural, the further step consisted in adding the recognition of multiple classes at the same time, as long as the classes belong to different joints (it is not possible to have simultaneous movements like WP & WS). Therefore the final configuration of the firmware can handle the combination of several movements (i.e. $HO + WE = 2^1 + 2^5 = 2 + 32 = 34$). The changes made to the firmware, for handling the combination of multiple classes at the same time, are reported and briefly described:

```
int oneVSall( float *hyp, int nClasses){
    int tmpClass = 0;
    int i;
    for( i = 0; i < nClasses; i += 2 ){
        if( hyp[i] > 0 & hyp[i+1] > 0 ){
            return tmpClass = 0;
        }
        else if( hyp[i] > 0 ){
            tmpClass = tmpClass + ( 1 << i );
        }
        else if( hyp[i+1] > 0 ){
            tmpClass = tmpClass + ( 1 << (i+1) );
        }
    }
    return tmpClass;
}
```

The function “*oneVSall*” takes as input the class membership probability vector (*hyp*) and the number of active classes, which depends on the prosthesis configuration (*nClasses*). Before using the “*oneVSall*” function, the *hyp* value is

computed through the logistic function [Eq. (11)] and the result is compared with a predefined threshold. If the class membership probability overcomes that threshold, the value is saved in the i-th position of the *hyp*-vector, on the contrary, if the threshold is not exceeded, the i-th *hyp*-value is multiplied by -1 . In this situation, the *hyp*-vector contains both positive and negative values. The “*oneVSall*” function is intended to convert the *hyp*-vector into a number (*tmpClass*), which corresponds to the predicted active classes. The *hyp*-vector is a row vector where the entries are all coupled to each other and each pair corresponds to a single DoF (row of the *VotingMatrix* in Figure 5.9). As a consequence, the for-loop starts its operations one every two values of the *hyp*-vector. At this point three different situations may occur:

- $\text{hyp}[i]>0 \& \text{hyp}[i+1]>0$: two classes, which belong to the same DoF, exceed the threshold. This is clearly a misclassification, since a single joint cannot perform opposite movements at the same time. As a matter of fact, the output value of the *oneVSall* function is equal to zero (*tmpClass* = 0);
- $\text{hyp}[i]>0$: if only the first class of a certain joint exceeds the threshold, the class is correctly predicted and the output value of the *oneVSall* function is updated (*tmpClass* = *tmpClass* + ($1<<i$));
- $\text{hyp}[i+1]>0$: if only the second class of a certain joint exceeds the threshold, the class is correctly predicted and the output value of the *oneVSall* function is updated (*tmpClass* = *tmpClass* + ($1<<(i+1)$)).

This function is intended to examine the class membership probabilities of each DoF, such that if several classes are contemporarily present, the algorithm can handle both and the prosthesis can perform the combination of the movements.

Since the algorithm gives 300 outputs per second, the references for the electric motors are synthetized over a sliding window of *nV* values. Every time that the algorithm outlines a class (*VotingArray*), the correspondent counter (*VotingResult*) is increased, as reported in the following lines of code:

```

for( i = 0; i < nV; i ++ ){
    if( VotingArray[i] == 0 ){
        VotingResult[VotingArray[i]]++;
    }
    else{
        for( k = 0; k < nClasses; k ++ ){
            if( VotingArray[i] & ( 1 << k ) ){
                VotingResult[k+1]++;
            }
        }
    }
}
if( VotingResult[0] < nV*percentage ) outClass = 0;
else{
    for(i = 1; i < nClasses; i ++ ){
        if( VotingResult[i] > nV*percentage ){
            outClass = outClass + ( 1 << (i-1) );
        }
    }
}

```

The *VotingArray* is a row vector corresponding to the binary representation of the output value of the *oneVSall*

function (*tmpClass*). The for-loop is implemented to build the vector of *VotingResult* which contains a counter for each class. When the *i*-th value of the *VotingArray* is different from zero, the corresponding counter in the *VotingResults* is increased. A graphic representation of the conversion logic is shown in Figure 5.9. Once the whole sliding window is completed, the last part of the code aims to verify if the number of identified classes exceeds a percentage of the total number of values available in the sliding window (*nV*). In the case of *Hannes* control the *percentage* value is often chosen as 70%, but it can be set through Bluetooth® software (GUI – Graphical User Interface).

The firmware is responsible for both identifying the output class and synthesizing the reference (PWM signal) of the electric motors. It is important to underline that the reference type is not the same for all the available DoF: the HO/HC and the WF/WE reference is in position, whilst the WP/WS reference is in velocity. As a matter of fact, the HO/HC and the WF/WE motion can be controlled proportionally to the RMS value of the sEMG signals, conversely, the WP/WS movement is a one-speed control (electric motor rotates always at the maximum velocity: 15 rpm).

The control system is based on a Low Threshold (*LTh*), which can be set offline based on the individual patient. The reference of the electric motors is generated if only if the RMS value [Eq. (3)] of the input sEMG signals overcomes the *LTh*; this technique is implemented for increasing the robustness of the control system in such a way that some artifacts can be discarded. The angular displacement is proportional to the RMS value of the sEMG signals. The higher the RMS value, the larger the variation.

A further feedback control is implemented for the current of the electric motor in order to both increase the control robustness and avoid possible faults. As a consequence, the hand and wrist in F/E are nested feedback control system (Figure 5.7), where encoders measure the joint position and the motor current time by time, whereas, the wrist in P/S is a feedforward control system.

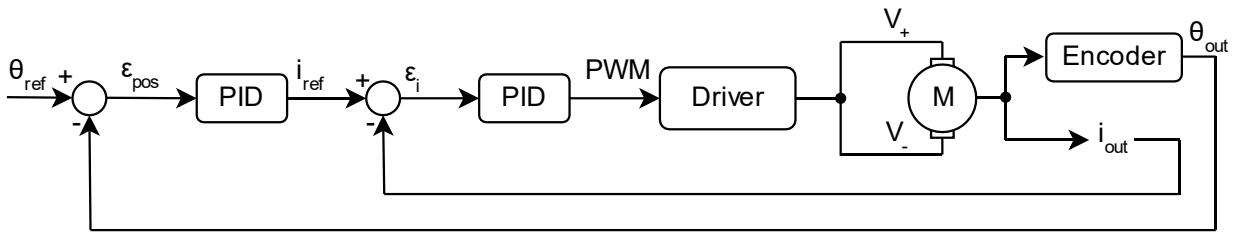


Figure 5.7: Nested feedback control system

There is not a direct match between each class output of the classifier and the reference of the electric motor, but the latter is synthesized on the basis of the most frequent output among a *sliding window* of *nV* values. The class output frequency of the entire system is 300 Hz, hence for each second 300 classes are outlined. As a result, the final outcome is different from the abstention-case if and only if the desired output has been detected for at least a certain percentage of the times. A good choice for this threshold is 70% but it can be re-designed according to the specific requirements of the final application. It is important to underline that in this configuration, multiple classes can be simultaneously detected and activated, thus, the output could be a combination of several classes. The decision window is described

as “sliding” because the final output class (prosthesis movement) is evaluated on outcome values from time t_0 till t_{nV} , whereas the following output class will be defined on values from time t_1 till t_{nV+1} . The entire nV -block of the already evaluated outputs is not discarded but only the first class event will be erased for considering a new one at time t_{nV+1} (as represented in Figure 5.8).

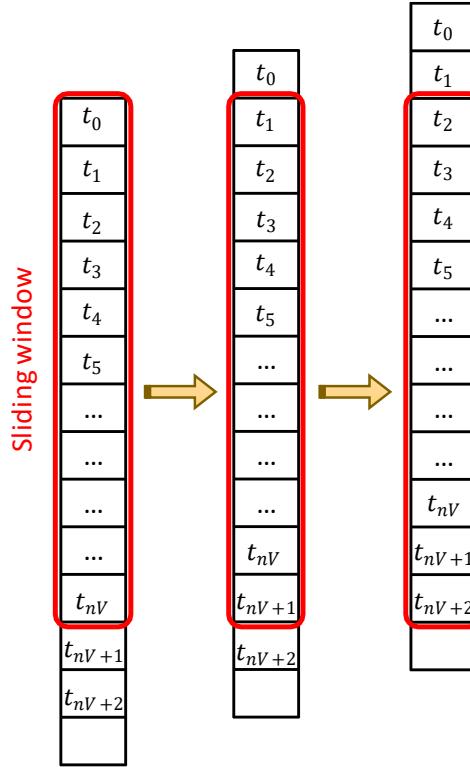


Figure 5.8: Sliding window mechanism

As previously mentioned, the firmware can handle the combination of several classes. As represented in Figure 5.9, each output of the classifier model is converted into a binary array of dimensions $\mathbb{R}^{1 \times nClasses}$ (row of the *Voting Matrix*). The bit inside this row corresponds to a specific class (*WE* \rightarrow *bit5*, *WF* \rightarrow *bit4*, ...). The single class has its corresponding binary representation ($HO = 2^1 \rightarrow [000010]$), whilst the classes combination is given by the addition of the integer numbers ($WS + HO = 2^3 + 2^1 = 10 \rightarrow [001010]$). The huge *Voting Matrix* is exploited for attaining the array, called *Voting Result*, which takes into account all the occurred class events. Whenever a class is present inside the row of the *Voting Matrix* the counter of the corresponding class inside the array *Voting Result* is increased. If class 0 is outlined, the abstention counter is increased (*VotingResult[0]*). At the end of the sliding window, if the class events inside the *Voting Result* overcome a certain percentage threshold (usually 70% of the total nV -values), the prosthesis will perform the corresponding movement.

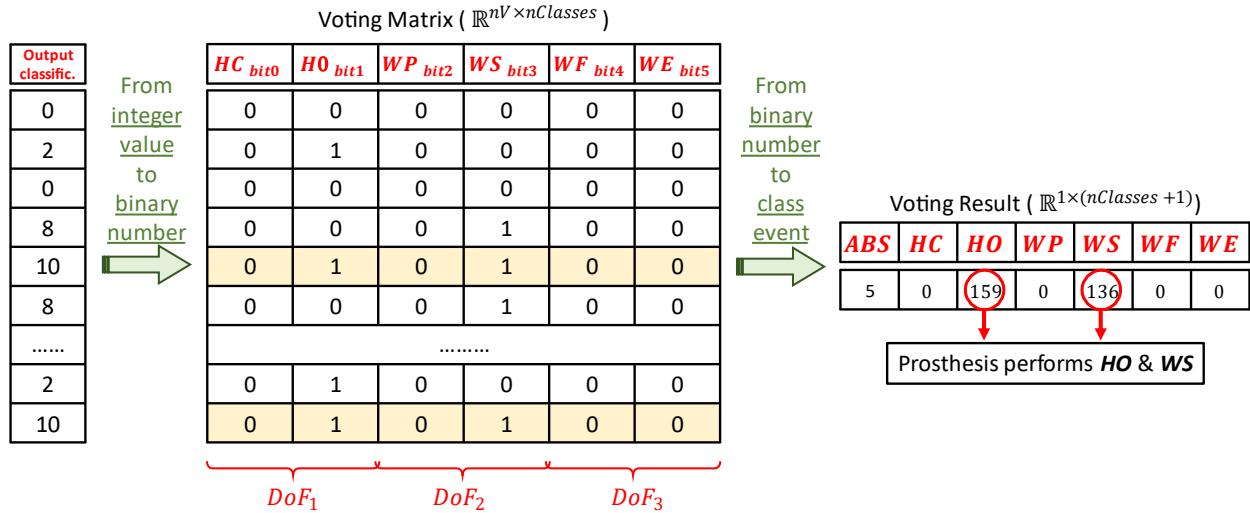


Figure 5.9: Example of conversion logic

5.3 Software

The EMG-Data Acquisition & Training Software has been realized at the Centro Protesi of Vigorso di Budrio and it is customized for the prosthetic devices: Prensilia and i-Limb. The Rehab Technologies department of IIT has developed a new version of the PC software E-DATS, which is designed for the *Hannes* prosthesis.

5.3.1 E-DATS (EMG – Data Acquisition & Training Software)

The E-DATS software is an application realized from the building of the LabVIEW project. The software serves multiple purposes and these can be divided into four blocks:

- general settings;
- sEMG data acquisition & classifier training phase;
- real-time simulation of the newly built classifier;
- uploading the model of the classifier & testing the embedded system through the VR.

5.3.1.1 General settings

During the initial settings the software can create and manage folders related to each participant in the experiment. Since amputees can choose the most suitable configuration of the prosthesis according to their needs, it is taken into account the possibility of having different combinations of DoF. The initial decision of DoF availability is made before the dataset acquisition inside the E-DATS software to withdraw only the sEMG signals related to the corresponding gestures. The available combinations are:

- WP-WS & HO-HC (4 classes);

- WF-WE & HO-HC (4 classes);
- WP-WS & WF-WE & HO-HC (6 classes).

The red box in Figure 5.10 shows the GUI, in particular the focus is on the user selection of the joints movements. Through these two buttons indeed, the user can select the desired active joints while the hand movement is active by default. Moreover, the sampling frequency and the width of the recording window can be defined from the *Initialization Menu*.



Figure 5.10: GUI for choosing the DoF

5.3.1.2 Data acquisition & Training phase

Once the general settings are defined, the acquisition phase can start and the EMG-Master is connected via Bluetooth® to the LabVIEW software. The wireless connection allows the transmission and the recording of sEMG data, which are collected from the patient skin through EMG sensors (Ottobock and IIT). The panel in Figure 5.11 shows the training phase when three DoF are available. The user is asked to reproduce each gesture (HO, HC, WS, ...) several times to create a suitable dataset to be exploited in MATLAB® for the generation of the classifier model.

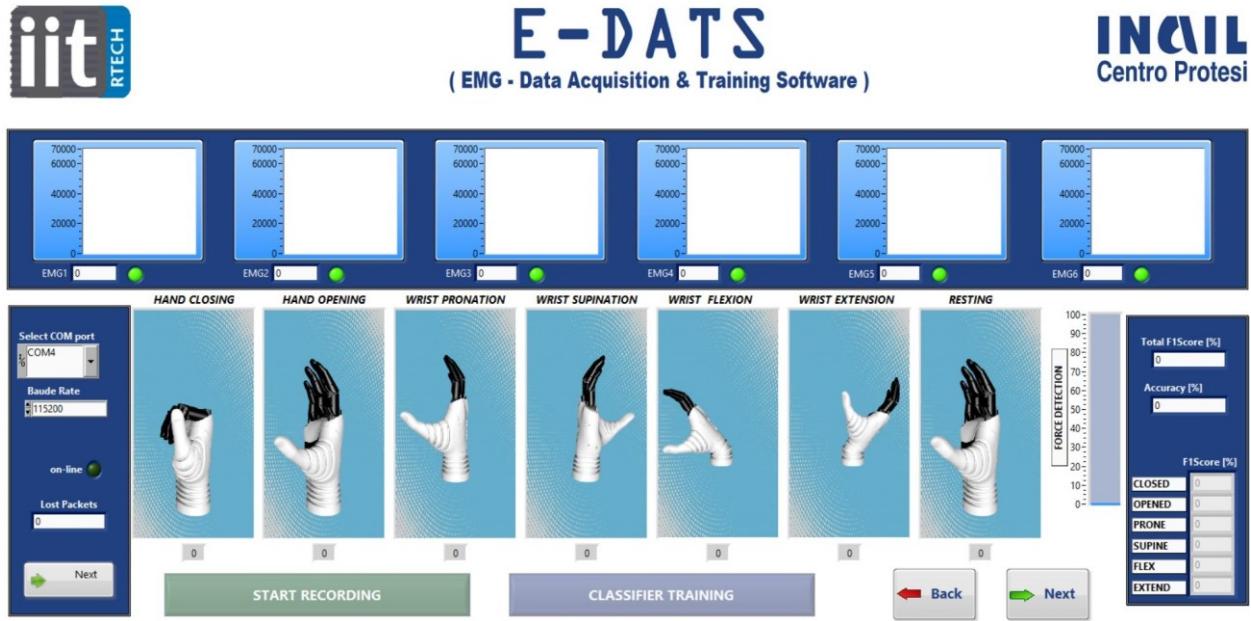


Figure 5.11: Training panel and generation of the classifier model

During the measurements, each movement is repeated 10 times except for the resting state, which is treated differently. The patient is asked to reproduce the gesture displayed on the screen and, during every movement, the sEMG signal is acquired for two seconds with a sampling frequency of 1kHz. For each recorded movement a column vector $G \in \mathbb{R}^{2000 \times 1}$ is finally generated.

The resting state is very important since it makes the prosthesis stop in the current posture without producing undesired movements. In the case of *abstention* the classifier does not give any output, thus the prosthesis does not perform any movement. Therefore, it is possible to exploit the *class isolation* technique by adding the resting state measurements inside each gesture class, creating an *unbound* region. In order to consider also the resting state, the resting measurement ($R \in \mathbb{R}^{2000 \times 1}$) is appended at the bottom of each column vector (G), as represented in Figure 5.12. In Figure 5.13 there is a graphical representation of the resting intersection zone which is common to all the possible gesture classes (HO, HC, WS, ...). The Figure 5.13 also shows the availability of different gestures combinations since the classifier is able to simultaneously detect two classes which are belonging to different DoF (WE+HO, WF+HC, ...).

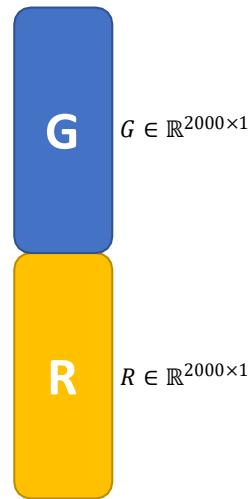


Figure 5.12: Gesture measurement (G) with the addition of resting state (R)

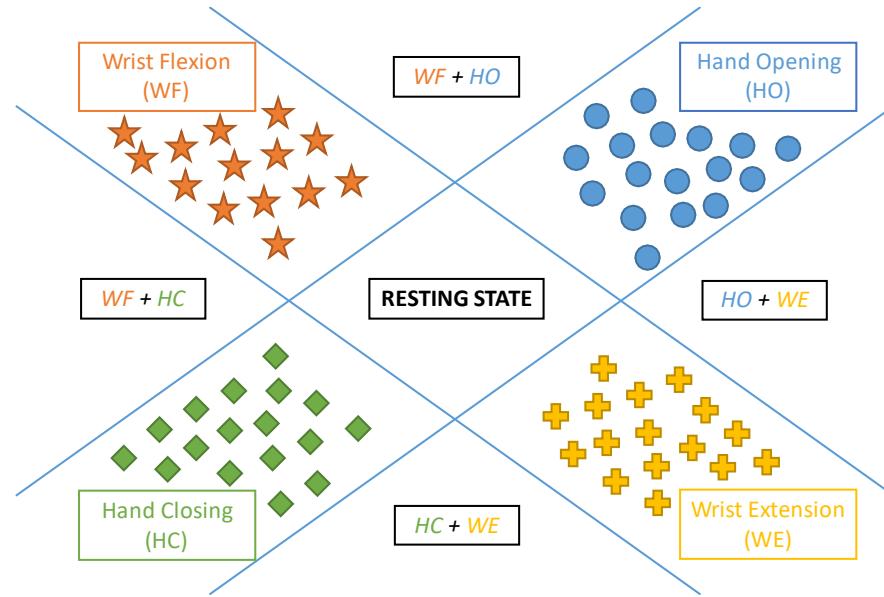


Figure 5.13: Combination of four classes and the common resting state

At this point an important discussion on the resting samples is needed. There are many ways to consider the resting measurements inside each gesture. Some techniques are described and the most suitable one will be taken into account.

- **ONE RESTING STATE ACQUISITION**

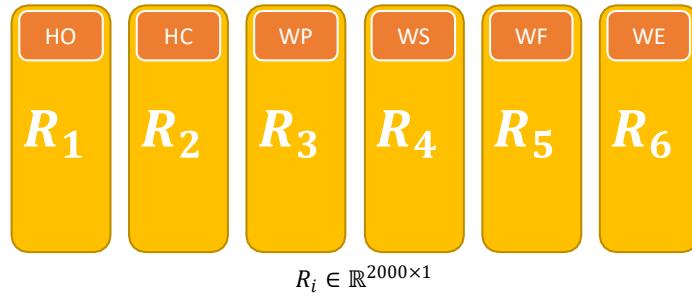
Since the indecision region should be the same for all the possible gestures, the resting state (R) is acquired once and it is appended at the bottom of the sEMG signal recorder for each gesture (G). It is important to underline that the resting samples are the same for all gesture measurements. This approach is faster with respect to all the other acquisitions due to the short time needed, but, on the other hand, it does not provide high variability to the system.



Figure 5.14: One resting state acquisition

- **'N' RESTING STATE ACQUISITIONS**

The second approach, conversely from the previously mentioned, is the most general one since it provides the acquisition of ' N ' resting measurements, where ' N ' corresponds to the number of active classes (4 or 6 in the case of this work). Each of the obtained measurements for the resting state (R_i where $i = 1, 2, \dots, N$) is appended to different gesture measurements, as shown in Figure 5.15. This approach is time-consuming due to the large number of required measurements. Nevertheless it provides the highest variability to the system.

Figure 5.15: ' N ' different resting state acquisitions

- **FEW (2 OR 4) RESTING STATE ACQUISITIONS**

This method is defined starting from the second one but it aims to reduce time for obtaining the desired measurements at the expense of variability. The number of measurements depends on the a-priori choice of available DoF. In the case of 2 DoF (HO/HC & WP/WS or HO/HC & WF/WE) is possible to acquire just 2 resting measurements and then attach them also to the other two gestures. As shown in Figure 5.16, the hand movements will have the same resting measurements as the ones recorded for the wrist. Alternatively, in the case of 3 DoF (HO/HC & WP/WS & WF/WE) is possible to acquire just 4 resting measurements and, afterward, attach 2 of them to the hand gestures, as shown in Figure 5.17.

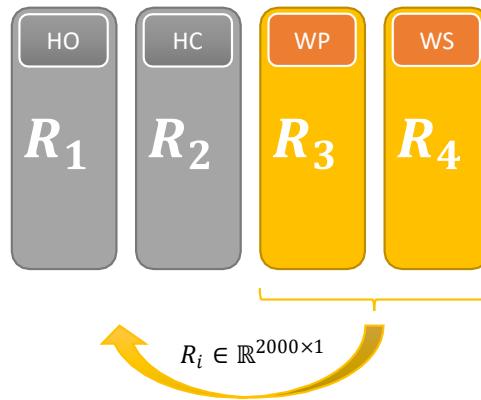


Figure 5.16: Resting measurements distribution for 2-DoF systems

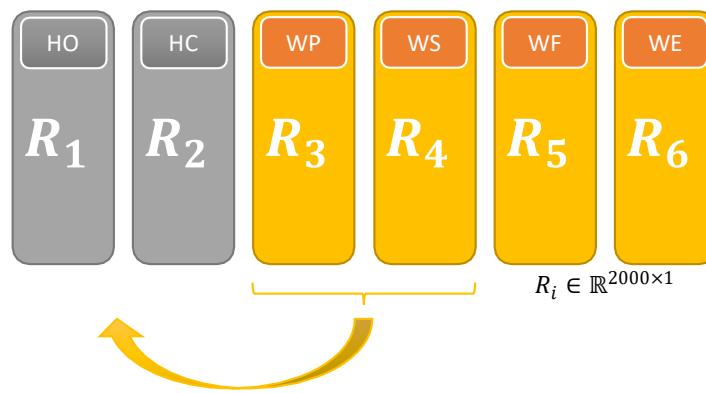


Figure 5.17: Resting measurements distribution for 3-DoF systems

- **FROM FEW (2 OR 4) RESTING STATE ACQUISITIONS INTO ‘N’ (4 OR 6) RANDOM CLUSTERS**

This method is a different interpretation of the third one. The number of resting measurements is lower than the total number of classes (N), however it gives higher variability to the system with respect to all the others. As in the previous method, it is possible to acquire 2 (for the 2-DoF systems) or 4 (for the 3-DoF systems) resting measurements and construct a unique resting vector of 4000 values (for the 2-DoF) or 8000 values (for the 3-DoF). From the huge resting-state vector is possible to randomly pick up 2000 values for building N *different* resting clusters (R_i). The 3-DoF case is represented in Figure 5.18.

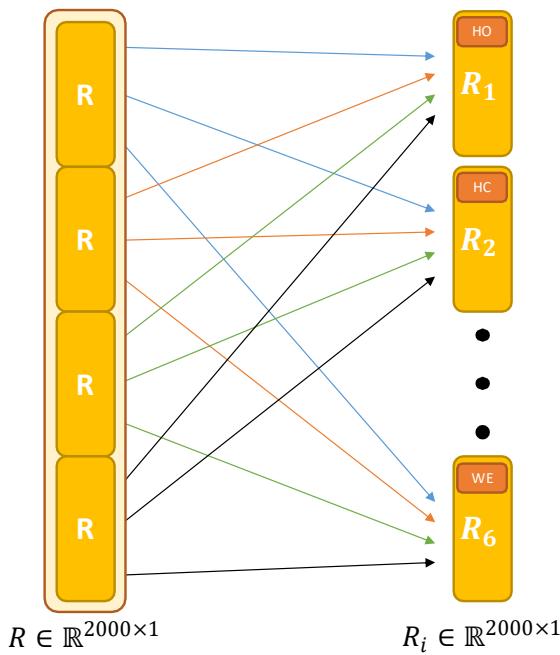


Figure 5.18: ‘N’ random resting clusters from 4 resting measurements

In the studies related to prosthetic control strategies, both measurements and results are strictly dependent on the patient attention and muscle contraction especially in the acquisition phase. It is important to reduce the time needed for the acquisition since it will help in attaining proper results. “During high-force and/or prolonged muscle contractions major intramuscular changes occur directly in terms of decreases in maximal muscle strength” [22].

For the aforementioned reasons, the fourth method appears the most suitable since it gives a reasonable trade-off among acquisition time and system variability.

With the already described techniques only one resting cluster for each movement is created. If the available classes are 4 (or 6), only 4 (or 6) resting measurements are obtained. In order to include more information, the classifier should be trained based on a larger number of samples so, to increase the resting section, it is possible to repeat the previous technique several times. As far as the random clusters generation is concerned, the 4 measurements (R in Figure 5.18) can be repeated 4 times to obtain 16 different resting measurements and, consequently, construct 24 random clusters (R_i). Then the 24 clusters will be split among the 6 gestures.

GESTURE	n° of acquisition	Total n° of samples
Wrist Flexion (WF)	10	20000
Wrist Extension (WE)	10	20000
Wrist Pronation (WP)	10	20000
Wrist Supination (WS)	10	20000
Hand Opening (HO)	10	20000
Hand Closing (HC)	10	20000
Rest	16	32000

Table 5.1: Number of samples and acquisitions for each gesture in 3-Dof

The 32000 samples of the resting state are organized in 6 clusters of 8000 samples through random allocation. In the end 48000 samples are generated from the 32000 samples. The final column vector for each gesture includes 20000 ($= 10 \times 2000$) samples coming from the 10 repeated movements (G_{TOT}) and 8000 ($= 4 \times 2000$) samples coming from the random distribution of the resting measurements among the 6 different classes (R_{TOT}). A graphical representation is shown in Figure 5.19.

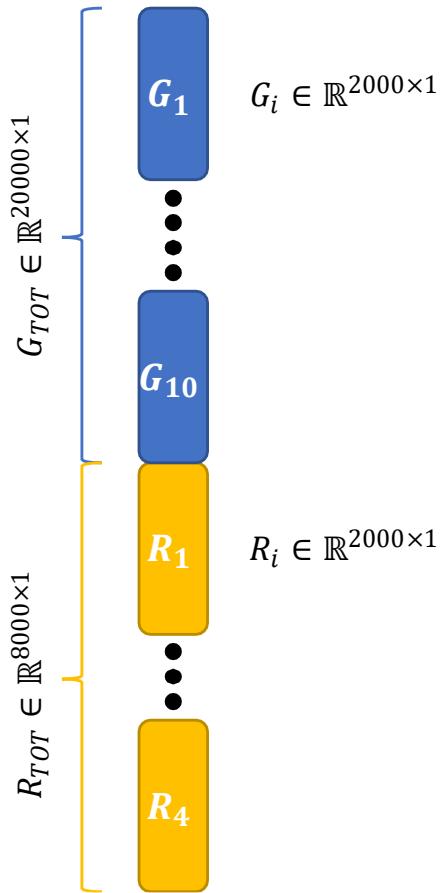


Figure 5.19: Training measurements for a single gesture

It is important to underline that for each movement it is not obtained a single vector (of dimension $\mathbb{R}^{28000 \times 1}$) but a matrix where each column is related to a single EMG sensor (6 n° of EMG sensors).

The final acquisition is hence described by a matrix of dimensions $\mathbb{R}^{28000 \times 6}$.

Although generally the more samples are considered the more accurate is the classifier, it is possible to say that the constructed dataset already appears too detailed for the adopted classifier. Therefore, it has been decided to perform a down-sampling technique (from 1kHz to 40Hz) to speed up the entire processing. In particular, for a down-sampling of 25 times (one sample each 25), the kept samples will construct the “Training group” (4%) and the discarded samples will compose the “Test Set” (TS: 96%). The TS set will be used at the end of the training phase for evaluating the model behavior and classification accuracy.

Since some of the considered classifiers (NLR, SVM, ANN) need an offline step for the model generation and parameters optimization, the “Training group” (4%) is divided into three different sets:

- Training set (TR: 60%);
- Validation set (VS: 20%);
- Thresholds-optimization set (ThS: 20%).

The Training set (TR) is exploited for the model generation, while the Validation Set (VS) has been used for parameters optimization in order to avoid the overfitting problems (the model is suitable for the already processed examples, but it is inefficient with the unknown data). The constructed model is characterized by 4 or 6 thresholds and 4 or 6 classes according to the initial choice of available DoF of the system. The Thresholds-optimization Set (ThS) is exploited to optimize the maximum likelihood thresholds for increasing system accuracy. As a matter of fact, the Test Set will contain the 96% of the initial data, the TR will contain the 2.4% of the initial data, the VS will contain the 0.8% of the initial data and the ThS will contain the 0.8% of the initial data (Table 5.2). This data splitting occurs for all the classifiers with the exception of the LDA case (Table 5.2). On the other hand, the results obtained from the Test Set (TS) represent an estimation of the classification ability when the machine learning algorithm is supplied with unknown EMG signals. Therefore, the TS set does not contain useless data. It is made by unknown samples for the classifier and it is used only for testing the performances. Instead the ThS is used for both optimizing thresholds and testing purposes, since the available samples inside the TS are already known by the classifier. The LDA algorithm does not need parameters optimization hence no down-sampling of the data is needed:

- Training Set (TR: 70%);
- Test Set (TS: 30%).

		<i>NLR</i>	<i>ANN</i>	<i>SVM</i>	<i>RLS</i>	<i>LDA</i>
Training Group	<i>TR</i>	2.4%	2.4%	2.4%	2.4%	70%
	<i>VS</i>	0.8%	0.8%	0.8%	0.8%	-
	<i>ThS</i>	0.8%	0.8%	0.8%	0.8%	-
Test Set	<i>TS</i>	96%	96%	96%	96%	30%

Table 5.2: Splitting of the dataset

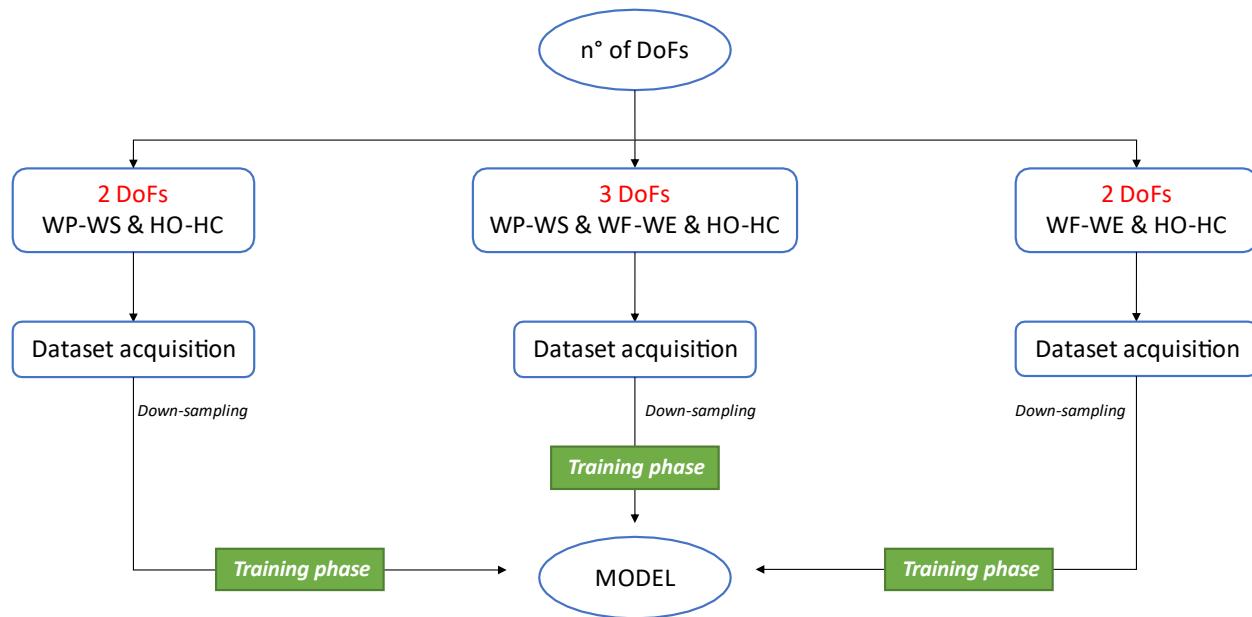


Figure 5.20: Block diagram for the model generation

As soon as the model is created, it can be tested offline for stating if its behavior is suitable or not for a prosthetic application. The model is tested through the thresholds-optimization set (ThS) in order to evaluate the classifier accuracy. During the offline evaluation only the single exceeding of the threshold is an acceptable result, while concurrent thresholds overcoming are considered as misinterpretation hence it leads to *abstention*.

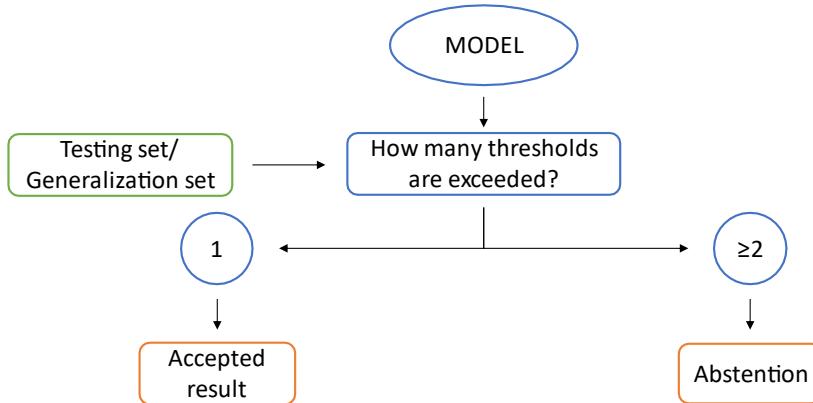


Figure 5.21: Block diagram for the classification phase

5.3.1.3 Real-time simulation

Subsequently, the real-time test is used to establish whether the newly generated classifier model is able to efficiently predict the desired prosthesis movement by taking as input the new real-time data. This test is an important intermediate step for establishing whether the model is generalized or it is affected by data overfitting.

At this point the classifier is tested online starting from a different (real) dataset which is directly withdrawn from the patient skin. By processing the new dataset, the classifier gives as output the class/classes which is/are exceeding the

corresponding optimized thresholds. Now, during the real-time test, different situations may occur with respect to the offline one:

- If only one class exceeds the threshold, the classifier allows the joint movement;
- If more than one class overcome the thresholds, the algorithm allows the movement only if the chosen classes belong to different DoF. In the case that not only the multiple classes exceed the thresholds but they also belong to the same DoF, the classifier output will be *abstention*.

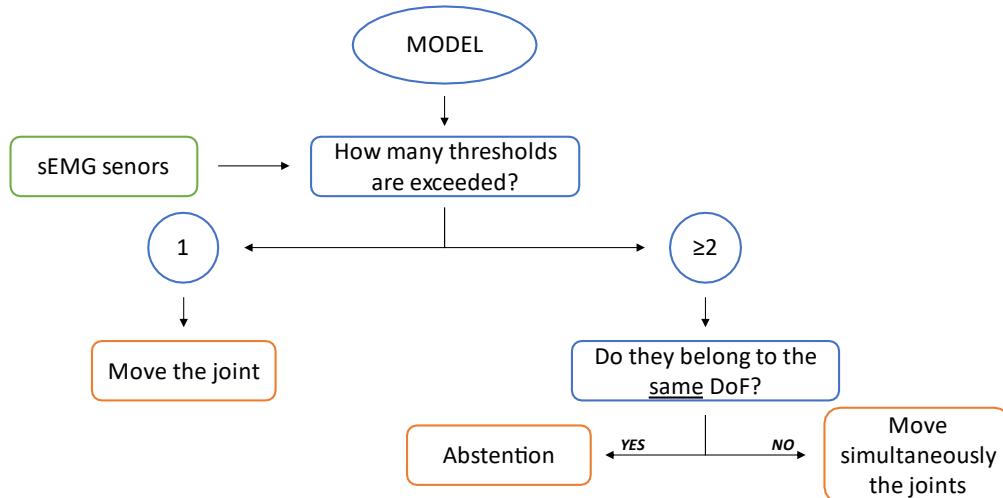


Figure 5.22: Block diagram for the classification phase

5.3.1.4 Model uploading & Simulation in Virtual Reality

The further step consists in uploading the classifier model into the EMG-Master via Bluetooth® (Figure 5.23). At this point, it is possible to use the EMG electrodes, the EMG-Master, the *Hannes* system and the battery pack as an embedded system. The last step, that must be activated during the *Uploading Phase*, is the control of *Hannes* system in virtual reality (VR). In this case the EMG-Master elaborates the input sEMG data inside the firmware and it sends via Bluetooth® the identified gesture class to the LabVIEW software. The E-DATS software, in turn, sends an encoded message via TCP/IP (Transmission Control Protocol/Internet Protocol) to the Unity software. The local IP address (127.0.0.1 with port 8889) is assigned to E-DATS in transmission mode, while Unity is set to the same address in listening mode. At this point, the C# script, related to the Unity animation, is able to decode the incoming message and it allows the movement of the prosthesis in the VR.

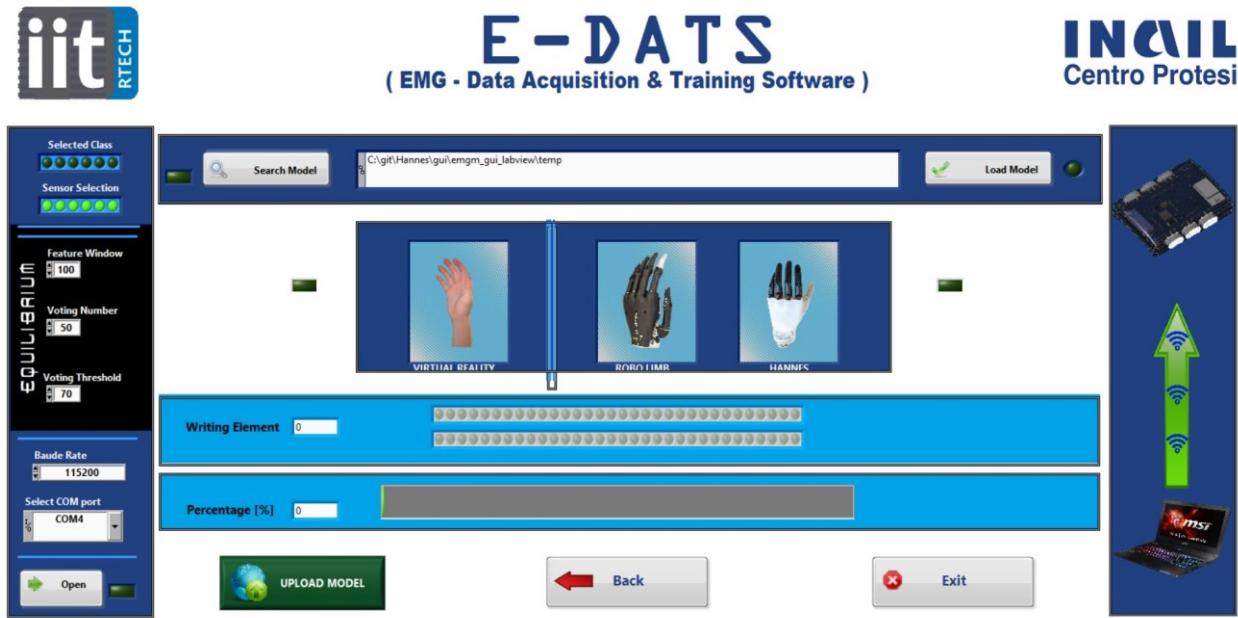


Figure 5.23: Uploading model interface

5.3.2 Virtual Reality

The virtual reality is realized through the software programs Blender and Unity. In the thesis, virtual reality has been used to have visible feedback on the correctness of the control algorithm. In the future, the VR can be exploited as a training or rehabilitation tool for the amputee. The patient can learn how to move the prosthesis by controlling the virtual *Hannes* first, and then the real device.

Blender is a free open-source 3D computer graphics software used for virtual reality, visual effects, interactive 3D applications and computer games [23]. In the framework of this thesis, Blender has been exploited for building the physical object of the prosthesis (Figure 5.24). The generation of *Hannes* system is based on simplified armature (called skeleton) which consists of links and joints. In addition, all the deformation areas of the body are defined through the *Vertex Paint* tool which can group some parts of the object and let them move simultaneously with a certain deformation parameter. Moreover, the definition of relative motion between links of the skeleton allows the parts of the object to move around the specific joint, like finger or wrist motion. In the 3D Blender environment, it is possible to assign to each joint its initial and final attitude (position and orientation), to generate a motion trajectory between the two poses. By bringing together all the trajectories, which were individually designed, it is possible to implement the desired gesture, as hand grasping or wrist flexion/extension. Basically, the body movements are defined in Blender, whereas the animation is implemented in Unity for allowing the control of the virtual reality through Bluetooth® connection with the EMG-Master.



Figure 5.24: *Hannes* system in Blender environment

Afterward, the Blender model is imported into the Unity environment in order to animate the physical object. Unity is a cross-platform game engine used to create 3D-2D virtual reality and augmented reality games [24]. In Unity, the Animator is applied to the virtual prosthesis and it is defined with three separate layers: the first one is connected to the hand Closing/Openning, the second one is connected to the wrist Pronation/Supination and the third one is connected to the wrist Flexion/Extension. The three layers of the Animator can be simultaneously activated in order to generate a combination of movements, which must belong to different joints. A C# script is written for controlling the virtual *Hannes* system (Figure 5.25) by receiving as input the class of the identified gesture. The number of the class is generated at the firmware level and it is the output of the classifier model. The control of the virtual *Hannes* system is defined as proportional since the position or velocity reference for the electric motors is related to the RMS value of the sEMG signals.



Figure 5.25: *Hannes* system in Unity environment

Finally, a standalone application of the Unity project is built, in order to directly control the virtual reality through the LabVIEW software simulation (described in paragraph 5.3.1). Therefore, the LabVIEW software has a double Bluetooth® connection: one with the EMG-Master and the other with the Unity application of virtual reality.

6. Study protocol

The primary goal of the thesis is to carry out a comparative analysis between different control strategies for upper limb prosthesis based on Pattern Recognition. In this work, a training protocol similar to that used in [25] is employed, but the involved prosthesis gestures are different. Each subject gave the informed consent before performing the experiments. Testers are ten able-bodied subjects, who have never performed the trial before, and four transradial amputees subjects (monoliteral and bilateral). The able-bodied participants are aged between 22 and 33 years ($\mu = 27.1$, $\sigma = 3.2$). For the healthy subjects the sEMG signals are recorded from the dominant forearm. The elastic band is applied on the forearm, approximately 5cm distal to the olecranon, and the sensors are placed, firstly, by asking the subject to reproduce a specific movement and, then, by a manual inspection of the desired muscle. The first step was focused on identifying the forearm muscles involved in opening and closing of the hand, and in pronation, supination, flexion, extension of the wrist. The electrodes are fastened at the forearm level (Figure 6.1) as:

- Electrode n°5: Brachioradialis muscle;
- Electrode n°4: Extensor Carpi Ulnaris muscle;
- Electrode n°3: Flexor Carpi Radialis muscle;
- Electrode n°2: Extensor Digitorum muscle;
- Electrode n°1: Flexor Carpi Ulnaris muscle;
- Electrode n°0: Extensor Carpi Radialis Longus muscle.

The two types of electrodes exploited in the thesis are described in paragraph 5.1.1 and they operate in the range of 0 – 5V.

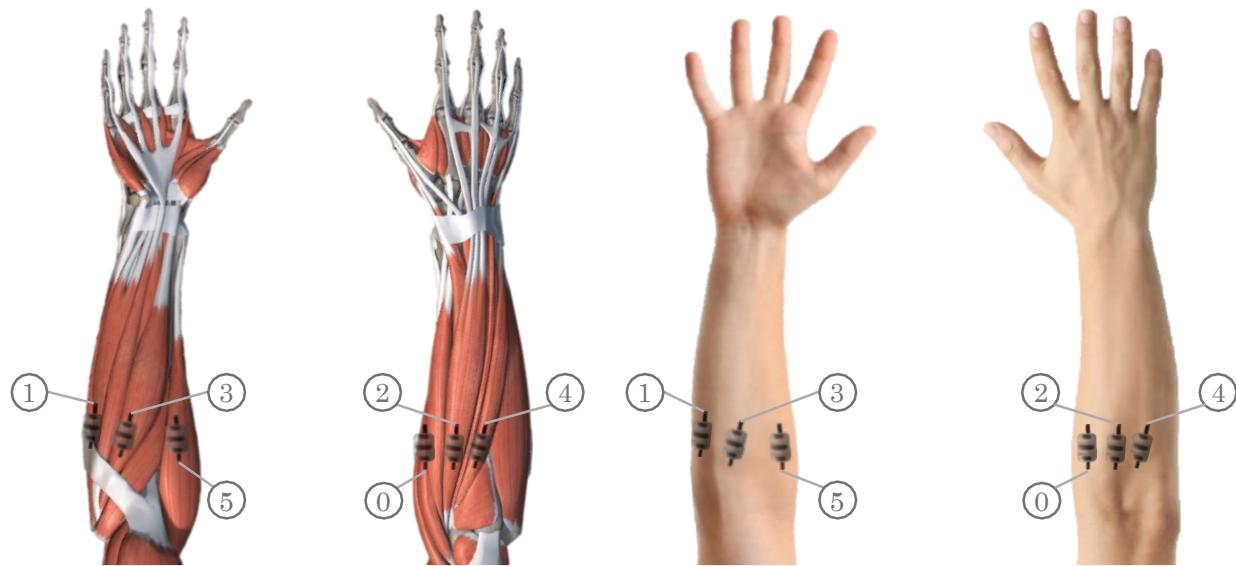


Figure 6.1: Electodes placement over the corresponding muscles on the right arm

Each subject was sitting on a chair in front of the PC monitor where, through the E-DATS software, the prosthesis gestures were displayed (Figure 5.11). The gestures involved in the experiment are shown in Figure 6.2. Participants were asked to reproduce for ten times the specific movement and for sixteen times the resting state. The acquisition of a single gesture consisted of starting the data recording and, subsequently, asking the subject to perform the movement. In this way, the acquired signals contained also the transient phase, from the resting state to the maximum contraction. This technique is performed to make the classifier faster in the real-time applications, in such a way that it could be able to quicker predict the desired movement. If this process had not been applied, the classifier would have been able to predict the human intention only when the muscle activity would have reached its maximum contraction. The recording phase of a single contraction lasted 2s (sampling frequency of 1kHz) with an inter-stimulus interval of about 5s during which the subject was asked to keep the resting position. Preliminary studies determined that the best gesture duration is equal to 2s, as described in [25]. The dataset acquired during the training phase was exploited for building the NLR algorithm (paragraph 7.1) with maximum complexity to allow the subject to control both the real Hannes system and the virtual one. The final step of controlling the prosthetic device was suitable for having a visual feedback of the actual operation of the system. The trial was performed twice on different days as it was necessary to create a dataset for both types of sensors (IIT and MyoBock). Each of the two sessions lasted for approximately 40min.

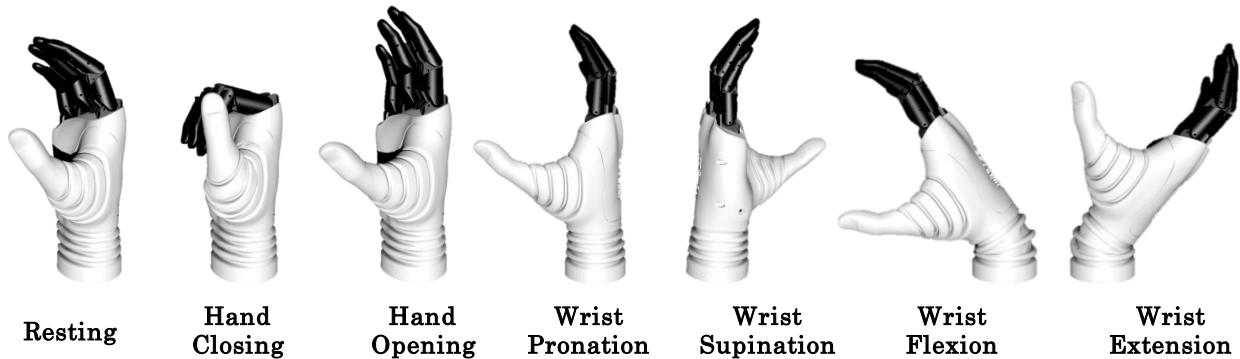


Figure 6.2: *Hannes* gestures

The exploited setup for the trials is shown in Figure 6.3 and it includes the elastic band with six sEMG sensors (MyoBock) (A), the power supply (B), the EMG-Master (C), the *Hannes* system (D), the E-DATS software (E) and the Unity software (F). The experiment process, in the presence of the participant, consists in the:

1. Presentation of the study protocol and thesis aims;
2. Fulfillment of informed consent and the consent of data processing;
3. Subject identification and evaluation of the dominant limb (only for able-bodied and bilateral amputees);
4. Application of the set of electrodes on the skin surface;
5. Data acquisition;
6. Creation of the NLR model;
7. Simulation in real-time for evaluating the NLR model suitability;
8. Uploading of the NLR model on EMG-Master;

9. *Hannes* system control (real and virtual).

The steps from the fourth to the ninth are repeated by using the other set of electrodes. [18] showed that the NLR model reached the best compromise in terms of EOF, ergo, as a first instance, this machine learning algorithm is used in the online experiment.

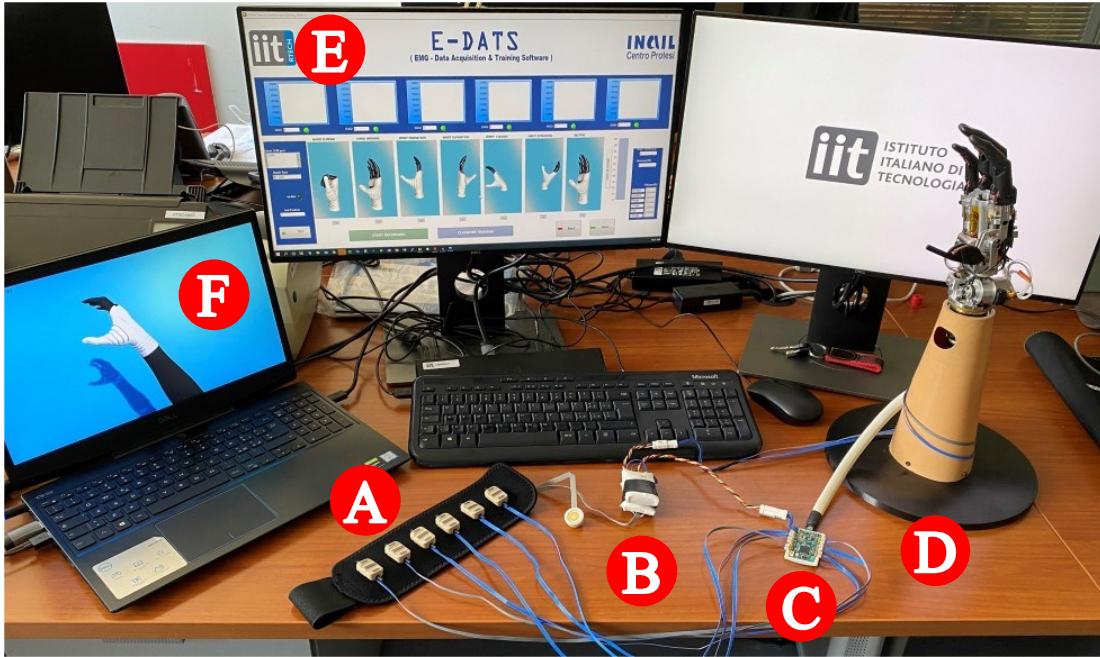


Figure 6.3: Setup pf the experiment

Subsequently, the offline analysis on the dataset of the single participant aims to state the most suitable classifier that can be used for upper-limb prosthetic applications. The process of the data analysis is divided into two steps:

1. Firstly, the single subject dataset is evaluated in order to separately obtain quantifiable results on performances and computational burden by varying the number of electrodes and the classifier complexity;
2. Secondly, a group analysis is performed to assess statistical results over the whole population involved in the experiments.

It is not needed to acquire different dataset in the case of different number of electrodes. To reduce the duration of the experiment, the trial is always conducted with six sensors, but the evaluation of the algorithms is performed offline by cancelling data coming from specific electrodes and by building a new model for each condition.

In this study, all the exploited classifiers fall into the class of supervised machine learning (chapter 7), hence a preceding training phase is needed. As mentioned in paragraph 5.3.1.2, the dataset used to build the model is down sampled from 1kHz to 40Hz, for reducing the computational time. During the single analysis, different models (for each type of classifier) are built by fixing the complexity to the highest value and by varying the number of electrodes (from 2 to 6). Each model is then associated with its own F₁-score and Embedding Optimization Factor (EOF). At this point a group analysis is conducted for evaluating the best number of electrodes for each classifier. The

optimization process may end up with different number of electrodes in the case of MyoBock sensors or IIT sensors, thus the group analysis is performed separately for the two types. As soon as the number of electrodes is optimized, a further single analysis is performed. This single analysis aims to optimize the complexity of NLR and ANN algorithm. By keeping fixed the optimized number of electrodes already found, the former is subject to a *D-value* variation (from 1 to 7) and the latter is subject to the variation of the number of layers (from 1 to 10). In that moment, the following group analysis aims to find the optimized values for *D* (NLR) and for the number of hidden layers (ANN). A further optimization is required only for the ANN algorithm since the number of neurons for each hidden layer still needs to be minimized. The number of neurons is changed (form 1 to 30) by keeping fixed both the number of hidden layers and the number of electrodes, which were previously optimized. At the end of the offline procedure, each classifier has its own optimized parameters.

The results of the evaluation process, which has been just outlined, are reported in part III.

7. Pattern-Recognition algorithms

The field of machine learning is focused on the construction of algorithms which are able to make predictions based on data. The machine learning is an application of the artificial intelligence (AI) that provides systems the ability to automatically learn and improve from the experience without being explicitly programmed. “Arguably, the problem of learning represents a gateway to understanding intelligence in brains and machines, for discovering how the human brain works, and for making intelligent machines that learn from experience and improve their competences as children do. In engineering, learning techniques would make it possible to develop software that can be quickly customized to deal with the increasing amount of information and the flood of data around us” [26].

It is important to emphasize that machine learning is a mathematical tool which relates a set of input values to a certain output, therefore, it is essential to have quantifiable values as input. In most problem domains, there is no functional relationship ($y = f(x)$) between y and x , and, usually, the input-output relation is described through a probability distribution ($P(y|x)$). Machine learning is based on computer programs which are able to access data and use them to learn. The learning process starts with data observation and it consists in looking for possible patterns to make better decisions in the future. Machine learning requires the right set of data to be applied to a learning process. It is unnecessary to have big data availability for building machine learning models, at most it can be used to improve their accuracy. Different types of machine learning algorithms are available:

- Supervised machine learning exploits labeled data (of the form $(x_i, y_i)_{i=1}^N$) to predict future events. Such a system is trained with set of examples, that is, a set of input-output pairs. After the data analysis, the inferred model should be able to match specific inputs to a specific class.
- Unsupervised machine learning can exploit only input data for finding regularities among the input set. One of the most widespread methods is the *clustering* which aims to find groupings in the input data.
- Regularized learning is a behavioral learning algorithm since it receives feedback on its decisions. As a matter of fact, this machine learning algorithm learns through a trial and error procedure.

In this thesis several machine learning algorithms are used for Pattern Recognition purposes and all of them fall into the category of supervised machine learning. As for [18], a comparative analysis is performed between the following algorithms: Non-Linear Logistic Regression (paragraph 7.1), Support Vector Machine (paragraph 7.4), Regularized Least Squares (paragraph 7.2), Artificial Neural Network (paragraph 7.3) and Linear Discriminant Analysis (paragraph 7.5). NLR, RLS and LDA are regressors, hence their output is the class membership probability coming from the likelihood function, while ANN and SVM are classifiers, thus their output is the predicted movement.

7.1 Non-Linear Logistic Regression (NLR)

The Logistic Regression is a linear and binary supervised classification algorithm that calculates the class membership probability using the following logistic function:

$$P(1|x, \theta) = \begin{cases} g(\theta^T \cdot x) = \frac{1}{1 + e^{-(\theta^T \cdot x + \theta_0)}} \\ 1 - P(y = 0|x, \theta) \end{cases} \quad (11)$$

where θ and θ_0 are the classification parameters vector and the bias term, and $g(\cdot)$ is the sigmoid (logistic) function. The Non-Linear Logistic Regression classifier is achievable with the addition of other input features which are called interaction terms. The NLR algorithm, exploited in the thesis, is characterized by polynomial input features which are obtained from the combination product of the starting input features (i.e. $x_1 \cdot x_2; x_1^2; x_2^2; \dots$). For both the algorithms NLR and LR, the class prediction is achieved by comparing the distribution $P(y|x)$ with a decision threshold (TH) as:

$$h_\theta(x) = \begin{cases} P(1|x, \theta) \geq TH & \rightarrow 1 \\ P(1|x, \theta) < TH & \rightarrow 0. \end{cases} \quad (12)$$

The complexity of the NLR classifier is determined by the internal parameter D , which encodes a structure of polynomial features as reported in Table 7.1. The polynomial features correspond to a new arrangement of the input variables (x_1, x_2, \dots, x_6): from the linear terms up to the D -power elevation and all the possible permutations without repetitions of a maximum number of elements corresponding to the indicated degree (D). Furthermore, a cost function J , called *cross-entropy* error, has been associated to the NLR algorithm to implement an internal parameters optimization:

$$J(\theta, \theta_0) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \cdot \ln g(\theta^T \cdot x^{(i)} + \theta_0) \right] - \frac{1}{m} \left[\sum_{i=1}^m (1 - y^{(i)}) \cdot \ln (1 - g(\theta^T \cdot x^{(i)} + \theta_0)) \right] \quad (13)$$

where m is the number of samples used to train the algorithm and $y^{(i)}$ is the known class membership of the $i - th$ sample. Since NLR is a binary classification algorithm, a 1vsAll approach is implemented to address the multi-class classification problem.

D	Description	Example
1	Linear case (LR)	$x_1, x_2, x_3, x_4, x_5, x_6$
2	max 2nd degree	$x_1, \dots, x_6, x_1 \cdot x_2, x_1 \cdot x_3, \dots, x_5 \cdot x_6, x_1^2, x_2^2, \dots, x_6^2$
3	max 3rd degree	$x_1, \dots, x_6^2, x_1 \cdot x_2 \cdot x_3, \dots, x_4 \cdot x_5 \cdot x_6, x_1^3, \dots, x_6^3$
4	max 4th degree	$x_1, \dots, x_6^3, x_1 \cdot x_2 \cdot x_3 \cdot x_4, \dots, x_3 \cdot x_4 \cdot x_5 \cdot x_6, x_1^4, \dots, x_6^4$
5	max 5th degree	$x_1, \dots, x_6^4, x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5, \dots, x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6, x_1^5, \dots, x_6^5$
6	max 6th degree	$x_1, \dots, x_6^5, x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6, x_1^6, \dots, x_6^6$
7	max 7th degree	$x_1, \dots, x_6^6, x_1^7, x_2^7, x_3^7, x_4^7, x_5^7, x_6^7$

Table 7.1: Encoding the variable D for NLR algorithm

7.2 Regularized Least Squares (RLS)

The Regularized Least Squares (RLS) is a binary and linear supervised classification algorithm, which evaluates the class membership probability using the following linear relation:

$$f(x) = w^T \cdot x \quad (14)$$

where x is the input sEMG data and w is a weight vector. In addition, several features (X) in the time domain are extracted from the electromyographic signal for enhancing the classification performances: Mean Absolute Value [Eq. (1)], Root Mean Square [Eq. (3)], Variance [Eq. (8)], Sum, Standard Deviation [Eq. (9)]. Input data, which are not easily classifiable by a linear model in the original input space, can be easily classified by a linear model in the feature space. Subsequently to the features extraction, a kernel method is applied. The kernel replaces the inner product of the input data ($x^T \cdot x'$) with a more general function. Typically, the positive definite kernels are:

- Linear kernel: $K(X, X') = X^T \cdot X'$;
- Gaussian kernel: $K(X, X') = e^{-\frac{\|X-X'\|^2}{2\sigma^2}}$;
- Polynomial kernel: $K(X, X') = (X^T \cdot X' + 1)^d$;

where the tuning parameters are: the polynomial degree d and the Gaussian width σ . The kernel application aims to move from the nonlinear input space to the linear feature space. Once the problem is linearized, the solution becomes:

$$f(X, X') = K(X, X')^T \cdot C \quad (15)$$

where C is defined as:

$$C = (K(X, X') + \lambda_i I)^{-1} Y \quad (16)$$

The parameter λ , which is chosen with a cross-validation technique, controls the invertibility of the matrix $(K(X, X') + \lambda_i I)$, Y is a vector containing the output classes as entries and I is the identity matrix. By knowing C , on the basis of the linear equation (15) it is possible to predict the output classes by considering unknown data as input. As it happens for NLR algorithm, the probability distribution of the predicted output $P(y|x)$ is compared with a decision threshold (TH) for achieving the value of the output class (h_θ). Since the RLS is a binary classification algorithm, to address the multi-class classification problem a 1vsAll approach is implemented. The improvement of RLS algorithm is based on the optimization of kernel parameters (i.e. d for the polynomial kernel, σ for the Gaussian kernel), and the regularization term (λ). The cost function, *expected loss* (J), is associated to the RLS classifier:

$$J(w, \lambda) = \min_{w \in \mathbb{R}^D} \frac{1}{m} \sum_{i=1}^m (y^{(i)} - w^T \cdot x^{(i)})^2 + \lambda \cdot w^T \cdot w \quad , \lambda \geq 0 \quad (17)$$

where m is the number of samples used to train the classifier and $y^{(i)}$ is the known class of the $i - th$ sample. The above analysis is not directly useful in practice since the data distribution is not accessible, hence, a data driven procedure (*hold-out cross validation*) is used to find a proxy of the cost function.

In the thesis, the RLS algorithm is implemented through the MATLAB software and it has a Gaussian kernel with maximum internal parameter σ_{max} equals to 5 and the maximum regularization term λ_{max} equals to 10.

7.3 Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is a supervised classification algorithm where each neuron inside the architecture implements a logistic function. Typically, the network architecture is composed by an input layer, one or more hidden layers (with the same number of neurons), and an output layer where each neuron is associated to a class. The complexity of ANN algorithm is based on the number of hidden layers (from 1 to 10) and the number of neurons (from 1 to 30). The output vector of the $l - th$ layer ($a^{(l)}$) of this classifier is computed through a forward propagation:

$$a^{(l)} = \begin{cases} x, & l = 1 \\ g(\Theta^{(l-1)} \cdot a^{(l-1)} + \Theta_0^{(l-1)}), & l = 2, 3, \dots, L \end{cases} \quad (18)$$

where L indicates the number of neurons in the output layer, $\Theta^{(l)}$ is the classification parameters matrix associated with the $l - th$ layer and $\Theta_0^{(l)}$ is the bias vector associated with the $l - th$ layer. Therefore the output of the ANN is a vector $Pv(y|x)$ whose elements represent the class membership probability expressed as:

$$Pv(y|x, \Theta^{(l)}, \Theta_0^{(l)}) = a^{(l)}, \quad l = 1, 2, \dots, L \quad (19)$$

Subsequently, the achievement of the predicted class value ($h_\theta(x)$: reported in equation (12)) is implemented by comparing each probability distribution of $Pv(y|x)$ with the decision threshold (TH). The optimization of the ANN classifier is based on maximizing the number of hidden layers and the number of neurons for each layer. The cost function, *mean square error* (J), is associated to the ANN classifier:

$$J(\Theta, \Theta_0) = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[y_k^{(i)} - (a_k^{(L)})^{(i)} \right]^2 \quad (20)$$

where m is the number of samples used to train the classifier, K is the number of output neurons (classes to be identified), $y_k^{(i)}$ is the known $k - th$ element of the class membership vector of the $i - th$ sample, and $a_k^{(i)}$ is the $k - th$ element of the evaluated membership probability vector of the $i - th$ sample.

7.4 Support Vector Machine (SVM)

The Support Vector Machine [27]-[28] is a linear and binary supervised classification algorithm, which considers only dichotomous classification problem, where the class labels are either 0 or 1. The SVM classifier is based on the following function:

$$h_\theta(x) = \begin{cases} (\theta^T \cdot x + \theta_0) \geq +1 & \rightarrow 1 \\ (\theta^T \cdot x + \theta_0) \leq -1 & \rightarrow 0 \end{cases} \quad (21)$$

where θ and θ_0 are the parameters vector and the bias term of the classifier. To obtain a nonlinear algorithm, a *kernel*

function needs to be included into the model. By using different *kernel functions*, satisfying the Mercer's Theorem, multiple degrees of nonlinearity and flexibility can be included in the model. The kernel function is a similarity function (f), which expresses the correspondence between a generic input vector (x) and the landmark (s). Indeed, each of the two compared classes has its own landmark (s). Usually, a selection of all the recorded vectors (x) are set as landmarks and the $j - th$ element of f for RBF (Radial Basis Function) kernel becomes:

$$f_j = \exp \left[-\frac{|x - s^{(j)}|^2}{2\gamma} \right], \quad j = 1, 2, \dots, n \quad (22)$$

where n is the number of landmarks chosen as representative vector of classes 0 and 1, and γ is the internal RBF parameter. The input data (x) are mapped to a higher dimensional space by performing a nonlinear projection (f_j), in order to linearly estimate the regression function. As a consequence, the input features vector becomes f and the class labels for a SVM with RBF kernel are assigned on the basis of the following relation:

$$h_\theta(f) = \begin{cases} (\theta^T \cdot f + \theta_0) \geq +1 & \rightarrow 1 \\ (\theta^T \cdot f + \theta_0) \leq -1 & \rightarrow 0 \end{cases} \quad (23)$$

where θ and θ_0 are obtained through a minimization of a particular cost function related to the SVM model. The cost function (J) related to the SVM classifier is:

$$\begin{aligned} J(\theta, \theta_0) = & -C \left[\sum_{i=1}^m y^{(i)} \cdot \ln g(\theta^T \cdot f + \theta_0) \right] - C \left[\sum_{i=1}^m y^{(i)} \cdot \ln (1 - g(\theta^T \cdot f + \theta_0)) \right] \\ & + \frac{1}{2} [\theta^T \cdot \theta + (\theta_0)^2] \end{aligned} \quad (24)$$

The optimized values of regularization parameters (C) and the internal RBF parameter (γ) are obtained through a cross-validation procedure: C belongs to the range $0 - 10^4$ with variable steps, starting with 0.01 and doubling each time, while γ varies in the range $0 - 50$ with a step of 0.1. The SVM classifier with RBF kernel is implemented by exploiting the open-source library *libsvm3.20*, where a *one vs one* algorithm is developed for solving multi-class classification problems [29]. The main disadvantage of SVM is that it considers only dichotomous distinction between two classes and no probability of class membership is given.

In the thesis, the SVM algorithm is implemented through the MATLAB software and it is characterized by a maximum internal RBF parameter γ_{max} equals to 50 and a maximum regularization parameter C_{max} equals to 1000.

7.5 Linear Discriminant Analysis (LDA)

The Linear Discriminant Analysis (LDA) is linear and binary supervised classification algorithm, which considers dichotomous classification problem, and assigns the class label 1 or 2 to unknown data item. The LDA classifier is based on the following function:

$$h_{\beta}(x) = \begin{cases} (\beta^T \cdot x + \beta_0) \geq 0 & \rightarrow 1 \\ (\beta^T \cdot x + \beta_0) < 0 & \rightarrow 2 \end{cases} \quad (25)$$

where β and β_0 are the classification parameters vector and the bias term. The classification parameters are evaluated through the following relations:

$$\begin{cases} \beta = \Sigma^{-1} \cdot (\mu_1 - \mu_2) \\ \beta_0 = -\beta^T \cdot \left(\frac{\mu_1 + \mu_2}{2} \right) + \ln \left(\frac{\Pi_1}{\Pi_2} \right) \end{cases} \quad (26)$$

where Σ is the covariance matrix, μ_1, μ_2 are the mean vectors of class 1 and class 2 and the Π_1, Π_2 are the prior probabilities of class 1 and class 2. The LDA classifier, conversely to all the aforementioned algorithms, does not have any associated cost function to be optimized, hence it does not need any validation process.

Since the LDA is a binary classification algorithm, to address the multi-class classification problem a 1vsAll approach is implemented. The class label (c) is evaluated as:

$$\begin{cases} h_{\beta}(x) = \max_c (c\beta^T \cdot x + c\beta_0) \\ c\beta = \Sigma^{-1} \cdot \mu_c \\ c\beta_0 = -c\beta^T \cdot \left(\frac{\mu_c}{2} \right) + \ln(\Pi_c) \end{cases} \quad (27)$$

where $c\beta$ is the classification parameters vector of c class and $c\beta_0$ is the bias term of c class. Furthermore, several features in the time domain are extracted from the sEMG signal for enhancing the classification performances: Mean Absolute Value [Eq. (1)], Root Mean Square [Eq. (3)], Slope Sign Change [Eq. (5)], Waveform length [Eq. (6)] and Variance [Eq. (8)]. EMG data are divided into time windows of 250ms with and overlap of 200ms [30]. These windowing parameters show a good trade-off between classification error and controller delay during real-time applications. A short time is needed to train the classifier since the LDA exploits features extraction, therefore, the down-sampling is not necessary.

8. Algorithms optimization

Before using sEMG signals as input of the algorithms, a unique operation is performed. The scaling technique is applied to each signal coming from the single electrode. It consists in subtracting the mean value (μ) from each sample (x_i) and dividing the result by the range ($\Delta x = x_{max} - x_{min}$), as:

$$\hat{x}_i = \frac{x_i - \mu}{\Delta x} \quad (28)$$

The resulting signal (\hat{x}) has a zero-mean. The plots in Figure 8.1 represent the six EMG signals acquired through the MyoBock sensors during the closing of the hand of a subject involved in the study. For each time instant i , the N -element vector (N coinciding with the electrodes number) is used as input for the classification algorithms. A special consideration is given to the classifier complexity to achieve a system with the lowest computational burden. For this reason each classifier algorithm is optimized in order to have approximately the same performances with the lowest complexity. As a matter of fact, for the NLR the D value is changed in the range 1 to 7, while for the ANN algorithm the number of hidden layer (from 1 to 10) and the number of neurons (from 1 to 30) for each hidden layer are varied. The performances are evaluated on the basis of the F_1 -score index, which is a measure of the classifier accuracy, while the computational burden is evaluated through the EOF index (Embedding Optimization Factor). By fixing the number of electrodes, the aim is to define the internal parameters of NLR and ANN for which the F_1 -score and EOF index flatten.

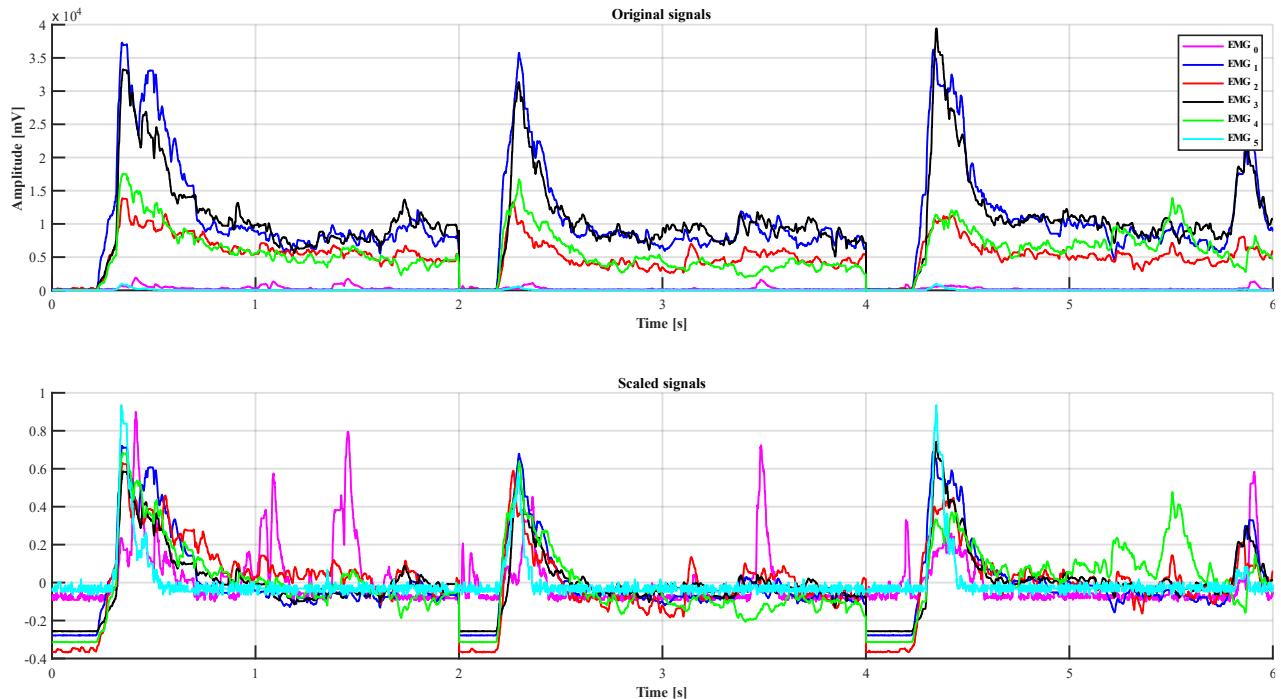


Figure 8.1: Scaling technique applied to EMG signals

The F_1 -score [31] is a common mode for evaluating results of machine learning algorithm and it is computed on the

basis of two properties: Precision (PR) [Eq. (29)] and Recall (RE) [Eq. (29)]. The Precision is the fraction of relevant instances among all retrieved instances, while the Recall (sensitivity) is the fraction of retrieved instances among the total amount of relative instances. The negative aspect is that these measures present specific biases, namely that they ignore performance in correctly handling negative examples. The confusion matrix of predicted outputs is:

	$y = 1$	$y = 0$
$h_\theta = 1$	nP	nFP
$h_\theta = 0$	nFN	nF

Table 8.1: Confusion matrix

where nP is the number of true positive, nF is the number of true negative, nFN is the number of false negative and nFP is the number of false positive. On the basis of the aforementioned terms, it is possible to compute the Precision, the Recall and the F_1 -score as:

$$\begin{cases} PR = \frac{nP}{nP + nFN} \\ RE = \frac{nP}{nP + nFP} \\ F_1 score = 2 \cdot \frac{PR \cdot RE}{PR + RE} \cdot 100 \end{cases} \quad (29)$$

The statistical analysis is performed through the Wilcoxon-Signed-Rank test which can be used to determine whether two dependent samples are selected from populations having the same distribution. This test is used instead of the *Student's t-test* when the assumption of normally distributed data is not satisfied. The Wilcoxon-Signed-Rank test is a nonparametric statistical test that calculates the difference between sets of pairs and it establishes if these sets are statistically significantly different from one another. The test, with a *p-value* lower than 0.05, is proven to be suitable for comparing several classifiers trained with a common datasets [32]. Actually, the statistical definition states that the lower the *p-value*, the greater the evidence against the null hypothesis, thus data are acceptably different.

The further optimization step is based on minimizing the classifier complexity and, consequently, the computational burden. Therefore the combined index, called EOF (Embedding Optimization Factor), is exploited. This index plays an important role when dealing with embedded systems, with limited memory storage. The EOF index takes into account both performance and computational burden and it is evaluated as follow:

$$\begin{cases} P = \frac{N\Theta - n\theta}{N\Theta} \cdot 100, & \text{if: } N\Theta > n\theta \\ P = 0, & \text{if: } N\Theta \leq n\theta \\ EOF = 2 \cdot \frac{F1Score \cdot P}{F1Score + P} \end{cases} \quad (30)$$

where $N\Theta$ is the maximum number of acceptable parameter [Eq. (18)] and $n\theta$ expresses the cardinality of classification vector θ [Eq. (11)- Eq. (23)]. The TiVa-microcontroller (Texas Instruments) is implemented on the EMG-Master microchip with a flash memory of 256KB, as a consequence, the value of $N\Theta$ is chosen as the maximum

number of parameters that can be stored in such memory. Since each parameter is coded as a float value (4 memory bytes), the maximum number of storable parameters is 64,000 (coming from $(256 \cdot 10^3 \cdot 8)/(4 \cdot 8)$).

III. Results

The part III of the thesis reports the results obtained after the analysis on data acquired during the experiments. In order to make a comparison between the two types of exploited sensors, the part III is divided into two different chapters. The chapter 9 is related to the results achieved through the Ottobock electrodes, whilst the chapter 10 is related to the results achieved through the IIT electrodes. Each of the two chapters is divided into several paragraphs: a first optimization is performed in terms of the best number of electrodes, subsequently, the complexity of the machine learning algorithms is minimized. The third paragraph (9.3 and 10.3) reports the comparative analysis between the aforementioned classifiers. All these paragraphs are related to the able-bodied subjects, whilst the last paragraph of each chapter (9.4 and 10.4) reports the performances of the classifiers obtained from experiments on amputees.

An important aspect to take into account is the number of electrodes used to acquire data, since they have a huge impact on battery consumption, mechanical encumbrance and also prosthetic device costs. The aim is to find a system with the lowest number of electrodes in the prosthesis, always ensuring a high level of performances. For this reason, the Wilcoxon-Signed-Rank test is exploited for evaluating the number of electrodes at which performances flatten by considering the highest level of complexity for each classifier (the computational burden is optimized afterward).

A parallel analysis is made between Ottobock and IIT sensors in order to evaluate the overall behavior of the system. As mentioned in paragraph 5.1.1, in the case of equivalent performances, the IIT electrodes allow to have several advantages like lower mechanical encumbrance in the prosthesis socket and lower costs. The electrodes n°0 and n°1 are placed respectively on the Extensor Carpi Radialis Longus muscle and the Flexor Carpi Ulnaris muscle, since they are the antagonistic muscles exploited for the conventional dual-site control. Therefore, when comparing the performances and the computational burden with different number of exploited sensors, these two electrodes will always be considered active. It is important to underline that the experiment is always conducted with six electrodes, but the evaluation of algorithms is performed offline by not considering the data of some sensors. As a consequence, the possible combinations of electrodes are:

- Six active sEMG sensors: n°5, n°4, n°3, n°2, n°1, n°0;
- Five active sEMG sensors: n°4, n°3, n°2, n°1, n°0;
- Four active sEMG sensors: n°4, n°3, n°1, n°0;
- Three active sEMG sensors: n°4, n°1, n°0;
- Two active sEMG sensors: n°1, n°0.

The results for able-bodied subjects are presented in boxplots where the central line represents the median value and the edges are the 25th and the 75th percentiles. The vertical whiskers indicates the variability outside the upper and lower quartiles without outliers (plotted as red plus signs), while the solid markers represent the mean values. The *NS*

indicators include all the cases in which the statistical difference is not significant in terms of Wilcoxon-Signed-Rank test with threshold set to 0.05.

9. Results with MyoBock sensors

The experimental procedure described in chapter 6 is used to obtain the results reported in this chapter, when the set of MyoBock sensors is exploited for the acquisition of the sEMG signals. It is important to underline that the models of the algorithms are built through the “Training Set” (TR), but the results are evaluated on the unknown data contained in the “Test Set” (TS). The two sets are obtained after the down-sampling technique, as described in paragraph 5.3.1.2.

The F_1 -score indexes, which are related to the performances, are used to optimize the number of Ottobock electrodes. The higher the F_1 -score, the better the performance. As soon as the number of electrodes is optimized, the obtained value is fixed for the subsequent operations. The further phase of optimization is related to the minimization of the classifier complexity. For the NLR algorithm, the polynomial degree (D) should be minimized. Considering that the classifier complexity is strictly related to the value of the variable D , the Embedding Optimization Factor (EOF) is used to search for the minimum degree of the polynomial. For the ANN algorithm, both the number of layers and the number of neurons should be optimized. As before, the minimization of the classifier complexity is based on the EOF index. Initially, the number of layers is reduced by considering the maximum number of neurons (30) as long as the performance degradation does not undergo important variations. The optimized value of the number of layers is fixed for the subsequent minimization of the number of neurons (per each layer). At the end of the sequential process of optimization, each classifier is characterized by its own setup with minimized parameters.

The comparative analysis between the five algorithms (paragraph 9.3) is performed on:

- the classifiers performances evaluated in terms of F_1 -score;
- the computational burden evaluated in terms of EOF;
- the classification (percentage of correctly predicted movements: $nP + nN$ in Table 8.1);
- the abstention (percentage of not assigned movements).

9.1 Optimization of the number of MyoBock electrodes

The first investigation concerns the minimization of the number of electrodes as long as the performances saturate and no statistical difference occurs anymore. For this type of analysis the F_1 -score index is taken into account since it gives an idea of the classifiers behavior in different conditions. The low statistical difference in terms of performances are pointed out by the NS indicators. To have a reasonable performance degradation, the minimum amount of needed electrodes corresponds to the smallest values shown by the NS indicators. In this chapter all the tables contain a row in bold which indicates the best choice of the number of electrodes for each classifier.

For the NLR algorithm, the best quantity of MyoBock sensors is 5 as summarized in Table 9.1. As mentioned in paragraph 7.1, the polynomial degree (D) can range from 1 to 7. During this preliminary phase of optimization, the computational burden is not considered, hence the training of the classifier is performed under the best conditions in terms of algorithm complexity ($D = 7$). In this case, the F_1 -score index is even higher when considering a lower

number of electrodes (the mean value of the F_1 -score with five sensors is higher than the one with six electrodes). Although the NLR is not considered as the ground truth for the prosthesis control with pattern recognition, the high values of F_1 -score confirm its suitability. The configuration with 5 electrodes, placed on Extensor Carpi Radialis Longus (EMG₀), Flexor Carpi Ulnaris (EMG₁), Extensor Digitorum (EMG₂), Flexor Carpi Radialis (EMG₃) and Extensor Carpi Ulnaris (EMG₄), is needed to reach acceptable performances.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
	2	85.0±10.1
	3	96.9±3.3
NLR _(D=7)	4	98.9±1.9
	5	99.9±0.1
	6	99.5±1.4

Table 9.1: F_1 -score of the NLR as the number of Ottobock electrodes varies

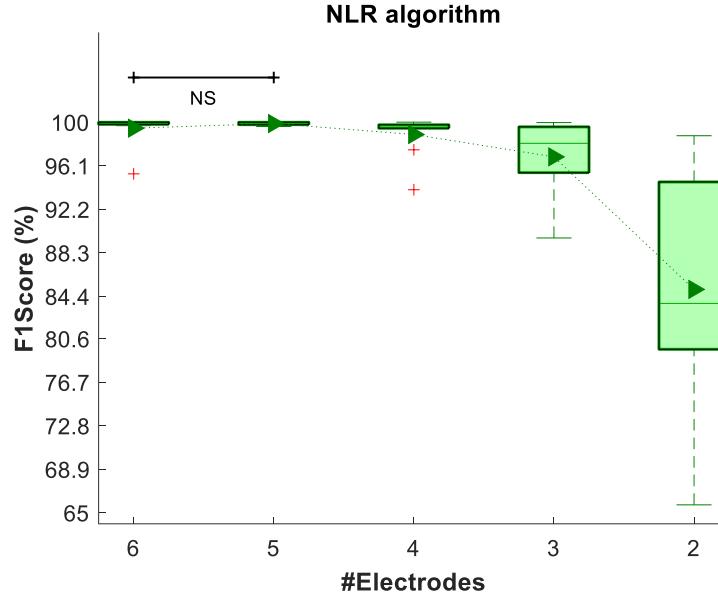
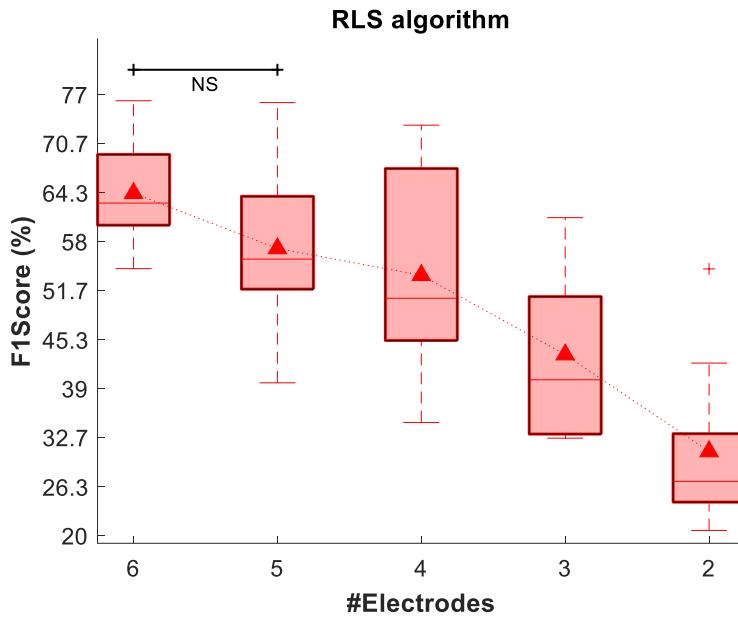


Figure 9.1: F_1 -score of the NLR as the number of Ottobock electrodes varies

The best quantity of MyoBock sensors for the RLS algorithm is 5 as shown in Table 9.2. The performances do not have relevant variations when using 5 or 6 electrodes, therefore the lowest number is considered. The performances of the RLS algorithm suffer a significant reduction when using one less electrode at a time. As a matter of fact, in the case of only two EMG sensors the prosthesis misbehaves.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
	2	30.9±10.3
	3	43.4±10.7
RLS	4	53.6±12.9
	5	57.1±9.7
	6	64.2±6.6

Table 9.2: F_1 -score of the RLS as the number of Ottobock electrodes varies

Figure 9.2: F₁-score of the RLS as the number of Ottobock electrodes varies

For the ANN algorithm, the best quantity of MyoBock electrodes is 4 as reported in Table 9.3. The performances do not have significant variations when using 4, 5 or 6 electrodes, hence the lowest number is considered. As mentioned in paragraph 7.3, the number of hidden layers ranges from 1 to 10, whilst the number of neurons for each hidden layer varies from 1 to 30. The ANN algorithm is evaluated with the highest complexity ($L = 10$ and $N = 30$) since in this preliminary phase the computational burden is not taken into account. Although the training is performed under the best conditions, the ANN algorithm is characterized by low F₁-score indexes. In the ANN case, the implemented configuration is characterized by 4 electrodes placed on Extensor Carpi Radialis Longus (EMG₀), Flexor Carpi Ulnaris (EMG₁), Flexor Carpi Radialis (EMG₃) and Extensor Carpi Ulnaris (EMG₄).

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F₁-score [%]</i>
ANN _(L=10,N=30)	2	53.8±2.7
	3	56.0±1.1
	4	56.7±1.2
	5	56.8±1.2
	6	56.7±1.0

Table 9.3: F₁-score of the ANN as the number of Ottobock electrodes varies

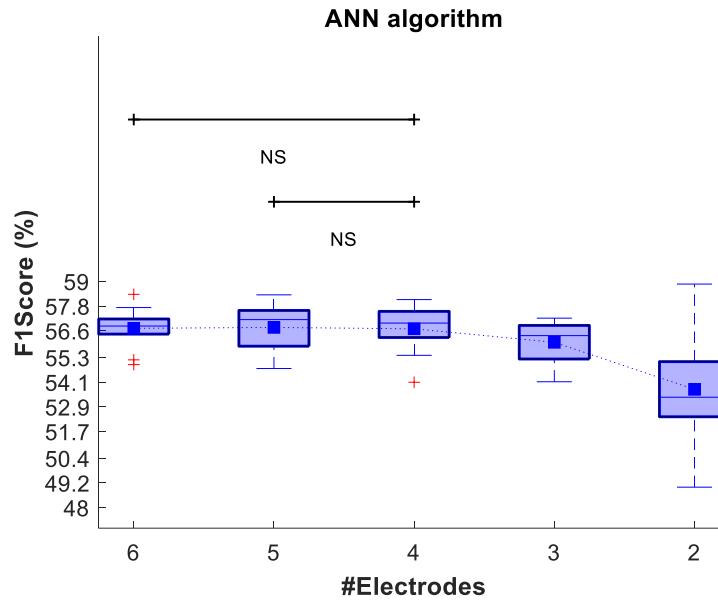


Figure 9.3: F_1 -score of the ANN as the number of Ottobock electrodes varies

For the SVM algorithm, the best quantity of MyoBock sensors is 4 as summarized in Table 9.4. The performances do not have relevant variations when using 4, 5 or 6 electrodes, thus the lowest number is considered. The SVM algorithm is characterized by low performances, as proven by its low F_1 -score indexes. In this case, the electrodes placement is the same as the ANN one, hence they are placed on Extensor Carpi Radialis Longus (EMG_0), Flexor Carpi Ulnaris (EMG_1), Flexor Carpi Radialis (EMG_3) and Extensor Carpi Ulnaris (EMG_4).

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
SVM	2	55.0±2.6
	3	58.0±1.2
	4	58.9±1.0
	5	59.1±0.9
	6	59.2±0.7

Table 9.4: F_1 -score of the SVM as the number of Ottobock electrodes varies

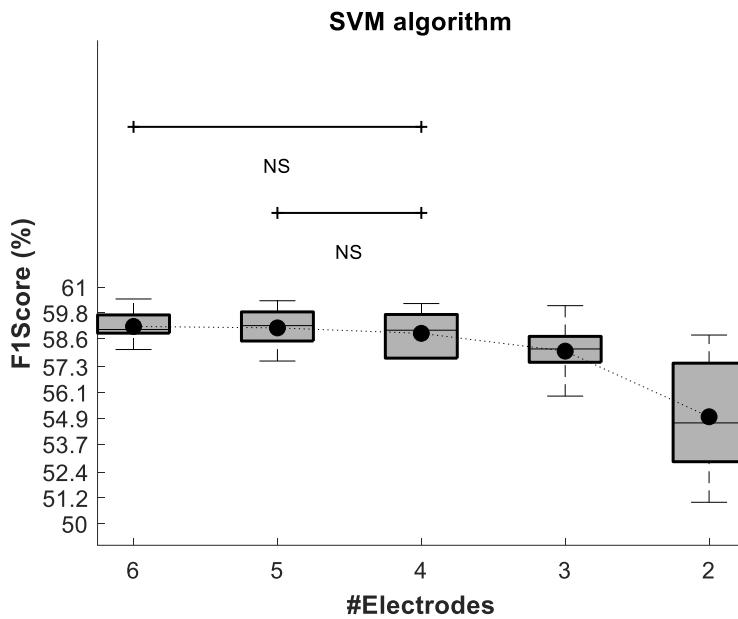


Figure 9.4: F₁-score of the SVM as the number of Ottobock electrodes varies

The best quantity of MyoBock sensors for the LDA algorithm is 5 as described in Table 9.5. In this case, the F₁-score indexes do not have important variations when using 5 or 6 electrodes, therefore the lowest quantity is considered. The LDA algorithm is the ground truth for the prosthesis control with pattern recognition, as demonstrated by its high F₁-score values. As for the NLR algorithm, in this case the configuration with 5 electrodes, placed on Extensor Carpi Radialis Longus (EMG₀), Flexor Carpi Ulnaris (EMG₁), Extensor Digitorum (EMG₂), Flexor Carpi Radialis (EMG₃) and Extensor Carpi Ulnaris (EMG₄), is needed to reach suitable performances.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F₁-score [%]</i>
LDA	2	81.2±11.1
	3	96.6±2.1
	4	98.7±0.9
	5	99.0±1.0
	6	99.5±0.5

Table 9.5: F₁-score of the LDA as the number of Ottobock electrodes varies

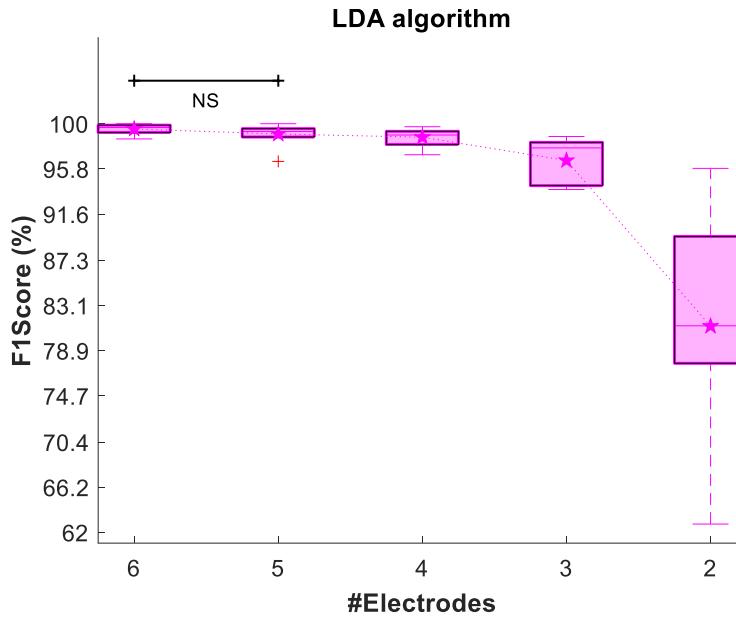
Figure 9.5: F_1 -score of the LDA as the number of Ottobock electrodes varies

Table 9.6 reports the best number of electrodes for each classifier obtained after the optimization process. The NLR and the LDA have higher F_1 -scores at the expense of higher number of electrodes. The RLS is the least suitable algorithm since it has high number of electrodes with low performances. Even though the ANN and SVM have lower number of electrodes, they are characterized by low F_1 -score indexes.

<i>Classifier</i>	<i>Number of MyoBock electrodes</i>
$NLR_{(D=7)}$	5
RLS	5
$ANN_{(L=10, N=30)}$	4
SVM	4
LDA	5

Table 9.6: Optimized number of MyoBock electrodes for different classifiers

9.2 Optimization of classifiers complexity

The second optimization phase starts from the already minimized number of electrodes which are defined in paragraph 9.1. Only the NLR and ANN classifiers are involved in the minimization of algorithm complexity. For the NLR, the value of the parameter D is swept from 1 to 7 in order to obtain the minimum polynomial degree which guarantees acceptable performances of the classifier. For the ANN, firstly, the number of hidden layers is minimized by considering the highest value of neurons (30). Then on the basis of the already minimized number of layers, the performances of the classifier are evaluated by varying the maximum number of neurons which is equal for each layer. The tested values of neurons are [1, 5, 10, 15, 20, 23, 24, 25, 26, 27, 28, 29, 30]. At the end of this process, the best configuration of the two classifiers will be defined in terms of the number of electrodes (NLR and ANN), polynomial degree (NLR), number of hidden layers (ANN) and number of neurons (ANN).

9.2.1 NLR: minimization of the D-value

As discussed in paragraph 7.1, the polynomial degree is changed in order to optimize the performances of the entire prosthetic device. The computational burden greatly affects the time needed to make the prosthesis move, so the goal is to reduce its effort as much as possible. The shorter the time delay between movement intention and prosthesis movement, the higher the naturalness of control. For the NLR, the EOF reaches satisfactory results starting from the polynomial degree D equal to 2. The Figure 9.6 reports the behavior of the EOF index when varying the polynomial degree of the algorithm. The NS whiskers represent the lack of statistical discrepancy in the results obtained with different D -values: [2, 3, 4, 5, 6, 7]. The linear algorithm ($D = 1$) clearly behaves worse than the others, as demonstrated by its lower EOF index.

<i>Polynomial degree</i>	<i>EOF [%]</i>
1	93.9±4.2
2	99.8±0.1
3	99.8±0.1
4	99.8±0.1
5	99.8±0.1
6	99.8±0.1
7	99.7±0.1

Table 9.7: Effect of the variation of polynomial degree on the EOF index

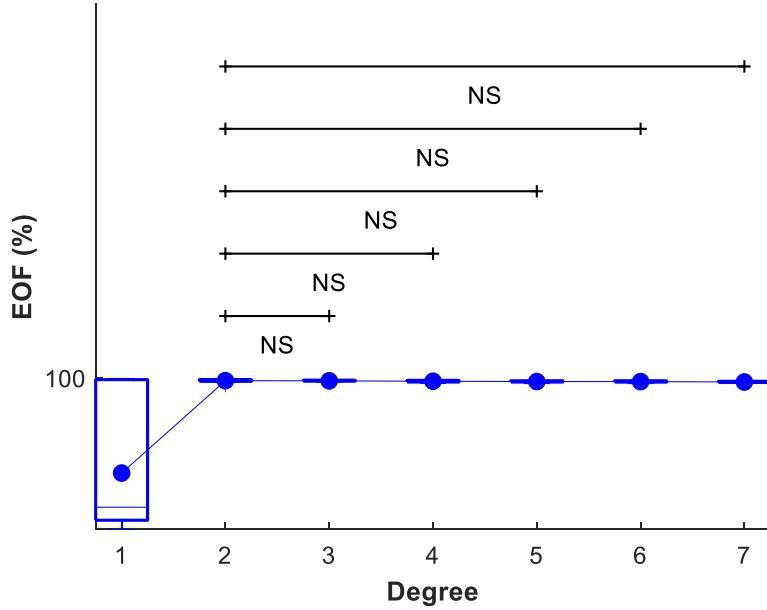


Figure 9.6: Effect of the variation of polynomial degree on the EOF index

9.2.2 ANN: minimization of the number of layers and neurons

As far as the ANN algorithm is concerned, the optimization process is performed in two sequential steps. The first

aims to minimize the number of hidden layers, and the second, on the basis of the previously optimized value, minimizes the number of neurons. For the MyoBock electrodes, the EOF indexes are evaluated to look for the best configuration in terms of computational burden. The minimum number of hidden layers, necessary to obtain admissible results, is 4, as shown in Table 9.8. The Figure 9.7 reports the EOF values that occur when considering different quantity of hidden layers.

<i>Number of Layers</i>	<i>EOF [%]</i>
1	71.8±1.6
2	71.7±0.9
3	71.7±0.8
4	71.7±1.0
5	71.3±0.9
6	71.5±0.9
7	71.5±1.2
8	70.7±1.0
9	71.2±1.1
10	71.0±1.4

Table 9.8: Effect of the variation in the number of layers on the EOF index

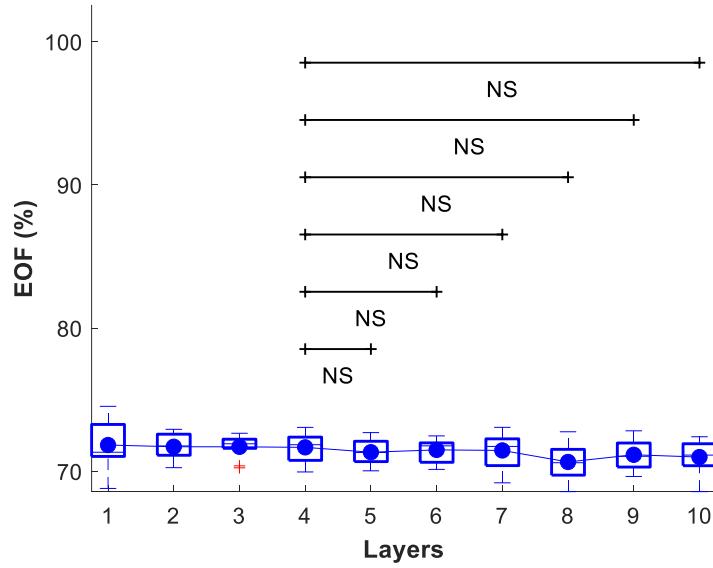


Figure 9.7: Effect of the variation in the number of layers on the EOF index

Once the first optimization has been completed, the variation of the neurons is studied. It is important to underline that the optimization of the number of neurons is not customized for each layer, but the same number is taken into account. As far as the neurons minimization is concerned, not all integer numbers between 1 and 30 are evaluated since there is no statistical difference between two contiguous values. As represented in Figure 9.8, the averages of EOF index for different numbers of neurons are not remarkably different, hence the lowest value is considered (as demonstrated also by the NS whiskers, the best solution is the one with only a neuron for each hidden layers).

<i>Number of Neurons</i>	<i>EOF [%]</i>
1	71.8±1.7
5	72.0±1.3
10	72.0±1.1
15	71.7±1.0
20	72.0±1.0
23	71.8±0.9
24	71.8±0.7
25	71.5±1.1
26	71.7±0.9
27	71.8±1.0
28	71.6±0.8
29	71.7±1.1
30	71.7±1.0

Table 9.9: Effect of the variation in the number of neurons on the EOF index

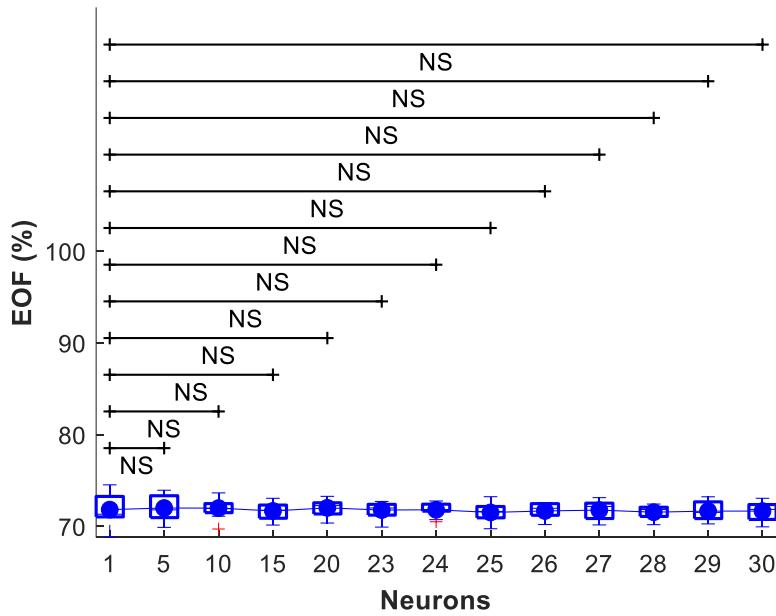


Figure 9.8: Effect of the variation in the number of neurons on the EOF index

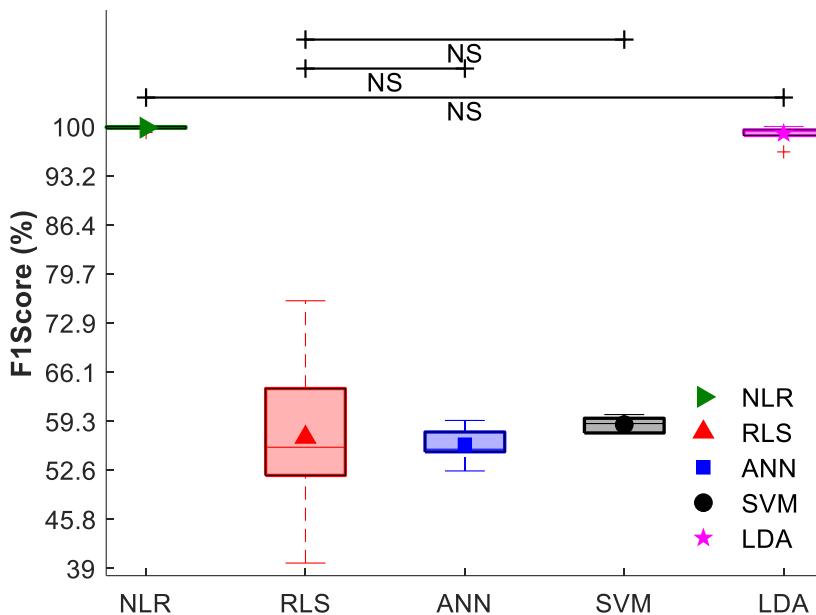
9.3 Comparative analysis between algorithms

The last part of the classifiers comparison is based on all the previous minimizations already discussed. The five algorithms are compared through the Wilcoxon-Signed-Rank test for a different number of electrodes. Table 9.10 reports the optimization settings and the corresponding relevant indexes for each classifier. The comparative analysis is conducted on the F_1 -score, the EOF, the classification and the abstention. Additionally, the performance scores are separately plotted from Figure 9.9 to Figure 9.12. Since the SVM and the ANN always have a class in output and they do not exploit any likelihood function, their abstention indexes are not computed. The results are summarized in Table 9.10, where the mean values and the standard deviations are highlighted.

Classifier	Number of MyoBock electrodes	F_1-score [%]	EOF [%]	Classification [%]	Abstention [%]
NLR _(D=2)	5	99.8±0.3	99.8±0.1	99.96±0.03	64.7±3.8
RLS	5	57.1±9.7	12.7±0.3	60.8±9.9	56.2±1.4
ANN _(L=4, N=1)	4	56.1±2.1	71.8±1.7	51.3±2.1	
SVM	4	58.9±1.0	62.1±1.6	54.5±0.9	
LDA	5	99.0±1.0	99.4±0.5	99.1±0.9	50.3±1.3

Table 9.10: Classifiers performance scores obtained with MyoBock sensors

The Figure 9.9 shows the performances of the classifiers in terms of F_1 -score. The NLR and LDA classifiers are characterized by high values of F_1 -score and also consistent results as indicated by small standard deviations. Moreover, the two algorithms do not have any statistical difference as demonstrated by NS whisker. The RLS classifier has low performances and also high variations in the results, as shown by the high standard deviation. The ANN and SVM are characterized by low F_1 -score but the results are more consistent than the RLS due to the small standard deviations. The NS whiskers underline the lack of statistical difference between RLS & ANN and RLS & SVM algorithms.

Figure 9.9: Algorithms comparison in terms of F_1 -score index

The Figure 9.10 represents the performances of the classifiers in terms of the Embedding Optimization Factor (EOF). As well as the F_1 -score case, the NLR and LDA classifiers are characterized by high values of EOF and also consistent results as indicated by small standard deviations. The RLS classifier has extremely low performances as demonstrated by its very low value of EOF. The ANN is characterized by better performances than the SVM classifier, although they have comparable standard deviations.

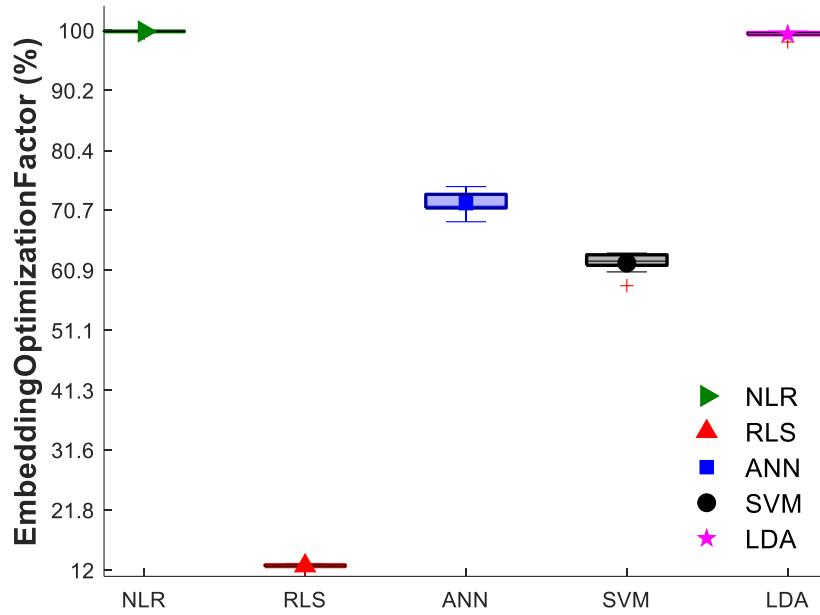


Figure 9.10: Algorithms comparison in terms of EOF index

The Figure 9.11 shows the accuracy of the classifiers in terms of classification index, hence the percentage of correctly predicted movements from all predictions made. As well as the F_1 -score and EOF cases, the NLR and LDA classifiers are characterized by high classification rates and also consistent results as indicated by small standard deviations. Conversely from the EOF cases, the RLS classifier has better performances than SVM and ANN but it has also more inconsistent results, as shown by its high standard deviation. The SVM is characterized by better performances and more consistent results than the ANN classifier.

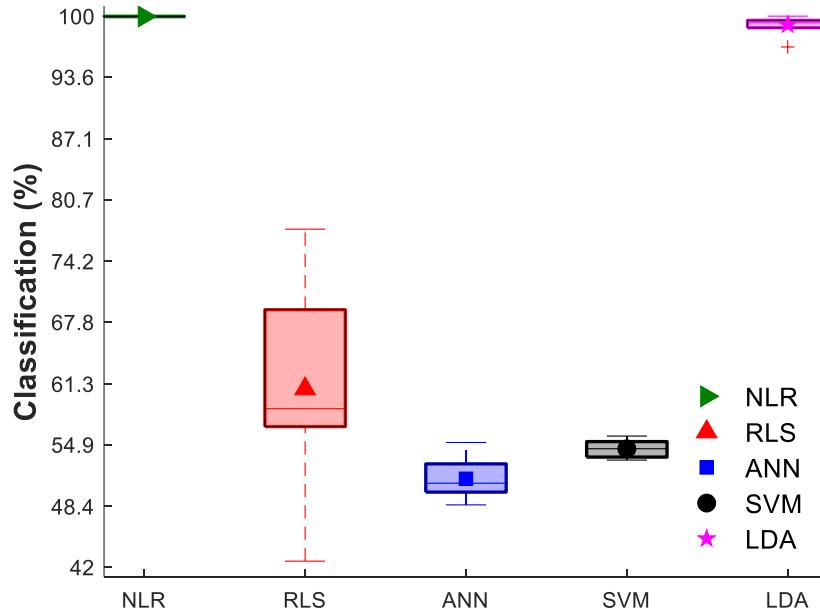


Figure 9.11: Algorithms comparison in terms of classification index

The Figure 9.12 represents the performances of the classifiers in terms of abstention index, thus the percentage of

unlabeled declarations. As already mentioned, ANN and SVM are not based on likelihood function, hence they do not choose to abstain from labeling inputs, therefore, their abstention rates are not computed. The NLR classifier is characterized by the highest abstention index and it could be considered an important drawback. The LDA has the lowest abstention index and also the better consistent results compared to NLR and RLS.

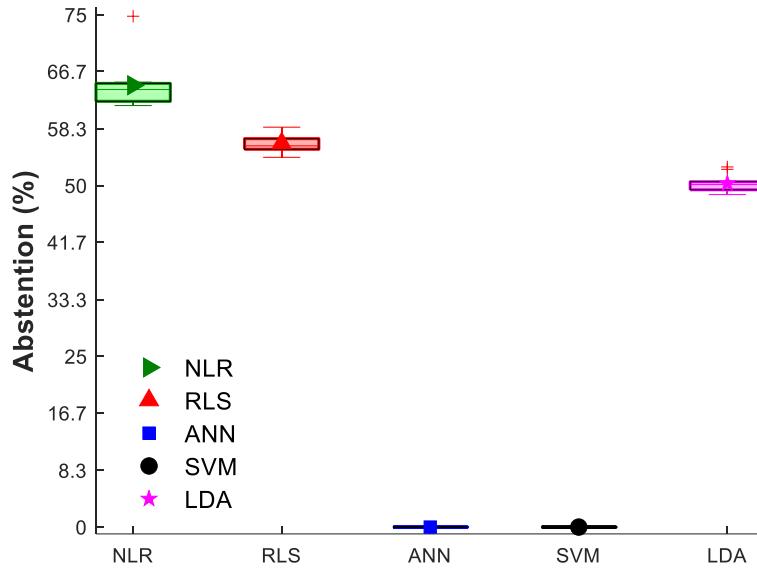


Figure 9.12: Algorithms comparison in terms of abstention index

9.4 Results on amputees whit MyoBock electrodes

Due to the low number of amputees involved in the study, no statistical analysis can be performed. Consequently, each dataset is exploited to create models of different algorithms based on parameters already minimized for able-bodied subjects. The optimized settings are used to build models for amputees, in order to evaluate whether these configurations have performances comparable to those obtained on healthy people. As already mentioned, the TR is used to build the algorithm model, while the TS is used to test the classifiers performances. In other terms, the results evaluated on TS represent an estimation of the classification ability when unknown EMG signals are supplied to the already built model. In the case of amputees, the NLR, RLS, ANN, SVM and LDA are trained with the minimum number of electrodes optimized for healthy people in terms of F_1 -score. Table 9.11 reports the performances of the five different classifiers in terms of F_1 -score, EOF, classification and abstention indexes when using the MyoBock sensors to acquire data on the stump of transradial amputees.

The potentiometer of the EMG electrodes is changed to increase/decrease the amplification of the myoelectric signals. By changing the position of the trimmer, the amount of resistance (R_{gain} in Figure 5.4) can be increased or decreased thereby controlling the amount of current. The variation of the resistance through the potentiometer aims at changing the gain of the instrumental amplifier circuit, as described in Eq. (10). Amputees people do not have high

muscular activity, therefore, the EMG signals must be amplified more than the able-bodied ones.

The performances of the classifiers obtained on the transradial amputees are not statistically analyzed since a small number of these subjects is involved in the experiments. Due to the lack of statistical studies, the results report just the performances of the classifiers separately for each amputee. This preliminary study can be further improved by optimizing all the parameters on datasets acquired from transradial subjects, in such a way to consider problems that do not occur with healthy individuals (i.e. low muscular activity, variation of the electrodes placement, difficulty in finding muscles, ...).

Subj.	Classifier	Number of electrodes	F1-score [%]	EOF [%]	Classification [%]	Abstention [%]
1	NLR _(D=2)	5	99.7	99.8	99.96	82.2
	RLS	5	41.0	12.2	43.9	52.7
	ANN _(L=4, N=1)	4	54.6	70.6	49.2	
	SVM	4	59.4	61.9	55.1	
	LDA	5	97.9	98.8	98.5	57.9
2	NLR _(D=2)	5	93.8	96.7	98.9	78.0
	RLS	5	32.4	11.7	34.3	56.0
	ANN _(L=4, N=1)	4	44.5	61.5	41.7	
	SVM	4	50.9	54.2	47.5	
	LDA	5	94.7	97.2	94.6	54.9
3	NLR _(D=2)	5	99.9	99.9	99.99	77.1
	RLS	5	37.4	12.0	42.6	57.2
	ANN _(L=4, N=1)	4	49.3	66.0	44.6	
	SVM	4	55.3	59.3	51.6	
	LDA	5	96.5	98.1	96.7	53.6
4	NLR _(D=2)	5	82.1	90.1	99.5	83.9
	RLS	5	27.6	11.4	39.8	65.1
	ANN _(L=4, N=1)	4	46.8	63.7	42.8	
	SVM	4	51.2	54.3	48.1	
	LDA	5	90.7	95.0	92.4	60.8

Table 9.11: Classifiers performance scores obtained with MyoBock sensors on amputees

10. Results with IIT sensors

The experimental procedure described in chapter 6 is used to obtain the results reported in this chapter, when the set of IIT sensors is exploited for the acquisition of the sEMG signals. It is important to underline that the models of the algorithms are built through the “Training Set” (TR), but the results are evaluated on the unknown data contained in the “Test Set” (TS). The two sets are obtained after the down-sampling technique, as described in paragraph 5.3.1.2.

The F_1 -score indexes, which are related to the performances, are used to optimize the number of IIT electrodes. The higher the F_1 -score, the better the performance. As soon as the number of electrodes is optimized, the obtained value is fixed for the subsequent operations. The further phase of optimization is related to the minimization of the classifier complexity. For the NLR algorithm, the polynomial degree (D) should be minimized. Considering that the classifier complexity is strictly related to the value of the variable D , the Embedding Optimization Factor (EOF) is used to search for the minimum degree of the polynomial. For the ANN algorithm, both the number of layers and the number of neurons should be optimized. As before, the minimization of the classifier complexity is based on the EOF index. Initially, the number of layers is reduced by considering the maximum number of neurons (30) as long as the performance degradation does not undergo important variations. The optimized value of the number of layers is fixed for the subsequent minimization of the number of neurons (per each layer). At the end of the sequential process of optimization, each classifier is characterized by its own setup with minimized parameters.

The comparative analysis between the five algorithms (paragraphs 10.3) is performed on:

- the classifiers performances evaluated in terms of F_1 -score;
- the computational burden evaluated in terms of EOF;
- the classification (percentage of correctly predicted movements: $nP + nN$ in Table 8.1);
- the abstention (percentage of not assigned movements).

10.1 Optimization of the number of electrodes

The first investigation concerns the minimization of the number of electrodes as long as the performances saturate and no statistical difference occurs. For this type of analysis the F_1 -score index is taken into account since it gives an idea of the classifiers behavior in different conditions. The low statistical difference in terms of performances are pointed out by the NS indicators. To have a reasonable performance degradation, the minimum amount of needed electrodes corresponds to the smallest values shown by the NS indicators. In this chapter all the tables contain a row in bold which indicates the best choice of number of electrodes for each classifier.

For the NLR algorithm, the best quantity of IIT sensors is 4 as summarized in Table 10.1. The statistical difference between F_1 -score results with 4, 5 or 6 electrodes is not relevant, therefore the lowest quantity of sensors is considered.

As mentioned in paragraph 7.1, the polynomial degree (D) can range from 1 to 7. During this preliminary phase of optimization, the computational burden is not taken into account, hence the training of the classifier is performed under the best conditions. As a consequence, the algorithm complexity is set as the highest possible value ($D = 7$). Although the NLR is not considered as the ground truth for the prosthesis control with pattern recognition, the high values of F_1 -score confirm its suitability. The configuration with 4 electrodes, placed on Extensor Carpi Radialis Longus (EMG₀), Flexor Carpi Ulnaris (EMG₁), Flexor Carpi Radialis (EMG₃) and Extensor Carpi Ulnaris (EMG₄), is needed to reach acceptable performances.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F₁-score [%]</i>
	2	82.4±12.9
	3	96.7±4.3
NLR _(D=7)	4	96.9±5.7
	5	99.8±0.5
	6	99.8±0.4

Table 10.1: F_1 -score of the NLR as the number of IIT electrodes varies

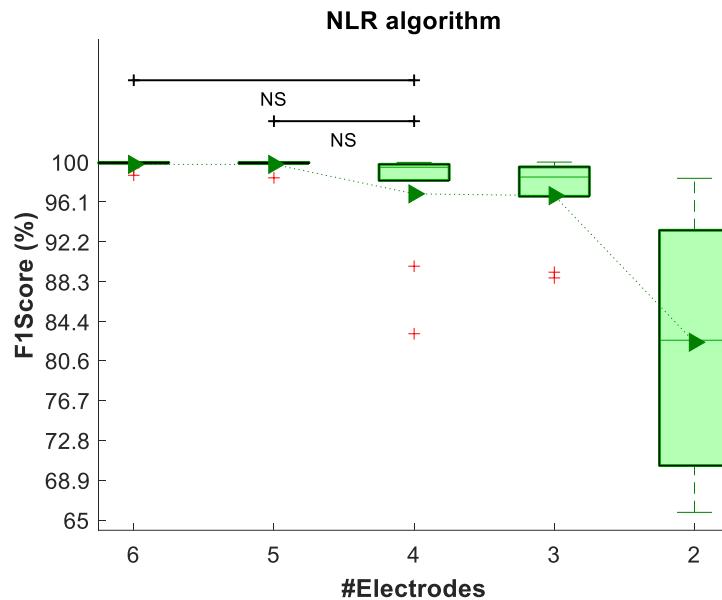
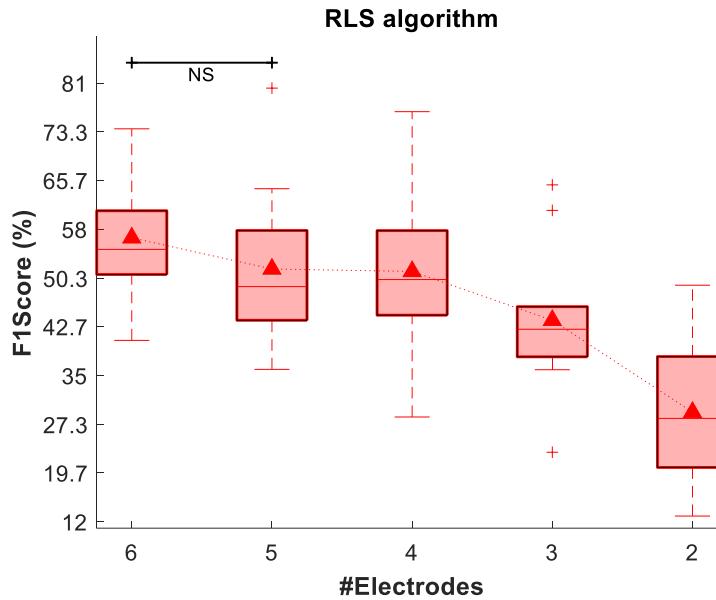


Figure 10.1: F_1 -score of the NLR as the number of IIT electrodes varies

For the RLS algorithm, the best quantity of IIT sensors is 5 as shown in Table 10.2. The NS whisker in Figure 10.2 shows that the performances do not have relevant variations when using 5 or 6 electrodes, therefore the lowest number is considered. The performances of the RLS algorithm suffer a significant reduction when using one less electrode at a time. As a matter of fact, in the case of only two EMG sensors the prosthesis misbehaves.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F₁-score [%]</i>
	2	29.2±11.2
	3	43.7±12.0
RLS	4	51.4±13.4
	5	51.8±12.9
	6	56.7±10.1

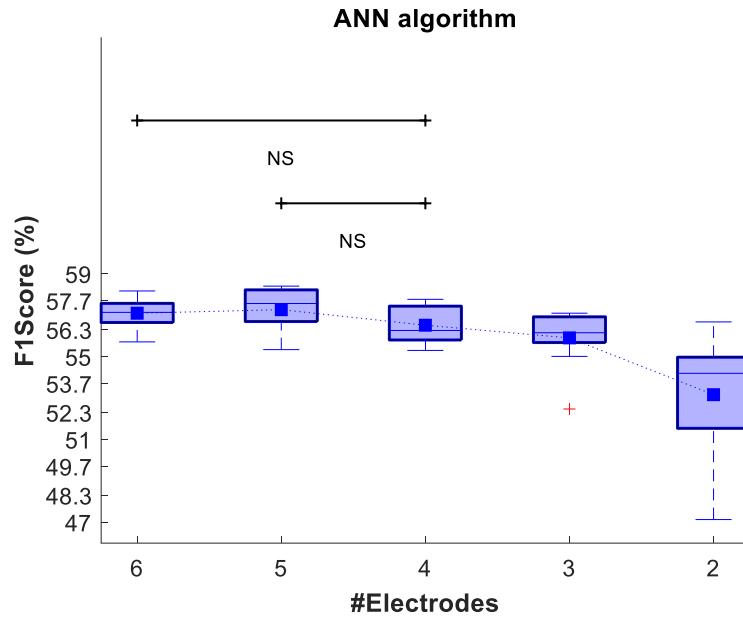
Table 10.2: F_1 -score of the RLS as the number of IIT electrodes varies

Figure 10.2: F_1 -score of the RLS as the number of IIT electrodes varies

The best quantity of IIT sensors for the ANN algorithm is equal to 4 as shown in Table 10.3. As mentioned in paragraph 7.3, the number of hidden layers goes from 1 to 10, whilst the number of neurons for each hidden layer varies from 1 to 30. The ANN algorithm is evaluated with the highest complexity ($L = 10$ and $N = 30$) since in this preliminary phase the computational burden is not taken into account. The NS whiskers in Figure 10.3 are the evidence that the performances do not have relevant variations when using 4, 5 or 6 electrodes, thus the lowest number is considered for the training of the classifier. Although the best conditions in terms of number of layers and neurons, the ANN algorithm is characterized by low F_1 -score indexes. In the ANN case, the implemented configuration is characterized by 4 electrodes placed on Extensor Carpi Radialis Longus (EMG_0), Flexor Carpi Ulnaris (EMG_1), Flexor Carpi Radialis (EMG_3) and Extensor Carpi Ulnaris (EMG_4).

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
	2	53.2 ± 3.0
	3	55.9 ± 1.4
$ANN_{(L=10,N=30)}$	4	56.5 ± 0.9
	5	57.3 ± 1.1
	6	57.1 ± 0.7

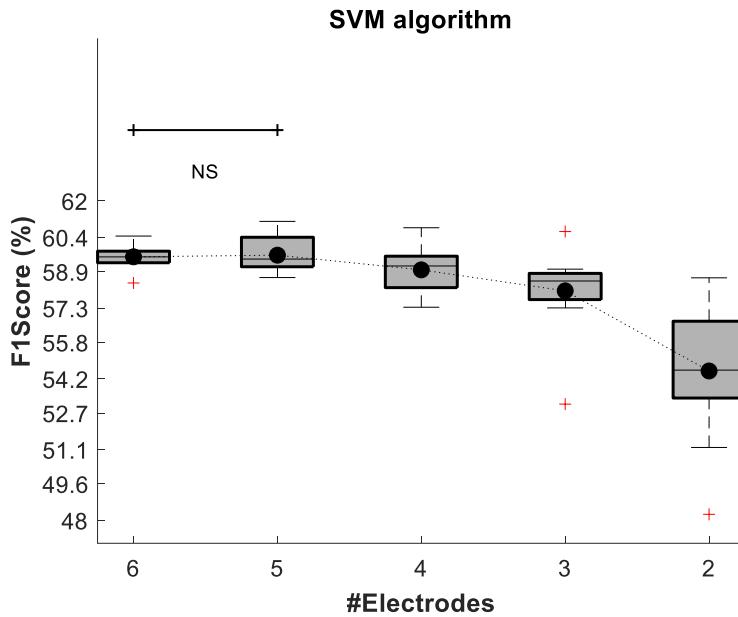
Table 10.3: F_1 -score of the ANN as the number of IIT electrodes varies

Figure 10.3: F_1 -score of the ANN as the number of IIT electrodes varies

The best quantity of IIT sensors for the SVM algorithm is 5 as summarized in Table 10.4. As shown by NS whisker in Figure 10.4, the performances do not have relevant variations when using 5 or 6 electrodes, thus the lowest number is considered for the training of the classifier. In this case, the SVM algorithm is characterized by low performances, as proven by its low F_1 -score indexes. In this case, the electrodes are placed on Extensor Carpi Radialis Longus (EMG_0), Flexor Carpi Ulnaris (EMG_1), Extensor Digitorum (EMG_2), Flexor Carpi Radialis (EMG_3) and Extensor Carpi Ulnaris (EMG_4).

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
SVM	2	54.6 ± 3.1
	3	58.1 ± 2.0
	4	59.0 ± 1.1
	5	59.6 ± 0.8
	6	59.6 ± 0.6

Table 10.4: F_1 -score of the SVM as the number of IIT electrodes varies

Figure 10.4: F_1 -score of the SVM as the number of IIT electrodes varies

For the LDA algorithm, the best quantity of IIT electrodes is 6 as described in Table 10.5. In this case, the results of the experiment outline the importance of all the EMG sensors to achieve acceptable performances in the classification. The LDA is the gold standard for the prosthesis control with pattern recognition, as demonstrated by its high F_1 -score values. The configuration with 6 electrodes, placed on Extensor Carpi Radialis Longus (EMG_0), Flexor Carpi Ulnaris (EMG_1), Extensor Digitorum (EMG_2), Flexor Carpi Radialis (EMG_3), Extensor Carpi Ulnaris (EMG_4) and Brachioradialis (EMG_5), is needed to reach suitable performances.

<i>Classifier</i>	<i>Number of electrodes</i>	<i>F_1-score [%]</i>
LDA	2	81.7 ± 14.7
	3	96.5 ± 2.2
	4	98.7 ± 0.8
	5	99.3 ± 0.5
	6	99.6 ± 0.3

Table 10.5: F_1 -score of the LDA as the number of IIT electrodes varies

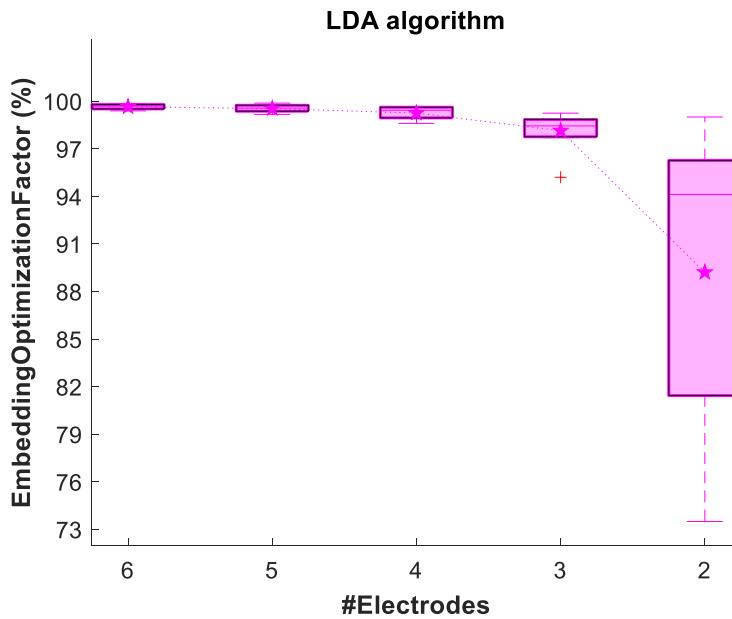
Figure 10.5: F₁-score of the LDA as the number of IIT electrodes varies

Table 10.6 reports the best number of IIT electrodes for each classifier obtained after the optimization process. The NLR and the LDA have high F₁-scores but the former relies on lower number of electrodes than the latter. The RLS and SVM are not particularly suitable as they have higher number of electrodes with respect to the NLR and also lower performances. Although the RLS and SVM have lower number of electrodes with respect to the LDA, their performances are worse. The performances of ANN algorithm are not high enough even though it has the same number of electrodes as the NLR.

Classifier	Number of IIT electrodes
NLR _(D=7)	4
RLS	5
ANN _(L=10, N=30)	4
SVM	5
LDA	6

Table 10.6: Optimized number of IIT electrodes for different classifiers

10.2 Optimization of classifiers complexity

The second optimization phase starts from the already minimized number of electrodes which are defined in paragraph 10.1. Only the NLR and ANN classifiers are involved in the minimization of algorithm complexity. For the NLR, the value of the parameter D is swept from 1 to 7 in order to obtain the minimum polynomial degree which guarantees acceptable performances of the classifier. For the ANN, firstly, the number of hidden layers is minimized by considering the highest value of neurons (30). Then on the basis of the already minimized number of layers, the performances of the classifier are evaluated by varying the maximum number of neurons which is equal for each layer. The tested values of neurons are [1, 5, 10, 15, 20, 23, 24, 25, 26, 27, 28, 29, 30]. At the end of this process, the best

configuration of the two classifiers will be defined in terms of the number of electrodes (NLR and ANN), polynomial degree (NLR), number of hidden layers (ANN) and number of neurons (ANN).

10.2.1 NLR: minimization of the D-value

As discussed in paragraph 7.1, the polynomial degree is changed in order to optimize the performances of the entire prosthetic device. The computational burden greatly affects the time needed to make the prosthesis move, so the goal is to reduce its effort as much as possible. The shorter the time delay between movement intention and prosthesis movement, the higher the naturalness of control. For the NLR, the EOF reaches satisfactory results starting from the polynomial degree D equal to 2. The Figure 10.6 reports the behavior of the EOF index when varying the polynomial degree of the algorithm. The NS whiskers represent the lack of statistical discrepancy in the results obtained with different D -values: [2, 3, 4, 5, 6, 7]. The linear algorithm ($D = 1$) clearly behaves worse than the others, as demonstrated by its lower EOF index.

<i>Polynomial degree</i>	<i>EOF [%]</i>
1	87.2±5.7
2	98.9±1.1
3	98.0±2.6
4	97.9±3.0
5	97.9±3.0
6	98.0±3.1
7	98.2±3.1

Table 10.7: Effect of the variation of polynomial degree on the EOF index

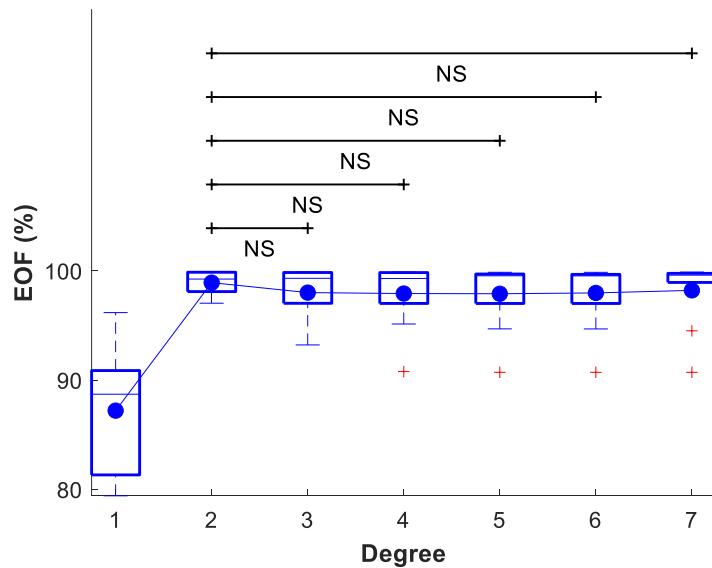


Figure 10.6: Effect of the variation of polynomial degree on the EOF index

10.2.2 ANN: minimization of the number of layers and neurons

As far as the ANN algorithm is concerned, the optimization process is performed in two sequential steps. The first aims to minimize the number of hidden layers, and the second, on the basis of the previously optimized value, minimizes the number of neurons. For the IIT electrodes, the EOF indexes are evaluated to look for the best configuration in terms of computational burden. Differently from the case of MyoBock electrodes, the minimum number of hidden layers, necessary to obtain admissible results, is 1, as shown in Table 10.8. The Figure 10.7 reports the EOF values computed when considering different quantity of hidden layers. As demonstrated by the NS whiskers, the lack of statistical discrepancy in the EOF indexes occurs, hence the minimum amount of needed hidden layers corresponds to the smallest value shown by such indicators.

<i>Number of Layers</i>	<i>EOF [%]</i>
1	71.9±2.4
2	71.6±0.8
3	71.7±0.9
4	71.6±0.6
5	71.4±0.5
6	71.2±0.8
7	71.5±1.1
8	71.2±0.9
9	71.3±0.9
10	71.2±1.1

Table 10.8: Effect of the variation in the number of layers on the EOF index

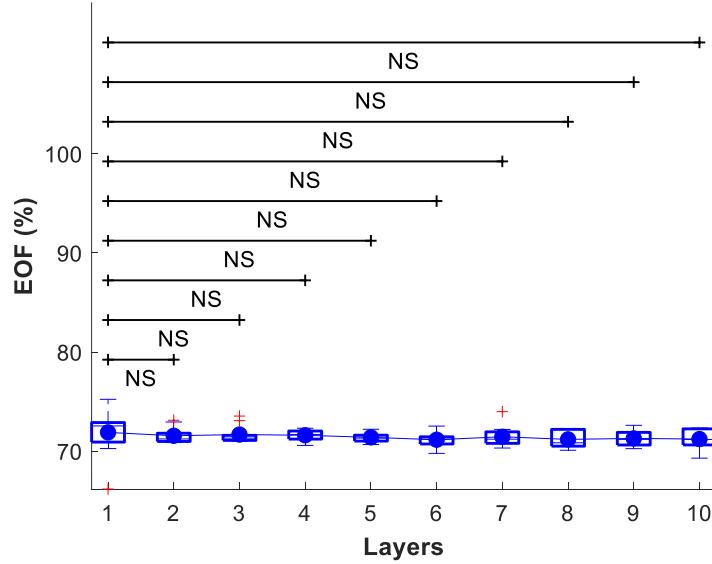


Figure 10.7: Effect of the variation in the number of layers on the EOF index

Once the first optimization has been completed, the variation of the neurons is studied. It is important to underline that the optimization of the number of neurons is not customized for each layer, but the same number is taken into account. As far as the neurons minimization is concerned, not all integer numbers between 1 and 30 are evaluated

since there is no statistical difference between two contiguous values. As represented in Figure 10.8, the averages of EOF indexes for different numbers of neurons are not remarkably different, hence the lowest value is considered (as demonstrated also by the NS whiskers, the best solution is the one with only a neuron for each hidden layers).

<i>Number of Neurons</i>	<i>EOF [%]</i>
1	71.9±2.4
5	71.9±2.4
10	71.9±2.4
15	71.9±2.4
20	71.9±2.4
23	71.9±2.4
24	71.9±2.4
25	71.9±2.4
26	71.9±2.4
27	71.9±2.4
28	71.9±2.4
29	71.9±2.4
30	71.9±2.4

Table 10.9: Effect of the variation in the number of neurons on the EOF index

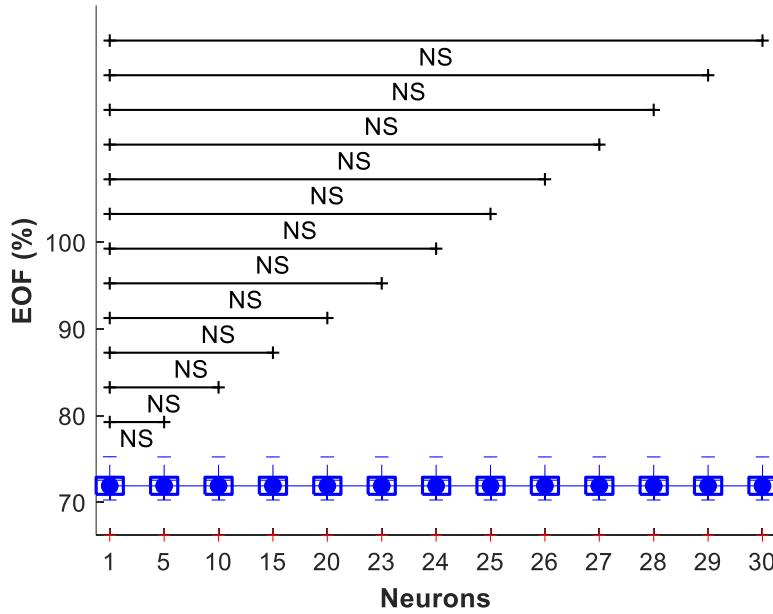


Figure 10.8: Effect of the variation in the number of neurons on the EOF index

10.3 Comparative analysis between algorithms

The last part of the classifiers comparison is based on all the previous minimizations already discussed. The five algorithms are compared through the Wilcoxon-Signed-Rank test for a different number of electrodes. Table 10.10 reports the optimization settings and the corresponding relevant indexes for each classifier. The comparative analysis is conducted on the F₁-score, the EOF, the classification and the abstention. Additionally, the performance scores are separately plotted from Figure 10.9 to Figure 10.12. Since the SVM and the ANN always have a class in output and

they do not exploit any likelihood function, their abstention indexes are not computed. The results are summarized in Table 10.10, where the mean values and the standard deviations are highlighted.

Classifier	Number of IIT electrodes	F_1 -score [%]	EOF [%]	Classification [%]	Abstention [%]
NLR _(D=2)	4	98.0±2.0	98.9±1.0	99.86±0.17	70.1±4.3
RLS	5	51.8±12.9	12.5±0.3	55.6±12.2	55.8±1.6
ANN _(L=1, N=1)	4	56.2±2.9	71.9±2.4	51.9±2.6	
SVM	5	59.6±0.8	60.9±1.5	55.4±0.7	
LDA	6	99.6±0.3	99.6±0.2	99.6±0.3	48.6±0.7

Table 10.10: Classifiers performance scores obtained with IIT sensors

The Figure 10.9 shows the performances of the classifiers in terms of F_1 -score. The NLR and LDA classifiers are characterized by high values of F_1 -score and also consistent results as indicated by small standard deviations. Moreover, the two algorithms do not have any statistical difference as demonstrated by NS whisker. The RLS classifier has low performances and also high variations in the results, as shown by the high standard deviation. The ANN and SVM are characterized by higher F_1 -score and more consistent results than the RLS, but the performances are still low for both. The NS whiskers underline the lack of statistical difference between RLS & ANN and RLS & SVM algorithms.

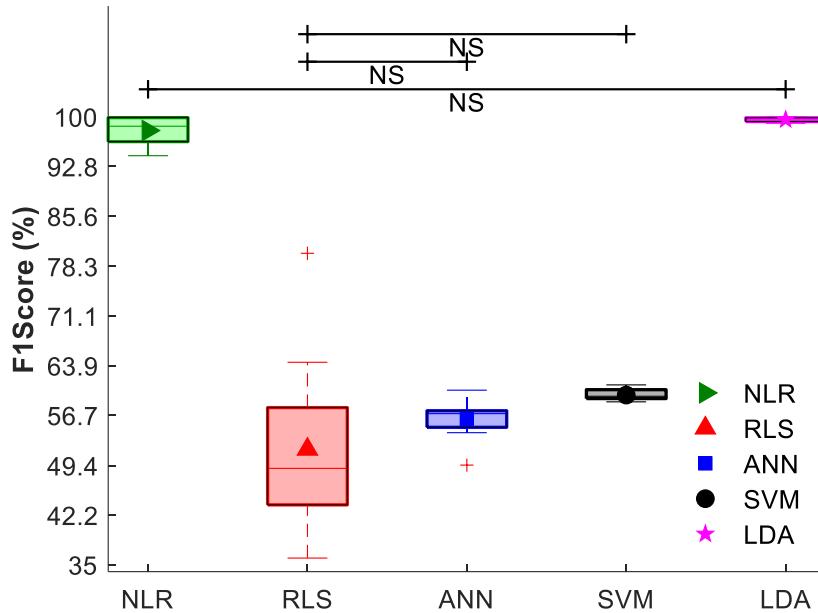


Figure 10.9: Algorithms comparison in terms of F_1 -score index

The Figure 10.10 represents the performances of the classifiers in terms of the Embedding Optimization Factor (EOF). As well as the F_1 -score case, the NLR and LDA classifiers are characterized by high values of EOF and also consistent results as indicated by small standard deviations. Moreover, the two algorithms do not have any statistical difference as demonstrated by NS whisker. The RLS classifier has extremely low performances as proven by its very low value of EOF. The ANN is characterized by better performances than the SVM classifier, although they have comparable standard deviations.

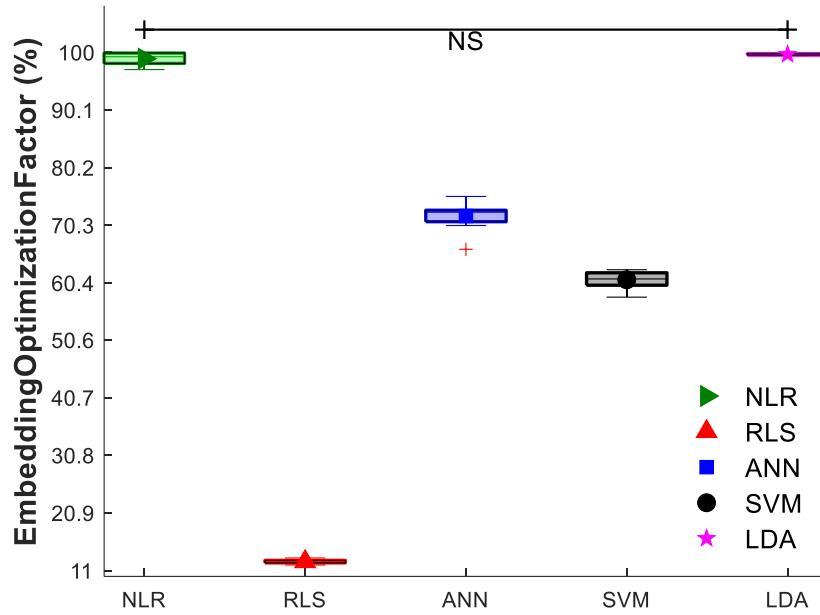


Figure 10.10: Algorithms comparison in terms of EOF index

The Figure 10.11 shows the accuracy of the classifiers in terms of classification index, hence the percentage of correctly predicted movements from all predictions made. As well as the F₁-score and EOF cases, the NLR and LDA classifiers are characterized by high classification rates and also consistent results as indicated by small standard deviations. Conversely from the EOF cases, the RLS classifier has better performances than ANN and comparable with SVM, but it has also more inconsistent results, as shown by its high standard deviation. The SVM is characterized by better performances and more consistent results than the ANN classifier.

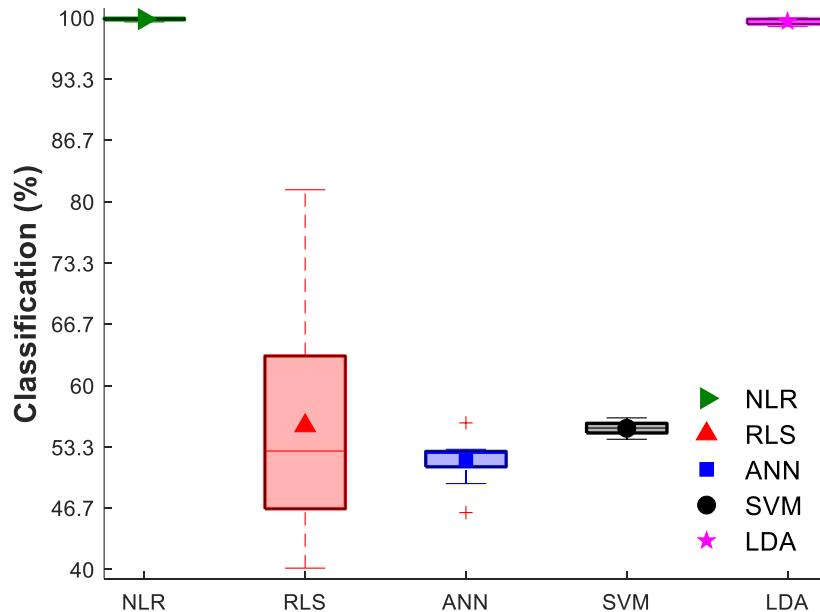


Figure 10.11: Algorithms comparison in terms of classification index

The Figure 10.12 represents the performances of the classifiers in terms of abstention index, thus the percentage of

unlabeled declarations. As already mentioned, ANN and SVM are not based on likelihood function, hence they do not choose to abstain from labeling inputs, therefore, their abstention rates are not computed. The NLR classifier is characterized by the highest abstention index and it could be considered an important drawback. The LDA has the lowest abstention index and also the better consistent results compared to NLR and RLS.

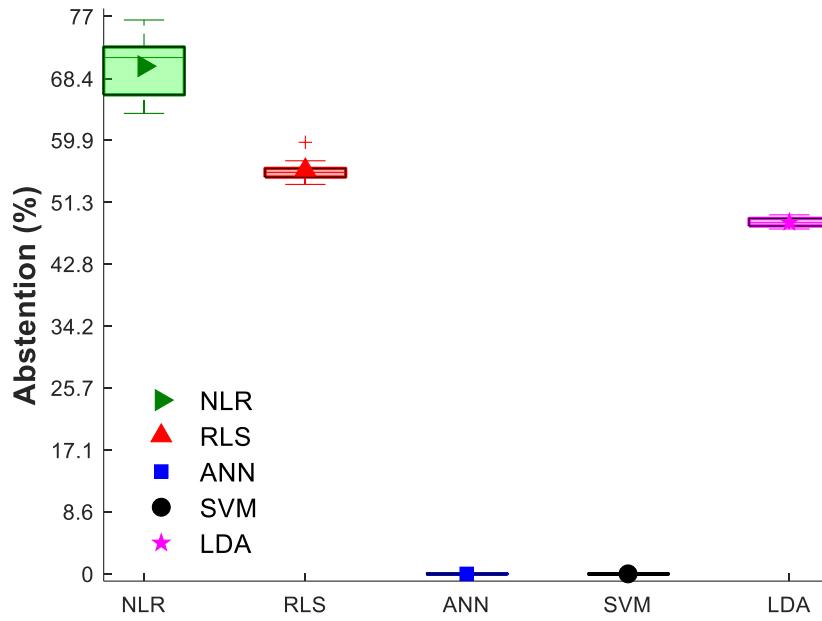


Figure 10.12: Algorithms comparison in terms of abstention index

10.4 Results on amputees whit IIT electrodes

Due to the low number of amputees involved in the study, no statistical analysis can be performed. Consequently, each dataset is exploited to create models of different algorithms based on parameters already minimized for able-bodied subjects. The optimized settings are used to build models for amputees, in order to evaluate whether these configurations have performances comparable to those obtained on healthy people. As already mentioned, the TR is used to build the algorithm model, while the TS is used to test the classifiers performances. In other terms, the results evaluated on TS represent an estimation of the classification ability when unknown EMG signals are supplied to the already built model. In the case of amputees, the NLR, RLS, ANN, SVM and LDA are trained with the minimum number of electrodes optimized for healthy people in terms of F_1 -score. Table 10.11 reports the performances of the five different classifiers in terms of F_1 -score, EOF, classification and abstention indexes when using the IIT sensors to acquire data on the stump of transradial amputees.

The potentiometer of the EMG electrodes is changed to increase/decrease the amplification of the myoelectric signals. By changing the position of the trimmer, the amount of resistance (R_{gain} in Figure 5.4) can be increased or decreased thereby controlling the amount of current. The variation of the resistance through the potentiometer aims at changing

the gain of the instrumental amplifier circuit, as described in Eq. (10). Amputees people do not have high muscular activity, therefore, the EMG signals must be amplified more than the able-bodied ones.

The performances of the classifiers obtained on the transradial amputees are not statistically analyzed since a small number of these subjects is involved in the experiments. Due to the lack of statistical studies, the results report just the performances of the classifiers separately for each amputee. This preliminary study can be further improved by optimizing all the parameters on datasets acquired from transradial subjects, in such a way to consider problems that do not occur with healthy individuals (i.e. low muscular activity, variation of the electrodes placement, difficulty in finding muscles, ...).

Subj.	Classifier	Number of electrodes	F_1-score [%]	EOF [%]	Classification [%]	Abstention [%]
1	NLR _(D=2)	4	80.2	88.9	99.3	86.2
	RLS	5	22.5	10.9	26.9	54.7
	ANN _(L=1, N=1)	4	46.3	63.3	45.6	
	SVM	5	54.9	56.5	50.7	
	LDA	6	96.8	98.2	96.9	52.7
2	NLR _(D=2)	4	85.6	92.2	99.9	82.5
	RLS	5	45.2	12.4	47.4	56.7
	ANN _(L=1, N=1)	4	42.3	59.5	43.4	
	SVM	5	55.1	58.3	54.6	
	LDA	6	91.5	95.4	93.3	52.0
3	NLR _(D=2)	4	95.3	97.5	99.9	82.8
	RLS	5	33.2	11.8	39.7	55.6
	ANN _(L=1, N=1)	4	48.2	65.1	44.7	
	SVM	5	52.5	54.2	48.5	
	LDA	6	87.3	93.1	89.3	53.8
4	NLR _(D=2)	4	82.7	90.4	99.7	83.3
	RLS	5	36.5	12.0	38.8	55.0
	ANN _(L=1, N=1)	4	51.5	68.0	49.7	
	SVM	5	58.9	58.0	54.8	
	LDA	6	94.9	97.2	95.7	51.5

Table 10.11: Classifiers performance scores obtained with IIT sensors on amputees

IV. Discussion

Several pattern recognition algorithms have been tested to evaluate the most suitable classifier for decoding human intentions when using a prosthetic device. The LDA is considered as the gold standard in myoelectric prosthesis control, therefore, all the others classifiers are compared with this ground truth. One of the advantages of the LDA algorithm is that it exploits features extraction as input, consequently, not only it does not need any down sampling process but it can also be trained in less time due to the absence of internal parameters to be optimized. The NLR, RLS, ANN and SVM algorithms are built through the training set (TR) while the performances are evaluated through the Test Set (TS). The analysis is performed on data acquired from both able-bodied subjects and amputees. For the former a statistical analysis is made, whilst for the latter only the performances are outlined due to the low number of samples. Although the number of recruited amputees is not sufficient to generalize the results obtained with healthy people, the thesis provides a significant contribution in finding a good compromise between performance and computational burden of machine learning algorithms for prosthetic applications. Two different types of electrodes are used for data acquisition (MyoBock and IIT) in order to evaluate how the prosthesis behaves in the use of one or the other.

The first minimization process aims to investigate the lowest number of electrodes that keeps acceptable performances. For the MyoBock sensors the optimized number of electrodes is always lower than six, which corresponds to the number of exploited sensors during the experiment. In this phase the maximum possible complexity of the algorithms is considered ($NLR_{(D=7)}$ and $ANN_{(L=10,N=30)}$). Although the $NLR_{(D=7)}$ and the LDA have higher number of electrodes than $ANN_{(L=10,N=30)}$ and SVM (Table 9.6), they are characterized by more consistent performances. Indeed, the F_1 -scores of $NLR_{(D=7)}$ ($99.9 \pm 0.1\%$) and LDA ($99.0 \pm 1.0\%$) far exceed those of RLS ($57.1 \pm 9.7\%$), SVM ($58.9 \pm 1.0\%$) and $ANN_{(L=10,N=30)}$ ($56.7 \pm 1.2\%$), which are approximately the same. Despite the high number of electrodes exploited by the RLS (5), this classifier is inappropriate for prosthetic applications. Even if the ANN is not characterized by high performances when considering the maximum possible complexity ($L = 10$ and $N = 30$), the number of layers and, subsequently, the number of neurons are optimized to reduce the computational burden required to the electronic board. This second minimization process consists in searching the best compromise in terms of number of hidden layers as long as the performances do not decrease. For the MyoBock electrodes, the best trade-off between performances and algorithm complexity is achieved for four hidden layers (EOF: $71.7 \pm 1.0\%$). A similar optimization process is implemented for the number of neurons, where all the configurations have no significant difference between them (Figure 9.8), therefore, the ANN with only one neuron is considered (EOF: $71.8 \pm 1.7\%$). For the configuration with only a neuron for each of the four hidden layers, the performances in terms of F_1 -score remain low (F_1 -score: $56.1 \pm 2.1\%$), as reported in Table 9.10. Right after, the study is focused on NLR algorithm to reduce as much as possible its complexity, thus the polynomial degree (D) is minimized. The Table 9.7 reports no significant difference in the EOF indexes obtained with different D -values: [2, 3, 4, 5, 6, 7], whereas the linear algorithm ($D = 1$) clearly

behaves worse than the others. Therefore, the optimal polynomial degree is equal to two (EOF: $99.8 \pm 0.1\%$). As soon as the optimization process is completed, all the classifiers are compared to each other in terms of four different indexes: F_1 -score, EOF, classification and abstention. As far as the F_1 -score is concerned, the NLR ($99.8 \pm 0.3\%$) is the most suitable algorithm since it has the highest performances immediately followed by the LDA ($99.0 \pm 1.0\%$). The SVM ($58.9 \pm 1.0\%$), RLS ($57.1 \pm 9.7\%$) and ANN ($56.1 \pm 2.1\%$) behave worse than the two just mentioned classifiers. Regarding the EOF index, the situation is similar to the previous case. The NLR ($99.8 \pm 0.1\%$) achieves the best performances followed by the LDA ($99.4 \pm 0.5\%$). The ANN ($71.8 \pm 1.7\%$), SVM ($62.1 \pm 1.6\%$) and RLS ($12.7 \pm 0.3\%$) have worse behavior than the LDA and NLR. The classification index is an indicator of the algorithm accuracy since it is computed on the correctly predicted movements. It is a further confirmation of the NLR ($99.96 \pm 0.03\%$) suitability due to its high percentage value. The ground truth LDA ($99.1 \pm 0.9\%$) has better performances with respect to the RLS ($60.8 \pm 9.9\%$), SVM ($54.5 \pm 0.9\%$) and ANN ($51.3 \pm 2.1\%$). Up to this point the NLR is the classifier that has achieved the best results, but, as regards the abstention index the situation is quite different. The NLR has the highest percentage of not assigned movements ($64.7 \pm 3.8\%$) followed by the RLS ($56.2 \pm 1.4\%$), while the LDA ($50.3 \pm 1.3\%$) behaves better than the others. It can be considered an important problem of the NLR algorithm but this apparent weakness is counterbalanced by the high classification frequency. The alleged downside of the NLR turns out to be undetectable by the end-users, as confirmed by experiments on amputees. Since the NLR and LDA have the same number of MyoBock electrodes (5), the former seems to be more suitable than the latter due to its higher performance scores.

The IIT electrodes are used to perform the same comparative analysis as that done for the Ottobock sensors to evaluate any discrepancies between the two. The first minimization process aims to investigate the lowest number of electrodes that keeps acceptable performances. In this phase the maximum possible complexity of the algorithms is considered ($NLR_{(D=7)}$ and $ANN_{(L=10,N=30)}$). Conversely from the analysis made for MyoBock case, different number of electrodes are identified as optimal for the $NLR_{(D=7)}$ (4), SVM (5) and LDA (6), while the $ANN_{(L=10,N=30)}$ (4) and RLS (5) have kept the same value. Although the LDA has the highest number of electrodes (6), it is characterized by more consistent performances in terms of F_1 -score than SVM ($59.6 \pm 0.8\%$), $ANN_{(L=10,N=30)}$ ($56.5 \pm 0.9\%$) and RLS ($51.8 \pm 12.9\%$). Despite the $NLR_{(D=7)}$ ($96.9 \pm 5.7\%$) has a slightly smaller F_1 -score index than the ground truth (LDA: $99.6 \pm 0.3\%$), the number of exploited electrodes (4) is significantly lower. Even if the ANN is not characterized by high performances when considering the maximum possible complexity ($L = 10$ and $N = 30$), the number of layers and, subsequently, the number of neurons are optimized to reduce the computational burden required to the electronic board. This second minimization process consists in searching the best compromise in terms of number of hidden layers as long as the performances do not decrease. For the IIT electrodes, the best trade-off between performances and algorithm complexity is achieved for only one hidden layer (EOF: $71.9 \pm 2.4\%$). A similar optimization process is implemented for the number of neurons, where all the configurations have no significant difference between them (Figure 10.8), therefore, the ANN with only one neuron is considered (EOF: $71.9 \pm 2.4\%$). For the configuration with only one neuron and one hidden layer, the performances in terms of F_1 -score remain low (F_1 -score: $56.2 \pm 2.9\%$), as reported in Table 10.10. Right after, the study is focused on NLR algorithm to reduce as much as possible its complexity, thus the polynomial degree (D) is minimized. The Table 10.7 reports no significant difference in the EOF indexes obtained with different D -values: [2, 3, 4, 5, 6, 7], whereas the linear algorithm ($D = 1$) clearly behaves worse than the others.

Therefore, the optimal polynomial degree is equal to two (EOF: $98.9\pm1.1\%$). As soon as the optimization process is completed, all the classifiers are compared to each other in terms of four different indexes: F_1 -score, EOF, classification and abstention. As far as the F_1 -score is concerned, the LDA ($99.6\pm0.3\%$) is the most suitable algorithm since it has the highest performances followed by the NLR ($98.0\pm2.0\%$). The SVM ($59.6\pm0.8\%$), ANN ($56.2\pm2.9\%$) and RLS ($51.8\pm12.9\%$) behave worse than the two just mentioned classifiers. Regarding the EOF index, the situation is similar to the previous one. The LDA ($99.6\pm0.2\%$) achieves the best performances followed by the NLR ($98.9\pm1.0\%$). The ANN ($71.9\pm2.4\%$), SVM ($60.9\pm1.5\%$) and RLS ($12.5\pm0.3\%$) have worse behavior than the LDA and NLR. The classification index is an indicator of the algorithm accuracy since it is computed on the correctly predicted movements. In this case the NLR ($99.86\pm0.17\%$) is more accurate than the LDA ($99.6\pm0.3\%$). The ground truth has better performances with respect to the RLS ($55.6\pm12.2\%$), SVM ($55.4\pm0.7\%$) and ANN ($51.9\pm2.6\%$). As regards the abstention index the NLR has the highest percentage of not assigned movements ($70.1\pm4.3\%$) followed by the RLS ($55.8\pm1.6\%$), while the LDA ($48.6\pm0.7\%$) behaves better than the others. It can be considered an important problem of the NLR algorithm but this apparent weakness is counterbalanced by the high classification frequency. The alleged downside of the NLR turns out to be undetectable by the end-users, as confirmed by experiments on amputees. Sometimes, the abstention is an important tool exploited to avoid misclassifications of unintended movements or artifacts. Despite the slightly lower performance scores, the NLR has a lower number of IIT electrodes (4) than the LDA (6), thus the former seems to be more suitable than the latter for the control of prosthetic devices.

Small differences appear between the results obtained with MyoBock electrodes and those obtained through IIT electrodes. The general behavior of the system is almost the same, the NLR and LDA are the most suitable classifiers among the five studied, hence they are the only algorithms considered in the next part of discussion. [33] obtained better performances for SVM, ANN and RLS classifiers due to the fewer movements to be identified considering the same amount of input information. As demonstrated by the results obtained in this thesis, the SVM, ANN, and RLS are not suitable for the control of *Hannes* system with 3-DoF. As far as the MyoBock electrodes are concerned, the same number of sensors are needed to obtain satisfactory results in term of performances. On the other hand, the IIT case requires different numbers of sensors for NLR and LDA. The number of electrodes is a crucial aspect in the upper limb prosthesis since they play an important role in the battery consumption, the computational burden of the electronic board, the mechanical encumbrance in the socket and also prosthesis costs. As a consequence, it could be a reasonable choice the use of the *Hannes* system with four IIT electrodes and the $NLR_{(D=2)}$ algorithm at the expense of a small reduction in the performances. Although the decrease in the number of electrodes causes a loss of information, the classification of NLR algorithm with only four sEMG sensors is still high, which means that the information carried by the muscular signals is still available. The study developed in the thesis shows that the nonlinear classification algorithms can be used in embedded applications of prosthesis devices with lower number of electrodes. In addition, a possible drawback of the LDA is that it relies on the features extraction from the sEMG signals, thus a possible delay in the classification process may occur.

V. Conclusion

The main goal of the thesis is to demonstrate the possibility of controlling upper limb prosthesis, with 3-DoF, through machine learning algorithms based on pattern recognition. The prosthetic device under consideration is *Hannes*, a poly-articulated hand able to restore over the 90% of lost functionality in people with transradial amputation. The five compared classifiers are the Non-linear Logistic Regression (NLR), Regularized Least-Squares (RLS), Artificial Neural Network (ANN), Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA), which is also considered as the ground truth. In the study are involved 10 healthy subjects and four amputees. The statistical analysis is performed on data acquired from able-bodied subjects, while the data obtained from amputees have been used for validation purposes only. The statistical analysis on amputees is not performed due to the small population involved in the experiment, but their contribution was fundamental to observe whether the classifier was able to behave correctly. The results of the experimental study highlight better behavior of two algorithms than the others. In particular, the two classifiers under consideration are the NLR and the LDA, characterized by high performances in terms of F_1 -score, EOF and classification. Even if the statistical comparison is carried out on healthy people, the thesis provides useful insights into the suitability of control strategies for upper limb prosthesis used by amputees. Although the control is still not smooth and fast, during the experimental tests, the amputees were astonished by this innovative method of translating human intentions into prosthesis movements.

As remarked by results obtained with both MyoBock and IIT electrodes, the NLR algorithm is the best choice since it guarantees high accuracy and, at the same time, low complexity. Similarly, [33] obtained that the NLR algorithm performed better than other classifiers when only two DoF were actuated. In the case of that study, the needed number of electrodes was three, due to the fewer classes to be identified. On the basis of the same information (six EMG sensors), the NLR algorithm implemented during this thesis had to be able to recognize two more movements than [33]. The greater the movements to recognize, the greater the difficulty of the algorithm, the greater the amount of needed information. As a consequence, the optimized number of electrodes, in the case of *Hannes* system control, is higher than that obtained by [33]. Even if the difficulty is increased, the performances in terms of F_1 -score, classification, and abstention remain almost unchanged. Therefore, during the trials the NLR is the algorithm trained for the online control of both the *Hannes* system and the VR. The polynomial degree of NLR algorithm ($D = 2$) remarkably reduces the computational burden and it has also a positive effects on the time delay between the movement intention and the actual prosthesis movement. The abstention is often implemented for reducing misclassifications, but the downside is that it leads to higher delays in the control. The NLR has the highest percentage of abstentions, however, this apparent weakness is counterbalanced by the high classification frequency. As a matter of fact, the subjects do not feel any delay in the use of the prosthesis.

To increase the accuracy of the classifiers, a future study can be carried out on an entire population of transradial amputees. Hence the internal parameters of the algorithms and the number of electrodes can be optimized directly on

the end-users of the prosthetic devices. Results of a such a study can be crucial to further improve *Hannes* system control through the pattern recognition. With the aim of making the experiments repeatable under the same conditions for each subject, a special tool can be exploited to search for the needed muscles on the stump. The identification of muscles by touch is very challenging on the amputees, since the residual muscles are not always anchored in the same way. This electronic tool could be useful during the electrodes placement as it is able to detect the muscle target through the EMG signals. It may be valuable to consider a custom amplification for each EMG sensor on the basis of the single amputee. As already mentioned, the muscle fibers do not have the same position and direction in the stump for all the subjects with transradial amputation as this is strictly dependent from the surgical reconstruction.

The main drawback of pattern recognition is its strong dependence from the training phase. The behavior of the classifier is related to the electrodes placement and the corresponding detected data. Changes in sensors positions after the algorithm calibration can lead to idleness of the classifier. If the sensors acquire different signals from those of the training phase, the classifier is no longer able to match these muscular patterns to the relative movement. Therefore, the donning and doffing of the prosthesis socket can be a huge problem for the control through machine learning algorithms, as it is difficult to place the electrodes always in the same positions. The first solution is the training of the algorithm every time that the amputee wears the prosthesis, but the negative aspect is that the training process lasts about 40 minutes. As a consequence, this solution is not feasible at all. Future development should aim to improve the robustness of the pattern recognition. A possible technique can rely on high density EMG sensors, which are matrices of flexible electrodes with 64 channels (8 electrodes for each row and interelectrode distance of 1cm). It is important to remark that not all the 64 channels are exploited for the classifier training otherwise an excessive and unnecessary computational effort is required to the EMG-Master. The training phase is always implemented with low number of electrodes (up to six) in order to reduce the amount of information provided to the classifier and, as a consequence, its computational burden. Once the amputee has donned the prosthesis, the high-density sensors can be exploited to identify the electrodes placed on the muscles involved during the training phase, and use them to acquire data for the prosthesis control. Therefore, the six electrodes exploited for the training phase are not always the one used for the online control. As implemented during the thesis, the number of exploited electrodes can be minimized in order to reduce both the computational burden required to the electronics components and also the time delay needed for the prosthesis control.

Another future development to increase the robustness of the machine learning classifiers rely on data fusion technique, where the information from muscular activity can be supported by those from additional sensors. One of the possible implementation is the IMU sensor, which is able to provide the attitude and configuration of the prosthesis during the activities of daily living (ADLs). The information of the IMU sensors can help in the dynamic adaptation of the prosthesis depending on the orientation of the device in the space.

The thesis is focused only on the afferent nervous system and, currently, there are no implementations of haptic feedback on *Hannes* system. Feedback sensations play an important role in ADLs as grasping, cooking, washing, or detecting shape, texture and compliance of the objects. Future developments can be focused on the efferent nervous system to provide touch, pressure, and temperature sensations to the amputee.

VI. Bibliography

- [1] I. Kyranou, S. Vijayakumar, and M. S. Erden, "Causes of Performance Degradation in Non-invasive Electromyographic Pattern Recognition in Upper Limb Prostheses," (in English), *Frontiers in Neurorobotics*, Review vol. 12, no. 58, 2018-September-21 2018, doi: 10.3389/fnbot.2018.00058.
- [2] I. Vujaklija, D. Farina, and O. C. Aszmann, "New developments in prosthetic arm systems," *Orthopedic research and reviews*, vol. 8, p. 31, 2016.
- [3] M. A. Powell, R. R. Kaliki, and N. V. Thakor, "User training for pattern recognition-based myoelectric prostheses: Improving phantom limb movement consistency and distinguishability," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 522-532, 2013.
- [4] A. D. Roche, B. Lakey, I. Mendez, I. Vujaklija, D. Farina, and O. C. Aszmann, "Clinical perspectives in upper limb prostheses: An update," *Current Surgery Reports*, vol. 7, no. 3, p. 5, 2019.
- [5] M. Laffranchi *et al.*, "The Hannes hand prosthesis replicates the key biological properties of the human hand," *Science Robotics*, vol. 5, no. 46, p. eabb0467, 2020, doi: 10.1126/scirobotics.abb0467.
- [6] C. Castellini, "UPPER LIMB ACTIVE PROSTHETICS: AN OVERVIEW," 2019.
- [7] R. Scott, "Myoelectric control of prostheses: A brief history," in *Proceedings of the 1992 MyoElectric Controls/Powered Prosthetics Symposium, Fredericton, NB, Canada*, 1992, vol. 1.
- [8] H. Flor, "Phantom-limb pain: characteristics, causes, and treatment," *The Lancet Neurology*, vol. 1, no. 3, pp. 182-189, 2002.
- [9] C. Behrend, W. Reizner, J. A. Marchessault, and W. C. Hammert, "Update on advances in upper extremity prosthetics," *The Journal of hand surgery*, vol. 36, no. 10, pp. 1711-1717, 2011.
- [10] D. R. Merrill, J. Lockhart, P. R. Troyk, R. F. Weir, and D. L. Hankin, "Development of an implantable myoelectric sensor for advanced prosthesis control," *Artificial organs*, vol. 35, no. 3, pp. 249-252, 2011.
- [11] M. Hichert, A. N. Vardy, and D. Plettenburg, "Fatigue-free operation of most body-powered prostheses not feasible for majority of users with trans-radial deficiency," *Prosthetics and orthotics international*, vol. 42, no. 1, pp. 84-92, 2018.
- [12] D. S. Childress, "Historical aspects of powered limb prostheses," *Clin Prosthet Orthot*, vol. 9, no. 1, pp. 2-13, 1985.
- [13] F. Cordella *et al.*, "Literature review on needs of upper limb prosthesis users," *Frontiers in neuroscience*, vol. 10, p. 209, 2016.
- [14] P. P. Newman, "Skeletal Muscle Tone and Posture," in *Neurophysiology*. Dordrecht: Springer Netherlands, 1980, pp. 227-237.
- [15] G. Khedkar, "Skeletal Muscle," 2016.
- [16] M. Ortiz-Catalan, R. Bränemark, B. Häkansson, and J. Delbeke, "On the viability of implantable electrodes for the natural control of artificial limbs: review and discussion," *Biomedical engineering online*, vol. 11, no. 1, pp. 1-24, 2012.
- [17] J. E. Cheesborough, L. H. Smith, T. A. Kuiken, and G. A. Dumanian, "Targeted muscle reinnervation and advanced prosthetic arms," in *Seminars in plastic surgery*, 2015, vol. 29, no. 1: Thieme Medical Publishers, p. 62.
- [18] A. D. Bellingegni *et al.*, "NLR, MLP, SVM, and LDA: a comparative analysis on EMG data from people with trans-radial amputation," *Journal of neuroengineering and rehabilitation*, vol. 14, no.

- 1, p. 82, 2017.
- [19] A. R. Tilley, *The measure of man and woman: human factors in design*. John Wiley & Sons, 2001.
- [20] M. Z. Jamal, "Signal acquisition using surface EMG and circuit design considerations for robotic prosthesis," *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*, vol. 18, pp. 427-448, 2012.
- [21] M. C. Garcia and T. Vieira, "Surface electromyography: Why, when and how to use it," *Revista andaluza de medicina del deporte*, vol. 4, no. 1, pp. 17-28, 2011.
- [22] A. K. Blangsted, G. Sjøgaard, P. Madeleine, H. B. Olsen, and K. Søgaard, "Voluntary low-force contraction elicits prolonged low-frequency fatigue and changes in surface electromyography and mechanomyography," *Journal of electromyography and kinesiology*, vol. 15, no. 2, pp. 138-148, 2005.
- [23] J. M. Blain, *The complete guide to Blender graphics: computer modeling & animation*. CRC Press, 2016.
- [24] S. Blackman, *Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development*. Apress, 2013.
- [25] F. Riillo *et al.*, "Optimization of EMG-based hand gesture recognition: Supervised vs. unsupervised data preprocessing on healthy subjects and transradial amputees," *Biomedical Signal Processing and Control*, vol. 14, pp. 117-125, 2014.
- [26] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," *Notices of the AMS*, vol. 50, no. 5, pp. 537-544, 2003.
- [27] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352-359, 2002.
- [28] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE transactions on biomedical engineering*, vol. 55, no. 8, pp. 1956-1965, 2008.
- [29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1-27, 2011.
- [30] A. J. Young, L. H. Smith, E. J. Rouse, and L. J. Hargrove, "A comparison of the real-time controllability of pattern recognition to conventional myoelectric control for discrete and simultaneous movements," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, pp. 1-10, 2014.
- [31] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.
- [32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1-30, 2006.
- [33] A. Marinelli *et al.*, "Performance Evaluation of Pattern Recognition Algorithms for Upper Limb Prosthetic Applications," presented at the 8th IEEE RAS/EMBS International Conference on Biomedical Robotics & Biomechatronics, 2020.

