



POLITECNICO DI TORINO

DIPARTIMENTO DI INGEGNERIA MECCANICA E AEROSPAZIALE
(DIMEAS)

Master's Degree in Biomedical Engineering

**Machine Learning Approaches for
Embedded Real-Time Gesture
Recognition**

Supervisors

Prof. Danilo DEMARCHI
Dott. Paolo MOTTO ROS

Candidate

Vincenzo BARRESI

TORINO, July 2020

Abstract

Gesture recognition is a research topic that is gaining more and more popularity, especially in recent years, thanks to technological advances in embedded devices and sensors. Today, especially in the clinical field, the most widely used method for the control of an active upper limb prosthesis is based on the ElectroMyoGraphic (EMG) Pattern Recognition, which allows to identify the type of movement that the subject intends to perform. In particular, the recognition of hand movements based on surface EMG (sEMG) signals is a promising approach for the development of Human-Machine Interfaces (HMI), such as robotic interfaces or poly-articulated prostheses, resulting in a more intuitive control and a quick tasks selection.

The goal of this thesis is to use some widespread machine learning techniques to create a system able to recognize and classify hand gestures in real-time, starting from the sEMG signals produced by the forearm muscles. The innovative event-driven approach of the Average Threshold Crossing (ATC) technique is implemented to drastically reduce the power consumption due to the minimal amount of data to be processed. This technique computes the ATC parameter by averaging the number of sEMG Threshold Crossing (TC) events in a pre-defined time window, obtaining a proper index for muscle activation monitoring.

The proposed system is composed by three acquisition channels, each one acquiring the sEMG signal and providing the related quasi-digital TC signal, and an Apollo3 Blue MicroController Unit (MCU), which feeds the embedded machine learning algorithms and classifies the performed gesture. The thesis work focuses on the comparison among Artificial Neural Network (ANN), Support Vector Machines (SVM) and K-Means ML techniques, exploring different capabilities during online prediction. Exploiting the ARM libraries features, a firmware optimization analysis has been carried out in order to reduce power consumption as much as possible and to achieve a latency suitable for real-time applications. In particular, with SVM algorithm an average classification accuracy of 94.49%, a latency of 109.85 ms and

an average power consumption of 1.67 mW were obtained. K-means has gotten an average classification accuracy of 95.14% , a latency of $62.6\text{ }\mu\text{s}$ and an average power consumption of 0.82 mW . Finally, the NN achieved an average classification accuracy of 97.59% , a latency of 2.58 ms and an average power consumption of 0.83 mW . Based on the obtained results, K-Means is the best algorithm analyzed for this study.

Summary

In this thesis work a system able to recognize and classify hand gestures in real-time, starting from the sEMG signals produced by the forearm muscles, is proposed.

In order to reduce the power consumption, the event-driven approach of the Average Threshold Crossing (ATC) technique is used. The proposed system is composed by three acquisition channels, each one acquiring the sEMG signal and providing the related quasi-digital TC signal, and a MicroController Unit (MCU), which feeds the embedded machine learning algorithm and classifies the performed gesture.

This document is composed by six chapters:

1. **Background Information:** In the first chapter the topics dealt during the thesis work are reported. This started with a brief introduction to the physiology of the skeletal muscular system, dealing in particular with the mechanism of contraction of the muscle fibers and the sEMG signal acquisition techniques. Subsequently, examples of applications that exploit the sEMG signal have been reported, both in medical and in other fields of study. Finally, the machine learning algorithms used in this thesis work for the recognition of movements have been described.
2. **State of Art:** This chapter analyzes the state of the art of modern gestures recognition techniques, which use some machine learning algorithms. The innovative ATC technique has been described, and finally the main works in which it has been used have been reported.
3. **Data Acquisition:** The first part of this chapter describes the acquisition channels implemented for the sEMG signal pick-up. The electronic components that make up the boards and the techniques implemented to solve problems such as power-line interference have been reported. A description of the Apollo3 Microcontroller was then made, including the main steps of the firmware that implements the three machine learning algorithms discussed in Chapter 1.

Finally, the entire process required to acquire the data is described. Particular attention was given to the acquisition protocol, describing in detail the electrode positioning procedure and the steps necessary for the training and testing phase.

4. **Gesture Recognition Algorithms:** This chapter describes the software that has been used to implement the classifier both offline and online. The Offline Training has been performed on a Matlab[®] platform, using ANN, SVM and K-Means algorithms in order to obtain the desired parameters. Online Prediction has been made directly on the Apollo3 board, exploiting the ARM libraries features.
5. **Experimental Results:** This part of the thesis reports the results obtained from the experimental tests of the three classifiers. An overview of the performances in terms of accuracy in the classification of the movements, latency and power consumption of the Apollo3 MCU is made.
6. **Conclusion:** This final chapter summarizes all the steps taken in the thesis to show the work done, and introduces some possible future works.

Contents

Abstract	IV
Summary	VI
List of Tables	X
List of Figures	XI
1 Background Information	1
1.1 Basis of Muscular System	1
1.1.1 The Skeletal Muscle	2
1.2 ElectroMyoGraphy (EMG)	7
1.3 sEMG Technique	9
1.3.1 Electrodes	9
1.3.2 Recording Mode	11
1.3.3 Electrodes Configuration	11
1.3.4 Electrical Noise in sEMG Signal	12
1.3.5 sEMG Acquisition Properties	13
1.4 sEMG Applications	14
1.4.1 sEMG for clinical/biomedical applications	14
1.4.2 Other applications	17
1.4.3 Recognition Control Methods	18
1.5 Machine Learning Algorithms	18
1.5.1 Artificial Neural Network (ANN)	21
1.5.2 Support Vector Machine (SVM)	23
1.5.3 K-means	24
1.6 Feature Extraction	26
1.6.1 Features in the Time Domain	27
1.6.2 Features in the Frequency Domain	27

2	State of Art	28
2.1	Gesture Recognition	28
2.2	Commercial Solutions for sEMG-based gesture recognition	29
2.3	Average Threshold Crossing (ATC)	31
2.4	ATC in sEMG Application	32
3	Data Acquisition	35
3.1	System Configuration	35
3.1.1	Acquisition Channel	36
3.1.2	Power-lines Interference Issue	38
3.1.3	Apollo3 Blue Board	42
3.1.4	Firmware	43
3.2	Performed Movements	45
3.3	Acquisition Protocol	49
3.3.1	Electrodes Placement	49
3.3.2	Training and Test Protocol	50
4	Gestures Recognition Algorithms	54
4.1	Offline Training	55
4.1.1	SVM Training	55
4.1.2	K-Means Training	58
4.1.3	ANN Training	58
4.2	Online Prediction	59
5	Experimental Results	60
5.1	Classifier Accuracy	60
5.1.1	Support Vector Machine	61
5.1.2	K-Means	62
5.1.3	Artificial Neural Network	62
5.2	System Latency	63
5.3	MCU Power Consumption	64
5.3.1	SVM Absorbed Current	64
5.3.2	K-Means Absorbed Current	66
5.3.3	ANN Absorbed Current	66
6	Conclusion	68
6.1	Future Works	69
A	Confusion Matrix	70

List of Tables

3.1	Apollo3 Blue technical sheet.	43
3.2	Apollo2 technical sheet.	52
4.1	Neural Network Architecture.	58
5.1	Statistical Result SVM (Exp.Func)	61
5.2	Statistical Result SVM (Taylor Series)	62
5.3	Statistical Result k-Means.	62
5.4	Statistical Result k-Means.	63
5.5	Comparison of the power consumption of the algorithms.	67
5.6	Comparison of the power consumption of the algorithms.	67
A.1	SVM Confusion Matrix (Exp.Func)	70
A.2	SVM Confusion Matrix (Taylor Series)	70
A.3	k-Means Confusion Matrix.	71
A.4	ANN Confusion Matrix.	71

List of Figures

1.1	Type of muscle fibers.	2
1.2	Skeletal muscle overview.	3
1.3	Physiology of the contractile mechanism of muscle fibers.	4
1.4	Mechanism of muscular contraction.	5
1.5	Type of Muscle Actions.	6
1.6	MUAP waveform main parameters.	7
1.7	Intramuscular ElectroMyoGraphy <i>iEMG</i>	8
1.8	Surface ElectroMyoGraphy <i>sEMG</i>	9
1.9	Example of Gelled sEMG Eleteode.	10
1.10	Example of Dry sEMG Eleteode.	10
1.11	Monopolar configuration.	11
1.12	Bipolar configuration.	11
1.13	Electrodes Positions	12
1.14	Example of the frequency spectrum of the sEMG signal	13
1.15	Use of the sEMG signal for prosthetic hand control.	14
1.16	Virtual Training System for rehabilitation.	15
1.17	Continuous control scheme for electric-powered wheelchair system.	16
1.18	Unvoiced Speech Recognition - arrangement of electrodes.	16
1.19	Flight simulator at NASA Ames using a neural interface that senses muscle contractions.	17
1.20	Closed loop for prosthetic myoelectric control.	19
1.21	Supervised Learning.	20
1.22	Unsupervised Learning.	21
1.23	Example of Feed-forward ANN.	22
1.24	Support Vector Machine example.	23
1.25	K-means example.	25
1.26	Feature Extraction example.	26
2.1	The Myo gesture control armband	30
2.2	The Oymotion gForce-Pro gesture control armband	30

2.3	Average Threshold Crossing transmission	31
2.4	Example of output signal of the comparator	32
2.5	ATC wireless system prototype	33
2.6	Low Power System for hand movement recognition, based on Surface ElectroMyo-Graphic Signals (sEMG)	34
3.1	Block Diagram.	36
3.2	Structure of the acquisition board.	38
3.3	Example of sEMG signal corrupted by power-line interference. . . .	39
3.4	Normalized signal PSD.	39
3.5	Model to illustrate displacement currents produced by electric field coupling [1].	40
3.6	DRL Schematic circuit.	41
3.7	Active low pass filter circuit for the reference-output INA feedback.	42
3.8	The AmbiqMicro Apollo3 Blue Evaluation Board.	43
3.9	Superficial muscle on medial section of the forearm.	46
3.10	Wrist extension.	46
3.11	Wrist flexion.	47
3.12	Wrist Radial Deviation.	47
3.13	Wrist Ulnar Deviation.	48
3.14	Grasp.	48
3.15	Idle.	49
3.16	Electrode placement on both sides of the forearm.	50
4.1	Dataset from one subject.	55
5.1	SVM Current consumption, measured with the DMM7510 digital multimeter.	65
5.2	k-Means Current consumption, measured with the DMM7510 digital multimeter.	66
5.3	ANN Current consumption, measured with the DMM7510 digital multimeter.	67

Chapter 1

Background Information

1.1 Basis of Muscular System

The *Muscular System* is composed by several muscles fundamental for human body movement. They also stabilize and maintain posture, protect underlying organs, produce heat and permit the transport of organic substances like blood and food [2]. All these functions are controlled by the *Nervous System* through electrical stimuli, called *Action Potential (AP)*. After receiving it, the muscle contracts, and the force of contraction determines the movement that can be involuntary or voluntary. This connection between the muscular system and the nervous system has led to the creation, in physiology, of the *Neuromuscular System*.

It is possible to make an initial categorization of the muscular system based on the type of muscle tissue that composes it.

There are three different types of muscle tissue in the body [3]:

- **Skeletal (or Voluntary) Tissue**

The skeletal muscles are voluntary muscles controlled by the peripheral part of the *Central Nervous System (CNS)*. In Figure 1.1A, it is possible to see that this tissue type appears as striated with multi-nuclei. They are the only multinuclear cells in the human body. These muscles are responsible of the body's mobility. In fact, the main functions are to allow the movement of the bones, maintain body posture and move the lymph and blood. The ends of skeletal muscles are usually connected to bone, skin or other muscles. In subsection 1.1.1 there are more details about this type of muscle.

- **Smooth (or Involuntary) Tissue**

The smooth tissues are called involuntary, because they are controlled by the autonomous nervous system. The cells of this type of muscle are not striated

and have a spindles shape (Figure 1.1B). Each cell has a single nucleus. This type of muscle has a slow but sustained contraction, in fact their main functions are to move organic substances through the digestive and urinary tract. Smooth tissue is found in the walls of the body's visceral organs, except in the heart.

- **Cardiac Tissue**

The cardiac muscle (or myocardium) works independently and rhythmically. This muscle is characterized by a self-exciting and rhythmic repetition that defines his rhythmic contraction. This muscle is composed of networked striated cells (Figure 1.1C). This particular structure allows the stimulation of one area to be transfered to another area of the heart muscle. This event is responsible of the rhythmic movement introduced above. As its name suggests, this type of muscle is only present in the walls of the heart.

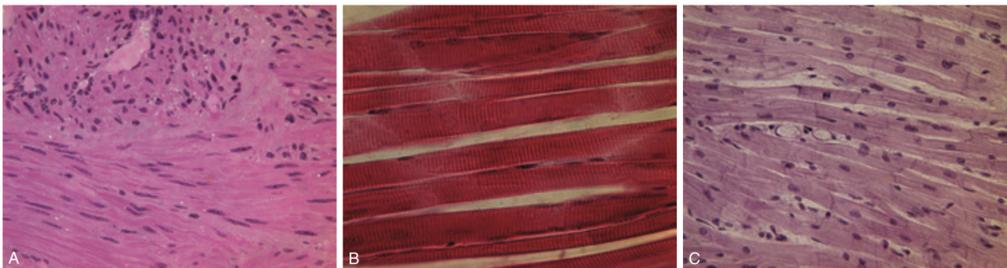


Figure 1.1: Type of muscle fibers:(A) skeletal, (B) smooth, and (C) cardiac [4].

1.1.1 The Skeletal Muscle

For this study it is very important to clearly understand the main characteristics of skeletal muscle, because these are related to body movement and therefore to the EMG signal. In fact, the EMG signal represents the electrical activity of skeletal muscles. The skeletal muscle accounts for about 40% of the body weight, while another 10% of the weight is made up of smooth and cardiac tissue. So, in a healthy body, half the total weight is muscle. These are composed of water (about 70%), protein (25%) and other organic and inorganic components such as salt, carbohydrates, fats, glycogen (5%). In this chapter, the different functions of skeletal muscles will be analyzed from two different points of view: *Metabolic* and *Mechanical*.

From a metabolic point of view, skeletal muscles play an important role in basal energy metabolism. In fact, this type of muscle works as a deposit for substances such

as amino acids and carbohydrates, but also as a consumer of nutrients and oxygen during contraction or to maintain the internal temperature. From a mechanical prospective the main objective of skeletal muscles is to convert chemical energy into mechanical energy to generate strength and power during physical activity.

Physiology of Skeletal Muscles

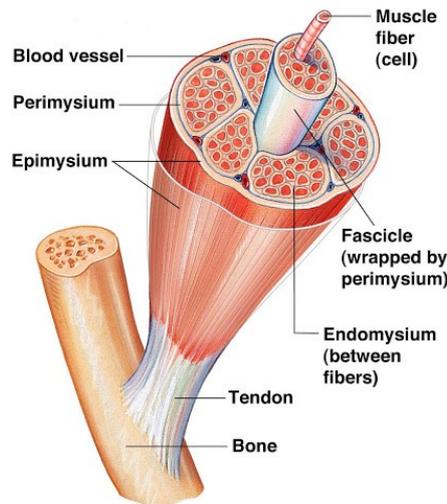


Figure 1.2: Skeletal muscle overview [4].

The skeletal muscles consist of numerous muscle fibres of various diameters (10 to 100 μm), held together by fibrous connective tissue. The external structure is called *muscular fascia*, which on the inside is composed of smaller elements. The *Epimysium* is a capsule of connective tissue that wraps around the *Perimysium*. The latter in turn groups the muscle fibres into *Fasciculi*. The single muscle fiber is covered by *Sarcolemma* (a plasma membrane). At the ends of the fibres, the sarcolemma and tendon fibres merge to form inelastic parts (*Tendons*) that connect the muscle to the bone.

Each muscle fibre consists of *Myofibril*, which in turn is composed of about 3000 *actin* and 1500 *myosin* filaments [5]. Among myofibril there is *Sarcoplasm*, an intracellular liquid rich in phosphate, magnesium and numerous *mitochondria* that provide energy in the form of ATP during contraction. These filaments are responsible of the muscle contraction as a result of their sliding movement. A longitudinal section of myofibril shows that the internal structure is organized in different areas:

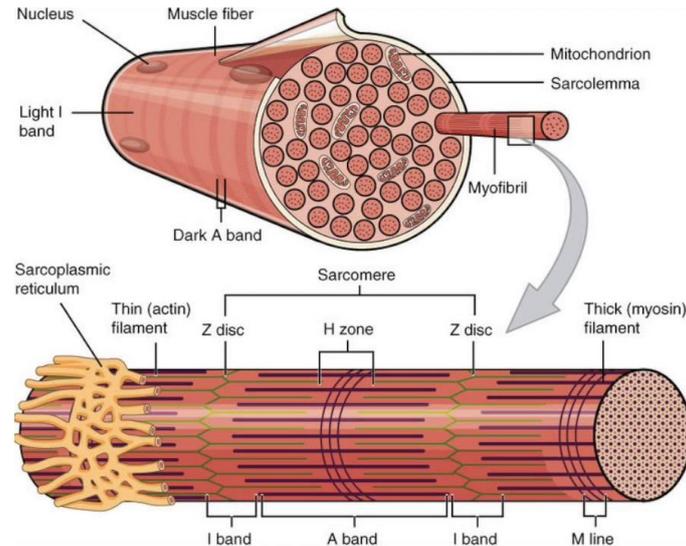


Figure 1.3: Physiology of the contractile mechanism of muscle fibers [4].

- **I bands** are composed of actin filaments. These bands are bright because they are isotropic to polarized light.
- **A bands** are dark bands made up of myosin filaments that are isotropic to polarized light.
- **Z disks** are mostly composed of alpha-actinin. This protein fixes the actin filaments. The portion of myofibril between two Z-disks is called sarcomere.
- **H zone** is the central part of the sarcomere. It is composed by thick filament.
- **M line** is a thickened area within the H zone that consists of proteins that hold the filament in place.

The position of the actin and myosin filaments is fixed by the *Titin Protein*, which has an elastic filamentous structure. These filaments are positioned between the Z disc and the M line and function as a spring during contraction.

Contraction Mechanism

Muscle contraction begins when the CNS sends an electrical stimulus through a motor nerve. The AP spreads through the *Motor Neuron* to the muscle fibres. Each end the nerve terminates the free *acetylcholine*. This neurotransmitter opens the cationic channels in the muscular membrane. This means that an important amount of sodium ions (Na^+) enter the interior causing local depolarization of the

membrane. For this reason there is a generation of AP within the membrane that reaches the centre of the muscle fibre where the sarcoplasmic reticulum releases calcium ions (Ca^{++}). These ions give rise to an attractive force between the actin and myosin filaments, which cause each other to slip. Another essential feature of myosin for contraction is the enzymatic property of obtaining energy from the phosphate bond with high energy content of ATP (which has become ADP). The contraction ends with the rebalancing of Ca^{++} within the sarcoplasmic reticulum. Figure 1.4 shows what happens to actin and myosin during contraction.

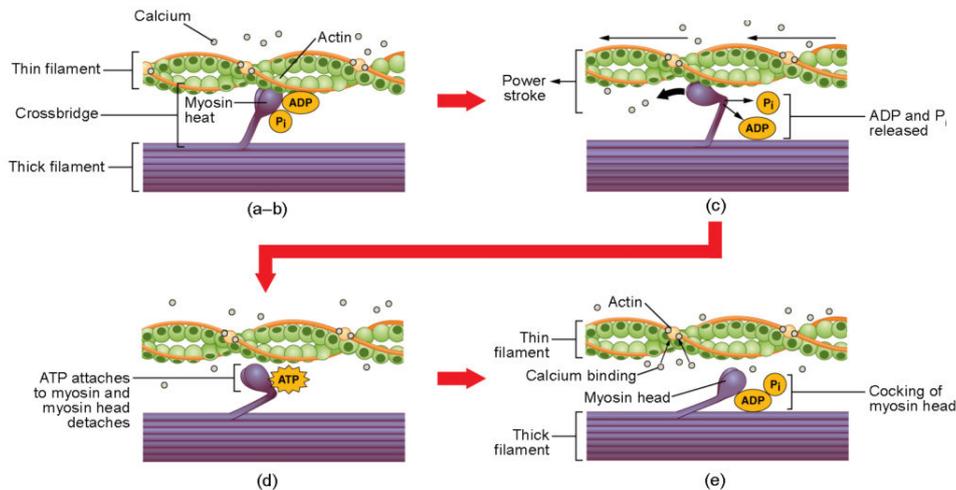


Figure 1.4: Mechanism of muscular contraction [3].

Type of Muscular Fiber

Each skeletal muscle consists of different types of fibres, classified into 3 types, based on their strength and contraction speed. Another difference between these types of fibres is the amount of myosin that causes a change in the colour of the fibre.

- **Type I or *Red fibers***

They are suitable for slow and long lasting work, because they have a good tolerance to fatigue and have the ability to stay contracted. This type of fiber involved in long and intensive efforts. They are the first type of fiber to be recruited during muscle contraction and are able to produce a small force. Red fibres have *aerobic metabolism*.

- **Type IIa or *Intermediate fibers***

Intermediate fibres are capable of rapid contraction. They are remarkably strong. This type of fiber has an *anaerobic metabolism*.

- **Type IIb or *White fibers***

White fibers are the most powerful muscle fibers. They are suitable for strong, short-term efforts. They are characterised by greater *conduction speed* and for this reason they have a quick response but a little resistance to tiredness. IIb fibres are useful for maximum contractions. This type of fibre, like IIa fibre, also has an anaerobic metabolism.

Muscle Movement

During movement each muscle does not work alone, but each movement is the result of the action of muscle synergy. It is possible to classify each group of muscles according to their function. The muscles that are activated mainly during movement are called *agonists* or primitives. The muscles that oppose movement are called *antagonists*. The role of muscles is not always the same, but the same group can be agonist or antagonist depending on the type of movement.

There are muscles that stabilize the origin (they hold a bone) and these muscles are called the *fixator*. Synergistic muscles help the agonist to perform the same movement. The two main types of muscle contraction are static, also called *Isometric Contraction*, and dynamic, defined as *Isotonic Contraction*. In the first type of contraction, the length of the muscles does not change. When a muscle contracts in isometric way, a tension is generated without shortening or lengthening the muscle fibers even if its sarcomers have shortened. Instead, the isometric contractions are those in which there is a change in the length of the muscles. These can be *Eccentric* or *Concentric*. Eccentric occurs when the muscles stretch and develop tension. In contrast to concentric where the muscles shorten to move the attachment closer.

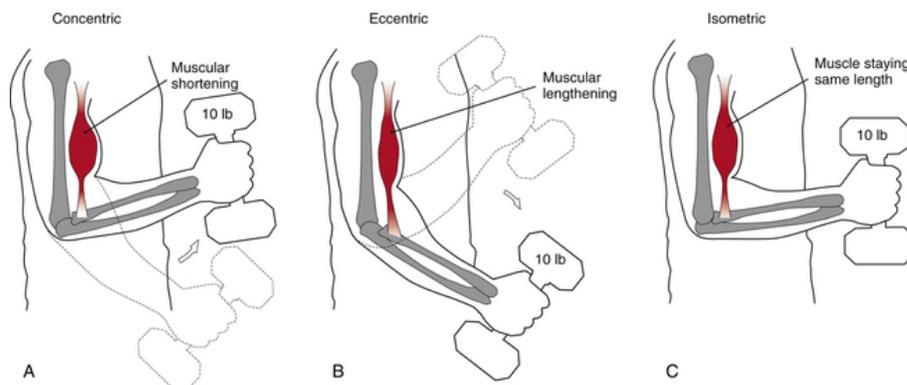


Figure 1.5: Type of muscle actions [3].

1.2 ElectroMyoGraphy (EMG)

The Electromyographic (EMG) signal provides bioelectric information related to muscle contraction, representing neuronal and muscular activities. The EMG signal is the result of the electrical potentials generated by the depolarization/ repolarization of the external membrane of muscle fibers.

The amplitude of these potentials depends not only on the anatomical characteristics of the muscles but also on the position and properties of the electrodes used for the collection. The resulting signal at the detection point is the spatial-temporal sum of the individual action potentials produced by the depolarizations of the muscle fibers of a motor unit and is called *MUAP (Motor Unit Action Potential)*.

Generally, in the electrode detection zone, there are contributions from other motor units, so a series of MUAPs are detected. Depending on the electrode used, there are variations in the shape, phase, and duration of MUAPs. The amplitude and shape of an observed MUAP are, therefore, a function of the geometric arrangement of the UM, muscle tissue and electrode properties used. The MUAPs generate a contribution so much greater the closer they are to the area of withdrawal giving rise to the electromyographic signal [2, 6].

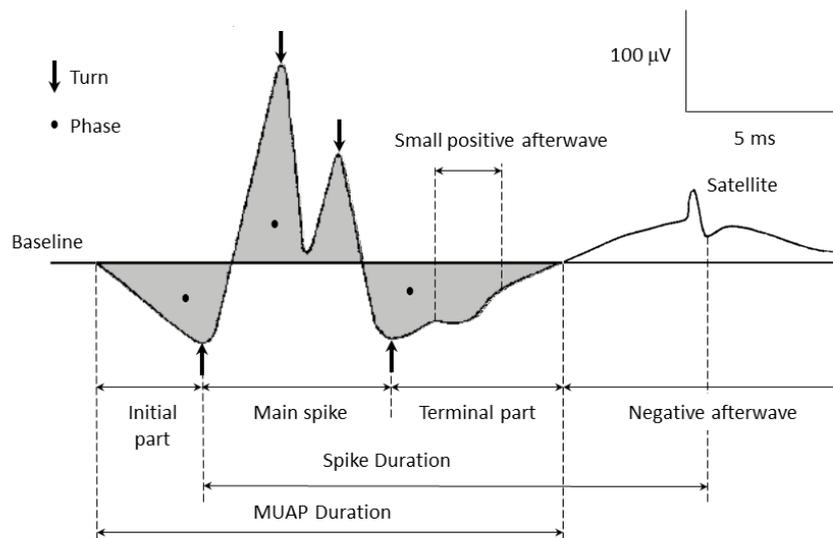


Figure 1.6: MUAP waveform main parameters: t = turn; amp = amplitude; p = phase; BL = baseline; dur = duration [6].

There are two different types of EMG signal:

- **Intramuscular ElectroMyoGraphy (*iEMG*)**

This is the classic technique for recording and evaluating the EMG signal. The electrodes used are small needles inserted directly into the muscle. This technique has a high selectivity and it is possible to study the single MUAP. The extracted signal allows to distinguish the contributions of the individual units (Fig. 1.7).

For this reason, it can be used to study both the morphology of the signal and the temporal activation. Another advantage of this technique is the absence of any type of artifact caused by the tissue, as the electrode is placed directly into the muscle. The disadvantages of this technique are certainly the invasiveness of the electrodes and the need to sterilize the needles. Furthermore, this type of sampling cannot be performed during dynamic motor activity. Nowadays, this technique is widely used in neurophysiology for the diagnosis of myopathies.

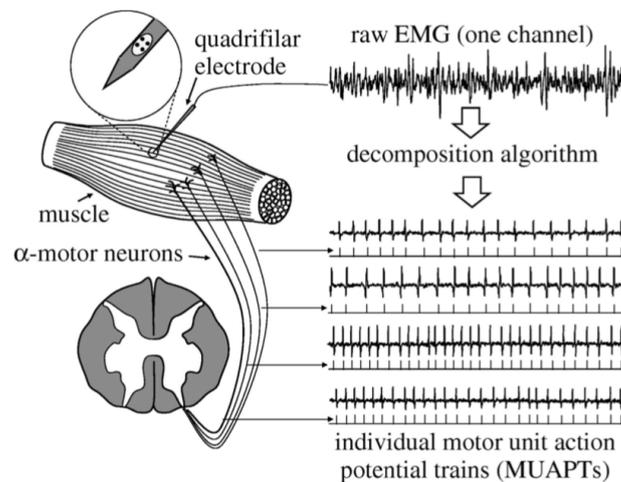


Figure 1.7: Intramuscular ElectroMyoGraphy (*iEMG*) [7].

- **Surface ElectroMyoGraphy (*sEMG*)**

Unlike the previous technique, this is a non-invasive methodology that consists of recording the EMG by placing the electrodes on the surface of the skin. The bio-signal recorded in this way is a signal given by all the motor units recruited in the movement. The signal obtained is influenced by the filtration of the tissues between the muscle and the electrodes. This technique is not good for studying the morphology of the SEMG signal, as the signal is composed of multiple sources that give a noisy appearance.

This technique is suitable for dynamic analysis, but to obtain good signals it is important to position the electrodes correctly.

When a signal is taken from surface electrodes, the total potential generated by a set of motor units is recorded, and this is called interference potential. Generally, surface electrodes can know the activity of about ten motor units simultaneously. The interference EMG signal can be traced back to those generated by the individual motor units and the time when they are activated. This procedure is carried out in order to know the recruitment method of the motor units and therefore the specific operation of the analysed muscle. sEMG is widely used in rehabilitation because it allows both to easily have information about the moment of activation of the muscle during a movement and the degree of contraction.

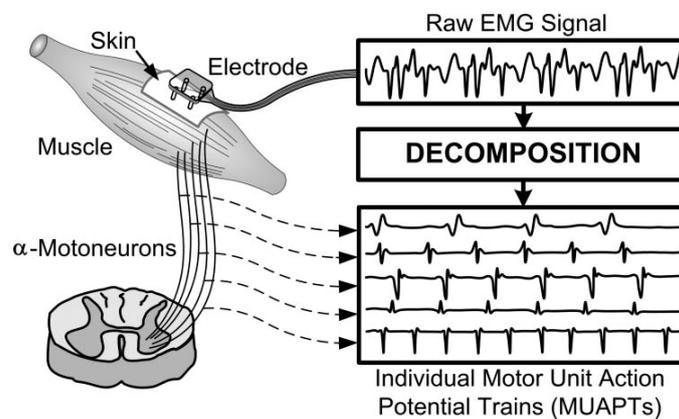


Figure 1.8: Surface ElectroMyoGraphy (*sEMG*) [8]

1.3 sEMG Technique

When an sEMG signal is acquired it is necessary to define the recording mode (monopolar or bipolar), the electrodes configuration (single or double differential), the type of electrodes to be used and especially their position.

1.3.1 Electrodes

The most used electrodes are made of silver/silver chloride (Ag / AgCl), silver chloride (AgCl), silver (Ag) or gold (Au). The Ag/AgCl electrodes are the most used and have the ability not to be polarized. In addition, thanks to the use of a *conductive gel layer*, it is possible to reduce the sensitivity to the movement artifacts caused by the sliding between electrode and skin. So these electrodes have good

stability at the electrode-skin interface. Low-cost disposable electrodes exist with the built-in gel state to simplify and lessen the timing of electrode positioning. As for the size of the surface electrodes, they vary from millimeters to a few centimeters in diameter or length. Each electrode has its particular field of application, and the size depends on both the area of acquisition and the distance between two electrodes. So the size of the electrodes determines the spatial resolution [2].

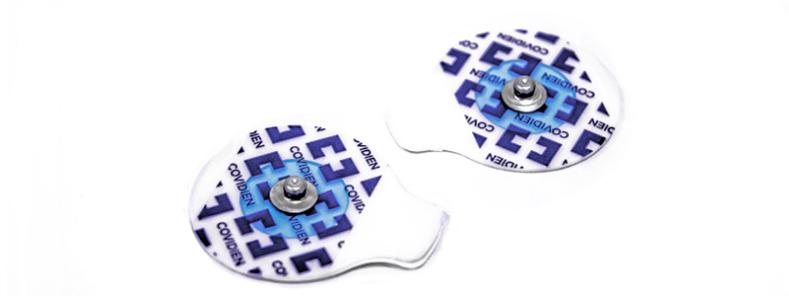


Figure 1.9: Example of Gelled sEMG Eelectrode.

An alternative to classic gel electrodes is the use of *Dry Electrodes*. These electrodes, which have been widely used in recent years for the acquisition of biopotentials, make it possible to avoid the problems caused by conductive gel on the skin, especially in long-term analysis.

The most common problems with wet electrodes are skin irritation, and the constant reapplication of the gel after several hours of use.

Dry electrodes allow to solve these problems, but not having any electrolyte medium between metal and skin, it is necessary to implement a good signal conditioning circuit to reduce the skin-electrode contact impedance.



Figure 1.10: Dry bar electrode.

1.3.2 Recording Mode

For the sEMG, there are two recording modes: monopolar and bipolar.

Monopolar configuration The monopolar mode consists of placing an active electrode at the muscle to be examined and a reference one at a neutral point.

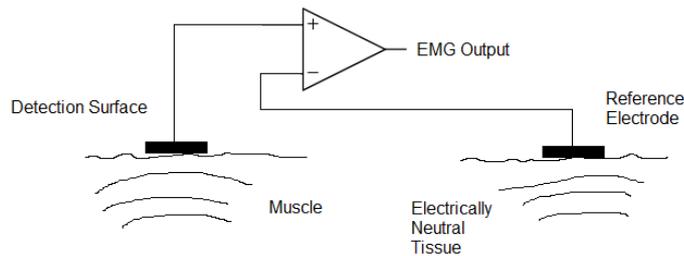


Figure 1.11: Monopolar configuration.

Bipolar configuration The bipolar mode instead requires the use of two active electrodes placed both on the examined muscle. The bipolar recording provides better immunity to disturbances but is characterized by less selectivity than the monopolar mode. In addition, more complex positioning is necessary for small muscles [2].

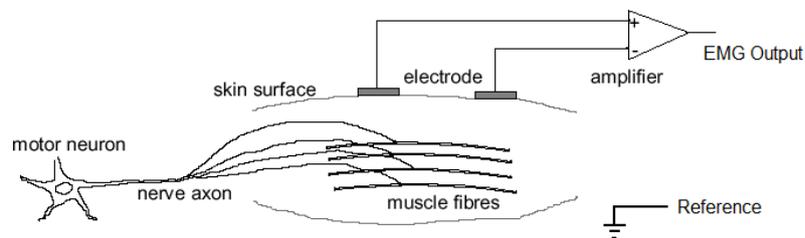


Figure 1.12: Bipolar configuration.

1.3.3 Electrodes Configuration

There are two electrode configurations: single or double differential.

In *Single differential*, the sEMG signal is equal to the output voltage of an Operational Amplifier that makes the difference between its inputs.

In the *Double Differential*, three amplifiers are used: this mode allows to create a spatial filter and therefore allows to detect the signal more on the surface.

Single differential sampling has a smaller size and is less noisy but less selective. The double differential requires an extra electrode and this increases the space but is more selective.

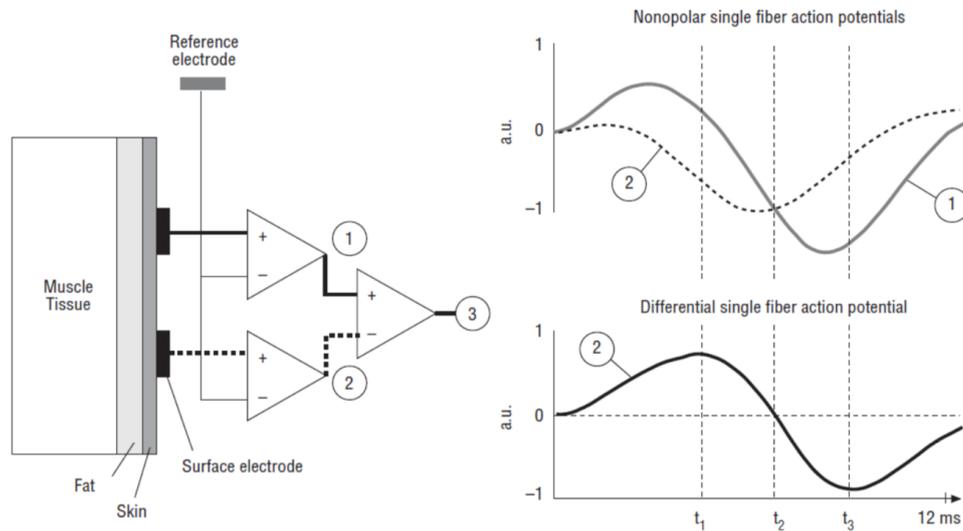


Figure 1.13: Electrodes positions[9]

1.3.4 Electrical Noise in sEMG Signal

During the acquisition of the sEMG signal the sources of noise that afflict the quality of the signal are multiple [10]:

- **Inherent Noise (electronic Components):** All electronic components generate noise throughout the band (white noise). It is not possible to completely eliminate this type of noise, but can be reduced with high-quality components and by properly designing a circuit;
- **Environmental Noise:** This noise is caused by interference between the acquisition device and the external environment (antennas, power supply, etc.). One of the most interesting interferences of the sEMG signal is network interference (50Hz in the EU, 60Hz in the USA);
- **Motion Artefact:** This noise can be generated either by the electrode sliding on the skin or by the movement of the cables connecting the electrodes to the amplification circuits. This noise is low frequency (below 15Hz);

- **Muscle Cross-Talk:** The simultaneous activation of several muscle fibres during the same movement creates cross-talk problems. So during the recording of the sEMG signal of one muscle, the electrodes detect also the activation of another muscle. This causes a wrong interpretation of the recorded sEMG signal. The factors that most influence this phenomenon are the inter-electrode distance and their size;
- **Electrocardiographic (ECG) Artefacts:** The activity of the heart is a strong source of the noise. There is no way to eliminate, completely, this noise. This can be reduced using a high CMRR acquisition channel with bipolar recording.

1.3.5 sEMG Acquisition Properties

The SEMG signal is a stochastic signal, which can be reasonably represented with a Gaussian distribution. The amplitude of the signal varies between 0 mVpp and 10 mVpp (Vpp: peak-to-peak amplitude), or from 0 to 1.5mV in rms. The spectral components are generally between 0 Hz and 500 Hz, but those with a higher power are in the 0 Hz - 150 Hz band. A typical example of a sEMG signal is shown in Fig. 1.14.

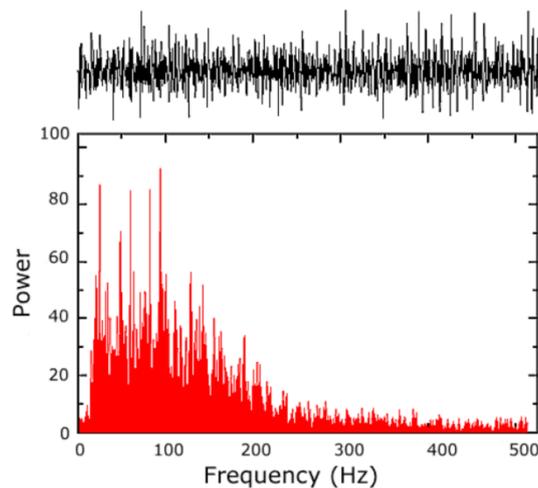


Figure 1.14: Example of the frequency spectrum of the sEMG signal [9]

A biopotential acquisition chain is typically used to acquire the sEMG signal: it consisting of a series of filtering and amplification blocks. The filtering blocks are used to record only the sEMG signal in the band of interest (usually 15 Hz -

400 Hz) and to attenuate the noise and artifacts that are been discussed earlier. Amplification is necessary to adapt the sEMG signal (usually below 10 mVpp) to the ADC dynamics useful for digitizing the signal. Therefore the typical acquisition channel consists of 4 main phases: *detection*, *amplification*, *conditioning* and *digitization* [2].

1.4 sEMG Applications

The EMG signal can be used in various research fields and in biomedical applications. In the last years research groups have been implemented many diagnostic tools for neuromuscular diseases, which exploit the electrical signal produced by muscles.

1.4.1 sEMG for clinical/biomedical applications

Rehabilitation and Prosthetic Hand Control

In the field of prosthetics and rehabilitation the EMG signal can be used for the development of modern *Prosthetic Hands* [11].

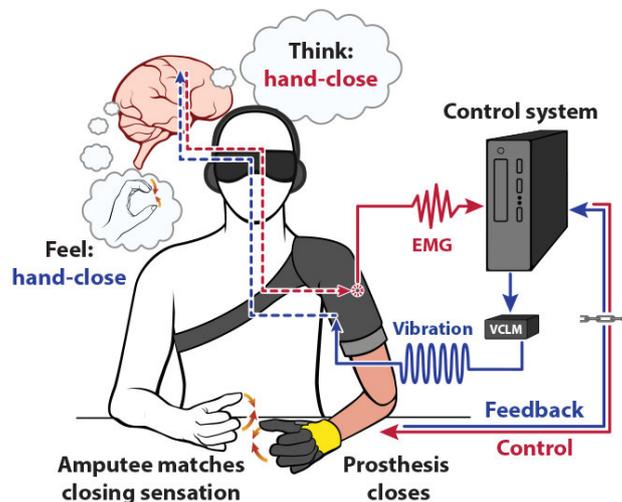


Figure 1.15: Use of the sEMG signal for prosthetic hand control. [11]

Many studies have led to a new type of games-based rehabilitation for the control of myoelectric prostheses [12]. At the beginning, the control of a myoelectric prosthesis can be a frustrating experience, especially after the already traumatic

event of limb loss. Due to the non-intuitive interface, which manages a complex mechatronic system, the cognitive demand for control of the prosthesis is high and further delays the actual use of the device in daily life [13]. Despite the progress the industry has made, especially during the 20th century, complete hand replacement is still a difficult challenge from both an engineering and clinical point of view. The first mechanical arm, called the Ballif arm, was invented in 1812, and was controlled by other upper arm movements [14].

The first myo-electric prosthesis was developed in Munich, and initially it was not a portable system, but was connected to a building power supply. This arm used vacuum tubes for the control system and had basic open and close functionality. Later progress incorporated the batteries into the prosthesis. The first clinically myo-electric prosthesis was developed by Russian experts in the 1960s and since then, innovations in prosthetic control schemes, especially in the upper limbs, have grown exponentially. There is still much to be improved in prosthetic control and, as a result, much attention in the academic community has focused on improving prosthetic control systems using motion recognition algorithms. At least 50% of upper limb amputees report problems with prosthetic control and function [15], which can be attributed to the need for more training in prosthetic handle. Virtual training systems in the form of videogames, as shown in Fig. 1.16, provide patients with a fun and intuitive way to improve muscle coordination and general control.

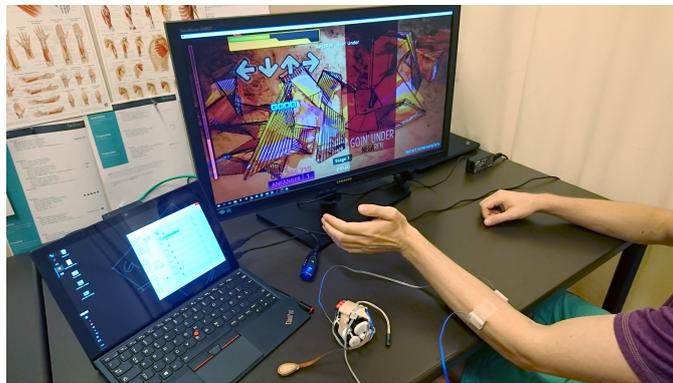


Figure 1.16: Virtual Training System for rehabilitation. [12]

Electrical Wheelchair

The standard way to drive an electric wheelchair involves the use of a hand to operate a kind of two-dimensional joystick. The use of the computer in communication with the muscles during contraction allow you to perform any type of action using EMG. Muscle contraction can be used to control an *Electrical Wheelchair* which

can be of great help for people with disabilities. [16].

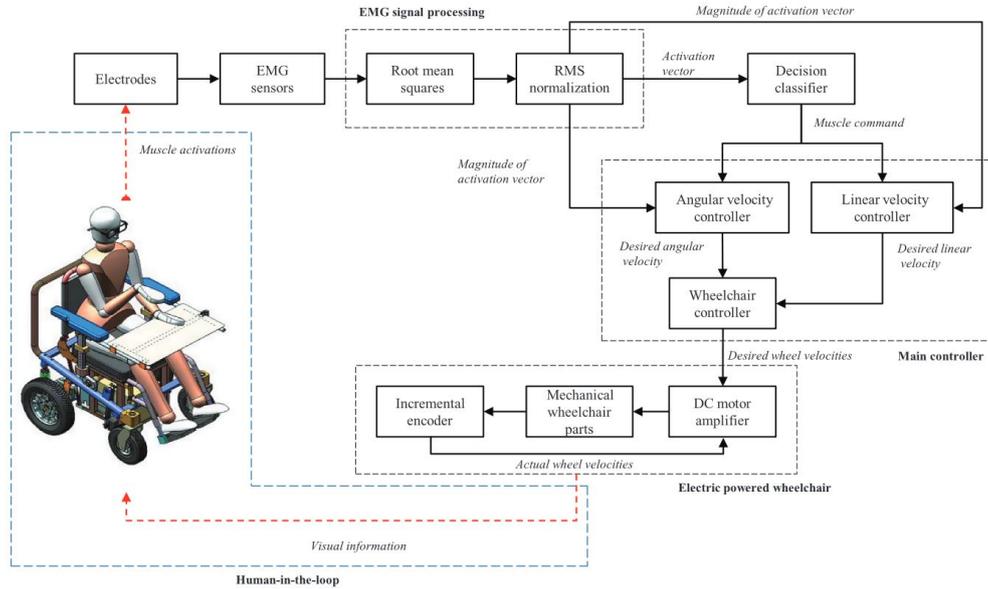


Figure 1.17: Overview of the continuous control scheme for electric-powered wheelchair system [16].

Unvoiced Speech Recognition



Figure 1.18: Unvoiced Speech Recognition - arrangement of electrodes. [17]

Another very important application is the use of the muscles associated with speech to recognize words. This type of technique is not based on any sound information, but only uses the EMG signal for communication.

1.4.2 Other applications

The use of the EMG signal can be a source of inspiration for even the most modern gaming techniques. In fact, it is possible to replace the joysticks commonly used to control video game characters with sensors that directly detect muscle activity, for a more intuitive type of entertainment. Joysticks are very cheap, but sometimes it's very difficult to memorise the sequence of keys to press to perform a certain action.

A man-machine interface device designed for a combat action game, *Muscleman*, has been developed. This device acquires the sEMG signal of the superficial flexor muscle of the forearm and the acceleration signal of the forearm movement. The device is able to classify hook and straight fist movements by analyzing forearm acceleration data [18].



Figure 1.19: Flight simulator at NASA using sEMG signal. [11]

At NASA's Research Center in California, researchers used the sEMG signal in their research program on *Flight Systems*.

NASA researchers are studying a lot of applications that exploit the muscle signal. Inside the NASA hangars, control stations have been created for aircraft that use the muscle signal to fly. Simulations have been carried out for the control of aircraft that carry fuel and, in addition, many studies are also involving the creation of uniforms that contain sensors for the EMG signal.

Charles Jorgense, master of NASA's Neuroengineering Laboratory, says that this type of communication with robots and airplanes, using biological signals directly,

will be fundamental in the future. Fig. 1.19 shows a control system for aircraft of the latest generation that exploits sEMG technology. The joystick is not used, but is replaced by an armband with eight pairs of dry electrodes that take the muscle signal. These signals have been used for simple landing maneuvers of airplanes. [19].

1.4.3 Recognition Control Methods

A conventional myoelectric pattern recognition control system consists of several steps responsible for taking the raw sEMG data and transforming it into a valid control signal for a peripheral device, such as wheelchairs [16], rehabilitation robots, prosthetic arms.

The sEMG signals are first amplified, filtered and digitized for use with a programmable microprocessor-based system. The modified signal data is then segmented, and the necessary characteristics can then be extracted from the segmented signal using a variety of time domain, frequency domain, time-frequency domain, and feature projection techniques.

Based on these extracted characteristics, a classifier is then trained to recognize patterns in signals and assign them to predetermined classes [20]. Based on the label outputs generated by the classifier, the controller generates output commands for external systems, such as a prosthetic limb. These algorithms are still limited in the performance they provide, as each of their possible control outputs, or classes, must be provided with the relevant training data. Pattern recognition algorithms have been the focus of prosthetic control research for decades. Most forms of the statistical and learning classifiers have been applied to myocentric control.

Techniques such as *Feed Forward Multi-layer Perceptrons*, *Convolution Neural Networks (CNN)*, *Fuzzy Logic*, *Linear Discriminant Analysis (LDA)*, *Support Vector Machines (SVM)*, *Hidden Markov Model (HMM)* and *K-Nearest Neighbours (KNN)* have all been applied to classify sEMG signals for gesture recognition applications. Three of the main machine learning algorithms used for movement prediction will be presented in the next section. These algorithms will be the ones that I will be used in my thesis work.

1.5 Machine Learning Algorithms

In this section three of the most popular ML algorithms are presented: Artificial Neural Networks (ANN), Support Vector Machines (SVM) and K-means. The theory behind these algorithms and their advantages and disadvantages will be briefly

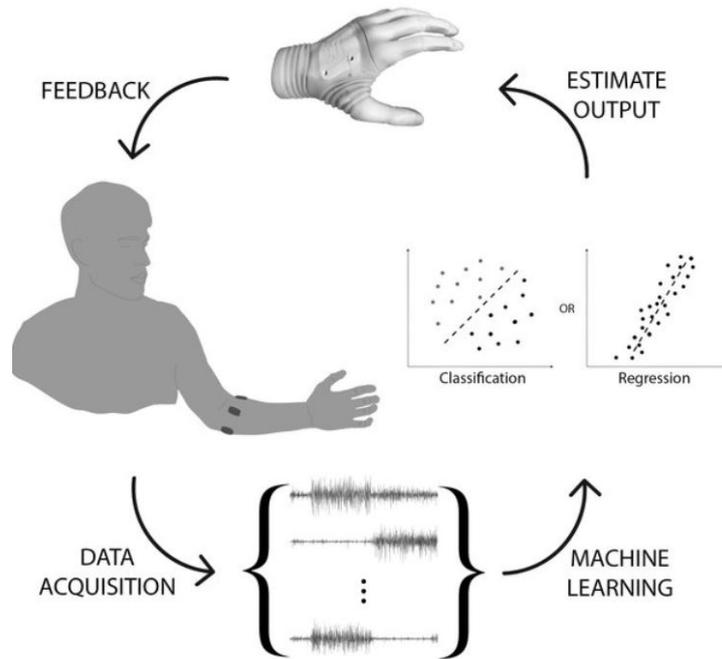


Figure 1.20: Closed loop for prosthetic myoelectric control.

described.

Before talking about these algorithms, however, it is important to analyze the difference between supervised and unsupervised learning algorithms.

Supervised Learning

In supervised learning, is given a dataset of which the correct output class is already known. The types of supervised algorithms are divided into *Classification* and *Regression* algorithms. In a classification algorithm, a result is predicted within a discrete output, while in a regression problem the output is continuous.

Since these types of algorithms work on an already "catalogued" dataset, it is easy at the end to define the performance of the classifier, in terms of process accuracy.

When a supervised learning algorithm is implemented, it is important to consider the compromise between Bias and Variance. Both are related to each other.

The balance between Bias and Variance also refers to the generalization of the model. In any model, there is a compromise between Bias, which is the tendency to deviate from the expected value, and Variance, which is the amount of the error in the label prediction between different data-sets. Bias and Variance behave in the opposite way: an increase in variance is generally linked to a reduction in Bias,

and vice versa.

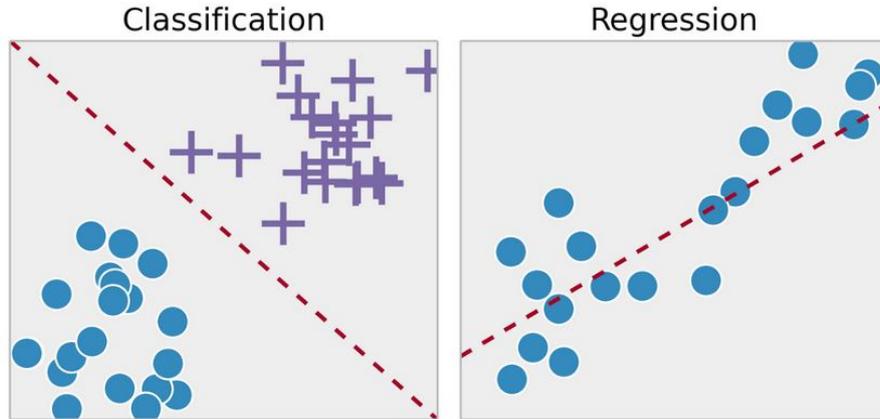


Figure 1.21: Supervised Learning.

Unsupervised Learning

In unsupervised learning, unlike supervised learning, there are unlabelled or unstructured data. With these techniques it is possible to observe the structure of the data and to extrapolate meaningful information. In these techniques, however, you cannot rely on a known result variable, such as an output class.

There are two techniques that help us in dealing with unsupervised learning problems: *Clustering* and *Data Dimensionality Reduction*.

Clustering is an exploratory technique that allows the aggregation within groups (called clusters) of data that have no previous knowledge of belonging to groups. There will therefore be large datasets where the data within them have similar elements. Within each single group (or cluster) you will therefore find those data that have many similar characteristics. Clustering is an excellent technique that allows us to find relationships between the data.

Dimensionality reduction is a widely used approach in the pre-processing of features, with the aim of eliminating "noise" from the data. This reduction can also cause a lower predictive performance, but can also make the dimensional space more compact in order to keep the most relevant information.

Large amounts of data generate a problem in storage space and a considerable decrease in computational performance.

The reduction in dimensionality can also be useful for data representation, such as within a high dimensional feature space, which can be projected onto 1D, 2D and

3D space.

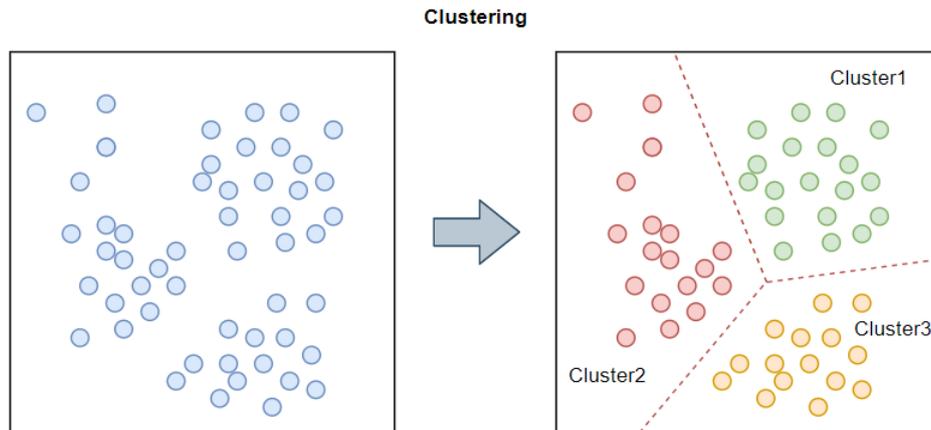


Figure 1.22: Unsupervised Learning.

1.5.1 Artificial Neural Network (ANN)

Neural networks are automatic learning models that try to imitate the structure and functioning of the brain. In this structure every neuron is connected with many others, and the connection can be of a strengthening or inhibiting type towards the activation of the units to which it is connected.

Each neuron contains a function used to combine among them the values of all its inputs and a function, called *Activation Function*, which returns the output of the neuron. The general form of the overall function contained in a neuron is represented by the following formula:

$$y = f\left(\sum_i w_i x_i + b\right) \quad (1.1)$$

In this, w_i are the *Weights* assigned to each input in the combination phase and b is a *Bias Term*. The set of Weights and Bias represents the information that the neuron learns in the training phase and that it keeps afterwards. The function f represents the *Activation Function*, which normally consists of a threshold or limit function that makes sure that only signals with values compatible with the imposed threshold or limit can propagate to the next neuron or neurons. Typically the activation function is a non-linear function, usually a step function, a sigmoid or a logistic function.

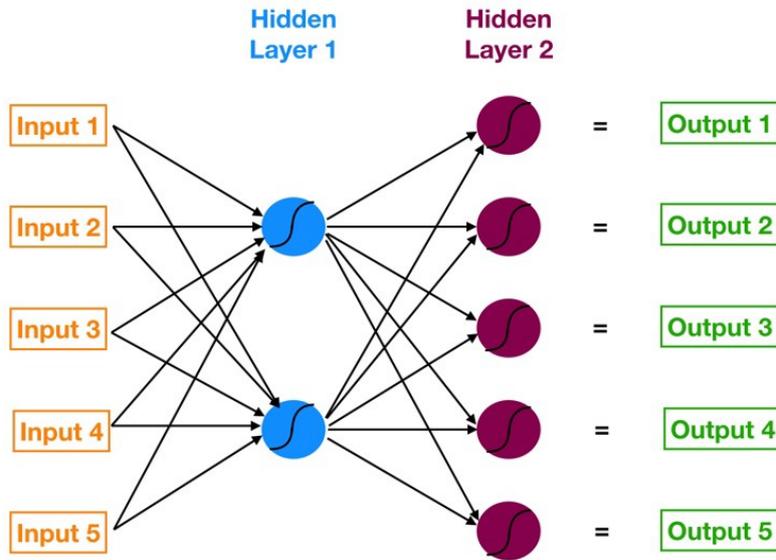


Figure 1.23: Example of Feed-forward ANN.

Neural networks are typically structured in three parts, containing distinct amounts of neurons:

- An input level;
- A more or less numerous set of hidden internal layers;
- An output level.

The input signals cross the entire network from the input layer to the output layer, as shown in the Figure 1.23. This ANN, called a *Feed-Forward ANN*, only allows data to travel from input layer to output layer.

Advantages and Disadvantages

Neural networks have a high speed in data analysis, and an ability to learn the solution from a given set of examples. However, the main disadvantages arise: (a) from the need to choose an adequate set of examples; (b) when you have to respond to inputs substantially different from those of the set of examples.[21].

1.5.2 Support Vector Machine (SVM)

A classification technique based on support vector machine (SVM) allows you to classify both linear and non-linear data collections. An SVM represents all instances of the training data collection on a plane formed by a number of axes (dimensions) equal to the number of attributes that constitute the instances. For example, if an instance consists of three attributes, then the training data will be represented on a three-dimensional plane. The three main characteristics of an SVM classifier are:

- Lines or Hyperplanes, depending on whether the classifier presents, respectively, two-dimensional or n-dimensional graph;
- Margins;
- Support Vectors.

A line or hyperplane constitutes a "boundary" that allows instances belonging to different classes to be classified by dividing them among themselves.

A margin is a distance between the two instances of different classes closer together. On the other hand, support vectors correspond to the instances that are the most difficult to classify for an SVM, since they are the ones within the margins of a hyperplane.

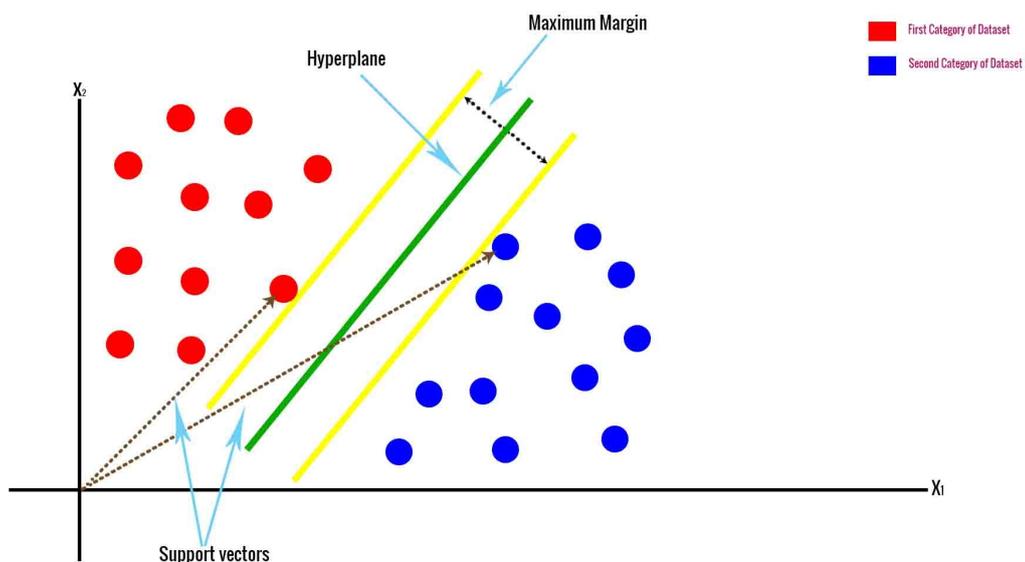


Figure 1.24: Creating the hyperplane thanks to the use of SVM.

An SVM applies differently depending on the type of instances of a data collection to be classified, linearly separable and not. In the presence of instances separable in a linear way, it is necessary to find among all the lines or among all the hyperplanes that separate them between the different classes those that maximize the value of the margin. In fact, it is selected the straight line or the hyperplane with maximum margin value, since it allows to minimize the classification error. In order to find maximum margin hyperplanes, SVM aims to maximize function 1.2 in relation to \bar{w} and b :

$$L_P = \frac{1}{2} \|\bar{w}\|^2 - \sum_{i=1}^t \alpha_i \gamma_i (\bar{w} \bar{x}_i + b) + \sum_{i=1}^t \alpha_i \quad (1.2)$$

Here, t represents training point quantity, α_i stands for Lagrangian multipliers and L_P exemplifies the Lagrangian. Vector \bar{w} and constant b characterize the hyperplane.

Advantages and Disadvantages

A SVM algorithm has the advantages to solve difficult and non-linear classification problems and to guarantee a high level of classification accuracy. Among the disadvantages are the high model creation time, which is shorter than that used by a neural network, and the non-interpretability of the model.

1.5.3 K-means

K-Means is a clustering algorithm that divides elements into K groups according to specific features. The characteristic element of each cluster is the *Centroid*, which corresponds to the average of the elements belonging to the cluster.

The main objective of the algorithm is to minimize the intra-cluster variance. Different K-means algorithms are available. The one used in this thesis work is the Hartigan-Wong Algorithm (1979), which defines the total intra-cluster variance as the sum of the squares of distances (Euclidean distance) between the elements of a cluster and its centroid.

This algorithm goal is to minimize an objective function:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c_j\|^2 \quad (1.3)$$

where

$$\|x_i^j - c_j\|^2$$

is a distance between the cluster centre c_j and a data value x_i^j . Its value must be as small as possible.

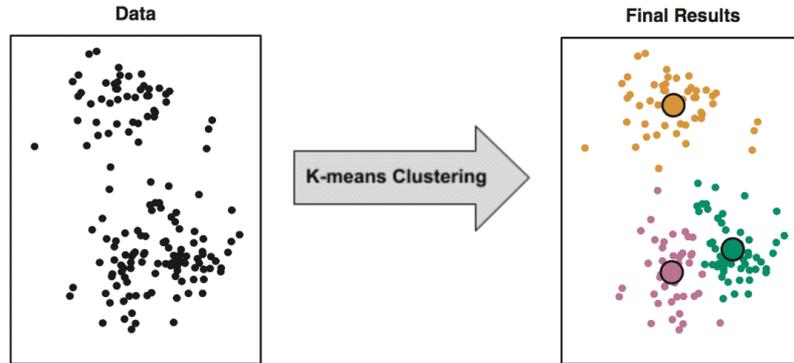


Figure 1.25: K-means example.

The main steps of the algorithm are now summarized: as a first step of the algorithm, the number of clusters needed to divide the analyzed dataset is indicated. The objects are selected in a random way from the dataset to be used as first centroids. Then, each of the remaining objects is assigned to the nearest centroid, according to the Euclidean metric (*Cluster Assignment Phase*).

After this phase, the algorithm calculates the new centroids of the total cluster (*Update Phase* of the centroids). Once all the centers have been recalculated, each assignment is re-trolled to see if it could be closer to a different centroid. All objects are then reassigned using the recalculated centroids.

The two phases of cluster assignment and centroid update are repeated iteratively until either convergence is reached (the clusters obtained in the current iteration are the same as those obtained in the previous iteration) or the maximum number of iterations chosen is reached.

The Hartigan-Wong algorithm can therefore be summarized as follows:

1. Define the number of clusters to generate;
2. select a number of values equal to the number of clusters chosen from the data-set, as centroids of the initial clusters;
3. Assign each data to the nearest centroid, according to the Euclidean distance;
4. Update the centroids by calculating the new average values of all the elements belonging to the cluster;
5. Minimize the total intra-cluster variance during a iteratively loop.

Advantages and Disadvantages

The main advantage of the K-means algorithm is undoubtedly the ease of implementation. The main problem instead is that the performance depends strongly on the initial conditions, i.e. the number of clusters chosen and the initialization of the centroids.

1.6 Feature Extraction

Despite the progress of machine learning, it is still not possible for amputees to initiate simultaneous control of individual fingers of the prosthesis, and instead control individual or combinations of finger movements based on predetermined gesture classes.

Feature Extraction should be used to increase information density of the EMG signals, retaining information that allows a classifier to identify different contraction patterns, while eliminating irrelevant data.

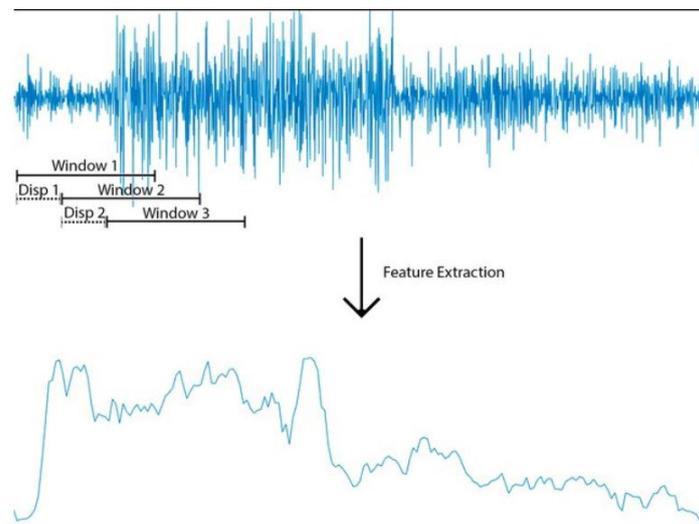


Figure 1.26: Feature Extraction example.

There are three major categories of features: time domain, frequency domain and time-frequency domain. The last domain is not considered in this introduction, due to its high complexity that make it not suitable for the training of a classifier. Only the features of the first two domains are then reported below.

1.6.1 Features in the Time Domain

This type of features extraction is largely use in literature, thanks to their straight-forward calculation, without needing any transform. Mayor disadvantage is that they assume no variations in the signal frequency.

- **Root Mean Square (RMS):** is related to the constant force and non fatiguing contraction.

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2} \quad (1.4)$$

- **Mean Absolute Value (MAV):** it is calculated by averaging the absolute values of the EMG data.

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_n| \quad (1.5)$$

- **Mean Absolute Value Slope (MAVSLP):** is calculated as the difference between consecutive MAV values.

$$MAVSLP_i = MAV_{i+1} - MAV_i \quad (1.6)$$

- **Integrated EMG (IEMG):** is defined by the summation of the absolute values of the sEMG signal.

$$IEMG = \sum_{n=1}^N |x_n| \quad (1.7)$$

All of these features above (RMS, MAV, MAVSLP, IEMG) are computed based on sEMG signal amplitude.

1.6.2 Features in the Frequency Domain

- **Median Frequency (MDF):** is the frequency value that divides signal power spectrum in two equal areas.

$$\sum_{j=1}^{MDF} P_j = \sum_{j=MDF}^M P_j = \frac{1}{2} \sum_{j=1}^M P_j \quad (1.8)$$

- **Mean Frequency (MNF):** is calculated as the sum of product of the sEMG psd and the frequency values, divided by the total sum of psd.

$$MNF = \frac{\sum_{j=1}^M f_j P_j}{\sum_{j=1}^M P_j} \quad (1.9)$$

Chapter 2

State of Art

2.1 Gesture Recognition

The recognition of gestures is not a new area for Information Technology (IT). In fact, in scientific literature this subject has been dealt with since the 1990s, and some of the techniques used to solve the problem of classification of gestures had already been developed in the 1980s for other purposes, such as voice or handwriting recognition, and were subsequently adapted.

However, in recent years, this topic is increasingly capturing the attention of the IT market, probably also due to technological developments in the electronic field. Initially, the issue of gesture recognition was brought to the public through film (mainly through fantasy films, including the well-known "Minority Report" and "Iron Man"), but it was the gaming market that gave the industry a real boost.

Later it became clear that gesture recognition technologies had potential outside of gaming, particularly in the clinical field. As a result, new devices were developed, both from large manufacturers and research groups around the world.

In the clinical field, there are several applications for the recognition of gestures, especially for the implantation of prosthetic limbs of the latest generation. The bottleneck of all these applications is represented by the time taken to acquire sEMG signals and pattern recognition (correct movement), which must not exceed 300 ms [22].

In [23] a system that allows the recognition of six different hand movements has been proposed. The scheme consists of 4 acquisition channels that allow to drive an exoskeleton.

The system has been validated offline and online on eight subjects with partial limb paresis. The accuracy of the classification was evaluated using the *Random Cross*

Validation (RCV). The total latency of the system is approximately 250 ms [23].

In [24] the research team proposed a real-time system for the classification of the movements of the fingers of the hand, using the sEMG signals of the forearm. Seven acquisition channels have been implemented with off-the-shelf components. The classification of the signal pattern was evaluated in real-time using *Linear Discriminant Analysis (LDA)* approach. The percentage of accuracy in classification reached 95.8% with a latency of 192 ms.

In [25], a research team has reduced the dimensions of the features vector, by using only the RMS, but to obtain a good result they had to assign a predefined set of movements. Features extraction has been performed on a windows made of 128 samples, with a sampling frequency of 2048 Hz. The acquisition window is slightly wider than 60 ms, being really suitable for a robot control in real time. This work has also introduced a *SVM classifier*, different from the one used in the previous work. This new classifier is able to disclaim between classes using complex non linear functions. Using a cross validation method this classifier can reach an accuracy of 90%.

Finally, in [26] an embedded system for real-time control of poliarticulated hand has been carry out. The online performance of the wearable node in terms of end-to-end recognition ratio are evaluated. Also in this work the *SVM Algorithm* has been used for the implementation of the classifier.

2.2 Commercial Solutions for sEMG-based gesture recognition

Myo Armband by Thalmic Lab

From a technical point of view, the Myo armband (Figure 2.1) is equipped with a 9-axis Inertial Measurement Unit (IMU), which allows it to detect the absolute orientation of the device and track its movements. However, the real evolution of the Myo armband is the possibility to detect hand and forearm movements without the use of video cameras or similar systems. For this purpose, the device is equipped with 8 sEMG sensors, which, in the same way of similar devices for medical use, measure the electrical signals produced by muscular activity; the Myo is an elastic band placed on the forearm, which is wrapped circularly by the 8 sensors. During

the execution of a gesture (e.g., a fist), the contraction of the forearm muscles allow to measure the intensity of the muscular electrical signal coming from different fibers with the possibility to identify the gesture performed.

The maximum sampling frequency for muscular signal acquisition is 200 Hz.



Figure 2.1: The Myo gesture control armband. [27]

Oymotion gForce-Pro



Figure 2.2: The Oymotion gForce-Pro gesture control armband. [28]

GForce 100 Armband is a smart wearable device very similar to the Myo product discussed above. The substantial change among the two products is the sampling frequency, to allow a higher accuracy in signal acquiring: it goes from 200 Hz of the Myo armband to 1000 Hz of this product. [28]

2.3 Average Threshold Crossing (ATC)

The *Average Threshold Crossing* (ATC) is a technique in which an event is generated each time that an input signal crosses a fixed threshold. Many papers in literature show the advantages of using this approach for sEMG signal [29] [30] [31]. A comparison between the classic way of transmitting data and the innovative ATC technique is shown in figure 2.3. The classic method involves the acquisition and the generation of TX Packets each fixed time (T_s) with a sampling frequency of $1/T_s$. Meanwhile, with the TC approach, only when the signal crosses the threshold (V_{th}), the TX events are created [2].

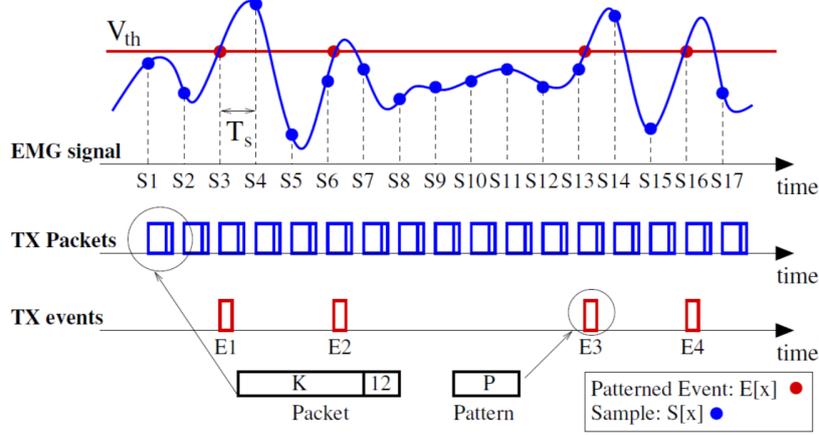


Figure 2.3: Comparison between classic sEMG transmission way and ATC technique [31].

On electronic point of view, this technique (applied to sEMG) is very easy to implement, in fact, in addition to the classic acquisition channel only an operational amplifier comparator is required. This type of components are able to compare two analog voltage or current inputs. The sEMG signal and a voltage reference are the inputs of this component, so in this way, every time that the signal is higher than the reference (threshold) the output take the value of the voltage supply. The output maintain this level until the signal falls below the threshold.

Therefore, the outputs signals of the comparator has the same shape of a digital signal like the one in Figure 2.4. It's more correct to define this signal a *Quasi-Digital Signal*, because the information is contained in the time between two consecutive events. The TC signal can be given directly as input to a micro-controller or a wireless transmission module, without using an *Analog-to-Digital Converter* (ADC).

This allow to reduce the number of component, and hence the space occupied by electronic components, in the device [2].

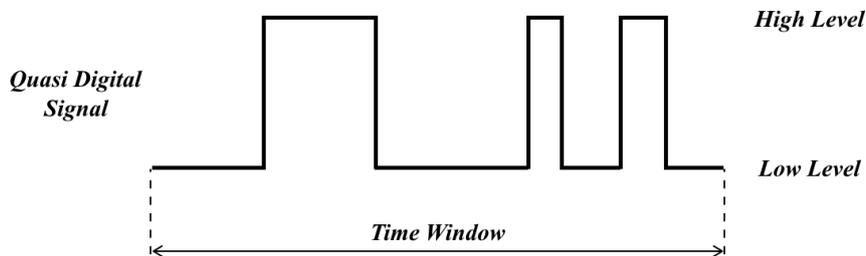


Figure 2.4: Example of output signal of the comparator.

The quasi digital signal allows to decrease the amount of data to be sent or in general, that has to be saved. Furthermore the ATC technique permits to obtain a low power consumption on transmitter perspective, compared to the standard way to process data. The consumptions of the ATC technique are very dependent on the choice of the threshold, because a threshold level too low would cause many events to be sent, but a too high level would be the cause of few information. For this reason, the correct selection of the threshold is a critical point of this technique. To obtain a good compromise between power consumption and amount of data, it could be useful to use a *Dynamic Threshold*, obtaining also a more efficient number of events. [2].

2.4 ATC in sEMG Application

Some studies have combined ATC technology with EMG acquisition devices. In particular, the *Istituto Italiano di Tecnologia* (IIT) has shown that by combining ATC technology with Impulse Radio - Ultra Wide Band (IR-UWB) communication technology it is possible to obtain and transmit information on the muscle strength developed [2].

In a first study [32], a portable wireless device for biomedical applications was built. This system is based on the wireless protocol IR-UWB. This technology consists of the transmission of short pulses to minimize transmission consumption. The system consists of an sEMG acquisition channel and allows to obtain in hardware the ATC parameter. In particular, this work demonstrated the correlation between system performance in terms of TC events (digital pulses) and system performance by looking at the ARV values calculated on the raw sEMG signal in

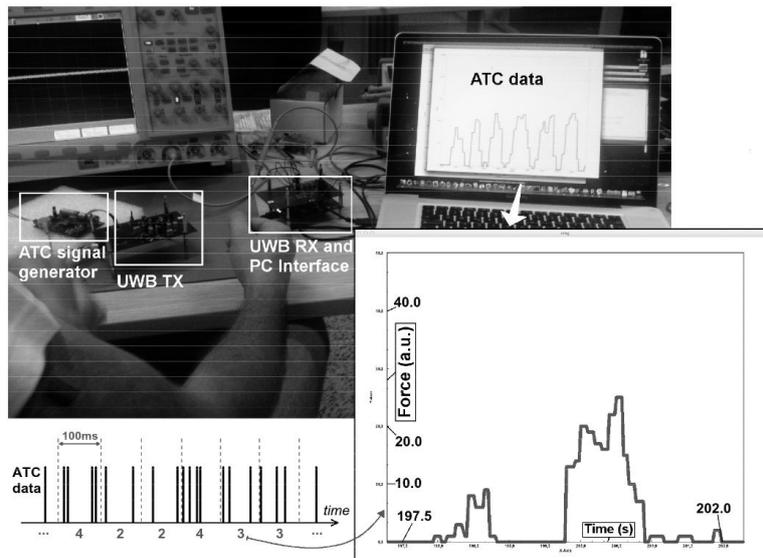


Figure 2.5: ATC wireless system prototype. In this example subsequent muscle contractions have been applied and force wirelessly transmitted [32].

the recognition of force levels. These values have been validated using also the signal of force recorded with a dynamometer during a maximum voluntary contraction [2].

An improvement of this system was developed by the same research team [33]. This second work concerns the extension of the previous ATC wireless system to a more complex version of multichannel acquisition. Data are transmitted using the AER (Address-Event Representation) approach which uses an encoder to form a packet of data that identifies the input channel and wireless transmitter. The pattern of this prototype is the same as the previous one: starting from the raw signal sEMG is obtained the signal almost-digital (TC) generating pulses for the IR-UWB transmitter that sends data accordingly. The work has determined a confirmation in the reduction of power consumption and in the dimensions of the board. This work also aims to evaluate the robustness of the ATC signal by varying the SNR, amplifier distortion, saturation and number of lost events. A tolerance of 5-6dB for SNR and 70% for lost events has been demonstrated [2].

In further work [34], a prototype of the multi-channel ATC/sEMG card was validated to define the parameters of the ATC (time window, threshold value). In this work, a threshold is fixed to 100mV above the noise on the baseline of the signal. The width of the time window must be sufficiently large to modulate the muscle activity, but not too large to not compromise the temporal resolution [2].

Another work [36] used the acquisition cards designed in [35] in order to control in real-time a functional electrical stimulation (FES), a therapy used in neuromuscular rehabilitation [2].

One of the latest works [37], a project conceived in collaboration between the Politecnico di Torino, the Massachusetts Institute of Technology (MIT) and Harvard Medical School had the objective of creating a wearable device to acquire and process sEMG data. The data obtained were then used to characterize muscle activity patterns in rehabilitative medicine. The innovative event-driven approach of ATC technique is implemented to drastically reduce energy consumption thanks to the minimum amount of data to be processed.

Finally, in work [38] a system that combines modern gesture recognition techniques and ATC technology has been developed. A low power system, able to recognize hand gestures through a built-in classifier acquiring sEMG signals from the forearm, has been designed. The ATC technique was used to pre-process the data and a *fully-connected Neural Network (NN)* has been chosen as classifier architecture. The final results of the implementation are suitable for real-time wearable applications due to low latency and low power consumption.

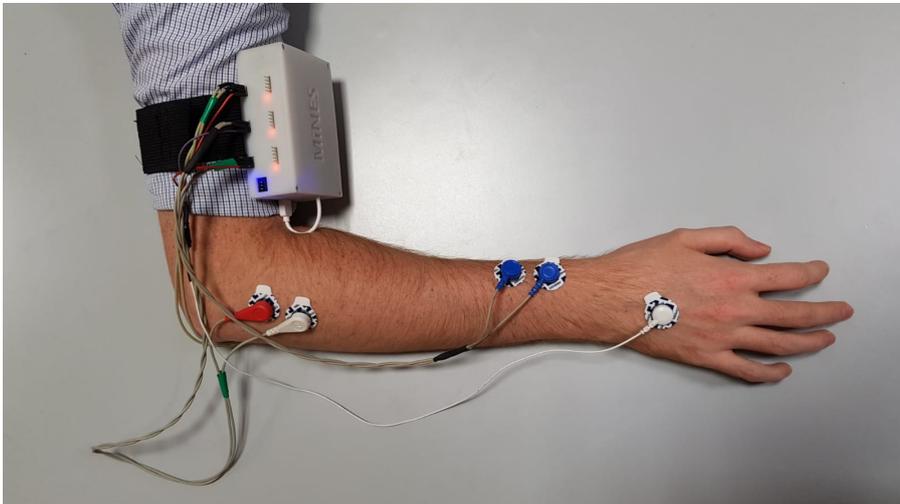


Figure 2.6: Structure of the system in [38] and wearability.

Chapter 3

Data Acquisition

This thesis work studies a possible implementation of hand gesture recognition, using a system based on the ATC event-driven feature of the forearm sEMG signals. The proposed system is composed of three acquisition boards (which acquire sEMG signals and process them to obtain the ATC values) and an Apollo3 Blue Micro-Controller Unit (MCU) with an ARM Cortex M4F DMA microProcessor (μ P). The thesis work focuses on firmware optimization on the ARM μ P as well as software on Matlab® environment, in order to obtain the lowest power consumption possible, with a latency suitable for real-time applications (< 300 ms) [39]. The thesis goal is to compare the performance of three machine learning algorithms in terms of consumption, latency and accuracy in predicting movements. The dataset used involved 25 healthy people, each of whom performed five movements in five repeated sessions.

3.1 System Configuration

The system analyzed in this thesis work is based on a previous work [38]. The *Analog Front-End (AFE)* is the part of an analog circuit that has the role of making the input signal suitable to be processed by the microprocessor. The AFE provide raw signal amplification (i.e., op-amp or INA) and filtering (i.e. Sallen-Key filter). In our case, AFE is responsible for both the conditioning of sEMG signal and the generation of TC events. The system is composed by two parts, as shown in Fig. 3.1:

- Three acquisition channels, which filter and amplify the sEMG signals acquired from the surface electrodes, also extracting the TC signals;
- The ultra low power *AmbiqMicro Apollo3* board [40], equipped with an ARM

Cortex-M4F DMA processor, which manages the sEMG signal acquisition and computes the online gesture prediction.

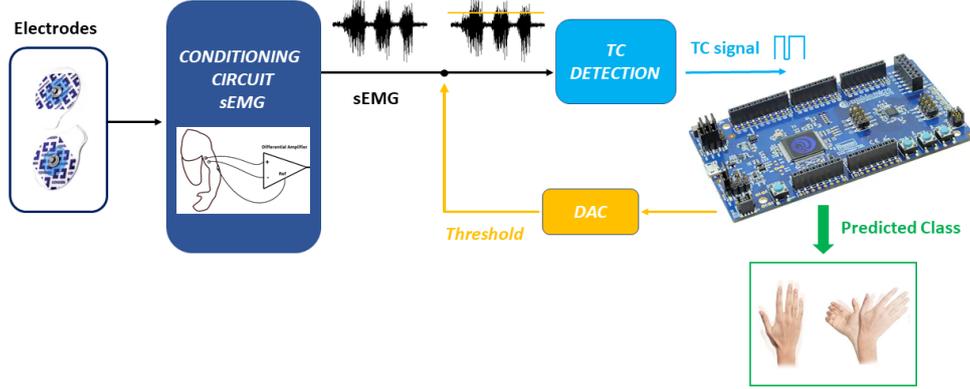


Figure 3.1: Block Diagram that shows the various steps of the sEMG signal acquisition, conditioning and analysis, for the correct prediction of the gesture.

3.1.1 Acquisition Channel

The acquisition of sEMG signals is performed applying standard pre-gelled surface electrodes on the skin of the forearm. The channel itself is a custom Printed Circuit Board (PCB) made with off-the-shelf components, obtained improving an older version [38]. The sEMG's conditioning has three input from three electrodes. As represented in Fig. 3.2, the AFE has three inputs because the sEMG signal is acquired using a differential approach. Therefore, in this configuration, two exploring electrodes are placed on the muscle belly to detect the muscular activity, and an additional one acting as reference electrode is located in an electrical-neutral area (i.e., bone prominence).

Overvoltage Protection

The overvoltage protection allow us to preserve the circuit from excessive out-range voltage inputs. In fact, the high or negative voltage could damage all the board because the components have a limited input voltages range.

Voltage Follower

Now, the signals can pass through a Voltage Follower to decouple the rest of the circuit. A voltage follower is a configuration of the Operational Amplifier (Op-Amp) featuring a negative feedback obtained by connecting the inverting input with the output. The features of this configuration are a unity gain and a very high input impedance (hundreds of M Ω) that allow isolation of the output from the signal source.

Differential High Pass Filter - 30 Hz

The 2 signals are filtered by means of a second order differential high pass filter to attenuate the movement artifact noise. The term 'passive' implies that no active component (an op-amp), but only resistors and capacitors are used to implement the filter.

Instrumentation Amplifier

Instrumentation Amplifier is a particular type of op-amp suited for amplifying biological signals, since it has a stable gain (typically adjusted via one or more resistor outside the chip), very high common-mode, differential mode impedance and low output impedance. This stage has a fixed gain of about 500 V/V, which will be further increased by next stage.

Gain Selector

The gain selector allows to set an adjustable gain on the board. In fact, it is possible to have an amplification of the input signal of a factor x1 (500 V/V), x2 (1000 V/V), x3 (1500 V/V), x5 (2500 V/V) or x6 (3000 V/V).

Low Pass Filter - 70 Hz

This filter is used to provide negative feedback to reference voltage of the INA and allow us to remove the low-frequency component introduced by the INA. In addition, compared to the previous version of the board [38], the cut-off frequency has been shifted from 10 Hz to 70 Hz, to solve the problem of power-line interference that will be analyzed in section 3.1.2.

This type of filter is defined as "active" because it is composed not only by resistor and capacitor (passive component) but also by an op-amp.

DRL Circuit

A Driven Right Leg (DRL) circuit is a circuit that is added to sEMG signal amplifiers to reduce common-mode interference. The middle dynamic voltage generated

in the INA loopback filter by the voltage divider feeds also the reference electrode through a DRL.

Low Pass Filter - 400 Hz

A second order Sallen-Key low pass filter is placed to eliminate the frequency over than 400Hz which are not part of the EMG signal.

Voltage Comparator

A voltage comparator is used to obtain the Threshold Crossing (TC) signal. It compares the pre-processed sEMG signal with an adjustable threshold that is usually set an hysteresis of 30 mV ensures a stable commutation between the digital high and the digital low state. This allows us to obtain a *quasi-digital signal*.

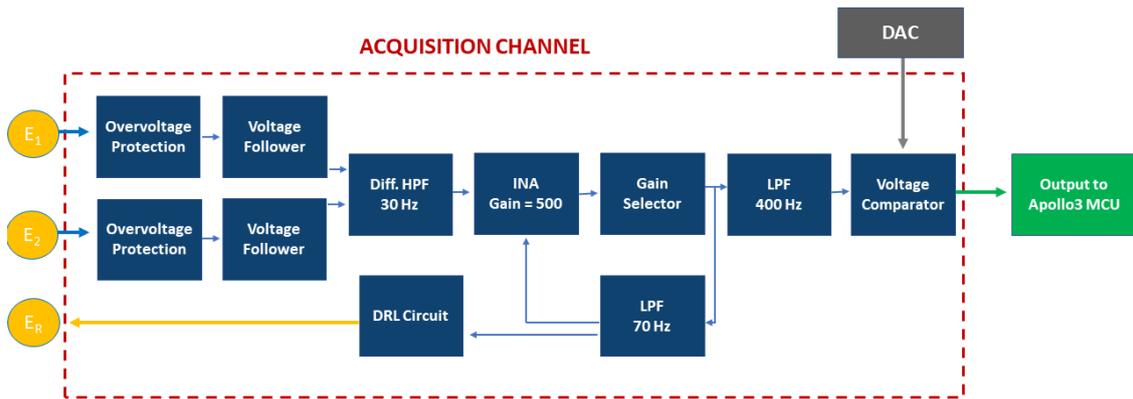


Figure 3.2: Structure of the acquisition board.

3.1.2 Power-lines Interference Issue

In this section the problem of power-line interference is presented and analyzed. Two analog solutions have been implemented in PCBs.

Overview

Modern biomedical amplifiers have a very high common mode rejection ratio. Nevertheless, recordings are often contaminated by residual power-line interference. In

order to better study this problem, at this stage sEMG signals were taken from many people and various forearm muscles.

A DAQ (model 6259) from National Instruments was used for signal acquisition. In Fig. 3.3 is possible to observe an example of sEMG signal corrupted by a very high presence of power-line interference noise.

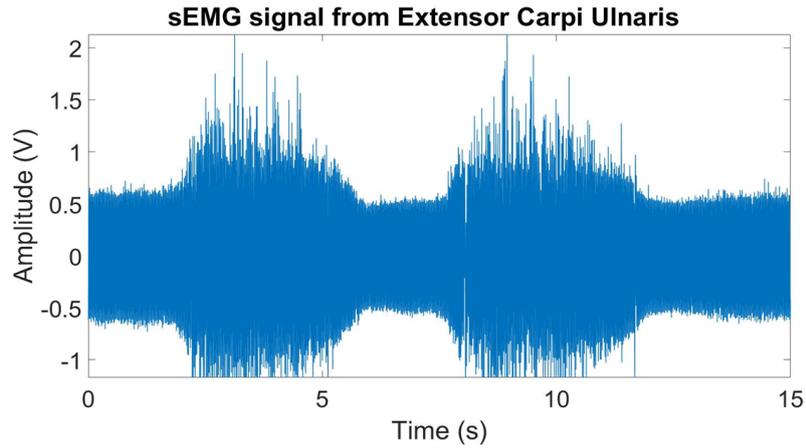


Figure 3.3: Example of sEMG signal extracted from *Extensor Carpi Ulnaris* corrupted by power-line interference.

Once the sEMG signals have been acquired, the entire processing phase has been carried out thanks to Matlab software. It was also possible to evaluate the *Power Spectral Density (PSD)* of the signal, as shown in Fig. 3.4.

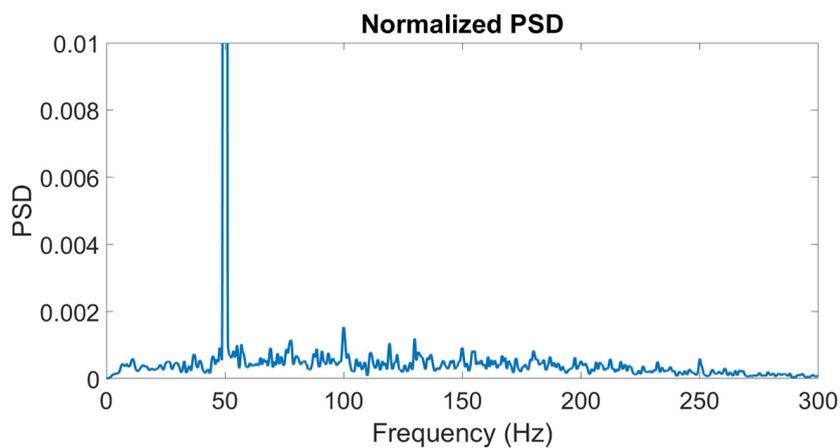


Figure 3.4: Normalized PSD.

It is clearly visible that almost all the signal power is concentrated at a frequency of 50 Hz, i.e. power-line frequency (in Europe and most of Asia). The problem has been greatly solved thanks to two innovations on the new acquisition boards:

- The use of the DRL circuit;
- The deviation of the Low Pass Filter cut-off frequency from 10 Hz to 70 Hz.

Solution 1: DRL Circuit

The DRL circuit is often used to reduce the problem of *Electromagnetic Interference (EMI)* in biopotential acquisition systems [1].

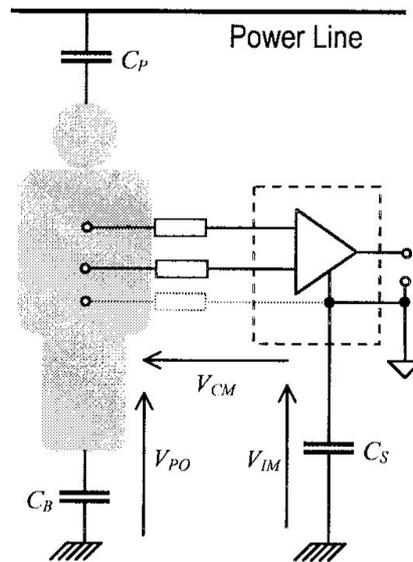


Figure 3.5: Model to illustrate displacement currents produced by electric field coupling [1].

Fig. 3.5 illustrates a power-line interference situation, where coupling has been represented with capacitors C_P and C_B : C_P identifies the parasitic capacitance that connects the subject to the network line; C_B identifies the one that connects the subject to the protection earth [1].

The displacement currents flowing through coupling capacitances, impose on the patient a nonzero potential V_{PO} (ground referenced) which can be decomposed into two voltages: an isolation mode voltage V_{IM} between ground and the amplifier's common, and a common mode voltage V_{CM} between patient and amplifier's common. By connecting the patient with a third electrode, it is possible to reduce the common mode voltage V_{CM} when there is a low impedance between the skin and the reference electrode. In order to overcome this constraint, the DRL circuit is

used, which allows the reduction of the common mode voltage through a negative feedback loop. The signal is amplified, reversed and injected back to the patient via the third electrode as shown in Fig. 3.6, reducing the impedance by a value dependent on the gain factor of the amplifier. [1].

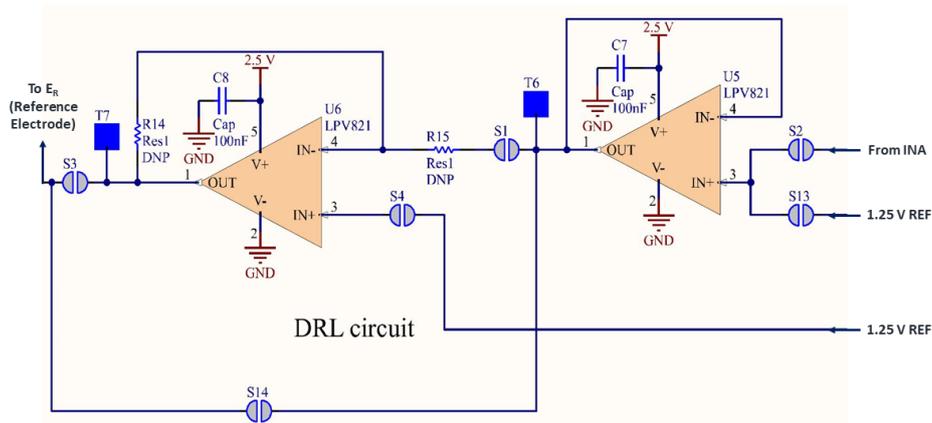


Figure 3.6: DRL Schematic circuit. The solder points have been inserted into the circuit to allow the user to choose whether or not to use the DRL circuit.

Solution 2: New LPF cut-off frequency

The Low Pass Filter, shown in Fig. 3.7, provides negative feedback to reference voltage of the INA and allow us to remove the low-frequency component introduced by the INA.

In this revised version of the channels, the filter cut-off frequency has been changed from 10 Hz (from the boards of the previous work [38]) to about 70 Hz.

This important change has allowed us to solve the problem of power-line interference whose characteristic frequency, as mentioned above, is 60 Hz (North America) or 50 Hz (Europe and most of Asia). It was decided to eliminate frequencies below 70 Hz, as the higher information content of the SEMG signal is concentrated for frequencies up to 150 Hz.

Moreover, thanks to the use of the ATC technique, it is not necessary to take all the information content of the signal. This advantage makes the filtering operation less critical.



Figure 3.8: The AmbiqMicro Apollo3 Blue Evaluation Board.

Table 3.1: Apollo3 Blue technical sheet.

Max operating frequency	48 MHz - TurboSpot 96 MHz
MCU	32-bit ARM Cortex-M4F DMA
MCU min power	6 $\mu\text{A MHz}^{-1}$
Flash/SRAM	1 MB/384 kB
VDD	1.8 - 3.6 V
I/O	I ² C/SPI (6x) - UARTS (2x)

precise, 32.768 kHz crystal has been used for the time window implementation needed by the ATC. The buck converters have been enabled to guarantee a really low power consumption by the μP .

3.1.4 Firmware

The firmware for the MCU was written using Keil $\mu\text{Vision IDE v5.30}$, as suggested by the AmbiqMicro developers, due to its high compatibility with the ARM product family. In fact, many specific ARM libraries are available directly in Keil Pack Installer, making it possible to add or delete any library package. The Digital Signal Processing (DSP) library was chosen for this application, based on its low calculation cost and ease of use; this library is a native ARM part of the CMSIS package and provides very useful support for matrix calculations. In addition, many support packages for the board are provided by AmbiqMicro itself, allowing

the developer to use high-level functions to interact with every single small component of the system, even performing complex tasks [39].

The program is divided in three main parts: firstly, all the needed constants and variables are defined, then useful routines are implemented and last the main function executes the desired commands. Here there is a brief list of the implemented routines useful for running the program in real-time:

- **Parameter Initialization:** The first part of the firmware is entirely dedicated to the initialization of useful parameters during real-time data computation.
- **Prediction Algorithm:** Immediately after the initialization and definition of the necessary variables and constants, its possible to make a choice on the type of algorithm to be used for the classification of movements in real-time. Through a simple control it is possible to use a supervised type classifier (ANN or SVM), or a non-supervised learning method (clustering through K-Means). All the data required by the algorithms have been written in dedicated sections within the program.
- **Timer Configuration:** a timer is set according to ATC requirements. The best trade-off to generate time window under 300 ms without dissipating too much energy it is to use the external high precision low frequency clock ($F = 32.768$ kHz). An interrupt is then set to a compare value, obtaining a specific time window, and corresponding service function is registered. REPEAT mode is activated, so the timer restarts from the beginning after the interrupt occurs. If a time measurement is required for performance reasons, another timer would be activated, with a frequency a quarter of the selected Hard Clock (HCLK), typically 6 MHz [39].
- **LEDs Configuration:** LEDs are configured according to library functions, they are made controllable from the μ P and then switched off.
- **GPIO Configuration:** the three pins required to collect data from the three AFEs are activated as input, with interrupt on the rising edge. For each of them an appropriate function is registered in the interrupt service table. No hazards occur in case of multiple concurrent interrupt are received. The interrupts are then cleared for security reasons and the GPIO master is enabled [39].
- **Interrupt Routines:** Each interrupt routine allows you to define specific actions that must be taken whenever a certain event occurs.

In particular, a routine that is called when a signal is detected at the input of a defined GPIO has been implemented .

Another routine is called each time the ATC window ends, to evaluate the number of detected TC events.

- **Initial Configuration:** some basic function are called depending on the value of environmental constants; if high frequency is required, the MCU would be set to 48MHz instead of 24 is enabled if needed, low power mode is entered, activating the buck converters and the debug interface is configured, if the mode is needed.
- **Running Loop:** The program enters a *while loop* for all the acquisitions set by the user. Inside the loop, if no interrupt is generated, the board is set to *Deep Sleep* mode. Up to the number of acquisitions set by the user, the program will execute all the operations useful for signal acquisition and movement classification.

The board could be powered via USB cable or with an external supply, like a battery or a voltage generator, to be connected directly on the power pin on the power header of the board.

3.2 Performed Movements

The list of gestures to be executed from the volunteers has been selected according to some recent literature works [25, 42] and without requiring an excessive effort from the subject. The final list of movements is the following:

1. *Wrist Extension*;
2. *Wrist Flexion*;
3. *Wrist Radial Deviation*;
4. *Wrist Ulnar Deviation*;
5. *Hand Grasp*
6. *Idle State/Rest Position* of the hand, necessary to consider as a class when no movements are done as well.

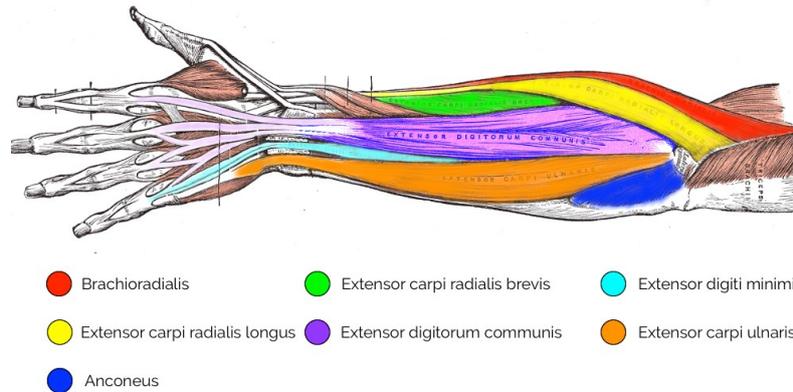


Figure 3.9: Superficial muscle on medial section of the forearm.

The muscles necessary to perform the desired movements are *hand extrinsic muscles* (called extrinsic because they are in the forearm, out of the hand area), generally originated in the elbow area and terminating in the metacarpal area. They are divided in superficial and deep ones; the second are not mentioned because they are useless for this work because their position in the forearm is not suitable for using surface electrodes. Only the superficial muscles have been then taken into account (Fig. 3.9). All the below match movement-muscles have been deduced from the Eaton Hand online book [43, 39].

Wrist Extension

The wrist extension is the act of move the back of the hand towards the distal forearm. The mainly used muscles are the *extensor carpi radialis longus*, the *extensor carpi radialis brevis* and the *extensor carpi ulnaris*, together with some deep muscles [39].

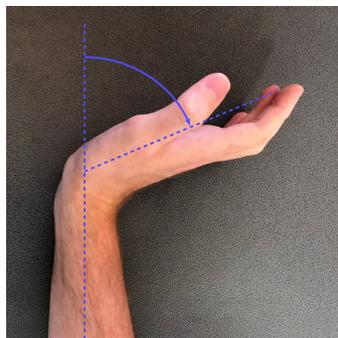


Figure 3.10: Wrist extension.

Wrist Flexion

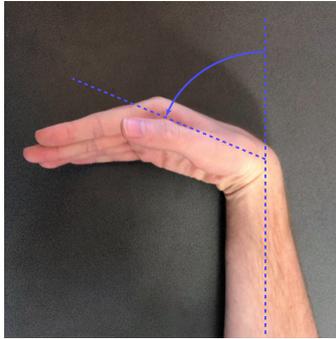


Figure 3.11: Wrist flexion.

The wrist flexion is the movement of the hand palm, towards the inner arm. The involved muscles are *flexor carpi radialis*, *palmaris longus* and *flexor carpi ulnaris*, as well as *flexor digitorum superficialis* and *profundus* [39].

Wrist Radial Deviation

The hand is moved up, following the thumb direction, in order to perform the radial deviation. Muscles involved are *abductor pollicis longus*, *flexor carpi radialis*, *extensor carpi radialis longus* and *brevis* [39].

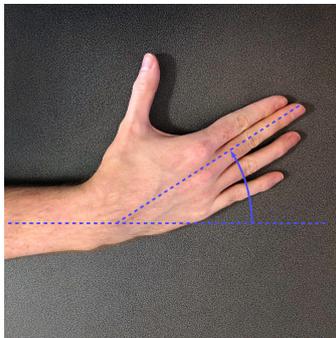


Figure 3.12: Wrist Radial Deviation.

Wrist Ulnar Deviation

The hand is moved down, in the little finger direction, to perform ulnar deviation. The useful muscles are *extensor carpi ulnaris* and *flexor carpi ulnaris*.

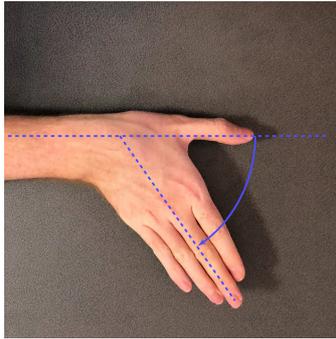


Figure 3.13: Wrist Ulnar Deviation.

Hand Grasp

Hand grasp is the action of closing all the finger towards the hand palm. *Flexor digitorum* and *palmaris longus* are the most used, together with many intrinsic muscles of the hand.



Figure 3.14: Grasp.

Idle State/Rest Position

The idle state is performed trying to relax all the above described muscles.

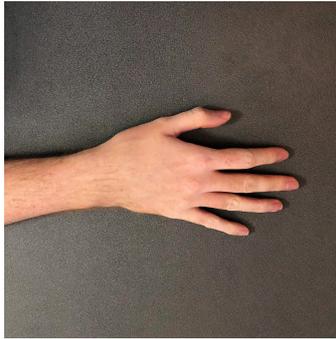


Figure 3.15: Idle.

3.3 Acquisition Protocol

In this section it will briefly describe the protocol used in the previous work for the collection of the sEMG signals from the 25 volunteers. The dataset acquired through this protocol will be the starting point of my thesis work, for the implementation of machine learning algorithms and the comparison of performance with the previous work [38].

3.3.1 Electrodes Placement

The collection of the sEMG signals presented many critical issues, due to many factors already discussed in 1.3.4. The electrode placement is one of the critical aspects. In fact, even using relatively small electrodes, the muscles are too close one to the other to not have *crossstalk* between them. Moreover, some muscles are so thin to be even smaller than the 24mm electrodes. Thus, an initial study on the optimal placement has been done, to adapt to the different morphology of the forearm [39].

This study considered the number of channels to be used, according to the five active movements determined in section 3.2. Tests realized resulted in a good discernment between classes: the final configuration of the electrodes placement is shown in Figure 3.16. In particular:

- First couple of electrodes is placed on the *palmaris longus*, which originates from the *medial epicondyle* and has its insertion on the proximal superficial *palmar fascia*. The electrodes should be positioned on the lower area of the palmaris, near to the *flexor carpi ulnaris*, in order to take into account some good cross action from it. The main contribution is to the *Flexion movement*, but some useful value are recorded also during *Ulnar Deviation* and *Grasp*.

- Second pair is placed on the superficial area of the *abductor pollicis longus*, which has its origin in the *radioulnar interosseous membrane* and terminates on the radial dorsal base of the *thumb metacarpal*. Placement of the electrodes should be made near the wrist, where the muscle become superficial. These electrodes are mainly used in *Radial Deviation*, but have nice effects on *Extension* and *Grasp*.
- The third and last pair has to be placed on the *extensor carpi ulnaris*, which has the origin in the lateral border of the *distal humerus* and the insertion on the dorsal base of the *small metacarpal*. Main effects are obviously on the *Extension movement*, but they are also necessary for *Ulnar Deviation*.
- Last, the reference electrode has been placed on the hand back, near the wrist, in a bony electrical neutral area, in a way that does not prevent correct execution of the movements [39].

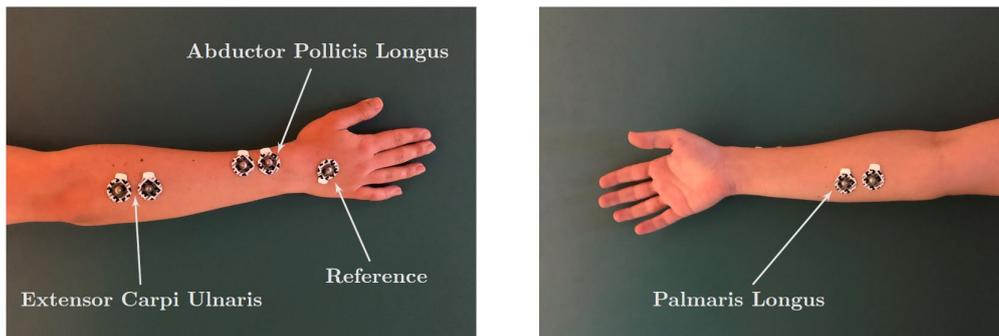


Figure 3.16: Electrode placement on both sides of the forearm [38].

As it is possible to see, the channels are used combined with others to obtain a possible movement, allowing to reduce the number of electrodes used, while keeping high the number of movements that are classified.

3.3.2 Training and Test Protocol

After the initial test phase, which it has been discussed in the previous section, an *in vivo* experimentation has been launched [38], in order to collect from different people enough data for the classifier to be trained and tested.

In this part 25 people have been involved, 16 males and 9 females (with an age between 23 and 37 years old). The participants have been divided in two different

groups: 20 people have been involved to the classifier training, while the remaining 5 have been enrolled in the online testing phase. The two sessions have been performed subsequently, in different days, without in any way replicating environment conditions, in order to guarantee that training data and testing ones were not dependent one to the other [39].

For both groups the initial calibration phase is in common: each subject has to sit, in a way that the right arm could stay above the table, in an horizontal position, and that the hand is free to move.

After the explanation about the study, the subject has the electrodes placed on the forearm. This placement is really critical, as written in Sec. 3.3.1, because of the narrowness of the muscles and their small cross section. A bad positioning could bring to a very bad quality of the acquired data, with an accuracy degradation up to 30% [44, 39].

To prevent this issue, a preliminary calibration is performed, acquiring sEMG signals for a small time period and visualizing the results:

- The volunteer execute one movement at a time, starting from the rest position and keeping the gestures once reached. The acquisition lasts 6.5 s, during which 50 values are acquired over the 130 ms time window.
- The hand returns in the rest position and a pause of 5 s is performed to avoid muscular fatigue.
- A different movement is acquired on the available 6.5 s period. The routine is repeated until all the five active movements are performed.
- Obtained data are then saved to a file and moved to Matlab environment. Here are 3D plotted and some visual observation takes place.
- If necessary, some electrodes could be placed in a slightly different way, to enhance classifier performances.

Once the calibration phase has been completed, the training and prediction phases have begun.

Training Protocol

During training each individual had to perform only the five active movements, always in the same order. The acquisition period is doubled with respect to the calibration phase, to ensure enough data is collected and to allow the subject to perform the movement in a clean way, without being too fast so no more noise is

introduced.

Data acquisition was made with the previous version of Apollo3 Blue EVB used in my thesis work. In fact, in [38] the Apollo2 EVB was used.

In Table 3.2 main features are reported.

Table 3.2: Apollo2 technical sheet.

Max operating frequency	48 MHz
MCU	32-bit ARM Cortex-M4F
MCU min power	10 μ A MHz ⁻¹
Flash/SRAM	1 MB/256 kB
VDD	1.8 - 3.6 V
I/O	I ² C/SPI (6x) - UARTS (2x)

The substantial difference between the two models mainly concerns consumption. The debug mode of the Apollo2 has been used to stop and restart the flow when necessary. No skin treatment is performed at the begin, to ensure robustness of the system even with not optimal condition on the forearm.

1. The supervisor remembers to the subject which is the movement to be performed.
2. A Start command is given to the volunteer and the debug is told to continue the program.
3. The person reaches the desired gesture, then comes back to the rest position and executes the movement again.
4. When the 13 s period is finished and all the 100 sets of data are acquired, a Stop command is given to the subject.
5. A rest of 5 s is observed. If there are still movements to be execute, the flow goes back to point 1.
6. If all the movements are done, a pause of 1 min is performed, letting the person to lay the arm on the table.
7. Data is saved on the computer. Flow restarts from 1, unless five session have already been finished.

Test Protocol

After the classifier is trained, the remaining 5 people performed the same movements plus Idle state for the test phase. The acquisition period has been set to 5.2 s (40 windows of 130 ms), in order to keep the execution low and not let the subject arm become tired, as well as to have 1000 acquisition packet for each subject. The debug mode is used like in training protocol.

1. The supervisor remembers to the subject which is the movement to be performed.
2. A Start command is given to the volunteer.
3. As soon as the person reaches the desired gesture, the debug is told to continue.
4. The gesture has to be kept steady for the 5.2 s period.
5. When the 40 sets of data are acquired, a Stop command is given to the subject.
6. A rest of 5 s is observed. If there are still movements to be execute, the flow goes back to point 1.
7. If all the movements are done, a pause of 1 min is performed, letting the person to lay the arm on the table.
8. Data is saved on the computer. Flow restarts from 1, unless five session have already been finished.

Chapter 4

Gestures Recognition Algorithms

The data used for the training phase of the three implemented ML algorithms were acquired in a previous work [38].

Once they are saved in the computer, the output label is added, depending on the movement that was performed, to communicate the classifier which the desired value is. For this thesis work ANN, SVM and K-means algorithms were chosen for three main reasons:

- To make a performance comparison between supervised and unsupervised learning algorithms;
- To make an easily implementation on the microcontroller chosen for online prediction. The CMSIS-DSP libraries of ARM Microcontroller, which has already been discussed previously, reduce computational cost significantly;
- To make a performance comparison with studies from other research groups. These algorithms are the most used by the scientific community for the recognition of hand movements.

The training of the algorithms has been performed offline on the Matlab[®] environment. This implementation allows us to make a relatively fast and repeatable training routine and provides the usual powerful tools of the software. The techniques used to implement the classifiers have been studied in [45]. The prediction has been implemented online on the Apollo3 Blue MCU.

4.1 Offline Training

The input datasets (e.g., the one in Fig. 4.1) are loaded separately, guaranteeing no correlation among data between them. Data matrices are divided into training set and validation set.

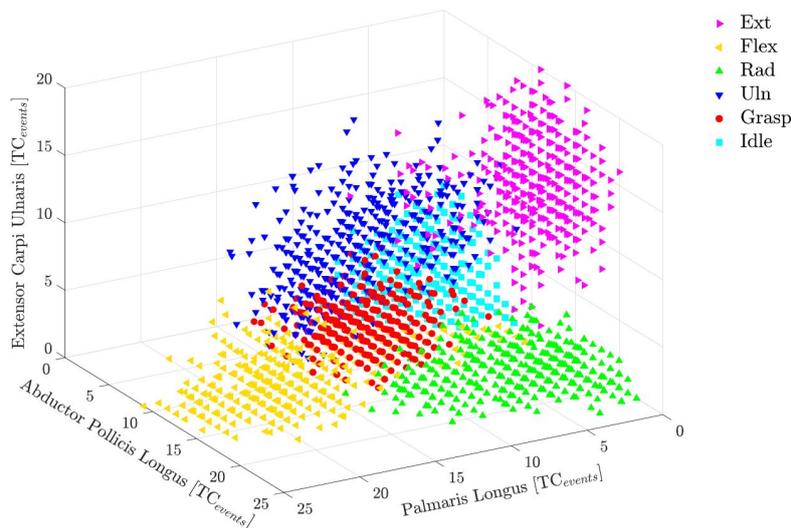


Figure 4.1: Dataset from one subject [38]

4.1.1 SVM Training

The main objective of the training phase for the SVM algorithm was to obtain the model to be imported into Apollo3 for online prediction. The implementation of the SVM algorithm required numerous steps and variations, given the problems related to its computational cost already discussed in Chapter 1.

In a first phase, exploiting the notions learned in [45], a script has been implemented to obtain the model. The main problem, however, is that the output model of the algorithm was too "heavy" for its online implementation on microcontroller.

Therefore I decided to use the *LIBSVM* library. It is an open source ML library, developed by National Taiwan University, for classification, regression and distribution problems [46].

The objective of the tool is to allow an easy use of SVMs. The package includes three functions: *svmtrain*, *svmpredict* and *svm-scale*. The first function trains the SVM through the training set, the second one tests the SVM through a test set

and the last is used to normalize the data in the proper range, usually $[0,1]$. The package can be used in a command line or exported to other languages like Java or, as in the case of my thesis work, Matlab.

The principal steps in using the tool are:

1. Data conversion into input in the format of the tool you intend to use;
2. Data normalization in the appropriate range;
3. Choice of the type of kernel function to use;
4. Choice of the C parameters and kernel parameters;
5. SVM training.

The tool provides several training options. First of all it allows you to choose the type of SVM problem to solve, through the *-s option*. You can choose among:

- C-SVC (multi-class classification) default parameter;
- v-SVC (multi-class classification);
- One-class SVM;
- SVR (regression);
- v-SVR (regression);

Through the *-t option* you can choose among 5 different kernel types:

- Linear;
- Polynomial;
- Radial Basis Function (RBF);
- Sigmoid;
- Precompiled Kernel;

Linked to kernels there are a number of options that allow you to set the desired parameters, such as:

- *-d*: to set the polynomial kernel degree;
- *-g*: to set the gamma in kernel functions;

- *-r*: to set r in the polynomial kernel;
- *-c*: to set the C parameter in multiclass classification problems.

The function to be used in Matlab is:

model = svmtrain(training-label, training-matrix , 'libsvm-options');

where *training-label* is an $L \times 1$ size vector containing the labels of the training examples; *training-matrix* is an $L \times n$ matrix containing the feature vectors, and n is their size; *libsvm-options* is a string containing the options with which you want to train the SVM.

In the case of my study, the SVM Multiclass Classification type and the RBF Kernel type has been used, having a number of output labels equal to 6. This function returns the *file.model*, containing the following model parameters:

- *Parameters*: the parameters;
- *nr-class*: number of classes;
- *totalSV*: the cardinality of the set of support vectors (SV);
- *rho*: variable b (bias) of the decision function;
- *Label*: class labels;
- *nSV*: the number of support vectors for each class
- *sv-coef*: is the value of α associated with SVs;
- *SVs*: the support vectors.

Due to the small memory size of the microcontroller, one of main design goals was the minimization of SVs number.

In order to obtain the best classification performance from the algorithm, it was decided to implement the *Cross-Validation Method*. This method consists in the subdivision of the dataset in a fixed percentage of data for training and validation set: a *for loop* has been created with a *Repetition Number* equal to 50 that allowed me to vary at each iteration the matrices within the training and validation set.

This process improved considerably the level of accuracy of the classification, reaching a percentage of 95.3%, using less than 3000 SVs. Finally, the model that obtained the maximum accuracy value was saved and imported into the microcontroller memory for online movement prediction.

4.1.2 K-Means Training

The objective of K-means training is to find the value of the centroids that allow to reach the highest accuracy in classification. As discussed in Chapter 1, one of the main problems of the K-means algorithm is to often fall in a local optima: this depends strongly on the centroids initialization and the number of iteration.

To solve this problem, the Cross-Validation method has also been used in this algorithm. Many tests have been made to find the right compromise among the number of iterations, the number of repetitions and the right computational cost. The final parameters used are:

- *Iteration Number* = 50. This value corresponds to the number of times the algorithm iterates to search for centroids.
- *Repetition Number* = 100. This value indicates how many times the matrices are re-initialized for the division of the training and validation set.
- *Matrices Number* = 20. It has been chosen to use the whole dataset because, since K-Means is an unsupervised learning algorithm, adding more data can improve performance in data separation.

The maximum accuracy reached at this stage is 83.35 %. The centroids that achieved this result were saved and transferred to the Apollo3 MCU.

4.1.3 ANN Training

As regards the implementation of the code for the NN algorithm some considerations and trials have been made in a previous work [38], looking for the best implementation possible.

With fewer data than the final ones, a preliminary study on the NN has been performed. 63 different NN have been trained and tested, each one considering 10 different regularization values (λ), starting from 1 hidden layer made of only 10 nodes and ending with 3 hidden layers with 30 nodes each.

Considering the performance achieved in these tests, the final NN architecture was configured as reported in Table 4.1

Table 4.1: Neural Network Architecture.

Layers	Nodes	Regularize(λ)	Val. Error	Accuracy(%)	Training Time(s)
2	26	0.010	0.630	92.31	2110

As in the case of the two algorithms previously analyzed, the main parameters of the neural network have been saved into the microcontroller memory, to allow the online prediction of the gestures.

4.2 Online Prediction

The characteristic parameters obtained offline from the three analyzed algorithms have been transferred to the Apollo3 MCU.

Every time the ATC window ends, the timer interrupt arises and the board exits the deep sleep mode. The service routine takes the TC values counted by the GPIO interrupts transforming them into the inputs for prediction algorithm.

Three routines have been implemented, one for each ML algorithm, giving to the user the possibility to select the desired algorithm for real-time gesture recognition.

- **SVM routine**, as in the training phase, uses the LIBSVM library. In this case, however, the library was suitably modified, in order to fit the hardware constraints of the microcontroller.

The prediction function receives the TC values in inputs and computes the output class according to the model specifications.

- **K-Means routine** allows to calculate the euclidean distances between the ATC values and the optimized centroids. The centroids are saved as an MxN matrix, where M corresponds to the number of classes and N corresponds to the number of TC channels. Once the euclidean distances are calculated, the result of the predicted class corresponds to the value of the lowest distance.
- **NN routine** NN routine computes the forward propagation, using the trained weights, and employing the sigmoid as activation function in order to avoid divergence.

Chapter 5

Experimental Results

This chapter presents the measurements carried out in order to evaluate the system performance in terms of accuracy, latency and power consumption. Power and Latency measurements were made using two different frequencies of the Apollo3 High Frequency RC oscillator (HFRC). It has a nominal frequency of 48 MHz but in the tests, I chose to use also the second available frequency (24 MHz) in order to assess performance at different working frequencies.

5.1 Classifier Accuracy

The accuracy of the classifier has been measured directly on the Apollo3 Blue MCU, implementing specific routines able to take count of the prediction and to understand if it has been correct or not.

One *Confusion Matrix* for each algorithm and movement has been created, considering the whole test set of 2401 elements. If all the movements were performed correctly, this value should be on the main diagonal.

The following parameters have been chosen as statistical assessment:

- **Accuracy:** measures the percentage of the exact predicted value in the total number of instances. It is the inverse of the error rate and varies from 0% (worst) to 100% (best).
- **Precision:** is the percentage of the corrected positive values out of the total positive model values (right or wrong). It varies from 0% (worst) to 100% (best).
- **Recall:** or sensitivity, is the percentage of the corrected positive values out of the total positive instances. It varies from 0% (worst) to 100% (best).

- **F1 score:** is the harmonic average of the Precision and Recall metrics. It varies from 0% (worst) to 100% (best).

5.1.1 Support Vector Machine

As already discussed in the previous chapter, the SVM algorithm has considerable critical issues from a computational cost point of view. Investigating the algorithm, it emerged that the most critical point is the calculation of the exponential function in the output class prediction operation.

To try to solve this problem, the exponential function has been replaced with *Taylor-Mc Laurin's Series of the Exponential Function*. This new solution has brought considerable advantages on the computational cost of the algorithm, greatly reducing the time in the prediction. On the other hand, however, the performance of the classifier has suffered some small deterioration. Below statistical parameters are analyzed using both solutions.

Prediction with Exponential Function

The statistics obtained are reported in Table 5.1. The average accuracy is 97.98%. It is possible to observe that Ulnar Deviation is the worst performing movements, but it is well compensated by the performance obtained by the other gestures.

Table 5.1: Statistical Result SVM (Exp.Func).

	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Ext	96.79	78.84	79.34	79.13
Flex	97.50	78.92	88.16	83.24
Rad	98.86	99.38	86.09	92.26
Uln	96.83	66.66	68.97	67.80
Grasp	98.08	80	81.36	80.67
Idle	99.83	99.94	99.82	99.87
Avg	97.98	83.95	83.05	83.87

Prediction with Taylor - Mc Laurin's Series Exponential Function

The statistics obtained for this algorithm are reported in Table 5.2. The average accuracy is 94.49%. Again, the Ulnar Deviation was the movement that achieved the worst performance. The percentages of accuracy and recall of the Grasp are low if compared to the other movements, which instead have excellent performance.

Table 5.2: Statistical Result SVM (Taylor Series).

	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Ext	94.67	98.28	30.98	47.10
Flex	95.79	97.22	41.42	58.09
Rad	98.08	94.34	80.21	86.70
Uln	94.46	44.21	56.03	49.43
Grasp	96.31	63.06	59.32	61.13
Idle	87.63	84.56	100	91.63
Avg	94.49	80.27	61.32	65.68

5.1.2 K-Means

The statistics for K-means algorithm are reported in Table 5.3. The average accuracy is 95.14%. This algorithm has some criticality in the classification of movements such Ulnar Deviation, Wrist Extension and Grasp.

Table 5.3: Statistical Result k-Means.

	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Ext	95	97.06	35.87	52.38
Flex	93.71	52.65	100	68.98
Rad	99.62	98.90	96.25	97.56
Uln	92.79	40	98.28	56.85
Grasp	93.46	62.14	35.16	44.91
Idle	96.25	99.74	94.72	97.16
Avg	95.14	75.08	76.71	69.64

5.1.3 Artificial Neural Network

The statistics obtained are reported in Table 5.4. The average accuracy is 97.14%. It is possible to observe that Ulnar Deviation and Wrist Extension are the worst performing movements. On the other hand, the classification of the other movements has achieved excellent results.

Table 5.4: Statistical Result ANN.

	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Ext	95.13	92.40	39.67	55.51
Flex	98.25	88.02	86.98	87.51
Rad	99.20	100	89.84	94.64
Uln	94.63	47.23	95.69	63.25
Grasp	98.37	81.10	87.28	84.08
Idle	99.92	100	99.87	99.94
Avg	97.59	84.79	83.29	80.82

5.2 System Latency

Latency has been measured on the Apollo3 Blue board, implementing a timer with a frequency of 6 or 12 MHz (dependent on clock frequency). The timer is started when the end of the ATC window occurs, it continues running through all the computations and it is stopped after the output class is defined. In this way it takes into account the whole computational phase, without neglecting anything.

As already explained at the beginning of the Chapter 5, the latency has been analyzed considering the two different frequencies of the HFRC oscillator.

In the case of 48 MHz clock, the average values obtained from the measurements are:

- 141.90 ms, for SVM algorithm with the use of the exponential function;
- 109.85 ms, for SVM algorithm with the use of Taylor - Mc Laurin's series exponential function;
- 62.6 μ s, for K-Means algorithm;
- 2.58 ms, for ANN algorithm;

Instead, in the case of a HFRC frequency of 24 MHz:

- 283.75 ms, for SVM algorithm with the use of the exponential function;
- 219.71 ms, for SVM algorithm with the use of Taylor - Mc Laurin's series exponential function;
- 124 μ s, for K-Means algorithm;
- 5.15 ms, for ANN algorithm;

Important considerations can be made thanks to the results obtained. The K-means and ANN algorithms can be used for real-time applications, because the total latency given by the sum of latency obtained and the time of the ATC windows is lower than the usual limit of 300 ms. As regard the SVM algorithm, the use of the Taylor exponential has allowed to decrease the total latency, but the margin is too small to use the algorithm implemented for real-time studies.

5.3 MCU Power Consumption

MCU power consumption analysis has been performed using *DMM7510* by *Tektronix*. The measurement has been made removing a small jumper on the board, usually inserted between board power and MCU VDD, and applying an high-resolution digital multimeter on its extremities. It has been downloaded the appropriate software that allowed to make the measurements and save the data directly on the PC.

Below are the power consumption results of the three algorithms. They are analyzed separately and, for each of them, a comparison in the use of the maximum HFRC clock frequency (48 MHz) and a frequency equal to half (24 MHz) is made. Obtained deep-sleep power consumption is very similar for the algorithms and is equal to 0.04 *mW*.

5.3.1 SVM Absorbed Current

In Figure 5.1 a comparison between the consumption of the two possible solutions for SVM algorithm is made. As it is possible to see, current absorption follows exactly the behavior of the μP , being low during the acquisition window and having a higher step during the calculation.

The computational cost of the algorithm is clearly visible in these figures: the microprocessor works on the class prediction calculation for almost all the ATC window, returning to deep-sleep mode for a few ms. The difference between the two solutions adopted affects power consumption. The average power consumption differs depending on the algorithm and the clock frequency used.

In the case of 48 MHz:

- 1.78 *mW*, for SVM algorithm with the use of the exponential function;
- 1.67 *mW*, for SVM algorithm with the use of Taylor - Mc Laurin's series exponential function;

Instead, in the case of 24 MHz HFRC:

- 1.28 mW , for SVM algorithm with the use of the exponential function;
- 0.93 mW , for SVM algorithm with the use of Taylor - Mc Laurin's series exponential function;

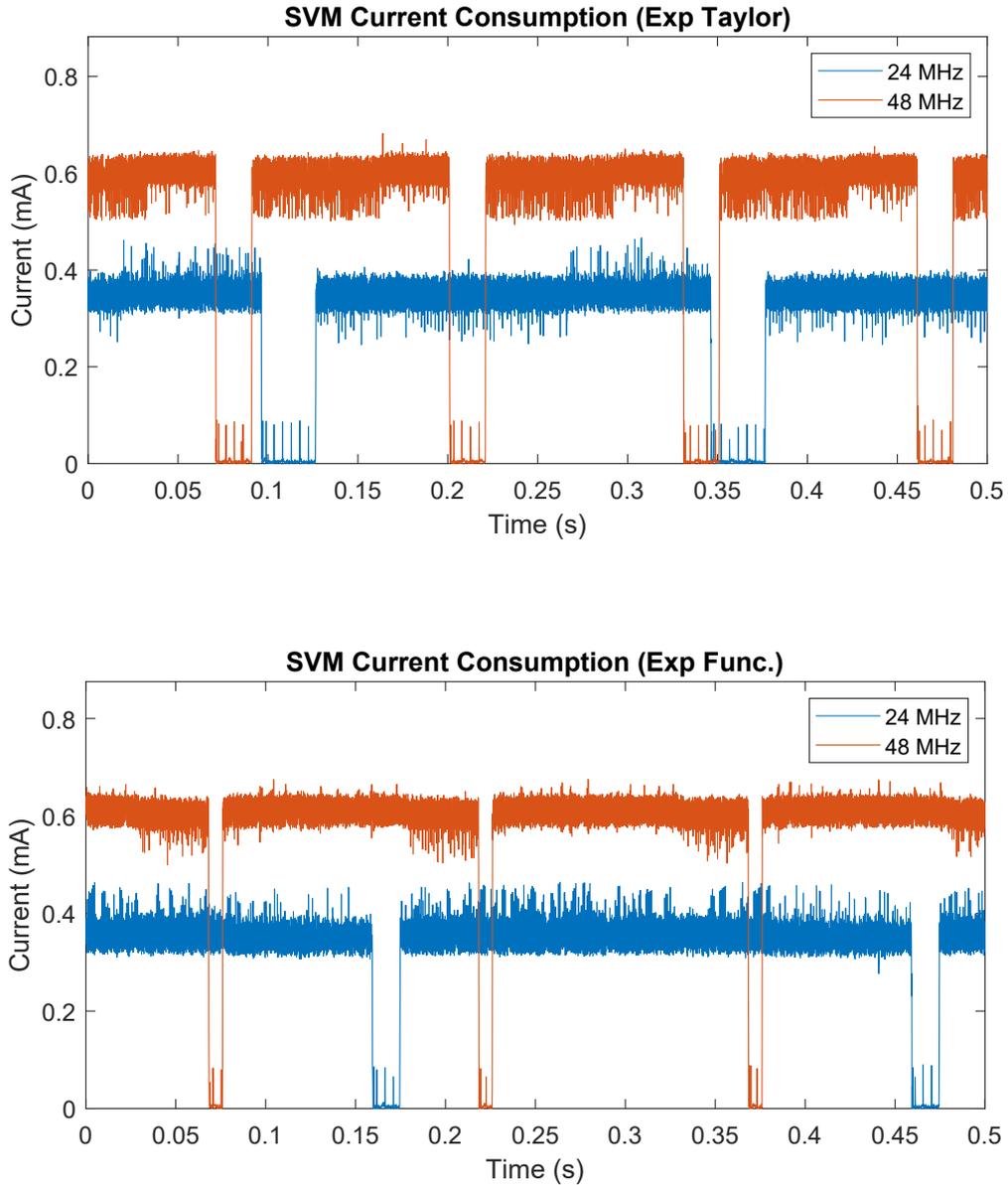


Figure 5.1: SVM current consumption, measured with the DMM7510 digital multimeter.

5.3.2 K-Means Absorbed Current

Figure 5.2 shows the current consumption related to the K-means algorithm. In this case the working period is less than 125 μs , and this also affects the result of average power consumption. In particular:

- 0.82 mW , for K-Means algorithm with a HFRC clock frequency of 48 MHz;
- 0.61 mW , for K-Means algorithm with a HFRC clock frequency of 24 MHz;

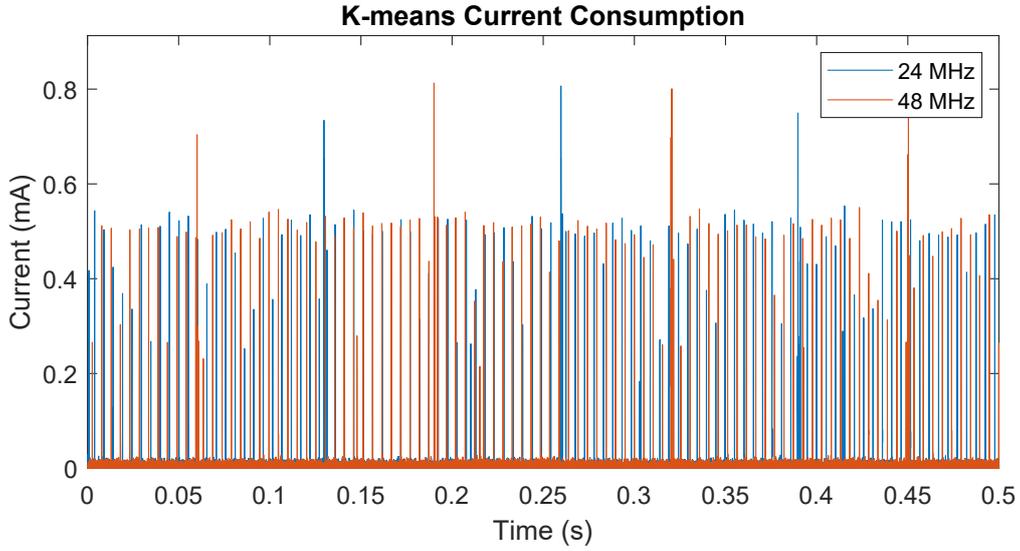


Figure 5.2: k-Means current consumption, measured with the DMM7510 digital multimeter.

5.3.3 ANN Absorbed Current

Figure 5.3 shows the current consumption of the ANN algorithm. In general ANN algorithm has a lower consumption than the K-Means, but totally the average consumption is comparable. In particular:

- 0.83 mW , for ANN algorithm with a HFRC clock frequency of 48 MHz;
- 0.53 mW , for ANN algorithm with a HFRC clock frequency of 24 MHz;

For a complete understanding, the average and the maximum power consumption of the 3 algorithms are shown in Table 5.5 and Table 5.6.

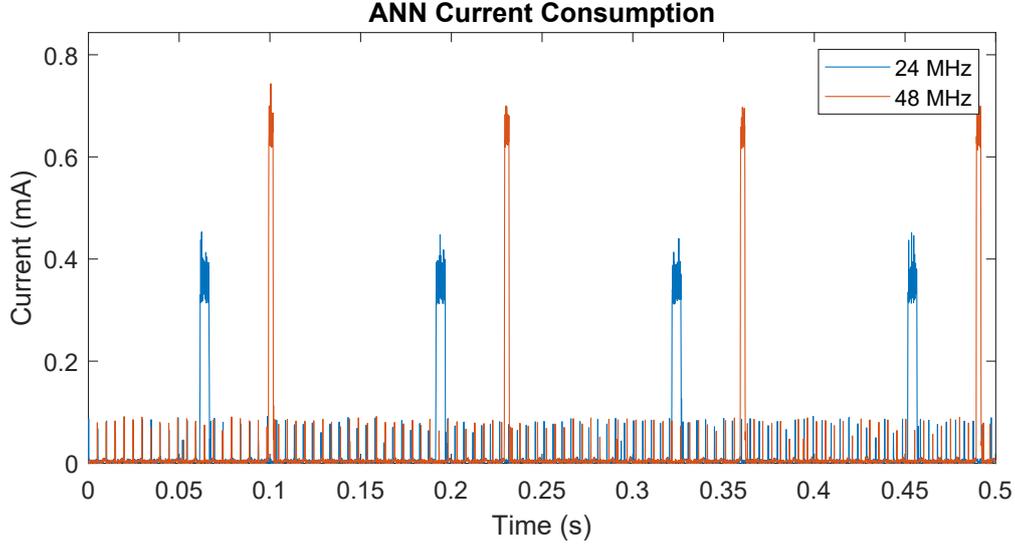


Figure 5.3: ANN current consumption, measured with the DMM7510 digital multimeter.

Table 5.5: Comparison of the power consumption of the algorithms with an HFRC clock frequency of 48 MHz.

	Avg Power Cons(mW)	Max Power Cons(mW)
SVM (Exp Func)	1.78	2.18
SVM (Taylor Exp)	1.67	2.04
K-Means	0.82	1.09
ANN	0.83	2.51

Table 5.6: Comparison of the power consumption of the algorithms with an HFRC clock frequency of 24 MHz.

	Avg Power Cons(mW)	Max Power Cons(mW)
SVM (Exp Func)	1.28	1.52
SVM (Taylor Exp)	0.93	1.39
K-Means	0.61	0.89
ANN	0.53	1.64

Chapter 6

Conclusion

In this thesis work a system able to recognize and classify hand gestures in real-time, starting from the sEMG signals produced by the forearm muscles, is proposed. In order to reduce the power consumption, the event-driven approach of the Average Threshold Crossing (ATC) technique is used. The final implementation results suitable for real-time applications, thanks to low latency and power consumption.

Three acquisition channels have been implemented, each one acquiring the sEMG signal and providing the related quasi-digital TC signals that are taken directly to three GPIO of the Apollo3 EVB. The TC events are counted with a simple interrupt routine and then averaged over a time window computing the ATC parameter. The values obtained were used for the training phase of three implemented ML algorithms. In particular SVM, K-Means and ANN were used for offline training and online prediction. The characteristic parameters obtained from this first phase, have been loaded into the memory of the Apollo3 MCU in order to perform a real-time movement prediction directly from the board.

The three algorithms were subsequently implemented on the microcontroller, in order to be able to make an online gesture recognition. Their performance in terms of accuracy in the classification of the movements, latency and power consumption of the Apollo3 MCU have been evaluated. Accuracy and latency measurements were carried out with specific firmware routines, while power consumption was calculated by an high-precision digital multimeter.

In particular with SVM algorithm an average classification accuracy of 94.49%, a latency of 109.85 ms and an average power consumption of 1.67 *mW* were obtained. K-Means has gotten an average classification accuracy of 95.14%, a latency of 62.6 μ s and an average power consumption of 0.82 *mW*. Finally, the NN achieved an

average classification accuracy of 97.59%, a latency of 2.58 ms and an average power consumption of 0.83 *mW*.

6.1 Future Works

There are several possible future developments concerning this thesis work. Next works could be focused on the improvement of ML algorithms in order to optimize the performance that has been analyzed. A possible improvement could concern the SVM algorithm, trying to optimize the offline training phase in order to reduce the number of support vectors needed for online gesture recognition. Regarding the K-Means algorithm, an alternative techniques to the euclidean distance for the clustering of data can be investigated: in this way it will be possible to obtain an improvement in the separation of these data during the training phase, in order to classify new data in a more consistent way in real-time mode. In addition, the use of unsupervised learning algorithms such as K-means could promote the addition of new output labels, in order to be able to recognise more hand movements.

Instead, from the hardware point of view, the increase of electrodes number and acquisition channels could achieve a better quality of the sEMG signal taken paying more attention to muscle synergies. Finally, the creation of wearable device with dry electrodes might be useful in order to carry out long-term monitoring of the myo-electric activity of the muscles, reducing a number of disadvantages related to the use of traditional electrodes.

Appendix A

Confusion Matrix

Table A.1: SVM Confusion Matrix (Exp.Func)

Actual		Predicted	
		Ext	Other
Actual	Ext	146	39
	Other	38	2178

Actual		Predicted	
		Flex	Other
Actual	Flex	149	40
	Other	20	2192

Actual		Predicted	
		Rad	Other
Actual	Rad	161	1
	Other	26	2213

Actual		Predicted	
		Uln	Other
Actual	Uln	80	40
	Other	36	2245

Actual		Predicted	
		Grasp	Other
Actual	Grasp	96	24
	Other	22	2259

Actual		Predicted	
		Idle	Other
Actual	Idle	1624	1
	Other	3	773

Table A.2: SVM Confusion Matrix (Taylor Series)

Actual		Predicted	
		Ext	Other
Actual	Ext	57	1
	Other	127	2216

Actual		Predicted	
		Flex	Other
Actual	Flex	70	2
	Other	99	2230

Actual		Predicted	
		Rad	Other
Actual	Rad	150	9
	Other	37	2205

Actual		Predicted	
		Uln	Other
Actual	Uln	65	82
	Other	51	2203

Actual		Predicted	
		Grasp	Other
Actual	Grasp	70	41
	Other	48	2242

Actual		Predicted	
		Idle	Other
Actual	Idle	1627	297
	Other	0	477

Table A.3: K-Means Confusion Matrix.

		Predicted				Predicted				Predicted													
		Ext	Other			Flex	Other			Rad	Other	Uln	Other	Grasp	Other	Idle	Other						
Actual	Ext	66	2	Actual	Flex	169	152	Actual	Rad	180	2	Actual	Uln	114	171	Actual	Grasp	64	39	Actual	Idle	1541	4
	Other	118	2215		Other	0	2080		Other	7	2212		Other	2	2114		Other	118	2180		Other	86	770

Table A.4: ANN Confusion Matrix.

		Predicted				Predicted				Predicted													
		Ext	Other			Flex	Other			Rad	Other	Uln	Other	Grasp	Other	Idle	Other						
Actual	Ext	73	6	Actual	Flex	147	20	Actual	Rad	168	0	Actual	Uln	111	124	Actual	Grasp	103	24	Actual	Idle	1625	0
	Other	111	2211		Other	22	2212		Other	19	2214		Other	5	2161		Other	15	2259		Other	2	774

Bibliography

- [1] E. M. Spinelli, N. H. Martinez, and M. A. Mayosky, “A transconductance driven-right-leg circuit,” *IEEE transactions on biomedical engineering*, vol. 46, no. 12, pp. 1466–1470, 1999.
- [2] D. Demarchi, D. P. M. Ros, and G. Manzella, “Design and implementation of wearable devices for semg detection and analysis of muscle synergies,”
- [3] H. McGuinness, “Anatomy & physiology (5th edition),” pp. 128–172, Hodder Education, 2018.
- [4] J. Moini, “Anatomy and physiology for health professionals (3th edition),” pp. 219 – 269, Jones & Barlett Learning, 2019.
- [5] J. E. Hall, “Guyton and hall textbook of medical physiology (12th edition),” pp. 71 – 90, Saunders, 2010.
- [6] E. Stålberg, S. D. Nandedkar, D. B. Sanders, and B. Falck, “Quantitative motor unit potential analysis,” *Journal of clinical neurophysiology*, vol. 13, no. 5, pp. 401–422, 1996.
- [7] R. Merletti and D. Farina, “Analysis of intramuscular electromyogram signals,” *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 367, no. 1887, p. 357, 2009.
- [8] C. J. De Luca, A. Adam, R. Wotiz, L. D. Gilmore, and S. H. Nawab, “Decomposition of surface emg signals,” *Journal of neurophysiology*, vol. 96, no. 3, p. 1646, 2006.
- [9] C. G. M. A and V. T. M. M, “Surface electromyography: Why, when and how to use it,” *In: Revista Andaluza de Medicina del Deporte*, 2011.
- [10] Carlo J. De Luca, *Surface ElectroMyoGraphy: Detection and Recording*, 2002. <https://www.delsys.com/downloads/TUTORIAL/semg-detection-and-recording.pdf>.
- [11] P. D. Marasco, J. S. Hebert, J. W. Sensinger, C. E. Shell, J. S. Schofield, Z. C. Thumser, R. Nataraj, D. T. Beckler, M. R. Dawson, D. H. Blustein, *et al.*, “Illusory movement perception improves motor control for prosthetic hands,” *Science translational medicine*, vol. 10, no. 432, 2018.

- [12] C. Prahm, I. Vujaklija, F. Kayali, P. Purgathofer, and O. C. Aszmann, “Game-based rehabilitation for myoelectric prosthesis control,” *JMIR serious games*, vol. 5, no. 1, p. e3, 2017.
- [13] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic myoelectric control strategies: a clinical perspective,” *Current Surgery Reports*, vol. 2, no. 3, p. 44, 2014.
- [14] E. A. Biddiss and T. T. Chau, “Upper limb prosthesis use and abandonment: a survey of the last 25 years,” *Prosthetics and orthotics international*, vol. 31, no. 3, pp. 236–257, 2007.
- [15] I. Dudkiewicz, R. Gabrielov, I. Seiv-Ner, G. Zelig, and M. Heim, “Evaluation of prosthetic usage in upper limb amputees,” *Disability and rehabilitation*, vol. 26, no. 1, pp. 60–63, 2004.
- [16] G. Jang, J. Kim, S. Lee, and Y. Choi, “Emg-based continuous control scheme with simple classifier for electric-powered wheelchair,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3695–3705, 2016.
- [17] H. Manabe, A. Hiraiwa, and T. Sugimura, “Unvoiced speech recognition using emg-mime speech recognition,” in *CHI’03 extended abstracts on Human factors in computing systems*, pp. 794–795, 2003.
- [18] D. G. Park and H. C. Kim, “Muscleman: Wireless input device for a fighting action game based on the emg signal and acceleration of the human forearm,” in *International Symposium on Neural Networks. ISNN*, Citeseer, 2011.
- [19] K. R. Wheeler and C. C. Jorgensen, “Gestures as input: Neuroelectric joysticks and keyboards,” *IEEE pervasive computing*, vol. 2, no. 2, pp. 56–61, 2003.
- [20] M. A. Oskoei and H. Hu, “Myoelectric control systems—a survey,” *Biomedical signal processing and control*, vol. 2, no. 4, pp. 275–294, 2007.
- [21] H. D. Beale, H. B. Demuth, and M. Hagan, “Neural network design,” *Pws, Boston*, 1996.
- [22] H. Mizuno, N. Tsujiuchi, and T. Koizumi, “Forearm motion discrimination technique using real-time emg signals,” in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4435–4438, IEEE, 2011.
- [23] Z. Lu, K.-y. Tong, X. Zhang, S. Li, and P. Zhou, “Myoelectric pattern recognition for controlling a robotic hand: a feasibility study in stroke,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 2, pp. 365–372, 2018.
- [24] R. Crepin, C. L. Fall, Q. Masclet, C. Gosselin, A. Campeau-Lecours, and B. Gosselin, “Real-time hand motion recognition using semg patterns classification,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2655–2658, IEEE, 2018.

- [25] P. Shenoy, K. J. Miller, B. Crawford, and R. P. Rao, "Online electromyographic control of a robotic prosthesis," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1128–1135, 2008.
- [26] S. Benatti, B. Milosevic, E. Farella, E. Gruppioni, and L. Benini, "A prosthetic hand body area controller based on efficient pattern recognition control strategies," *Sensors*, vol. 17, no. 4, p. 869, 2017.
- [27] U. Côté-Allard, G. Gagnon-Turcotte, F. Laviolette, and B. Gosselin, "A low-cost, wireless, 3-d-printed custom armband for semg hand gesture recognition," *Sensors*, vol. 19, no. 12, p. 2811, 2019.
- [28] "GForce 100 Armband gforce 100 armband." <http://www.oymotion.com>. Accessed: 2019-09-30.
- [29] S. Sapienza, M. Crepaldi, P. Motto Ros, A. Bonanno, and D. Demarchi, "On integration and validation of a very low complexity atc uwb system for muscle force transmission," *In: IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS*, 2016.
- [30] S. Sapienza, P. Motto Ros, D. A. F. Guzman, F. Rossi, R. Terracciano, E. Cordedda, and D. Demarchi, "On-line event-driven hand gesture recognition based on surface electromyographic signals," Institute of Electrical and Electronics Engineers Inc., 2018.
- [31] P. M. Ros, M. Paleari, N. Celadon, A. Sanginario, A. Bonanno, M. Crepaldi, P. Ariano, and D. Demarchi, "A wireless address-event representation system for atc-based multi-channel force wireless transmission," IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC, 2013.
- [32] M. Crepaldi, M. Paleari, A. Bonanno, A. Sanginario, P. Ariano, D. H. Tran, and D. Demarchi, "A quasi-digital radio system for muscle force transmission based on event-driven ir-uwb," in *2012 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 116–119, IEEE, 2012.
- [33] P. M. Ros, M. Paleari, N. Celadon, A. Sanginario, A. Bonanno, M. Crepaldi, P. Ariano, and D. Demarchi, "A wireless address-event representation system for atc-based multi-channel force wireless transmission," in *5th IEEE International Workshop on Advances in Sensors and Interfaces IWASI*, pp. 51–56, IEEE, 2013.
- [34] I. Furfaro, "Integration and validation of average threshold crossing (atc) applied to surface electromyography (semg)," *Torino, Italy, October*, 2015.
- [35] B. Sereni, "Desing and development of a low-power wearable device for the acquisition of surface electromyography (semg) signals with average threshold crossing (atc)," *Politecnico di Torino-DET*, 2016.
- [36] F. Rossi, *Low Power System for Event-Driven Control of Functional Electrical*

- Stimulation*. PhD thesis, Master Thesis-Politecnico di Torino, 2017 Jul, 2017.
- [37] D. Demarchi and B. L. Pugliese, “Low power long term semg monitoring modular system,”
- [38] A. Mongardi, P. M. Ros, F. Rossi, M. R. Roch, M. Martina, and D. Demarchi, *A Low-Power Embedded System for Real-Time EMG based Event-Driven Gesture Recognition*. PhD thesis, Politecnico di Torino, 2019.
- [39] M. Martina, P. M. Ros, G. Maserà, and A. Mongardi, “A low-power embedded system for real-time emg based event-driven gesture recognition,”
- [40] “Apollo3 evaluation board.” <https://www.ambiqmicro.com/mcu/>.
- [41] “Arm, “cortex-m series.”” <https://www.arm.com/products/silicon-ipcpu/cortex-m/cortex-m4>. 2019.
- [42] S. Sapienza, P. M. Ros, D. A. F. Guzman, F. Rossi, R. Terracciano, E. Cordedda, and D. Demarchi, “On-line event-driven hand gesture recognition based on surface electromyographic signals,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2018.
- [43] “E. hand, “the electronic text book of hand surgery.”” <http://www.eatonhand.com>. Accessed: 2019.
- [44] S. Benatti, E. Farella, E. Gruppioni, and L. Benini, “Analysis of robust implementation of an emg pattern recognition based control.,” in *BIOSIGNALS*, pp. 45–54, 2014.
- [45] “A.ng, “machine learning by stanford university.”” <https://www.coursera.org>. Accessed: 2019.
- [46] 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.