POLITECNICO DI TORINO

DIPARTIMENTO DI INGEGNERIA MECCANICA E AEREOSPAZIALE
(DIMEAS)

Master's Degree in Biomedical Engineering

Design and Integration of an Event-Driven Rehabilitation System for Functional Electrical Stimulation

Supervisors
Prof. Danilo Demarchi
Ph.D. Paolo Motto Ros
M.Sc. Fabio Rossi
M.Sc. Andrea Mongardi

Candidate
Andrea Prestia

TORINO, December 2020

Ringrazio Andrea Mongardi e Fabio Rossi per avermi guidato durante questo percorso, dimostrandosi nel contempo delle ottime persone e dei validi compagni di squadra. Ringrazio Danilo Demarchi e Paolo Motto Ros per avermi dato l'opportunità di dedicarmi a questo progetto. Ringrazio i colleghi di laboratorio con cui ho condiviso esperienze durante questo periodo. Ringrazio tutte le persone della mia sfera privata che mi sono state accanto nei momenti di difficoltà.

Abstract

Spinal cord injury (SCI), stroke and multiple sclerosis are some of the main causes that lead people to paralysis, worsening their daily-life quality. In such scenario, physiotherapeutic procedures require novel techniques in order to improve effectiveness and suitability of rehabilitative applications. A proper response to this challenge can be the *Functional Electrical Stimulation* (FES), a technique born in the 1960s which is continuously evolving to satisfy medical requirements. The principle behind FES is to induce muscle contraction by applying low-energy pulses to the motor nerves of the muscle of interest, stimulating their activation through the use of non-invasive electrodes. Therefore, the development of a system able to enhance the benefits of FES application covers a central role in clinical and medical research programs.

In this thesis project, a versatile multi-platform application for FES real-time control has been designed. Integrating previous and parallel works, the proposed software architecture allows a more efficient and embedded link between the FES stimulator and a *surface ElectroMyoGraphy* (sEMG) acquisition system. The sEMG is a non-invasive technique to acquire the electrical signals generated by skeletal muscles during contraction, from which it is possible to extract several parameters indicating the behavior of a muscle, such as its activity or its fatigue. However, the amount of information contained within these parameters and the processing complexity are constraints for low-power and real-time performance. For this reason, the proposed system exploits an event-driven approach applying the *Average Threshold Crossing* (ATC) to the sEMG signal. This technique works by generating an event whenever the sEMG signal exceeds a defined threshold. The average number of TC events inside an observation window has been proved to be an indicator of muscle activity; thus, this is the only value provided by the acquisition system. The central task of the application is to modulate the FES parameters (e.g., pulse amplitude, pulse width), according to the ATC values, and send them to the electrical stimulator.

The application has been developed in Python with the ambition to satisfy the following requirements: modularity, scalability, extensibility, reliability. In order to accomplish this task, *Object-Oriented Programming* (OOP) has been exploited. The application is designed on three layers: at the bottom layer, two classes define the sEMG acquisition system and the electrical stimulator; at the middle layer, another class integrates the two previous ones, putting them in communication with each other; finally, at the top layer the *Graphical User Interface* (GUI) is defined. Implementing the multithreading paradigm, safety, usability and timing performance are achieved fulfilling the constraints of real-time application.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Skeletal Muscle

Skeletal muscle allows vertebrates to move and maintain posture. It differs from heart and smooth muscles since it is under voluntary control. In humans, skeletal muscle tissue constitutes about 40% of the weight of an adult and its content is approximately divided into 75% of water, 20% of proteins and the remaining 5% of minerals, carbohydrates and fat [1]. Muscle tissue cells, also referred as *muscle fibers* or *myocytes*, can increase their mass/volume (but not their number) through continuous training. Muscle fibers can contract to bring closer the bone segments, to which they are connected by tendons, or they can relax to allow the action of other muscles. Excluding some special cases (e.g., diaphragm), muscles work in groups: from one side, agonist muscles collaborate in the execution of a movement while, on the other side, antagonist muscles exert the opposite behaviour. For example, as shown in Figure 1.1, the biceps plays the role of agonist muscle in the elbow flexion while the triceps is the antagonist one since it allows the elbow extension.

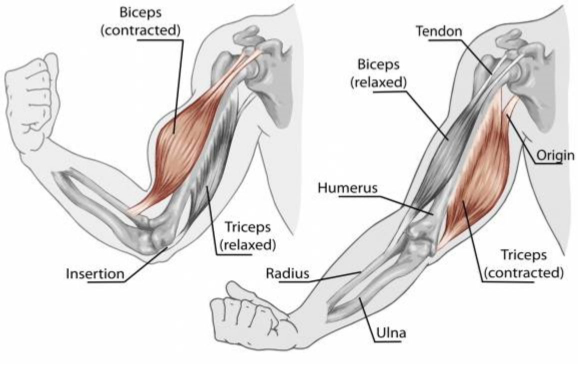


Figure 1.1: The biceps and triceps are antagonistic muscles between them; indeed, during the elbow flexion-extension, the activity of the two muscles is alternated: when one is contracted, the other one is relaxed [2].

1.1.1 Muscular Structure

Skeletal muscle, as well as cardiac muscle, belongs to the striated muscle category since its structure, being composed by several fascicles, presents cross-striation bands. As reported in Figure 1.2, each muscle fiber is covered by a layer of connective tissue called *endomysium*. In the same way, each fascicle is surrounded by a layer of connective tissue called *perimysium*. Different fascicles are then wrapped together in a layer of connective tissue called *epimysium*, which forms a continuous tissue with tendon extremities. This connective structure allows the exerted force to be transmitted among individual muscle components in form of relative movement. Besides, nerves and blood vessels pass through the connective tissue, branching off and reaching the muscle fibers [3].

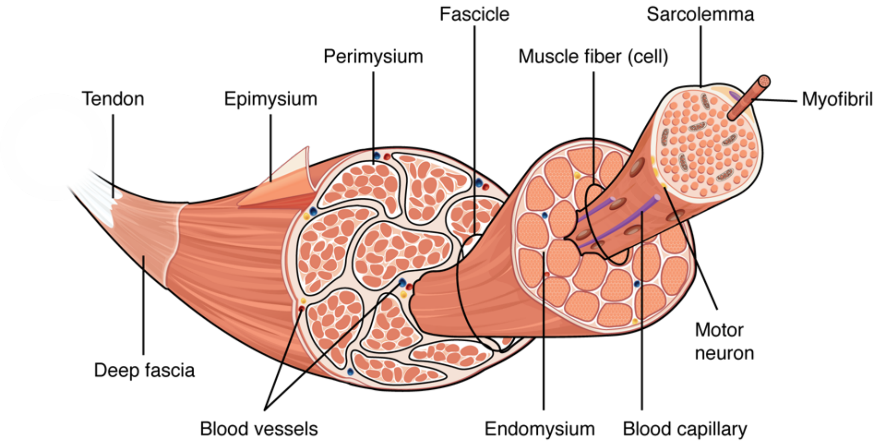


Figure 1.2: Striated muscle structure [4].

Internally, each muscle fiber has multiple nuclei and they are located near the *sarcolemma*, which is the plasma membrane that covers the muscle fiber [5]. The cytoplasmic fluid inside a muscle fiber is called *sarcoplasm* which contains multiple mitochondria and *myofibrils*. Mitochondria are responsible for *Adenosine TriPhosphate* (ATP) production. Through ATP hydrolysis, the human body is able to produce chemical energy. The conversion of chemical energy into mechanical energy is the method used by muscle fibers to contract [6]. Myofibrils are the contractile component of the muscle fiber. Each myofibril is enveloped by the *sarcoplasmic reticulum* which is a tubular complex acting as a reservoir of calcium ions [5]. The latter is the chemical specie responsible for the activation of muscle contraction. Looking a myofibril, its structure is composed by an alternation of light and dark bands, oriented orthogonally to its longitudinal axis. Figure 1.3 illustrates this behavior. The light bands are called *I bands* and are divisible in two by the *Z line*. The dark bands instead are called *A bands* and are divisible in two by the *M line*. The tract of myofibril between two Z lines is called *sarcomere* and constitutes the base unit of the muscle as it is the smallest element capable of contracting [5].

The sarcomere is made up of two types of protein filaments: the thick filaments are composed of *myosin* and are contained in the A band; the thin filaments are composed of *actin* and are contained between the Z line and the nearest end of the *H zone* (see Figure 1.3). Inside the sarcomere, the two types of filaments are partially overlapped. During muscle

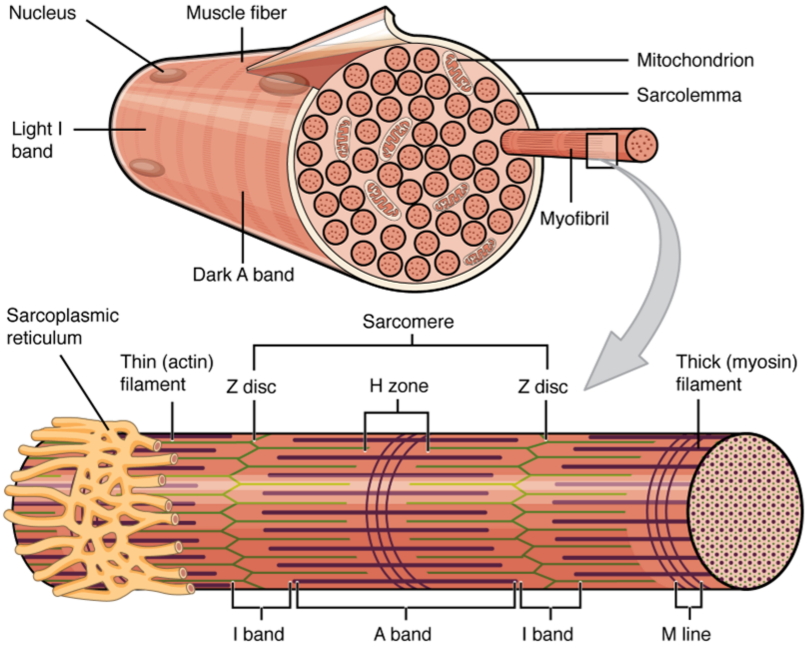


Figure 1.3: Sarcomere structure [4].

contraction, the thin filaments slide over the thick ones increasing the overlap between them and thus shortening the length of the sarcomere. Hence, during the contraction, I band and H zone dimensions decrease while the width of the A band remains unchanged.

1.1.2 Muscular Contraction Mechanism

Myosin is the motor protein of muscle contraction. The myosin molecule is composed of a long tail and a globular head on which the binding sites for actin and those for ATP are exhibited. Specifically, the thick filaments of the sarcomere are composed by *myosin-II* molecules, which consists of two identical myosin filaments wrapped together [3]. The myosin molecules are organized to expose the heads at the lateral ends of the sarcomere. Actin, on the other hand, is a globular protein characterized by myosin binding sites. The polymerization of the globular molecules of actin forms the main structure of the thin filaments of the sarcomere. In addition to actin, two other proteins are present in the thin filaments: *tropomyosin* and *troponin*. These two proteins work together to control muscle contraction. Tropomyosin is a filamentous protein that is helically arranged around actin filaments. Troponin, on the other hand, is a protein complex consisting of three subunits that have binding sites for actin, tropomyosin and calcium respectively. In the absence of calcium ions in the sarcoplasm, tropomyosin sterically blocks the binding sites between actin and myosin. When calcium ions bind troponin, proteins conformational changes allow the actin-myosin bond to be established [3]. Figure 1.4 summarizes the structural information of the myo-proteins and represents their interaction inside the sarcomere.

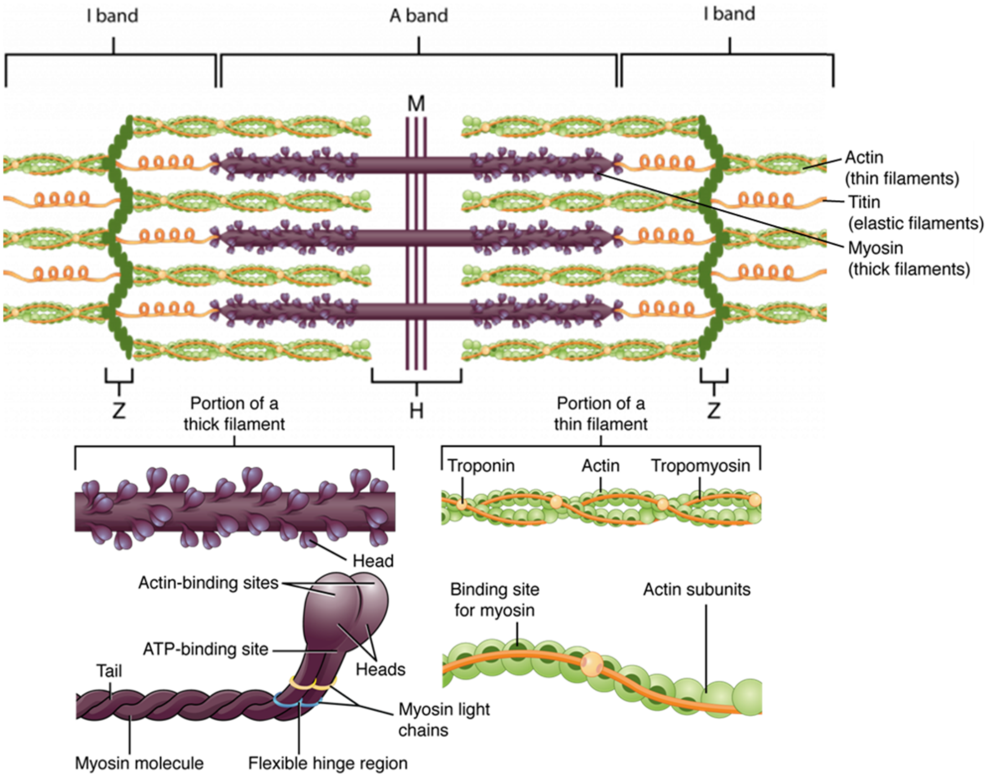


Figure 1.4: Proteins inside the sarcomere [4].

The muscle contraction mechanism can be described by the cross-bridge cycle theory, which consists of the following phases [7] (see Figure 1.5):

(a) Energization of the myosin head: the myosin-bound ATP is hydrolyzed to *Adenosine DiPhosphate* (ADP) and phosphate (Pi), consequently realising chemical energy. Some of this energy is stored by the myosin heads, which reach their high-energy conformational state.

(b) Attachment of myosin to actin: in the high-energy state, myosin has a high affinity for actin. If calcium ions are released into sarcoplasm, myosin and actin filaments bond together forming the cross-bridge interactions.

(c) Force stroke: the cross-bridge formation determines the release of ADP and Pi from the ATPase site, triggering the myosin heads rotation and pulling the actin filament towards the middle of the sarcomere. Then, the myosin returns to its low-energy conformational state, also called *rigor status* since the actin bonds are still intact.

(d) Detachment of myosin from actin: a new ATP molecule binds to the myosin head, inducing a conformational change that causes a decrease in affinity for actin, causing its detachment.

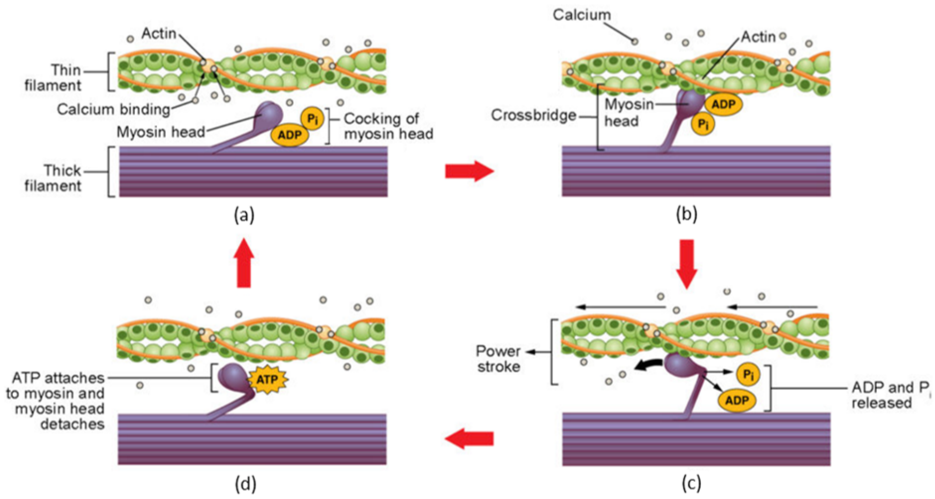


Figure 1.5: Cross-bridge cycle [8].

1.1.3 Action Potential

The increasing of the sarcoplasmic calcium concentration, which allows actin and myosin filaments to bond each other, is the final result of the control exerted by the somatic nervous system (SNS). The smallest neuro-muscular functional entity under the control of the SNS is the *motor unit*, which includes a motor neuron and all the muscle fibers it innervates. Therefore, when a stimulus is sent, it reaches all the muscle fibers within the motor unit. Motor neurons are the pathways used by nerve impulses responsible for muscle control to reach the *neuromuscular junctions*. The neuromuscular junction is where the nerve fiber joins the muscle fiber. In order for the impulse to pass the synapse, which is the space that separates the terminal of the axon's terminal from the surface of the muscle fiber (the sarcolemma), the terminal of the nerve fiber releases a neurotransmitter called *acetylcholine*. Acetylcholine crosses the synapse and is picked up by specific receptors on the sarcolemma. By interacting with acetylcholine, these receptors open the sodium ion channels allowing them to enter the cell and consequently generating a depolarization of the sarcolemma. If the depolarization is such that it exceeds a physiological threshold, the action potential is triggered. When the potential of the sarcolemma reaches 30 mV, the sodium channels close and the repolarization of the sarcolemma begins thanks to the outflow of potassium ions which allows the sarcolemma to return to its resting potential of -70 mV [3]. Figure 1.6 summarizes the described phases of action potential development. The action potential generated is propagated within the muscle fiber through invaginations of the sarcolemma called *T-tubule* (transverse tubules). These T-tubules are located near the terminal cisterns of the sarcoplasmic reticulum which contain calcium ions. With the arrival of the action potential, particular receptors are activated and calcium is released into the sarcoplasm where it triggers muscle contraction. When the stimulus ceases, the release of acetylcholine ends. In the absence of the action potential, the calcium ions are reabsorbed by the sarcoplasmic reticulum using an active pump. Then the muscle contraction ends [3].

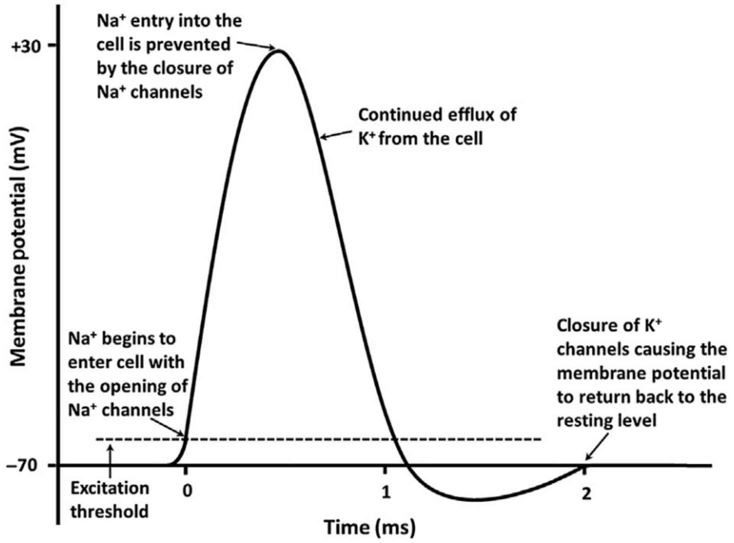


Figure 1.6: Phases of action potential [3].

1.1.4 Types of Muscular Contraction

A muscular contraction can be *isometric* or *isotonic*. An isometric contraction occurs when the muscular tension is equal to the applied load. During this contraction, the muscle length remains unchanged and no movement is executed. Hence, this type of contraction does not perform work, as the displacement is null. An isotonic contraction instead involves a movement because the muscular tension is greater than the applied load. If the muscle shortens to sustain the load, the contraction is defined *concentric*. If instead the muscle lengthens, the contraction is defined *eccentric* [3]. Figure 1.7 summarizes the cases described.

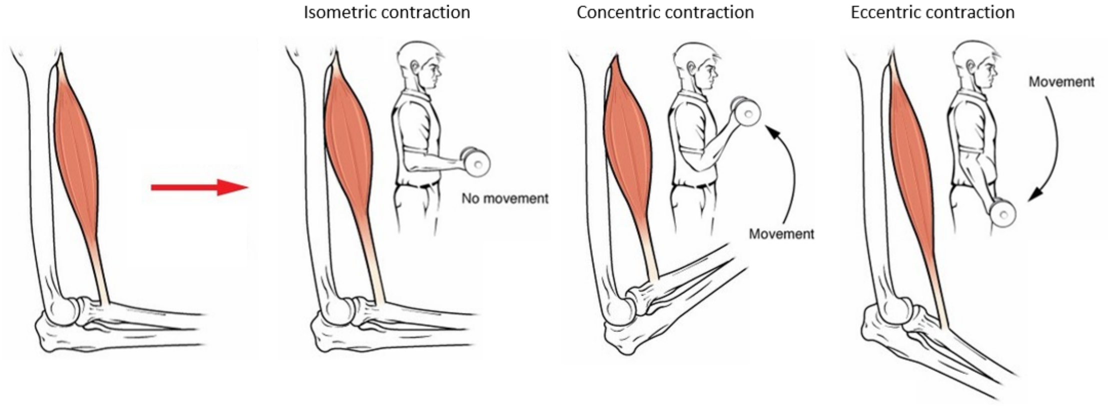


Figure 1.7: Difference in muscle lengthening during isometric, concentric or eccentric contractions [9].

1.1.5 Types of Muscular Fibers

Muscle fibers are not all the same but they can be classified based on their contraction velocity and fatigue resistance [6].

- *Type I*: they are those with the slowest contraction and are the most fatigue resistant. They are also called red fibers, since thier myoglobin content gives them the characteristic color. Generally, additional typical features of this fibers type are an elevate number of mitochondria, which results in oxidative metabolism (suitable to sustain aerobic activities), and small diameter of innerving motor-neuron.
- *Type IIA*: they have intermediate characteristics between type I and type IIB muscle fibers. Metabolism is both oxidative and glycolytic.
- *Type IIB*: they are those with the fastest contraction and are the least fatigue resistant. They are also called white fibers. They have a high amount of glycogen and the metabolism is glycolytic. These muscle fibers are used during anaerobic activities. The motor neurons innervating these muscle fibers have a large diameter.

Within a motor unit, the muscle fibers are all of the same type. The motor units can be identified in three different types: *slow*, *fast fatigue resistant*, *fast fatigable.* Each of these type of motor units contains respectively the three types of muscle fibers listed above. The slow motor units are therefore more suitable for low intensity and prolonged activities over time, while the fast fatigable motor units are better suited for intense and short duration activities [10].

1.1.6 Regulation of Muscular Force

The regulation of the force developed by the muscle occurs by varying the number of motor units recruited and the frequency of stimulation [11].

- Motor unit recruitment: the recruitment order respects the *Henneman size principle.* This principle explains how the first motor units to be recruited are those with motor neurons with smaller diameter, i.e. slow motor units. Then, as the necessary muscle tension increases, the fast fatigue resistant motor units are recruited and finally the fast fatigable ones. In Figure 1.8 is shown a graph representing force production vs motor unit recruitment.
- Stimulation frequency: if an action potential arises before the end of the contraction, a second contraction occurs which is added to the previous one increasing the force developed. As the firing rate of action potentials increases, the muscle tension increases. The tetanic contraction is the maximum condition that can be reached through this summation mechanism. In Figure 1.9 is shown what has been described.

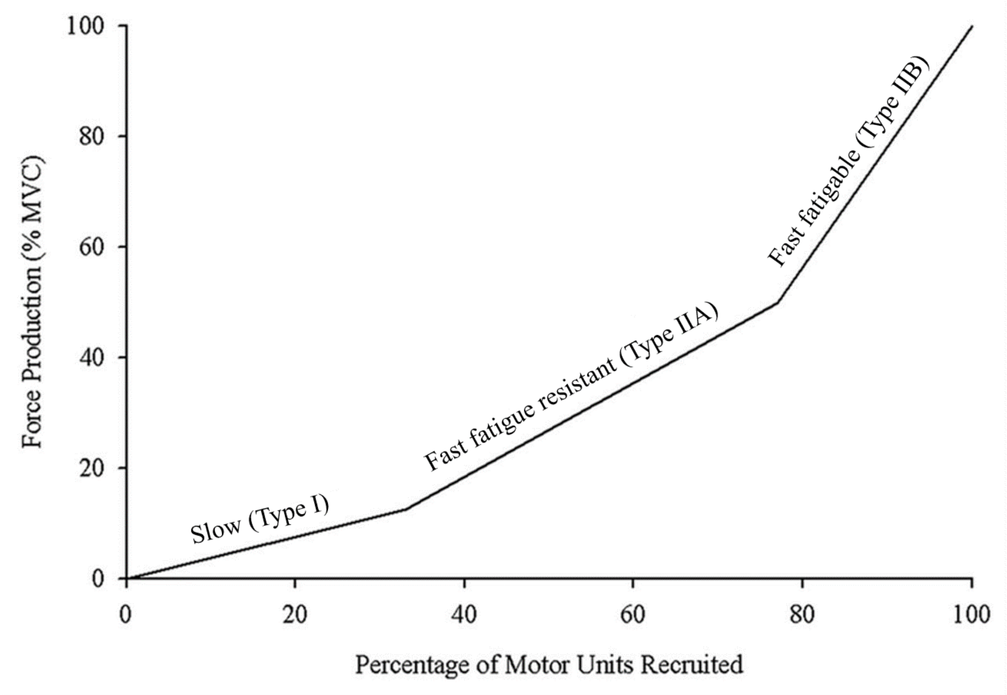


Figure 1.8: Force production vs motor unit recruited [12]

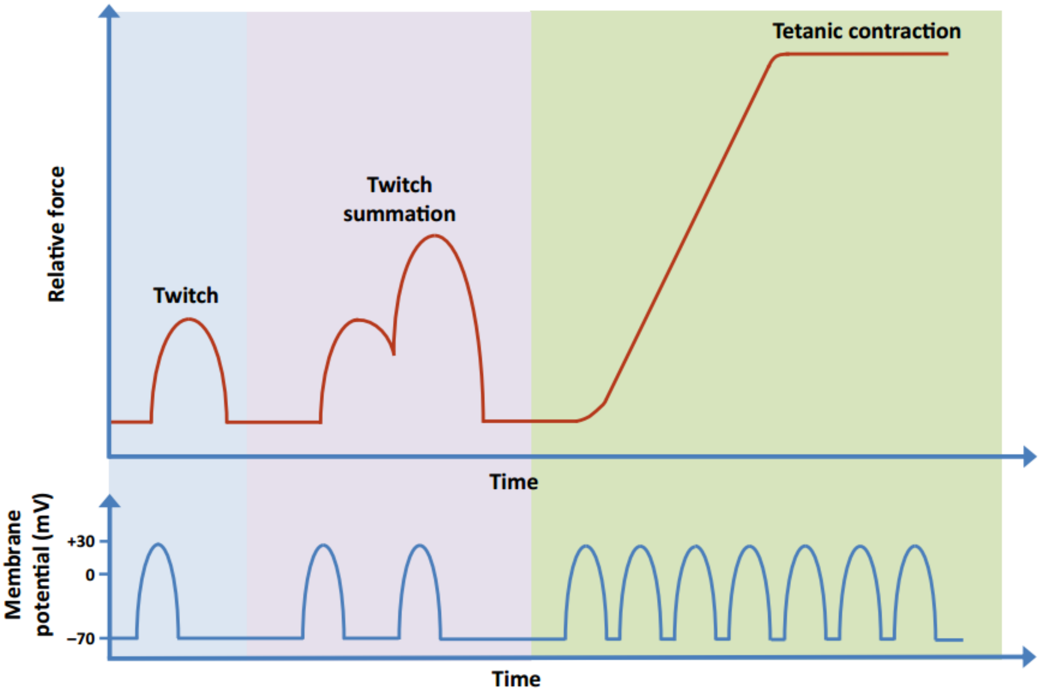


Figure 1.9: Force production vs action potentials firing rate [3]

1.2 Surface ElectroMyoGraphy (sEMG)

Surface ElectroMyoGraphy (sEMG) is a technique used to acquire an electrical signal from a muscle (Figure 1.10) in a non-invasive way using electrodes positioned on the subject's skin in correspondence with the muscle of interest. The signal consists of the sum of the motor units action potentials (MUAPs) within the sampling region. A constraint of the surface electromyography is that the high number of MUAPs does not allow the analysis of a single MUAP, unlike intramuscular electromyography which, however, is an invasive technique. Surface electromyography provides information about muscles and nerves. For this reason, it can be used to study the progression of neuromuscular disorders. In addition to diagnostic purposes, sEMG can be used as control for systems such as a muscle electrical stimulators [13], prosthesis [14] or exoskeletons [15]. Indeed, the non-invasiveness of surface electromyography allows its use even in non-hospital contexts.

1.2.1 Signal Morphology

The peak-to-peak amplitude of the electromyographic signal reaches values up to 10 mV. The bandwidth is up to 500 Hz, but the predominant energy content is between 50 an 150 Hz [16][17]. The nature of the signal is stochastic and this makes it mathematically describable through Gaussian probability distribution functions.

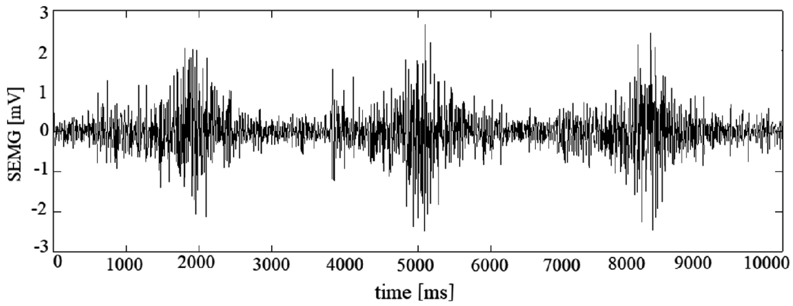


Figure 1.10: Example of surface electromyographic signal [15].

1.2.2 Acquisition Electrodes

In biological tissues the electric current is given by the flow of ions. In signal conditioning circuits, on the other hand, the current is given by the flow of electrons. For this reason, in order to sample biopotentials such as the electromyographic signal, it is necessary to use electrodes whose purpose is to convert the ionic current into an electronic one. The electrode for biopotentials is composed of an electrolytic solution that interfaces on one side with the skin and on the opposite side with a metal connector. Thanks to the electrolyte, oxidation-reduction reactions take place at the interface between the electrode and the skin, resulting in the release of free electrons in the metal connector in relation to the ionic activity of the biological tissue. The distribution of anions and actions at the interface generates a potential called half-cell. This potential results in a dc offset in the acquired signal. Silver/silver chloride (Ag/AgCl) electrodes have a very low half-cell potential of around 220 mV and for this reason they are the most popular electrodes used for biopotential sampling, together with their simple manufacturability [18]. The interface between the electrode and the skin can be modelled with a non-linear RC circuit, such as the one represented in Figure 1.11.a. The model can be simplified using the circuit represented in Figure 1.11.b, where: E_{hc} represents the half-cell potential; C_p and R_p represent the capacity and resistance respectively due to the displacement and conduction phenomena present at the interface; Rs represents the resistance of the materials making up the electrode. The circuit obtained is frequency dependent. For frequencies below about 100 Hz, the overall impedance is given by $R_p + R_s$, while for higher frequencies it is given only by Rs (Figure 1.12) [19]. The electrodes can be non-polarizable or perfectly polarizable. In the first case the charge flows through the electrode for conductive phenomena while in the second case it exploits displacement phenomena. Non-polarizable electrodes are better for rejection of motion artifacts. Ag/AgCl electrodes are an example of non-polarizable electrodes [18].

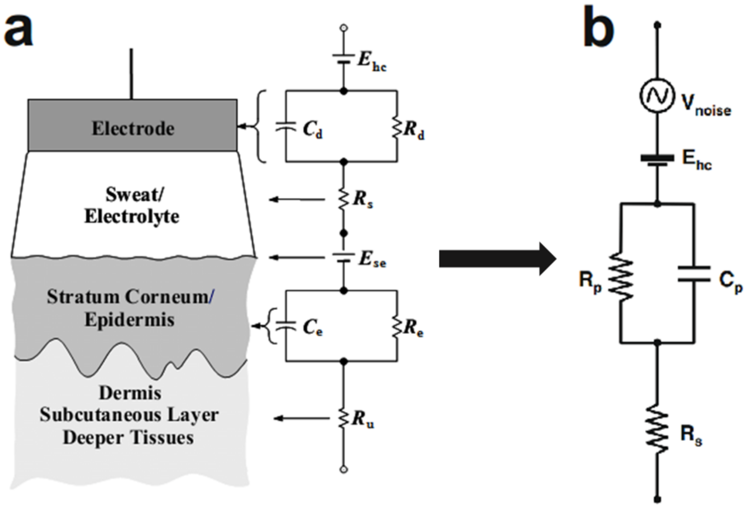


Figure 1.11: Electrode-skin interface model: a) generalized model; b) simplified model [19].

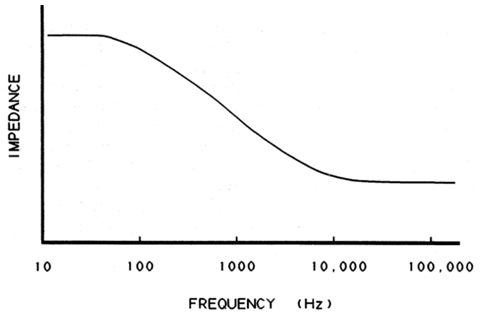


Figure 1.12: Electrode-skin impedance versus frequency [20].

1.2.3 Acquisition Techniques

The biopotential acquisition circuits need the following characteristics [21]:

- Safety for the patient, who must not be subjected to high potential differences;
- High input impedance, to avoid signal distortion due to the electrode-skin interface;
- High CMRR (Common Mode Rejection Ratio);
- Low noise, to not corrupt the signal of interest;
- High gain, because the acquired signals have very low amplitudes;
- Selective bandwidth, to acquire only the required information.

In order to satisfy all the necessary requirements to acquire the sEMG signal, typically the analog signal conditioning circuit is composed of the following modules:

- An overvoltage protection, which is useful in case of spurious spikes overcoming or coupling with electrical stimulation of the muscle;
- An instrumentation amplifier (Figure 1.13) for differential acquisition. The latter works by subtracting the two input signals and thus overcoming common mode interference problems (an example is power line interference, since it acts on both inputs). Common mode signal rejection is measured with the CMRR. Acceptable values of CMRR are around 100 dB [22]. Moreover, thanks to the two input buffer amplifiers, it is possible to obtain a high input impedance. The amplification of this stage is low, to avoid saturating the circuit due to a DC offset;
- A high pass filter to remove DC offset and frequency components lower than the sEMG signal band, due for example to motion artifacts;
- An amplification stage that adapts the dynamics of the signal to that of the ADC;
- A low pass filter to remove interferences with frequency components above the sEMG signal band.

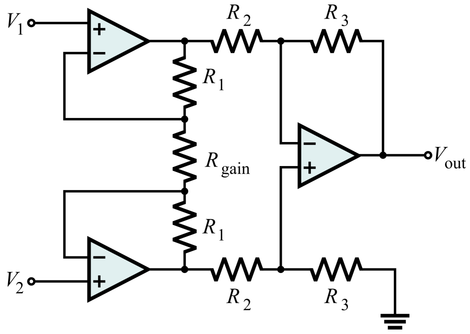


Figure 1.13: Instrumentation amplifier output: $V_{out} = A_d \cdot (V_2 - V_1) + \frac{1}{2} \cdot A_{cm} \cdot (V_2 - V_1)$ where $A_d = \left(1 + \frac{2 \cdot R_1}{R_{gain}}\right) \cdot \left(\frac{R_3}{R_2}\right)$ is the differential gain and A_{cm} is the common-mode gain. The common mode rejection ratio is computed as $CMRR = 20 \cdot log_{10}\left(\frac{A_d}{|A_{cm}|}\right) dB$.

Literature standard sEMG acquisition approached can be summarized into three configurations [23]. These modes are represented in Figure 1.14. Each of them also involves a reference electrode, which must be applied on an electrically neutral zone. For example, if you are interested in acquiring the electromyographic signal from the biceps, the reference electrode can be positioned on the elbow or alternatively on the back of the hand.

- Monopolar sampling: simpler configuration but not recommended considering its susceptibility to noise. In this mode, in addition to the reference electrode, only one sampling electrode is used which also acquires the surrounding noise.
- Bipolar sampling: this is the most common configuration and consists in the use of two sampling electrodes, in addition to the reference electrode. The two electrodes are positioned about 2 cm away from each other and are connected at the input to an instrumentation amplifier. This mode is called single differential acquisition.
- Multipolar sampling: this configuration uses at least three sampling electrodes, in addition to the reference one. Basically, with this mode, instrumentation amplifiers are applied in cascade, using the outputs of two different amplifiers as inputs to another instrumentation amplifier. In case only three electrodes are used, this mode is called double differential acquisition.

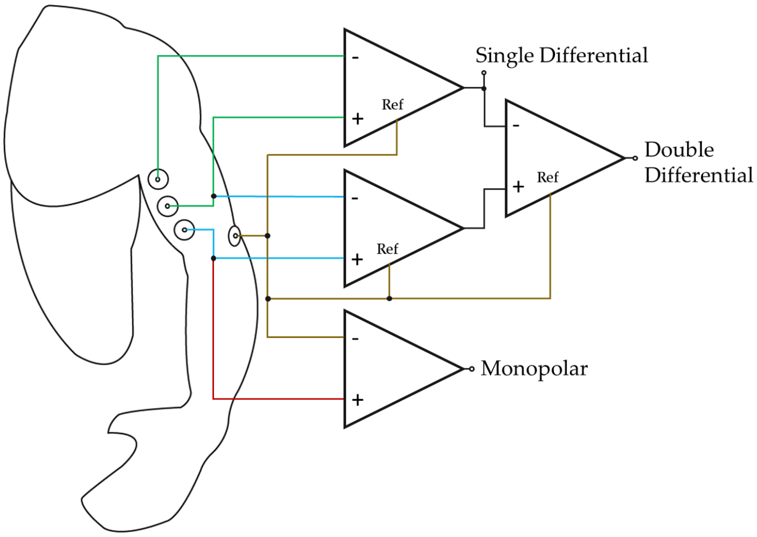


Figure 1.14: Possible electrodes configurations for sampling the EMG signal: Single Differential (SD) is the most commonly used as it offers a good trade-off between circuit complexity and noise robustness.

1.2.4 Noise Sources

The surface electromyographic signal is corrupted by the noise caused by various possible sources of interference, listed below together with the possible methods for the respective resolution [17][21][24].

- Internal noise: the body tissue between the muscle of interest and the acquisition electrodes is an obstacle for the propagation of the sEMG signal. The skin is not particularly conductive and therefore the capacitive effects come into play. Along with the latter, dispersive phenomena are also of particular importance, thus compromising the amplitude of the acquired signal.

- Quality of the acquisition electrodes: the electrodes in poor condition compromise conduction due to the high and inhomogeneous contact impedance. It is therefore necessary to use certified, clean and intact electrodes, as well as to apply them correctly and on clean skin.

- Movement artifacts: movement of the cables connecting the electrodes to the acquisition circuit corrupts the signal with low frequency interference. This problem can be addressed by properly designing the acquisition circuit using, for example, a high pass filter with a cutoff frequency of 20 Hz.

- Crosstalk: this type of interference occurs when, in addition to acquiring the EMG signal of the muscle of interest, the signal from a nearby muscle is also involuntarily acquired. To solve this problem it is recommended to reduce the distance between the acquisition electrodes and their size, since in this way the sampling volume is reduced.

- ECG artifacts: during the acquisition of the EMG signal, involuntary acquisition of the electrocardiographic signal is often encountered. The main problem is that the spectrum of the ECG overlaps that of the EMG, so removal by filtering is complicated. To reduce contamination by the ECG artifact, it is possible to operate on the common mode rejection.

- Stimulation artifacts: if electrostimulation is carried out near the sampling area of the EMG signal, the latter will be corrupted by a spike-shaped artifact. Through special algorithms, the artifact can be suppressed [25][26].

- Electromagnetic noise: even if no element of the electronic instrumentation used during signal acquisition is powered by the power line, the electromagnetic noise of the surrounding environment is able to propagate through the human body which acts as an antenna. This means that the frequency component of the power line (50 Hz in Europe or 60 Hz in USA) corrupts the EMG signal with amplitudes up to three times greater. This problem can be faced by using selective filtering techniques such as the Notch filter or more complex adaptive algorithms. Furthermore, it is good practice to twist the cables connected to the sampling electrodes in order to minimize the difference in the parasitic capacitances present between the noise source and the cables. In fact, if the capacitive effects on the two cables were the same, the interference would be common mode and would not be differential amplified.

- Ground loop: if a subject were connected to two devices that do not have the same reference potential, a current would flow through the subject that would cause common mode interference, as well as compromising the safety of the subject. To avoid this problem it is necessary to adequately design the instrumentation and have the skills to interface it with the subject.

1.2.5 Feature Extraction

Many features can be extracted from the electromyographic signal for classification purpose [24][27] or to evaluate, for example, muscular fatigue. The most common features are listed below, in which x_n and N represent a sample and the length of the EMG signal, respectively.

- Integrated EMG (IEMG), computed summing each sample of the rectified EMG signal, obtaining the area under the curve. This is a time feature used as index for the muscle activity.

$$IEMG = \sum_{n=1}^{N} |x_n|$$

- Mean Absolute Value (MAV), named also Average Rectified Value (ARV) and computed as the average of the rectified EMG signal. This is a time feature used to detect muscle contraction levels.

$$MAV = \frac{1}{N} \sum_{n=1}^{N} |x_n|$$

- Simple Square Integral (SSI), computed summing each point of the rectified EMG signal squared. This is a time feature that indicates the energy of the signal.

$$SSI = \sum_{n=1}^{N} |x_n|^2$$

- Variance of EMG (VAR), computed summing each squared point of the EMG signal and dividing the result by the length of the signal subtracted by one. This is a time feature that indicates the power of the signal.

$$VAR = \frac{1}{N-1} \sum_{n=1}^{N} x_n^2$$

- Root Mean Square (RMS), computed square rooting the average of the rectified EMG signal squared. This is a time feature used to evaluate muscular fatigue.

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^{N} |x_n|^2}$$

- Willison Amplitude (WAMP), computed as the number of times the difference between two sample of the EMG signal exceed a defined threshold. This is a time feature used to evaluate the motor unit firing rate.

$$WAMP = \sum_{n=1}^{N} f|x_{n+1} - x_n|$$

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

- Waveform Length (WL), computed as the cumulative length of the waveform. This is a time feature that indicates the waveform complexity.

$$WL = \sum_{n=1}^{N-1} |x_{n+1} - x_n|$$

- Slope Sign Change (SSC), computed as the number of changes between positive and negative slope. This is a time feature and the threshold is used to face noise.

$$SSC = \sum_{n=2}^{N-1} f\left[(x_n - x_{n-1}) \times (x_n - x_{n+1})\right]$$

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

- Zero Crossing (ZC), computed as the number of times the EMG signal crosses the zero value. This is a time feature and the threshold is used to face noise.

$$ZC = \sum_{n=1}^{N-1} \left[sgn\,(x_n \times x_{n+1}) \cap |x_n - x_{n+1}| \geq threshold\right]$$

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

- Median Frequency (MDF), computed as the frequency value that divides the Power Spectral Density (PSD) of the EMG signal in two equal areas. This is frequency feature used to estimate muscular fatigue.

$$\sum_{f=f_{min}}^{MDF} PSD_f = \sum_{MDF}^{f=f_{max}} PSD_f = \frac{1}{2} \sum_{f=f_{min}}^{f=f_{max}} PSD_f$$

- Mean Frequency (MNF), computed as the frequency value resulting from the weighted average on the PSD. This is a frequency feature used to estimate muscular fatigue.

$$MNF = \sum_{f=f_{min}}^{f=f_{max}} f \cdot PSD_f \Bigg/ \sum_{f=f_{min}}^{f=f_{max}} PSD_f$$

- Dimitrov spectral index (FInsmk) [28], computed as the ratio between the spectral moment of order -1 and the one of order k which can be 2, 3, 4 or 5. This is a frequency feature to estimate muscular fatigue.

$$FI_{nsmk} = \sum_{f=f_{min}}^{f=f_{max}} f^{-1} \cdot PSD_f \Bigg/ \sum_{f=f_{min}}^{f=f_{max}} f^k \cdot PSD_f$$

- Conduction Velocity (CV), computed dividing the inter-electrode distance by the conduction delay. The latter can be estimated in several ways [21]. Some of these methods require the acquisition of the EMG signal from two different channels. For example, by applying a method such as zero crossing to the two acquired signals, it is possible to obtain reference points. The time difference between the reference points of each signal is an estimate of the delay. The problem with this approach is its susceptibility to noise. Another method consists in using the cross-correlation between the two signals. This method is more robust to noise but the accuracy of the delay is dependent on the sampling resolution. Typical values of conduction velocity are around 4 m/s. This feature can be used to study muscle fatigue, as the conduction velocity decreases as the latter progresses.

The study of muscle fatigue is one of the applications for which electromyography can be very useful. With the onset of muscle fatigue, a leftward shift in the frequency content of the EMG signal spectrum is observed [29][30]. To study muscle fatigue it is therefore possible to analyze the decrease in MDF and MNF. Figure 1.15 shows the variation of MDF in the presence of muscle fatigue.

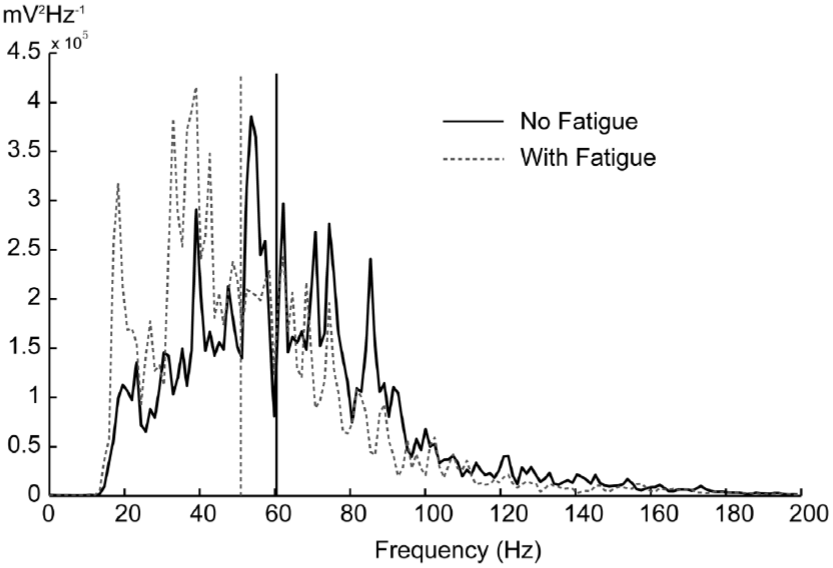


Figure 1.15: PSD median frequency variation with muscular fatigue [30].

MNF is more sensitive than MDF but the latter is more robust with respect to noise. However, both indices are not sensitive enough and for this reason other indices have been developed, such as the one described by Dimitrov, which increases as muscle fatigue increases [28]. Figure 1.16 shows a comparison between the Dimitrov spectral index with order k equal to 5 and the median frequency. Observing the variation of the Dimitrov index as the number of repetitions of the exercise increases, it appears that this index is more sensitive to muscle fatigue than the median frequency.

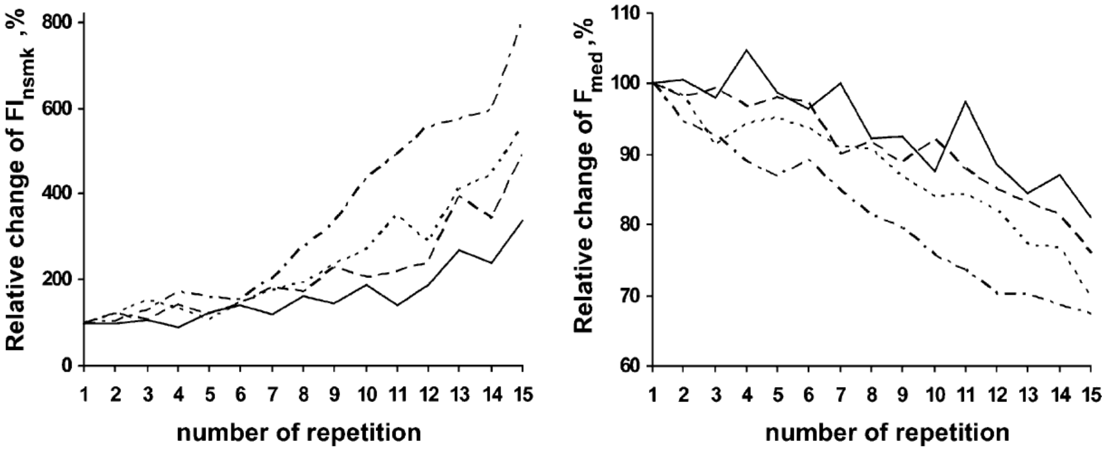


Figure 1.16: Comparison between Dimitrov spectral index and median frequency studying muscular fatigue. The exercise protocol consists of four sets, represented respectively from the first to the last with: solid lines, dashed lines, dotted lines, dashed dotted lines [28].

1.3 Average Threshold Crossing (ATC)

The Average Threshold Crossing (ATC) is a technique useful to reduce the amount of signal information to the minimum necessary, in order to reduce circuit complexity and power consumption. In order to detect the onset and offset of a muscular contraction from the sEMG signal, features such as Root Mean Square (RMS) can be extracted or algorithms such as Sample Entropy (SampEn) [31] can be applied. However, these methods result in high computational cost, low real time performance and high power consumption. For this reason, it is extremely advantageous to use a bio-inspired approach by minimizing the information involved. Bio-inspired design is an engineering paradigm used to mimc the evolutionary strategies of nature, as it is extremely efficient [32]. The ATC technique works by generating an event whenever the signal exceeds a defined threshold. The number of events generated is then averaged over a time window with a duration defined according to the system requirements. The event-driven sampling performed by this technique allows the system to considerably simplify the information broadcasting, as shown in Figure 1.17, and to reduce power consumption [33].

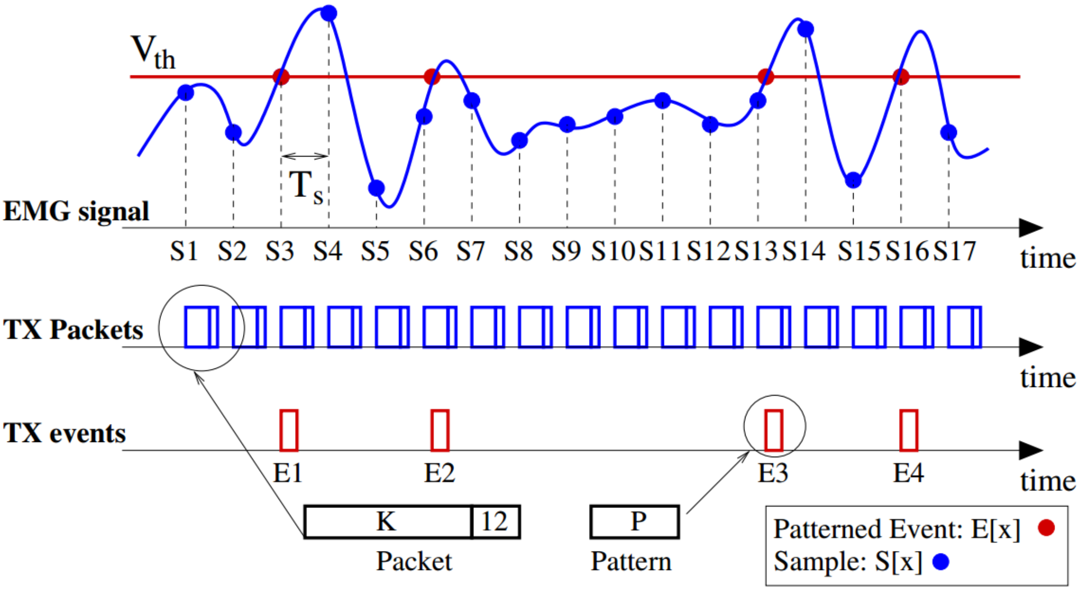


Figure 1.17: Difference in broadcasting between periodic sampling and event-driven one [33].

A voltage comparator is used to perform this technique. The output obtained is a quasi-digital signal where the logic level is high in correspondence with the threshold crossing (TC) events. A quasi-digital signal is a signal with digital shape that carries frequency-encoded analog information, as shown in Figure 1.18.

In order for a microcontroller to be able to read an analog signal, it is necessary to use an Analog to Digital Converter (ADC). Since the TC signal has only two logic levels, it can be managed by the Interrupt Capture Unit (ICU) of the microcontroller and the ADC is

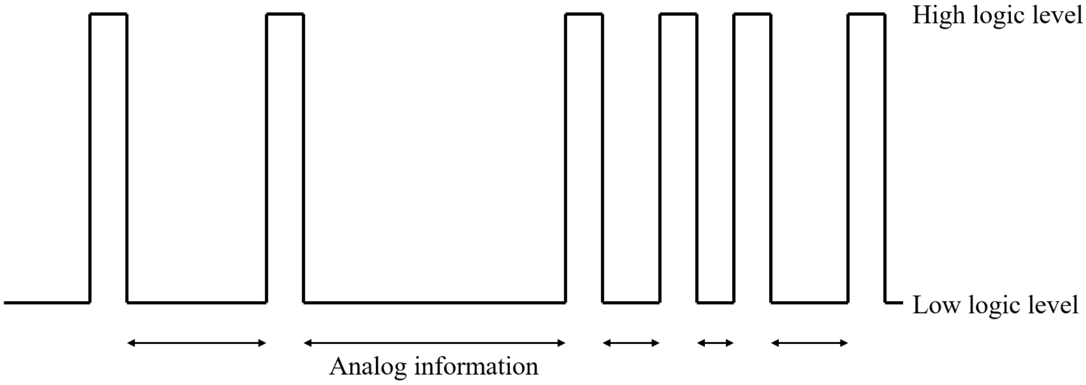


Figure 1.18: Quasi-digital signal illustration.

therefore no longer needed, thus reducing the circuit area and its complexity in general [34]. The number of TC events generated by this technique has been shown to be highly correlated with the muscle strength developed during both isometric and isotonic contractions [35]. However, the ATC technique involves a complete loss of signal morphology and consequently precludes the possibility of extracting the characteristic features of the sEMG signal. This constraint can be harmful in some applications, so the ATC technique is not always a valid solution. A further critical aspect of this technique is the choice of an optimal threshold: if the threshold is too high, too much information is excluded; if instead the threshold is too low, the intrinsic variability of the sEMG signal corrupts the information. A previous study proposed an algorithm for using a dynamic threshold and demonstrated an increase in the robustness of the ATC technique compared to using a fixed threshold [36].

1.4 Functional Electrical Stimulation (FES)

Every year in USA, above 17.000 people undergo a spinal cord injury (SCI) [37] and more than 795.000 suffer a stroke [38]. While in the latter a person's brain may no longer be able to generate a motor command, in case of SCI the path of motor axons is damaged and the motor command is unable to reach the target muscle. As a consequence of these neurological diseases, considering the inability of the human body to restore neural tissues, a person's quality of life can be gravely compromised. Functional electrical stimulation (FES) can be a solution to allow people with paralysis to perform a movement. Indeed, if the muscular innervation is intact and so are the elements that mechanically allow the movement, such as the joint, through electrical stimulation it is possible to contract muscle tissue and achieve movement [39].

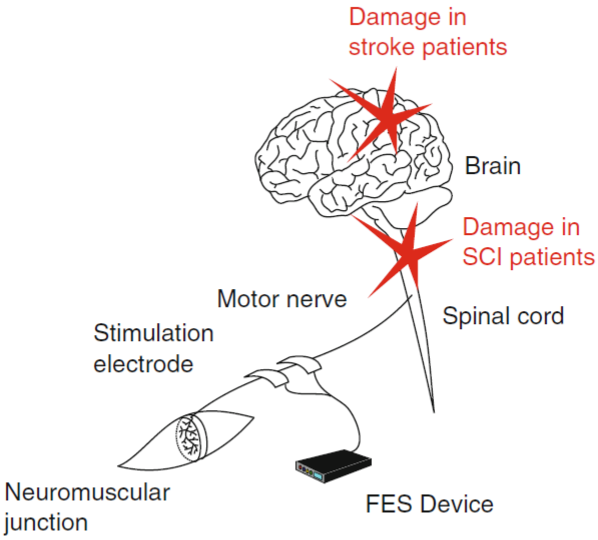


Figure 1.19: FES overcoming the effects of stroke and SCI on paralysis [40].

1.4.1 FES Overview

FES is a technique increasingly used in the rehabilitation field. The first FES application dates back to the 1960s [41] and was aimed to correct drop foot in hemiplegic patients. The principle of the system was to stimulate the *common peroneal nerve* whenever the heel was not in contact with the ground during walking. Hence, the stimulation allowed the *tibialis anterior* muscle to activate, inducing dorsiflexion. Whenever possible, FES is used to support the subject in re-educating the execution of a voluntary movement and therefore has the purpose of making the subject independent [13]. FES cannot regenerate damage to nerve tissue and therefore it is not a cure for the neurological disorders cited previously. In these latter cases, however, it can be integrated into neuroprostheses whose purpose is to permanently replace the physiological mechanism of movement [39]. FES is not even able to prevent the deterioration of motor axons but is able to prevent muscle atrophy due to inactivity [42]. The reason why electrical stimulation is applied to the motor nerve (i.e. a bundle of motor axons) rather than directly to the muscle, is that in the latter case it would require the use of energies up to three orders of magnitude greater [39]. Therefore, one of the requirements to be able to use FES is that the target muscle is innervated. If this is not the case, it is possible through surgery to replace the denervated muscles with those innervated [42]. In some applications, not only the efferent motor nerves but also the afferent sensory nerves are stimulated, in order to induce a reflex response that involves a physiological contraction [39]. FES can be performed using either non-invasive surface electrodes or implantable ones. In the latter case, it is possible to overcome some of the aspects that will be described in Section 1.4.3, since the surface available for electrodes positioning is no longer a constraint and therefore it is possible to create more complex patterns. However, being an invasive technique, its use is reserved for advanced neuroprostheses. The use of surface electrodes, on the other hand, makes the technique more accessible, especially for rehabilitation purposes, but involves some constraints and greater variability among the subjects. For example, the intensity of the stimulation necessary to obtain a contraction varies from person to person as it depends on various factors such as the thickness of the fat layer, which contributes to the dispersion of the energy supplied by the stimulus. Moreover, the positioning of the stimulation electrodes itself is a source of variability in the results obtained even on the same subject. Incorrect positioning could result in skin discomfort or ineffectiveness of stimulation if the nerves reached are not those of the muscle of interest.

1.4.2 FES Mechanism

Electrical stimulation is performed by injecting rectangular pulse trains into motor neurons, as depicted in Figure 1.20. Muscle contraction is achieved if the membrane potential of the latter exceeds the threshold and generates action potentials. The stimulation waveform is characterized by parameters such as pulse width, pulse width, stimulation frequency. These parameters affect both the intensity of muscle contraction and muscle fatigue. A pulse can be both monophasic and biphasic. In the latter case, the impulse is composed of a positive and a negative phase, in order to remove the injected charge from the tissues. If the energy of these two phases is the same, the pulse charge is balanced, resulting in two advantages:

- An accumulation of charge in the tissues is prevented, which can be harmful to them due to galvanic effects [39].
- The electrodes ionization is prevented, which would lead to the deposition of the material of the latter in the tissues [42].

A biphasic pulse can be both symmetrical and asymmetrical, in the latter case the positive and negative phase have different waveforms (as in Figure 1.20.g). Muscle contraction can be achieved using either voltage-regulated or current-regulated pulses. However, the use of the latter proves to be more advantageous, as it is ensured that the charge injected into the tissues is independent of an impedance variation, due for example to a partial detachment of the stimulation electrodes. The use of voltage regulated pulses would instead require a constant adjustment of the stimulation voltage on the basis of the impedance seen by the electrodes.

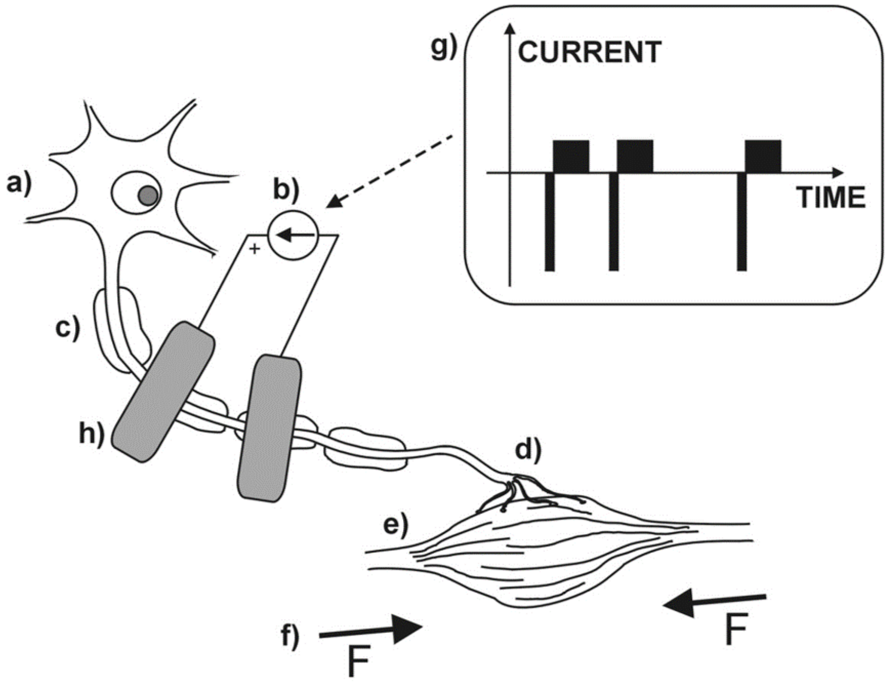


Figure 1.20: Motor unit response to FES: in the presence of neurological disorders, the motor neuron nucleus may not receive stimuli from its dendrites (a). By injecting charge from the outside (b), it is possible to generate action potentials that propagate through the intact axon (c) until they reach the neuromuscular junction (d). Muscle fibers (e) belonging to the motor unit contract and generate muscle force (f). Using this stimulation waveform (g), the membrane depolarization occurs at the negative phase of the pulse (h), corresponding to the entry of electrons into the axon [43].

1.4.3 Differences between Physiologic and FES Contractions

Contractions artificially induced by FES have several disadvantages compared to physiological ones. These differences are listed below.

- Muscle fibers innervated by a motor neuron with a larger axon diameter have lower electrical resistance and therefore have a lower excitation threshold [44]. This implies that the FES recruitment order is opposite to the physiological one, which instead respects the *Henneman size principle* as described in 1.1.6. Therefore, since the motor units are recruited by the FES starting from the most fatigable ones, this type of contraction implicitly involves more muscle fatigue than a physiological one. However, an opposite recruitment order to the physiological one has an advantage in a rehabilitation context: the muscle fibers activated by the FES are those that physiologically would be used for medium-high efforts, and are therefore those most prone to atrophy in subjects who have difficulty in make a movement.
- The motor units recruited by the FES are always the same, i.e. those placed in proximity to the stimulation electrodes. As in the next point, this leads to an increase in muscle fatigue [39].

- FES synchronously stimulates all the motor units within its range of action. This is different from what happens physiologically, where there is an asynchronous recruitment of the different motor units, in order to share the muscular effort and reduce fatigue. The repercussions of synchronous recruitment are the need to use higher stimulation frequencies in order to obtain constant muscle tension. The range of frequencies at which action potentials are physiologically generated is between 4 and 12 Hz. To obtain a continuous muscle contraction with FES it is instead necessary to use stimulation frequencies higher than 20 Hz (see Figure 1.21), resulting in a greater effort for the active motor units [39].

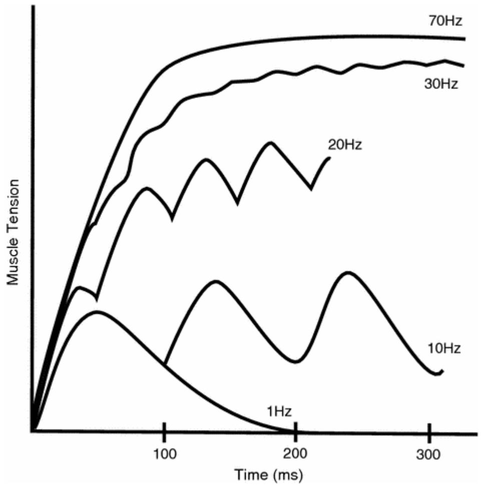


Figure 1.21: Stimulation frequency to obtain a tetanic contraction with FES [45].

1.5 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a standard for short-range wireless communication. It was introduced with the Bluetooth 4.0 specification and its purpose is to minimize energy consumption. BLE is not compatible with classic Bluetooth, but they can coexist in dual-mode devices [46]. Nowadays, Bluetooth technology has widely established itself in the market and is a powerful player in the Internet of Things (IoT).

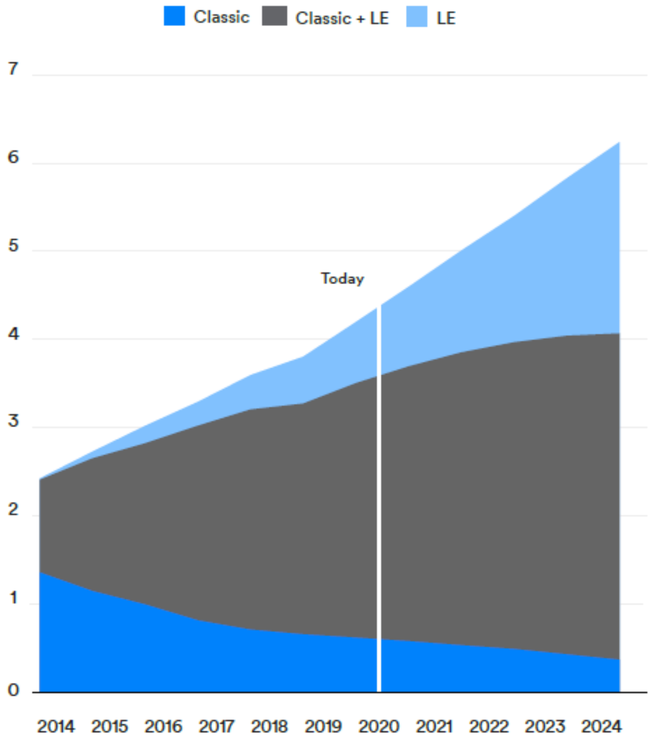


Figure 1.22: Annual Bluetooth device shipments in billions [47].

1.5.1 BLE Protocol Stack

The BLE protocol stack consists of several parts: the main two are the *Controller* and the *Host*. The former is implemented at the hardware level and comprises the Physical Layer and the Link Layer, while the latter, implemented via software, comprises the higher level layers. These two parts communicate with each other via the *Host Control Interface*. At the top of the stack there is the Applications Layer which includes *non-core profiles*, i.e. functionalities customized by the user and not defined by the Bluetooth specifications. Figure 1.23 shows all the layers belonging to the stack described below [46][48].

- *Physical Layer* (PHY). The frequencies used by BLE are part of the 2.4 GHz Industrial, Science and Medical (ISM) band. As shown in Figure 1.24, the BLE channels are centered in frequencies from 2402 to 2480 MHz spaced 2 MHz each, for a total of 40 channels. Among the latter, 3 channels are dedicated to advertising, i.e. for device

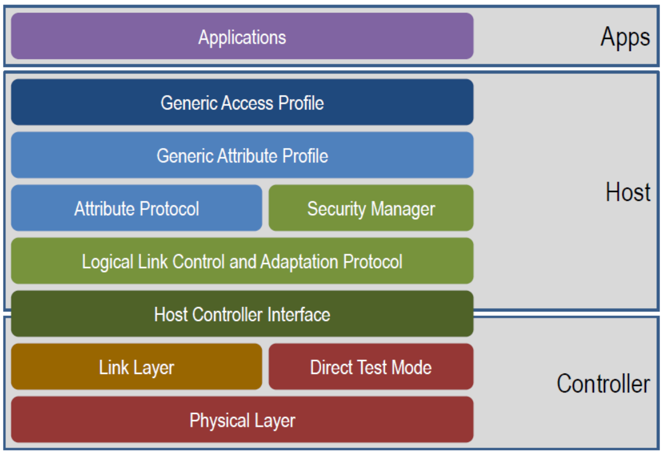


Figure 1.23: Bluetooth Low Energy Protocol Stack [49].

discovery and connection establishment, while the other 37 are used for data transfer between connected devices. In order to circumvent interference due to the presence of other overlapping signals in the band (such as Wi-Fi), an adaptive frequency hopping mechanism is used. Through the latter, one of the 37 available channels is selected whose carrier frequency is not disturbed. The PHY works with a data rate of 1 Mbps.

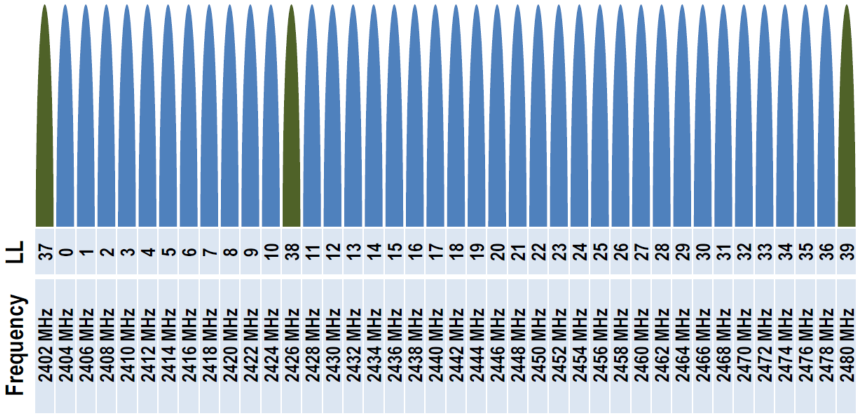


Figure 1.24: Bluetooth Low Energy communication channels: those represented in green (37-38-39) are the advertising channels, while those represented in blue are the channels used for data transfer [49].

- *Link Layer* (LL). Each Bluetooth device plays a role during connection establishment. During the initial phase, a device can be either an *advertiser* or a *scanner*. In the first case the *peripheral* is signaling its presence while in the second case the *central* is looking for devices with which to establish a connection. When the connection is established, the central assumes the role of *master* and the peripheral that of *slave*. A Bluetooth communication network, called *piconet*, can contain up to 8 devices: 1 master and 7 slaves. A master can be connected to multiple slaves but each slave can only be connected to one master. In addition to the definition of these roles, the LL also deals with the setting of three fundamental parameters for the connection:

  - *Connection Interval*: time between two connection events. It can range between 7.5 ms and 4 s in steps of 1.25 ms.
  - *Connection Slave Latency*: number of connection events that the slave can skip without losing the connection with the master. It can range between 0 and 499.
  - *Connection Supervision Timeout*: time limit between the reception of two valid packets beyond which the connection is interrupted. It can range between 100 ms and 32 s in steps of 10 ms.

- *Host Controller Interface* (HCI). This protocol allows communication between the Controller and the Host.

- *Logical Link Control and Adaption Protocol* (L2CAP). This protocol is basically a multiplexer between the lower (LL) and the upper (SMP and ATT) layers. It works encapsulating the raw data arriving from the LL in the BLE packet format for ATT and SMP (*recombination*) and vice versa (*fragmentation*).

- *Security Manager Protocol* (SMP). The purpose of this protocol is to encrypt and decrypt data packets using specific security algorithms.

- *Attribute Protocol* (ATT). This protocol defines how the server expose its data to the client and how they are structured within the server. Each data inside the server is called *attribute* and is composed of: a handle, an *Universally Unique Identifier* (UUID), a set of permission and a value.

- *Generic Attribute Profile* (GATT). This protocol defines the roles and interactions of the client and the server. The former is the device that ask for a *service*, while the latter is the device that expose it. A service is given by a set of attributes that satisfy a functionality. If the attribute holds a value, it is called *characteristic*. Each characteristic also contains a *descriptor* that provides information about it. A characteristic can be:

  - *Readable* if the client can read the characteristic value;
  - *Writable* if the client can write the characteristic value;
  - *Notifiable* if a notification can be received by the client every time the server updates the characteristic value. In order to receive notifications, the client first have to start them writing 1 on the *Client Characteristic Configuration Descriptor* (CCCD). To stop the notifications instead it is necessary to write 0 on the CCCD.

- *Generic Access Profile* (GAP). This is the upper layer of the stack and its purpose is to manage devices discovery, connection establishment and security.

Chapter 2

State of Art

2.1 sEMG Wearable Acquisition Systems

The use of surface electromyography for diagnostic, rehabilitation, research or sport-related applications has become widely established in recent years and the existing systems have an acquisition chain with a structure that can be generalized like the one in Figure 2.1. The raw signal is first acquired using electrodes placed on the surface of the subject's skin, mainly using a bipolar configuration (see Section 1.2.3). The signal is then filtered and amplified. Typically the high pass filter cutoff frequency is around 20 Hz while the low pass filter one is around 400 Hz. The gain typically has a value between 500 and 1000 [23], since the ADC power supply voltage is usually a few volts. The reason why these components are needed has already been described in Section 1.2.3. The resulting analog signal is converted into a digital one by the Analog to Digital Converter (ADC) [50] so that it can be read by the microcontroller inside the acquisition system. At this point, the microcontroller can be in charge of processing the signal with some technique, and finally has the task of transmitting the data to the end-user. Transmission can take place in different ways, but in wearable systems a wireless method is typically used, such as Bluetooth for example.

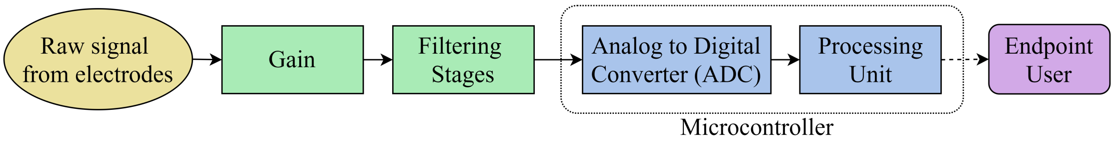


Figure 2.1: General structure of the sEMG signal acquisition chain: the blocks in green represent the analog front end (AFE), while the blocks in blue represent the digital component. The latter is also responsible for the transmission of the acquired data to the end user.

Considering the need to use a comfortable and wearable acquisition system that does not hinder the movements of the subjects and that is not corrupted by related artifacts, on the market there are devices such as those produced by *Delsys* or *Cometa*, reported in Figure 2.2. Furthermore, thanks to the absence of cables, the signals acquired by these sensors are not corrupted by capacitive effects from the surrounding environment. However, the autonomy of these devices is still limited: the two examples shown have nominal autonomies of 8 hours and 12 hours respectively.

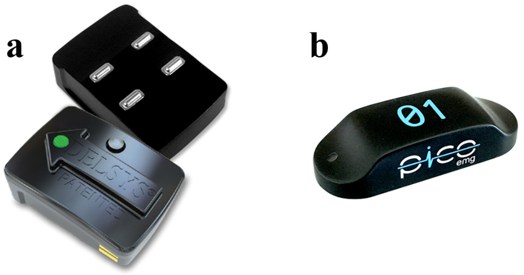


Figure 2.2: Commercial sEMG sensors: a) *Trigno* by *Delsys* [51]; b) *PicoEMG* by *Cometa* [52].

2.2 ATC Technique Applied to sEMG

In recent years the Average Threshold Crossing (ATC) has proved to be a valid technique to be applied to the electromyographic signal and this has also led to a simplification of the hardware used, as shown in the Figure 2.3.

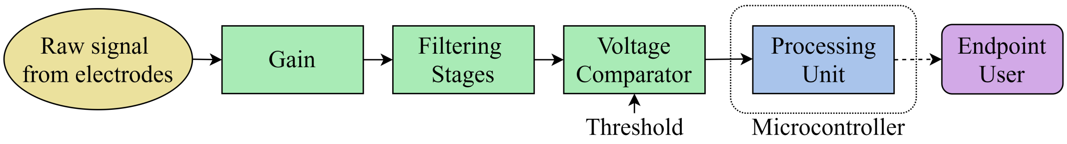


Figure 2.3: General structure of the sEMG signal acquisition chain if ATC is used: compared to the previous case (Figure 2.1), the presence of the ADC is no longer necessary. The voltage comparator used to generate the ATC signal is included among the components of the analog front end (AFE). The threshold is set by the microcontroller.

In [33], ATC was used for a biomedical application where data transfer was achieved using the Impulse-Radio Ultra-Wide Band (IR-UWB) wireless technology. In this work, it has been demonstrated an advantage in the use of ATC with regard to energy consumption. Indeed, instead of transmitting the entire sEMG signal, a pulse is transmitted every time a TC event occurs, obtaining a quasi-digital signal (see Figure 2.4). The result was validated with experimental tests using a dynamometer and carrying out a contraction up to 70% of the Maximum Voluntary Contraction (MVC). ATC was then compared with Average Rectified Value (ARV), computing the correlation of the two parameters with the force signal obtained. The ATC-force correlation was 0.95 ± 0.02, while the ARV-force one was 0.97 ± 0.02. These early results encouraged the possibility of using ATC to estimate muscle force. Further tests were performed in [34] during which the correlation between ATC and muscle strength was calculated as in the previous work, but the data was artificially corrupted to demonstrate the ATC robustness:

- As the Signal to Noise Ratio (SNR) varied, the maximum correlation value was reached starting from 6 dB;
- ATC was found to be independent by the amplification distortion effects;
- As the loss of TC events varied, the maximum correlation value was maintained up to 70% loss.

In addition to these tests, it was also demonstrated the possibility of reducing the size and complexity of the circuit by removing the ADC. Indeed, the latter component is no longer necessary if there is no need to acquire the entire information content of the sEMG signal.

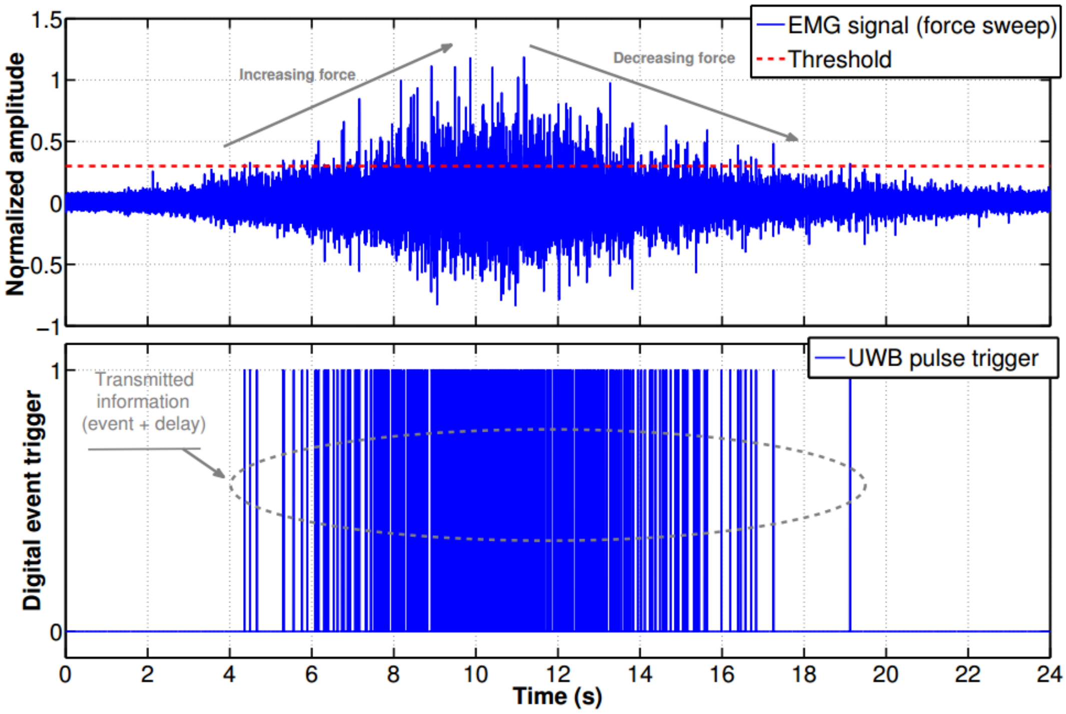


Figure 2.4: Application of average threshold crossing (ATC) to the electromyographic signal during a contraction. Each TC event is wirelessly transferred with Ultra-Wide Band (UWB) technology. The receiver obtains both the information on the number of events and their temporal distance [33].

In [35], the previously described system was tested *in vivo* and it was demonstrated that the variation of the ATC applied to the sEMG signal is dependent on the variations of force both during isometric and isotonic contractions. In order to accomplish this task, a set of weights (i.e., 2 kg, 4 kg, 6 kg, 8 kg and 10 kg) has been used. Using the ATC it was possible to discriminate without errors the contractions performed using weights of at least 6 kg of difference, as well as with respect to the resting condition. In [53] a sEMG wearable acquisition board was therefore developed and experimental tests were carried out for its validation.

ATC has also proved to be a valid technique in machine learning applications, such as hand gesture recognition, as described in [54] and [55]. In [54], a model was trained with Support Vector Machine (SVM) on the ATC data of four wrist movements, acquired using three input channels. The accuracy of the classifier obtained was 92.87%, with a latency of 160 ms. The latter proved to be a promising result, compared to that obtained using the sEMG signal, with which the accuracy obtained was 5.34% higher. Additionally, the use of ATC has significantly reduced the data rate to 30 B/s, compared to the 6.14 kB/s required using the sEMG signal. As regards energy consumption, the use of four channels that transmit ATC data via Bluetooth Low Energy resulted in an energy usage of 20.23 mW, which is approximately 14% lower than that measured with a single sEMG channel. In [55] instead, an embedded system was developed for the classification of six hand gestures, using three acquisition channels and the Apollo2 microcontroller from Ambiq Micro. Here the classifier implements a Neural Network (NN) obtaining an accuracy of 96.34%, with a latency of 268.5 ms and a power consumption of 2.9 mW.

In [56], ATC was implemented to control Functional Electrical Stimulation (FES). The acquisition board used is the one described in [53] and mentioned above. To obtain the ATC value, it was chosen to use a time window of 130 ms, obtaining a data throughput of about 30.7 B/s using four acquisition channels (the ATC data from each channel can be represented in 1 Byte). Considering the ATC acquisition phase alone, in which four channels were involved, a power consumption of 5.126 mW was measured. If Bluetooth transmission is also considered, the power consumption reached 20.23 mW. For the system management, a GUI was developed using MATLAB® and Simulink®. This system has been validated in [57] by comparing the movements from which ATC is acquired with the ones performed thanks to FES. The cross correlation value obtained was overall equal to 0.91. Since then, the system has evolved further. In [58], an embedded version was presented in which the event-driven control enabled latencies of 140 ms to be achieved in the FES parameter definition process.

2.3 FES Systems Controlled by sEMG

The use of the sEMG signal to control FES is a valid possibility in the rehabilitation field as it allows the patient to actively participate in the exercise. This does not happen if the electrical stimulation is carried out on command, for example by pressing a button, without the patient activating the neural pathways involved in motor activity. The sEMG-based FES control systems typically exploit two possible configurations:

- Therapist controlled FES: in this configuration, the stimulation electrodes are applied to the patient while the sEMG electrodes are applied to the therapist. Hence, the stimulation is controlled by the therapist and the patient repeats the movement with the FES aid. In [59] this configuration was explored obtaining an average latency of 400 ms between the movement performed by the therapist and the patient one.

- Controlaterally Controlled FES (CCFES): in this configuration the sEMG signal is acquired from the same subject who receives the electrical stimulation. The acquisition takes place on the opposite side to the stimulated one. For example, if the subject needs rehabilitation to be able to extend the left wrist, it is possible to use the sEMG signal acquired from the right wrist if the latter is healthy (see Figure 2.5). This configuration is particularly useful for the rehabilitation of subjects with hemiplegia or hemiparesis due to stroke [60]. An example of a system where this configuration was used is described in [61], in which, thanks to a classifier, 6 possible hand gestures are first discriminated and then FES is applied obtaining a movement execution latency of about 600 ms.

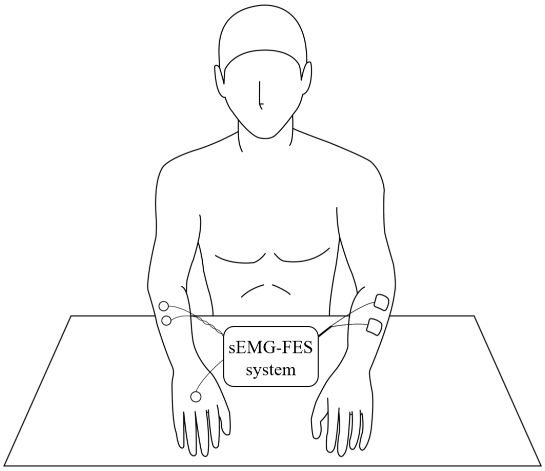


Figure 2.5: Experimental setup for Controlaterally Controlled FES (CCFES).

In [62] using a classification algorithm it is identified which movement is being made by the controller, among 4 possible hand gestures. Stimulation is then applied on the controlled side, in order to repeat the classified movement. The stimulation parameters selection is carried out by another algorithm called MNDC [63], which allows the system to modulate pulse width and stimulation frequency reducing muscle fatigue. The results obtained show that the angular trajectories performed have the maximum value of the cross correlation coefficient higher than 0.84 and a latency lower than 300 ms. This study is applicable to both CCFES and therapist controlled FES. In any case, the sEMG is acquired on the healthy side. The same research group published another work later [64] in which Sample Entropy is used for the identification of muscle contraction. The MNDC algorithm is still used for stimulation parameters modulation (frequency and pulse width). The objective of this study is oriented to gait and in particular to dorsiflexion and plantarflexion. The stimulation is triggered by the contraction of the antagonist muscle on the healthy side in order to have the correct alternation of movements during walking. The angular trajectories of the ankle were reproduced with a maximum cross correlation coefficient higher than 0.82 and a latency lower than 300 ms. However, the system is reported to be relatively bulky.

On the other hand, by applying the Average Threshold Crossing technique (ATC) to the sEMG, the system developed in [58] allowed to reproduce elbow flexion and knee extension movements with cross-correlation values around 0.87 and 0.93 respectively. Moreover this is an embedded system, which simplifies its integration in the rehabilitation field.

Most of the control systems currently available do not perform the sEMG acquisition in the same area as the stimulated one. However, it would also be useful to exploit the information of the sEMG signal evoked by the stimulation, in order to evaluate further parameters such as the patient's muscle fatigue during FES and consequently modulate the stimulation parameters. The problem in acquiring and stimulating the same area at the same time lies in the presence of the stimulation artifact, which is a spike-shaped signal that corrupts the electromyographic one. The systems in which the control of the FES is carried out by acquiring sEMG from the same area require custom modules for the stimulation artifact suppression and are typically constituted as the one represented in Figure 2.6.

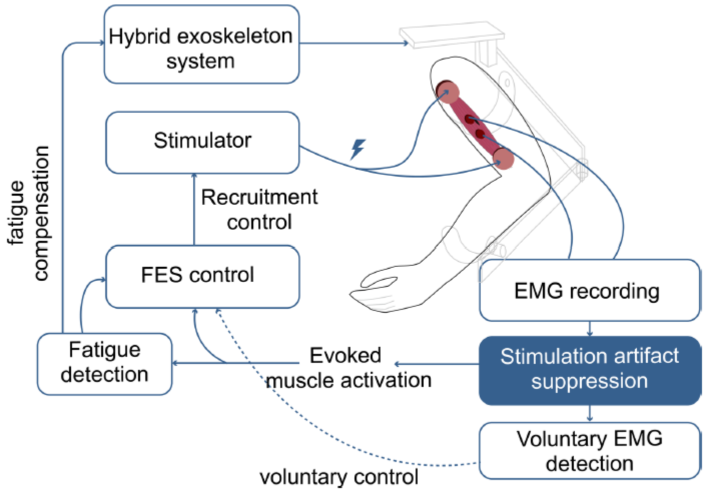


Figure 2.6: General system structure to acquire the sEMG signal from the same muscle on which the FES is applied [26].

2.4 Works related to FES Parameters Modulation

Muscle fatigue, for the reasons described in Section 1.4.3, is one of the main FES constraints. Several studies have been carried out to identify which stimulation parameters could be best suited to minimize it. The most relevant aspects of some of the studies carried out are described below.

- In [65], three stimulation protocols on 9 subjects with SCI were compared, in order to evaluate the best combination of stimulation frequencies to obtain 50 degrees of knee excursion during non-isometric contractions. The protocol in which the greatest number of excursions was successfully obtained on most subjects (6 out of 9) involves the use of an initial stimulation frequency of 20 Hz which is increased to 66 Hz when the subject cannot reach the target excursion. In the other protocols, 2 out of 9 subjects achieved better performance using 33 Hz as the initial stimulation frequency and then switching to 66 Hz. A single subject, on the other hand, performed the greatest number of excursions using a constant frequency at 66 Hz throughout the duration of the exercise. What this study suggests is that it is best to start exercise at low stimulation frequencies and then switch to higher frequencies when the subject begins to show the first signs of muscle fatigue. However, considering the inter-subject variability, it is recommended to determine the frequency values to be used for each subject.

- In [66], the effect of pulse width and stimulation frequency on muscle fatigue was studied testing three different stimulation protocols on 12 healthy subjects while performing isometric contractions. The combination of stimulation parameters of each protocol was chosen by keeping one parameter constant and varying the other one to have the same initial muscle force:

  - Protocol 1: fixed pulse width = 600 μs; average frequency = 11.5 Hz
  - Protocol 2: average pulse width = 150 μs; fixed frequency = 30 Hz
  - Protocol 3: average pulse width = 131 μs; fixed frequency = 60 Hz

  With protocol 1, the least decline in muscle force was achieved. This result suggests that using high pulse width and low stimulation frequency may be the winning strategy to face muscle fatigue. From a theoretical point of view, considering that using a high pulse width results in a greater number of motor units recruited, this means that fewer action potentials are required by each motor unit to generate the same force. Therefore, lowering the stimulation frequency decreases the ATP consumption for each motor unit and consequently decreases muscle fatigue since this is related to the metabolic demand.

- In [63] the *MAV/NASS Dual Coding* (MNDC) algorithm was developed, which is also the one cited in Section 2.3. This algorithm is used to control the stimulation parameters based on some features extracted from the sEMG signal. In particular, the algorithm consists of a transfer function which in input receives the mean absolute value (MAV) and the number of slop sign changes (NSS) and in output provides the pulse width and the stimulation frequency. The MAV is used as an indicator of the amount of motor units recruited and therefore it was chosen as the pulse width controller; the NSS instead reflects the firing rate of the motor units and consequently is used as a controller for the stimulation frequency. The mapping between the sEMG features and the FES parameters was performed using the isometric wrist torque as intermediate variable. This algorithm has proved to be valid both in reproducing the muscle force developed during a voluntary contraction and in reducing muscle fatigue. Indeed, with regard to the latter, it has been observed that by using higher values of pulse width and lower values of the stimulation frequency, muscle force is maintained for longer periods of time. Among the possible explanations of this result, there is also what was described in the work [66] reported in the previous point. This study was tested on 8 healthy subjects and inter-subject variability was observed implementing the transfer function to map the sEMG time features into the stimulation parameters.

- Since an increase in the number of motor units recruited is observable both increasing the pulse amplitude and the pulse width, in [67] these two types of modulation are compared with the same stimulation frequency (30 Hz). The experimental tests of this study were performed on 10 healthy subjects in isometric conditions. The two protocols tested consist of using short pulse width (200 μs) with high pulse amplitude or long pulse width (1000 μs) with low pulse amplitude. The stimulation currents used for each subject are such as to operate under the same muscle force developed. Muscle fatigue was greater in the protocol with short pulse width. A possible explanation of the result obtained lies in the possibility of triggering a voluntary contraction using reflex pathways, stimulating the type *Ia* afferent nerves. The latter are excitable by electrical stimulation by applying impulses of duration between 500 and 1000 μs [68]. Therefore, since the motor units are recruited in a physiological way, the developed muscle contraction is less affected by muscle fatigue than those triggered by electrical stimulation applied to motor nerves.

- In [69], the difference in pulse width modulation compared to pulse amplitude one was investigated. The study analyzes the effects of different pulse width values on 28 healthy subjects and concludes that modulating the pulse width while keeping the current constant is more advantageous than vice versa. The justification for these results is that if the pulse width is low there is a greater discrimination between the excitatory thresholds of nociceptive and non-nociceptive nerve fibers. Furthermore, with the pulse charge being equal, the use of short duration pulses results in an impedance decrease compared to the use of long duration ones, so the amount of charge required to obtain the contraction is less. Both aspects result in less discomfort for the subject. Therefore, the use of a pulse width modulation would allow the electrical stimulator to adapt to the subject using the shortest possible pulse duration values. This also translates into less battery consumption of the stimulator, facilitating its integration into a wearable system. Considering the results obtained from [69] and those obtained from the articles described above, it is possible that there is a trade-off around the pulse width between discomfort and muscle fatigue.

Chapter 3

System Description

The purpose of the system is the real-time control of Functional Electrical Stimulation (FES), using as input the result of the Average Threshold Crossing (ATC) applied to the surface ElectroMyographic Signal (sEMG). Therefore, when the muscular signal is acquired, the ATC data are firstly computed and then transmitted via Bluetooth Low Energy (BLE) to a computer. A dedicated software, described in details in Chapter 5, allows the user to manage the entire acquisition system, to process ATC data, to define the suitable FES pattern and to communicate with the electrical stimulator. Moreover, the system also comprises two Bluetooth electrogoniometers able to acquire articulations/limbs trajectory, useful to have a monitoring visual feedback. This system is suitable for use with both a therapist-patient or a CCFES approach (see Section 2.3).

The first version of this system was developed in [70], while in [71] it was validated. In [72], the system control software moved from MATLAB® to Python®, resulting in a multi-platform system. In this thesis project, the control software has evolved, allowing the adoption of a new acquisition system developed in parallel with this work. Furthermore, a new prototype of electrogoniometers has been developed and experimental tests were carried out to validate the system. Figure 3.1 shows a representation of the system: the following sections describe all the parts composing it.

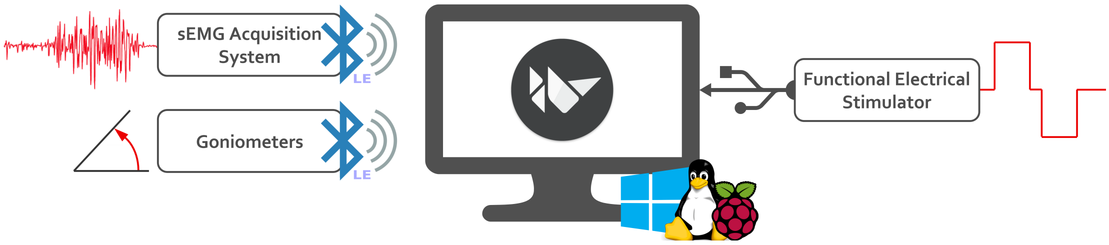
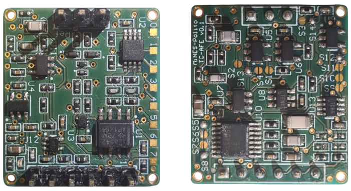


Figure 3.1: System overview: the involved hardware comprises the sEMG acquisition system, the goniometers and the functional electrical stimulator. The first two communicate with the control unit through Bluetooth, while in the latter the communication is via USB cable. The software has been developed using a multi-platform approach, so it runs on different Operating Systems (OSs) (i.e., Microsoft® Windows®, Linux® or Raspbian).

3.1 Bluetooth sEMG Acquisition System

The sEMG acquisition system is composed of the Analog Front End (AFE) channels and their digital control unit. Different setups are available: the AFEs can be coupled with the nRF52840 microcontroller [73] from Nordic Semiconductor®, as reported in [72], or they can be connected to the Apollo3 Blue ultra-low power microcontroller [74] from Ambiq Micro. Both microcontrollers are equipped with a Bluetooth antenna. The difference in using the two setups is that, with the same number of connected Bluetooth devices, while in the first scenario the acquisition channels are limited to one, the second setup allows the possibility to increase the total numbers of channels up to nine. Anyway, the application developed during this thesis project is able to manage both acquisition systems, even simultaneously. Figure 3.2 shows the double-layers PCB of an AFE of the latest setup, whose functional blocks diagram, representing the fundamental steps of sEMG conditioning circuit and ATC extraction, is reported in Figure 3.3.



(a) (b)

Figure 3.2: Single Analog Front End (AFE) channel of the sEMG acquisition system.

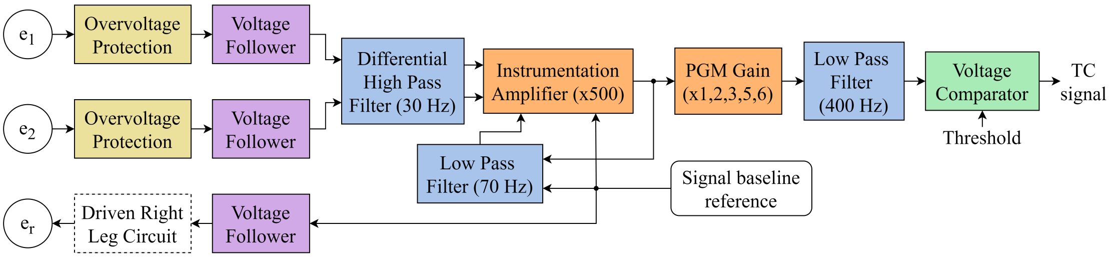
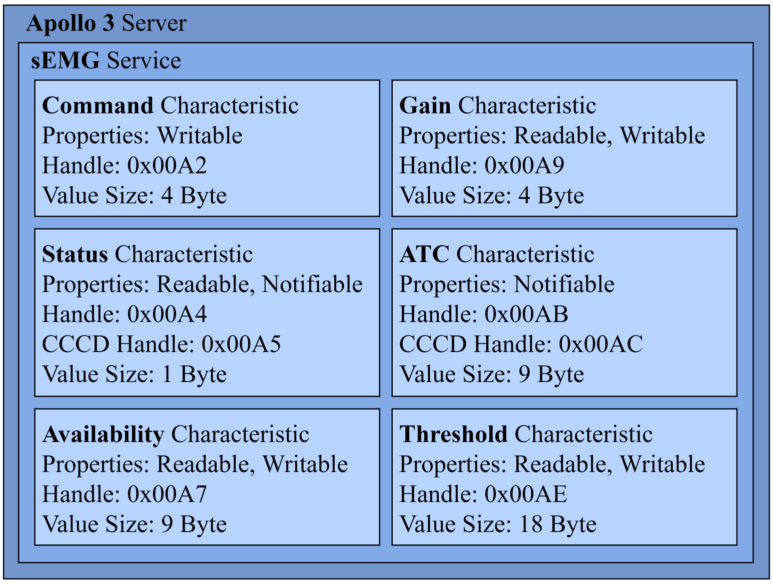


Figure 3.3: Block diagram of the Analog Front End (AFE) channels. The sampling configuration used is the single differential, whose electrodes are connected to an overvoltage protection, for patient safety. Each one of the three inputs is connected to a voltage follower, for decoupling from the electrode-skin impedance. Optionally, the reference electrode can be connected to the Driven Right Leg (DRL) circuit [75]. Using a passive differential high pass filter, with a cut-off frequency of 30 Hz, motion artifacts are adequately attenuated. The instrumentation amplifier provides a gain of 500 V/V and, in order to remove the electromagnetic noise corrupting the signal, a second order low pass filter (30 Hz) acts as feedback on its reference. The signal can then be further amplified using a programmable gain stage. The useful signal bandwidth is limited by a second order low pass filter with a cut-off frequency of 400 Hz. Finally, using a voltage hysteresis comparator, the threshold crossing (TC) signal is generated.

As regards the latest setup, in order to establish a dedicated Bluetooth communication with the sEMG acquisition system, Apollo3 implements the sEMG service represented in Figure 3.4. This service is made following the GAP/GATT specifications (see Section 1.5). Basically, the server is defined on the acquisition side, while the client has been implemented by the application developed in this thesis project, so the user control unit can read the ATC data and process them to define the FES pattern. The ATC value is assessed using a time window of 130 ms, as it represents a good trade-off between low data throughput and the discrimination of muscle activity [53]. Therefore, considering that this acquisition system can use up to 9 channels and the ATC value can be represented in 1 Byte, the ATC notifications data rate is about 69.23 B/s.



Apollo 3 Server

sEMG Service

Command Characteristic
Properties: Writable
Handle: 0x00A2
Value Size: 4 Byte

Gain Characteristic
Properties: Readable, Writable
Handle: 0x00A9
Value Size: 4 Byte

Status Characteristic
Properties: Readable, Notifiable
Handle: 0x00A4
CCCD Handle: 0x00A5
Value Size: 1 Byte

ATC Characteristic
Properties: Notifiable
Handle: 0x00AB
CCCD Handle: 0x00AC
Value Size: 9 Byte

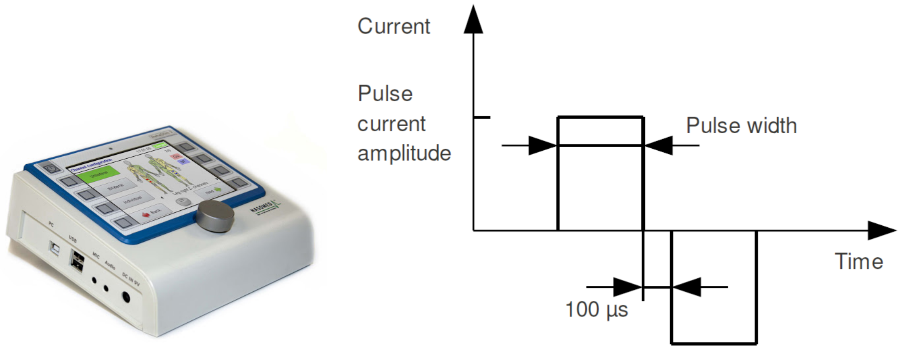
Availability Characteristic
Properties: Readable, Writable
Handle: 0x00A7
Value Size: 9 Byte

Threshold Characteristic
Properties: Readable, Writable
Handle: 0x00AE
Value Size: 18 Byte

Figure 3.4: Apollo3 server used for communication with sEMG channels. Through the *Command* characteristic, the user can control system functionalities, e.g., setting the threshold for ATC or programming the AFE gain (see Figure 3.3). The *Gain* and *Threshold* values are stored in two respective characteristics. Instead, through the *Availability* characteristic it is possible to assess which channels are connected and can be used. Both the *Status* characteristic and the *ATC* one allows the client to receive notifications: the former contains feedback about the executed commands, the latter holds the ATC value computed from TC signal.

3.2 Functional Electrical Stimulator

The electrical stimulator is the RehaStim2 [76] from HASOMED® (see Figure 3.5a). This is a medical device of class IIa according to EU guidelines MDD 93/42/EWG with applied part type BF (body floating) [77]. Thanks to built-in rechargeable battery, this stimulator is portable, making it suitable to be coupled with bio-signal acquisition system (usually strongly affected by power-line interference). The stimulator can interface with a PC through the ScienceMode2 serial communication protocol [78], allowing the user to have the direct control of the stimulation. RehaStim2 has 2 independent current generators, each one connected to 4 stimulation channels, for a total of 8 stimulation channels able to operate simultaneously. For safety reasons, the stimulator performs a tissue impedance check by sending a small test pulse before each stimulation pulse. If the impedance is outside the expected range, the stimulation is stopped. The stimulator is also connected to an emergency button which, when pressed, instantly stops stimulation. The stimulation waveform consists of biphasic rectangular pulses with balanced charge, as represented in Figure 3.5b. There is a delay of 100 μs between the positive and negative phases of the pulse in order not to nullify the action potential generated during the first pulse phase. The user-controlled stimulation parameters are reported in Table 3.1.



(a) (b)

Figure 3.5: (a) RehaStim2 electrical stimulator [76]; (b) RehaStim2 pulse waveform [78].

Table 3.1: RehaStim2 stimulation parameters [77].

Parameter	**Range**	**Step**
Current Intensity	0 - 130 mA	2 mA
Pulse Width	20 - 500 μs	10 μs
Frequency	10 - 50 Hz	5 Hz

The RehaStim2 features three operating modes:

- Single Pulse: in this mode, the stimulator applies a single pulse to one selected channel as soon as the command is processed.

- One Shot Channel List Mode (OSCLM): in this mode, a group of pulses is applied. There are three group modes: *single*, *doublet*, *triplet*. These three consist of the application of 1, 2 or 3 pulses respectively. The *interpulse interval* (i.e., the delay between two successive pulses if doublet or triplet has been selected) has to be set in the initialization command. Since each stimulation channel reserves a time slot equal to 1.5 ms and every generator is connected to 4 channels, the minimum value for the interpulse interval is 8 ms (4 x 1.5 ms + 2 ms communication buffer).

- Continuous Channel List Mode (CCLM): in this mode, like in OSCLM, a group of pulses is applied. Unlike the previous mode, the pulse group is automatically repeated at the set frequency. This value is computed as the inverse of the *main stimulation interval*, which has to be set in the initialization command. The OSCLM considerations about the interpulse interval are the same also for this mode. As regards the main stimulation interval, this condition must be respected: $t1 \geq t2 \times n_{p_{gr}}$ where $t1$ = main stimulation interval, $t2$ = interpulse interval, $n_{p_{gr}} = 1$ if single, 2 if doublet, 3 if triplet. Figure 3.6 shows an example of this mode.

  In this thesis project, CCLM is the primarily used stimulation mode because it allows the application to have a continuous stimulation, enabling real-time pulse parameters update (e.g., current intensity, pulse width)

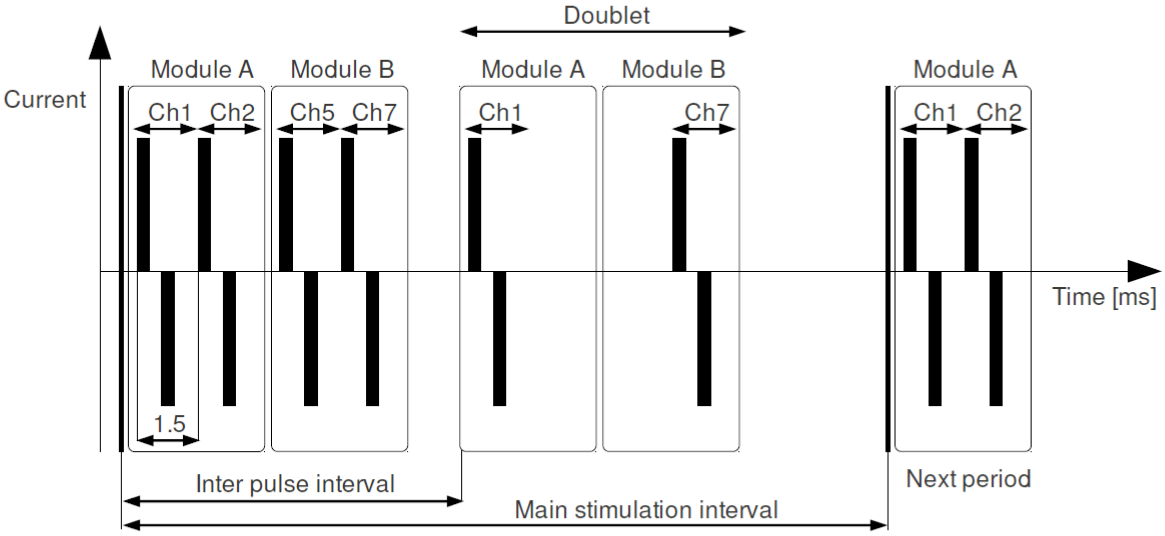


Figure 3.6: Continuous Channel List Mode example: channels 1 and 7 are set to use doublet as group mode, while channels 2 and 5 use single mode [78].

3.3 Bluetooth Goniometers

The analysis of the human movements related to the FES application has fundamental relevance for the proposed system. Therefore, in order to obtain digitalized and reliable data for post-processing, electrogoniometer prototype has been designed. The use of these devices is not mandatory but, anyway, it is strongly recommended, as they allow the user to monitor in real-time both the controller movements and the controlled ones. These electrogoniometers are the evolution of those developed in [71], with the following differences:

- Serial communication has been replaced with Bluetooth communication, in order to reduce the number of cables involved during the system usage, as they can be an obstacle to the patient's rehabilitation activity;
- The goniometer structure has been redesigned to have more than one Degree of Freedom (DOF) and to adapt to the kinematics of the joint.

The Bluetooth server used by these electrogoniometers is a remodelling of the one developed for the sEMG acquisition system (see Section 3.1). Figure 3.7 shows a photo of the electrogoniometer prototype, whose design is described in Chapter 4 and the results are discussed in Chapter 8.

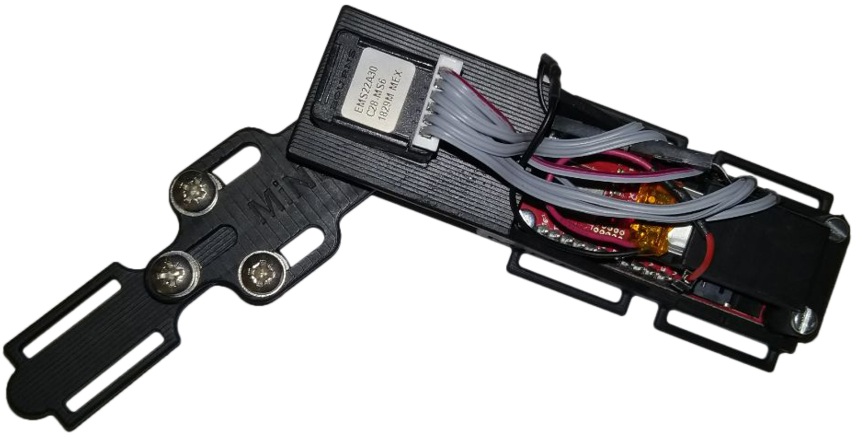


Figure 3.7: Articular electrogoniometer developed in this thesis project.

Chapter 4

Bluetooth Goniometers Design

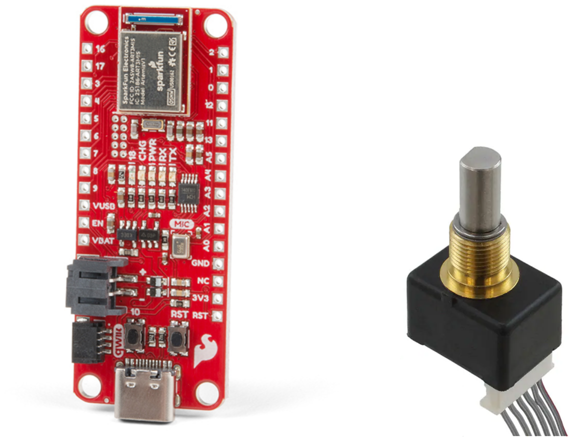
4.1 Conceptualization

Since skeletal joints do not possess a single Degree Of Freedom (DOF), the challenge of this design is the development of a goniometer architecture that allows the system to obtain an accurate and reliable measurement of the Range Of Motion (ROM) of the joint of interest. In particular, the center of rotation of a joint is not constantly located in space, but moves with the rotation of the joint. As regards the knee, for example, during its flexion the femoral condyles perform both a rolling and a sliding over the tibial plate [79]. An interesting architecture to address this problem is used in both [80] and [81] and consists of an articulated parallelogram that allows the measurement to be independent of the positioning of the goniometer axis with respect to the center of rotation of the joint. Hence, the first development attempt was to reproduce this structure.

4.2 Prototyping

Since the need to use Bluetooth electrogoniometers in order to have a wireless instrument that does not hinder the movements of the subjects, the Artemis Thing Plus [82] from Sparkfun Electronics® (see Figure 4.1a) has been chosen as the control module. Indeed, it integrates an Ambiq Apollo3 microcontroller and an antenna for Bluetooth Low Energy communication. The articulation angular value is monitored by means of an absolute encoder, i.e., EMS22A30-C28-MS6 [83] from Bourns® (see Figure 4.1b), featuring magnetic technology and 1024 resolution levels per revolution. Due to standard 3.3 V operating voltage, the encoder has been easily interfaced with Artemis Thing Plus board through SPI protocol (see Figure 4.2). A 100 mAh battery is used to power the electrogoniometer, obtaining about 6 hours of autonomy.

The CAD development of the goniometer structure was carried out using SOLIDWORKS®. The components realization has been done through 3D printing using Form 3 [84] from Formlabs with a standard biocompatible resin. The designed architecture consists of two main parts. The first one, whose CAD is shown in Figure 4.3, integrates the encoder, the Artemis thing plus and the battery.



(a) (b)

Figure 4.1: Commercial electrogoniometers components: in Figure 4.1a the Artemis Thing Plus board; in Figure 4.1b the EMS22A30-C28-MS6 encoder.

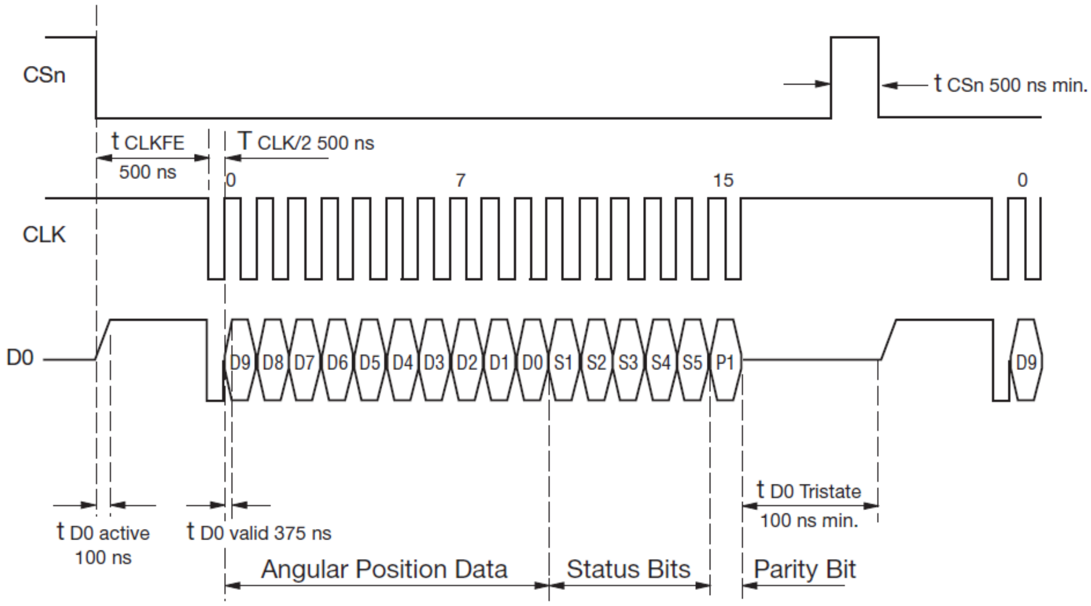


Figure 4.2: The encoder digital output (DO) is made up of 16 bits. The first 10 bits contain information about the absolute angular position. The other 6 bits are status bits, useful for error handling. The last is the event parity bit, for detecting errors in the previous fifteen ones. As represented, read operations are automatically triggered by controlling chip select (CS) pin.

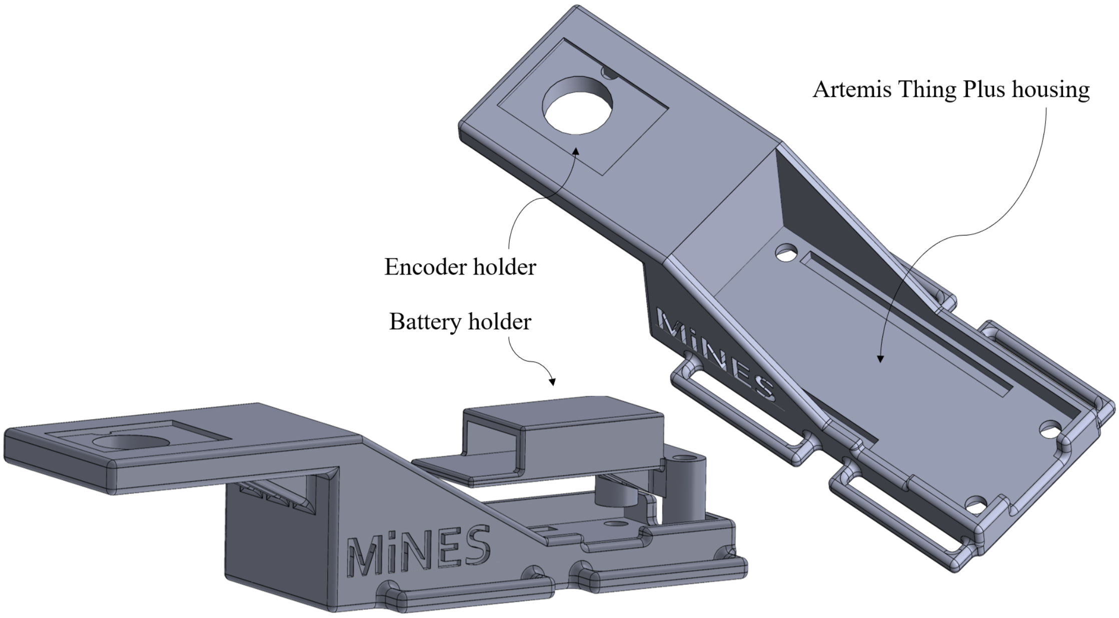


Figure 4.3: Fixed part of the electrogoniometers structure. It consists of the housing for the Artemis Thing Plus board and the one for the encoder. Furthermore, it offers a practical battery holder. The lateral slits are used to apply the goniometer on the subject limbs by means of elastic bands.

The second part of the goniometer structure has been created by applying what was studied during the conceptualization phase. Hence, taking inspiration from the articular goniometers used in [81], a structure with an articulated parallelogram has been designed. The obtained result is shown in Figure 4.4. However, this structure did not find a satisfactory outcome, since it was not able to match the movement of the subject throughout its ROM. This result can be explained considering two factors: the first is that the resin used may not be the ideal material, due to the friction present during movement; the second is that for high ROMs, such as those involved in this thesis project, often motion of the parallelogram rods (see figure 4.4) reached the respective limit, constraining the structure in the remaining phase of the movement.

Therefore, a new structure was designed, in which some notions learned during the development of the previous structure were implemented. Indeed, it has been observed that during the flexion-extension of a joint, since it is characterized by a variable center of rotation, the part of the goniometer bound to the movable articular segment tended to approach or move away from the encoder axis, which instead is fixed. Therefore, the new structure allows the device to perform this adaptation through sliding tracks, as reported in Figure 4.5. It is important to note that this structure, unlike the previous one, does not make the positioning of the goniometer independent from the center of rotation of the joint.

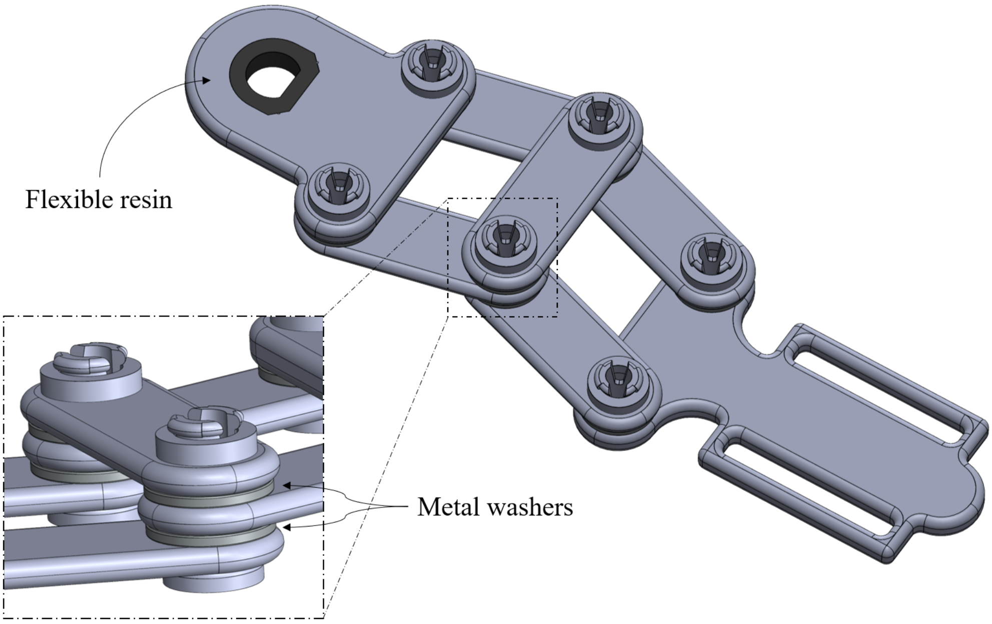


Figure 4.4: First version of the mobile part of the electrogoniometers structure. The articulated parallelogram allows the goniometer placement to be independent from the center of rotation of the joint. Indeed, the motion of the five rods inside the parallelogram allows it to modify its conformation to adapt to the joint. In order to reduce the friction between the elements of the structure, metal washers were inserted between them. The female of the encoder shaft (whose housing is on the fixed part of the goniometer) was made using the flexible resin, in order to clutch the shaft.

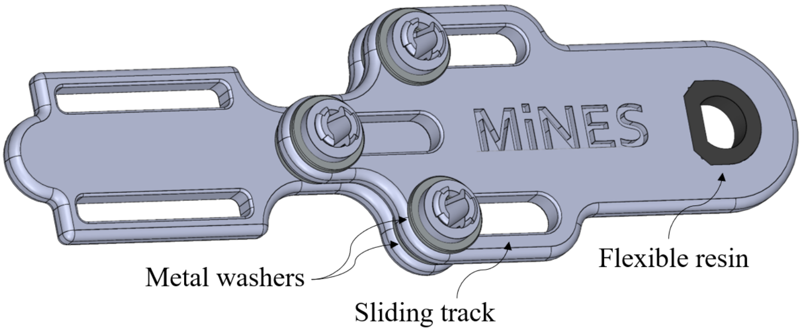


Figure 4.5: Second version of the mobile part of the electrogoniometers structure. As in the previous version in Figure 4.4, the metal washers were used to reduce the friction between the moving components of the structure and the flexible resin was used to clutch the encoder shaft. The structure is composed of three tracks, to allow the goniometer to lengthen and shorten during the joint movement. It has been chosen to use a triple track in order to prevent rotation between the two components, as it would falsify encoder measurement.

Chapter 5

Software Design

5.1 Software Requirements

- **Reliability**: since the proposed system is intended for healthcare users in the rehabilitation field, it is necessary to design a reliable application, with deterministic behavior and, ideally, with a zero probability of failure.

- **Extensibility**: the system is still evolving, and it is therefore necessary that the application be upgradable over time, not necessarily by the same person who designed it, by adding new features or correcting any bugs found in the current ones.

- **Scalability**: due to the versatility of human movements, a different number of acquisition channels can be requested to adequately record the total activity of the muscles related to an action. In order to accomplish this task, the application have to maintain its performance independently by acquisition hardware configuration.

- **Modularity**: designing an application made up of independent modules simplifies the system integrity. Indeed, when each module exposes its *Application Programming Interface* (API), if during the development of the system it is decided to replace a device (e.g., the electrical stimulator), it is not necessary to redo the entire application but it is sufficient to replace the respective module, creating APIs compatible with previous ones.

5.2 Software Overview

In order to satisfy the requirements listed in Section 5.1, the application has been designed in Python® following the Object Oriented Programming (OOP) paradigm. Furthermore, by exploiting the potentiality of multithreading, the application is able to control the entire system in real-time, obtaining the latency times described in Chapter 6. This project is an evolution of what has been developed in [72]. The application is designed on three layers, as shown in Figure 5.1. At the bottom layer, the *BLE* and *FES* objects are instantiated for managing both Bluetooth communication and electrical stimulator control, respectively. At the middle layer, the *System* object is instantiated in order to integrate the two previous ones, putting them in communication with each other. Finally, at the top layer, the *Graphical User Interface* (GUI) is defined by inheriting from the classes of the *Kivy* Python framework [85]. This library has been chosen in order to make the application multi-platform, therefore able to run on different OSs (Microsoft® Windows®, Linux®, Raspbian). In particular, in order to obtain an embedded system, the application has been developed also taking into consideration its proper functioning on low-computational device, i.e., Raspberry Pi. Depending on the computer OS, the Bluetooth communication has been implemented using a different hardware: in the Microsoft® Windows® OS case, the USB dongle CC2540 [86] from Texas Instrument™ is used as external antenna; otherwise (i.e., Linux®, Raspbian OSs) integrated Bluetooth antenna is employed. In Figure 5.2, the UML diagram of this application is represented.

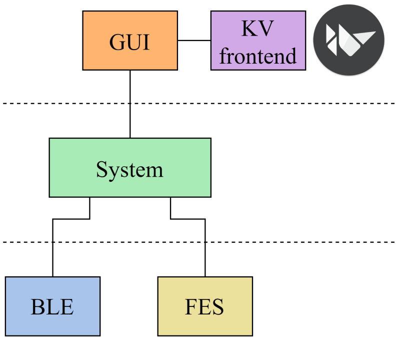


Figure 5.1: Software modules overview: the application is built on three independent layers. The GUI module, which is at the top layer, is coupled to the application frontend. Indeed, Kivy has its own KV language [87] which allows the designer to separate the interface logic (written in Python) from its aspect (written in KV), supporting the modularity of the system.

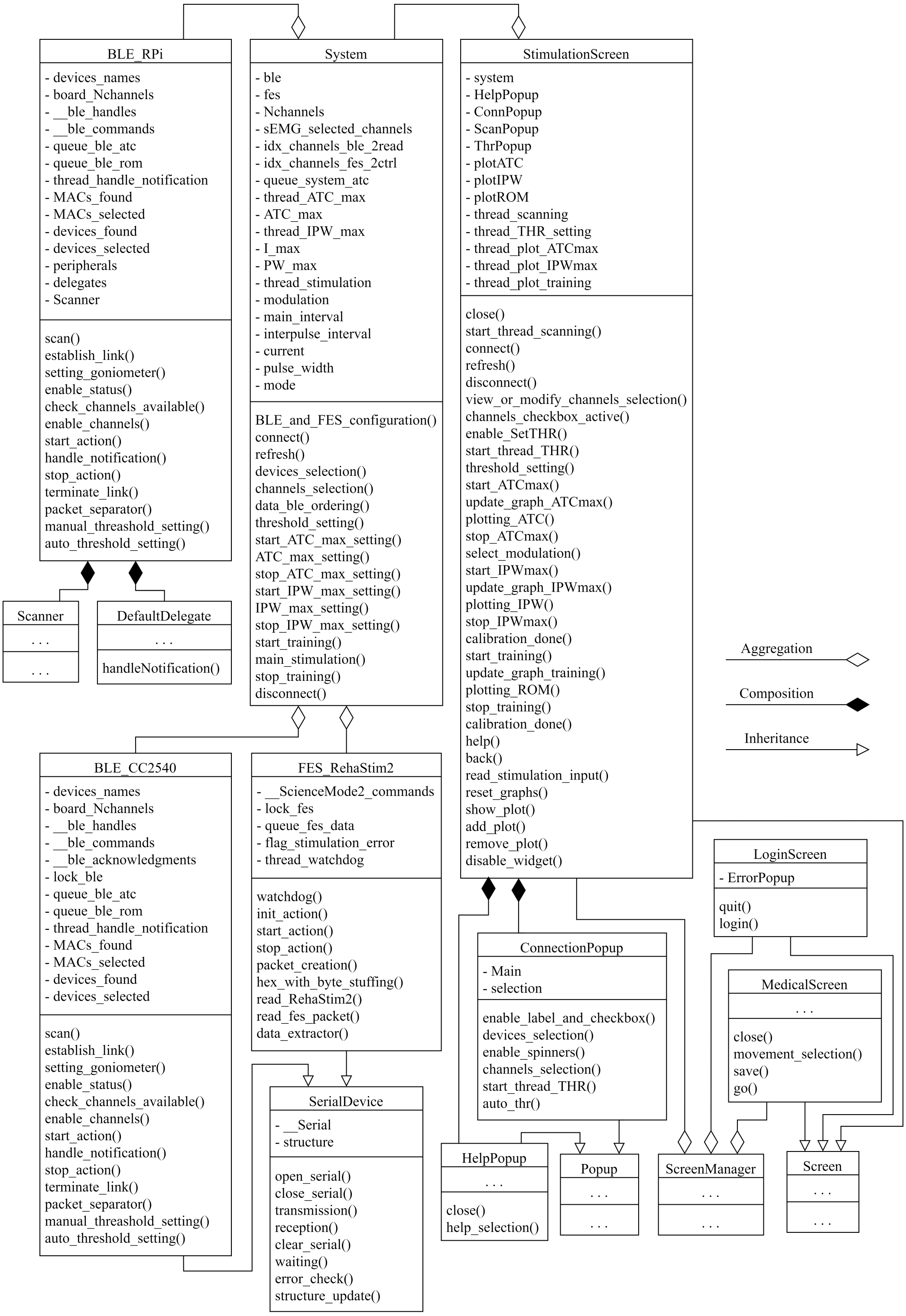


Figure 5.2: UML diagram of the application designed.

5.3 Serial Device Class

This class has been created in order to be the parent of the FES class and the BLE one when dongle CC2540 is used. Indeed, both the Bluetooth USB dongle and the electrical stimulator interface to the personal computer via serial communication. Hence, this class encapsulates the Serial class of the pySerial library [88] and has the following methods:

- open_serial: this method tries to open the serial port and writes information about it on the logger. The management of possible errors during the opening phase of the serial communication, such as the missing COM, is carried out by means of an attribute which acts as a flag and which is subsequently checked.
- close_serial: close the serial port.
- transmission: sends the desired bytes through the serial port.
- reception: returns the data present on the serial port. If the purpose is to read a defined number of bytes, this number can be supplied as input to the method. Otherwise, the method will read until the *end of line* (EOL) or until the reading is stopped by the timeout.
- clear_serial: clears the serial input buffer.
- waiting: checks for data availability on serial port.
- error_check: this method is used to check the status of the received data in order to handle the error if the status is not successful. The implementation of this method is mandatory, so an *Exception* is raised if the child of this class does not define it.
- structure_update: this method is used to update the logger about the data exchanged with the device. Like the method described above, this is also mandatory.

5.4 BLE Classes

The purpose of this class is the management of the Bluetooth communication between the central device and the peripherals. As reported in Chapter 3, the peripherals used are the sEMG acquisition system and the electrogoniometers. The former can be either individual channels or a board to which multiple channels are connected. If the application is running on Windows, the central device must be the USB dongle CC2540. Otherwise, if the application is running on Linux, it is possible to choose whether to use the USB dongle or BlueZ, which is the official Linux Bluetooth protocol stack [89]. In order to manage these two alternatives, two BLE classes have been created: the first uses the USB dongle, therefore it is inherited from the Serial Device class; the second interfaces with BlueZ, so it uses the bluepy library [90]. These two classes use the same methods to be equally compatible with the other classes belonging to the application. The internal implementation of these methods is clearly different depending on whether they interface with the USB dongle CC2540 or with BlueZ using the bluepy library.

- open_serial: this method is inherited from the Serial Device class and is used only by the class that interfaces with the USB dongle. In addition to establishing the serial connection with the latter, the *MinConnInterval* and the *MaxConnInterval* are set respectively equal to 7.5 ms and 55 ms. These two parameters were chosen as the best results able to guarantee timely receipt of notifications.

- scan: this method scans for available Bluetooth devices looking for those on the white list. The latter is written in the configuration file and is the list of allowed devices.

- establish_link: this method establishes the Bluetooth connection with the devices selected from those found by the scan method. Selected devices are indicated in an attribute of this class. The setting_goniometer method is called for each electrogoniometer among the selected devices. If the sEMG acquisition system using Apollo3 is among the selected devices, the enable_status method is called. In the case of the class that uses the bluepy library, for each peripheral with which the connection is established, a *DefaultDelegate* object is instantiated that will be responsible for handling notifications.

- setting_goniometer: the sampling settings of the electrogoniometers are the responsibility of this method, which has to write them on the respective characteristics of the server. The parameters involved are the sampling period (i.e. how often the reading is carried out via SPI) and the number of samples to be averaged before notifying. The client will then be notified every $SamplingPeriod \times Sample2Average$.

- enable_status: this method writes 1 to the CCCD of the status characteristic in order to enable the notifications about the board status, useful for signalling to the user the onset of a problem.

- check_channels_available: since the sEMG acquisition system with Apollo3 is able to manage up to 9 channels, but typically those connected are fewer, through this method it is possible to know which channels are available. This information is provided by the server, so this method reads the respective characteristic and returns the obtained value.

- enable_channels: like the previous method, this one has been implemented for using the acquisition system with Apollo3. Since a sEMG channel have to be enabled to be used, using this method it is possible to write in the same characteristic of the previous method to enable one or more channels among those available. It is not possible to enable an unconnected channel.

- start_action: this method receives as input the list of devices that have to start sending notifications. Therefore it writes 1 to the CCCD of the respective characteristics: as regards the sEMG acquisition systems, the characteristic of interest is the one containing the ATC value; for electrogoniometers it is the one containing the angular value. Finally, a thread targeting the handle_notification method is started.

- handle_notification: the logic of this method is fundamentally different for the two BLE classes. In both cases the method consists of a while loop that ends when the stop_action method is called. In the case of the class that uses the CC2540 dongle, inside the while loop the data are read from the serial buffer. Each time a notification is received, the origin device is first identified by checking the connection handle (whose information is contained in the received packet). Then, the value of interest (the ATC data or the angular data) is extracted from the packet and is put in the queue of the respective device. Instead, in the case of the class that uses bluepy, for each device with notifications enabled, this while loop does nothing but wait for the object instantiated by the *DefaultDelegate* class to receive the notification. Indeed, this class has a callback, which is a method that is called every time a notification is received. Within this method, the data of interest is then extracted and inserted in the respective queue. The queues used (not only here but throughout the application) are Python objects belonging to the queue library. Queue management is asynchronous: there is a queue for each device, so even if all the devices send data with the same frequency (for example, both the acquisition system with the nRF and the one with Apollo3 are connected, both sending data every 130 ms), the data are managed by two separate queues, so that if one device slows down, the other is not bound.

- stop_action: first, this method ends the while loop of handle_notification. Then, operating on the same characteristics as method start_action, it writes 0 in order to disable notifications. Finally, it cleans up the queues of the used devices.

- terminate_link: this method terminates all the active Bluetooth connections. In the case of the class that uses bluepy, the *DefaultDelegate* objects instantiated for each peripheral are also deleted.

- close_serial: this method is inherited from the Serial Device class and is used only by the class that interfaces with the USB dongle. Its purpose is to close the serial communication with the CC2540.

- packet_separator: this method belongs only to the class that interfaces with the USB dongle. The purpose of this method is to separate the different packets which are however read together using the reception method of the Serial Device class. Indeed, the latter is not able to distinguish two immediately following packets. Therefore, this method separates the received packets knowing that each packet transmitted by the CC2540 starts with *0x04FF*. The method then returns a buffer containing the received packets.

- manual_threshold_setting_nRF: receives as input the new threshold value to be used for ATC and writes it in the respective characteristic of the nRF server.
- auto_threshold_setting_nRF: starting from a threshold value equal to the supply voltage of the Digital to Analog Converter (DAC) connected to the nRF, this method automatically searches for the threshold. It basically consists of two while loops. The main loop lasts until the threshold is found. Within this loop, first the ATC notifications are started to receive the ATC data. Then, the second loop is started with a duration of 1 second within which the ATC value is read. If a value greater than 0 is found, the main loop ends. Otherwise, the threshold is lowered. To set the threshold at the end of each main loop step, the manual_threshold_setting_nRF method is called.
- manual_threshold_setting_Apollo3: receives as input the information about the channels with the threshold to be set and the desired threshold value; then sets the threshold by writing on the characteristic dedicated to commands. The command is coded on 5 bytes: the first byte is the identifier of the command; the second and third bytes contain the coding of the channels for which to set the threshold; the fourth and fifth bytes are the threshold value. As regards the channel coding, if it has been chosen to set the threshold for channels 1, 2 and 9 for example, the result would be *0b0000000100000011*, that is *0x0103*.
- auto_threshold_setting_Apollo3: this method, like the previous one, uses the command characteristic. The commands that can be used by this method are two: the first automatically search the thresholds for all the enabled channels, the second instead performs the search only for the selected ones between them. In the second case it is therefore necessary to provide as input also the information about the channels whose threshold needs to be calibrated. Unlike the case of nRF, here the threshold searching is implemented on the Apollo3 firmware in order to embed this functionality. Hence, the client only has to send the command and wait to receive the Apollo3 status indicating whether the threshold setting was successful or not.
- error_check: the status of the received packet is inspected by this method. Each possible status byte received indicates if there have been issues interfacing with the Bluetooth protocol stack. This method belongs only to the class that interfaces with the USB dongle.
- structure_update: this method updates the logger about the commands exchanged with the CC2540 dongle to interface with the Bluetooth protocol stack. For each response received from the USB dongle, it keeps track of the command name, its status and the timestamp at which the response was received. In the case of establish link command, the device name and its connection handle are also saved. This method belongs only to the class that interfaces with the USB dongle.

5.5 FES Class

This class has been created in order to communicate with the RehaStim2 electrical stimulator using the ScienceMode2 communication protocol cited in 3.2. This class is inherited from the Serial Device class and has the following methods:

- open_serial: this method is inherited from the Serial Device class. Once the serial connection is open, communication protocol version is checked. If it is correct, the RehaStim2 enters the *Stimulation start mode* (see Figure 5.3) and a thread that targets the watchdog method is started. For writing to the serial port, the transmission method of the parent class is used.

- watchdog: the purpose of this method, working as a thread, is to maintain the connection with the RehaStim2. Indeed, if the latter does not receive anything for 1200 ms, it disconnects itself. So, what this thread does is to check how much time has passed since the last serial write and if at least 1000 ms have passed then it sends a watchdog command. In this method, as well as in init_action, start_action, stop_action and close_serial, the *threading lock* is used in order to make sure that this thread does not write data to the serial port at the same time as the other methods are doing (for example during the stimulation).

- init_action: this method allows RehaStim2 to enter the *Stimulation initialized* mode (see Figure 5.3). The command sent with this method (i.e., *InitChannelListMode*), contains the setting of the main stimulation interval (i.e., the inverse of the stimulation frequency), the interpulse interval and the activated channels. It also contains additional information such as setting the low frequency factor, which however is not used in this application.

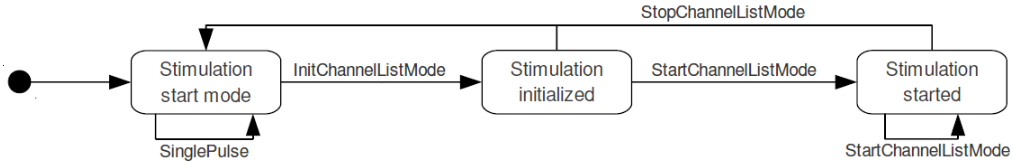


Figure 5.3: State diagram of RehaStim2 [78].

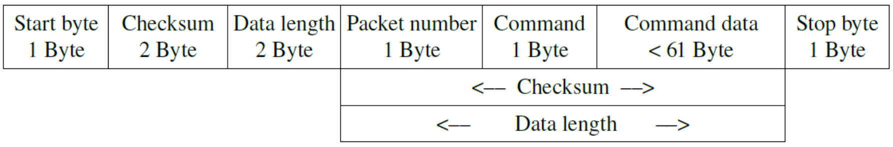
- start_action: this method allows RehaStim2 to enter the *Stimulation started* mode (see Figure 5.3). The command sent with this method (i.e., *StartChannelListMode*) contains the setting for current intensity, pulse width and group mode for each activated channel. This command allows both to start the stimulation and to update its parameters. Each time this command is transmitted, the stimulation parameters are inserted into a Python queue. These parameters will be retrieved from the queue by a method of another object in charge of plotting them.

- stop_action: this method sends the command to stop the stimulation (i.e. *StopChannelListMode*) and return to the *Stimulation start mode* (see Figure 5.3).

- close_serial: this method stops the watchdog thread and close the serial communication between PC and RehaStim2 inheriting the parent class method.

- packet_creation: this method builds the packet to be transmitted according to what is defined in the ScienceMode2 protocol and represented in Figure 5.4 (the structure of the packet is this both in transmission and reception). Each packet consists of the following parts:

  - Start byte: it is equal to *0xF0* and serves to delimit the beginning of the packet.
  - Checksum: this is performed with a cyclic redundancy check (CRC), which is *CRC-8-CCITT* and the polynomial used is $x^8 + x^2 + x + 1$. The checksum is carried out on the payload, which is composed by the packet number, the command identification value and the command parameters. The CRC returns a result of 8 bits, so 1 byte, but the ScienceMode2 protocol uses byte stuffing on the checksum, so 2 bytes are used.
  - Data length: the number of bytes composing the payload. As for the checksum, the result is 1 byte but byte stuffing is applied so 2 bytes are used.
  - Packet number: this is a counter from 0 to 255 (1 byte) of packets exchanged. When the overflow is reached, the count restarts from 0. When the counter is equal to 15 or 240, the byte transmitted is respectively *0xF0* or *0x0F*, therefore it is necessary to apply byte stuffing in order not to misidentify the beginning or the end of the packet.
  - Command: the ScienceMode2 protocol allows the application to exploit several commands, the use of which is described in the protocol specifications [78]. Each of these commands is provided with an identification number, which can be represented on 1 byte.
  - Command data: this piece of the packet contains the necessary parameters for the command. The dimension of this piece are therefore variable and command-dependent. For example, the *watchdog* command requires no parameters while *InitChannelListMode* requires at least 7 bytes (if byte stuffing is not required).
  - Stop byte: it is equal to *0x0F* and serves to delimit the end of the packet.



Start byte 1 Byte	Checksum 2 Byte	Data length 2 Byte	Packet number 1 Byte	Command 1 Byte	Command data < 61 Byte	Stop byte 1 Byte
			<— Checksum —>			
			<— Data length —>			

Figure 5.4: Structure of a communication packet of RehaStim2 [78].

- hex_with_byte_stuffing: since the ScienceMode2 protocol uses constants such as the start byte and the stop byte, it is necessary to manage the possibility that another byte of the packet is equal to these values. For this reason, byte stuffing is implemented. Byte stuffing requires the use of a *stuffing key* and an additional byte constant: the *escape character* (ESC). Performing this technique requires applying an

XOR operation between the stuffing key and the byte with value equal to a constant. The result obtained is stuffed by the ESC. The value assigned to the stuffing key is *0x55*, while the ESC value is *0x81*. This method is called whenever during the creation of the transmission packet it is necessary to convert values into hexadecimal in order to be expressed in bytes. Example: if during the transmission of the *StartChannelListMode* command, the pulse width assigned to a channel is 240 μs, when it is converted to hexadecimal it is *0xF0*. To ensure that this byte is not interpreted as the start byte, the *XOR* operation is performed with *0x55* whose result is *0xA5*. Finally, with the application of the ESC, *0x81A5* is obtained.

- read_RehaStim2: this method allows to read and manage RehaStim2 responses such as acknowledgments or stimulation errors. In order to do that, this method sequentially calls read_fes_packet, data_extractor, error_check and structure_update.
- read_fes_packet: this method reads bytes from the serial port starting from the start byte up to the stop one and returns the complete packet. For reading from the serial port, the reception method of the parent class is used.
- data_extractor: this method checks that the received packet has the correct size and returns the command acknowledgment number and the status position within the received packet.
- error_check: the status of the received packet is inspected by this method. For each possible status byte received, a respective message is reported on the ScienceMode2 protocol [78].
- structure_update: this method updates the logger about the command acknowledgments received from the RehaStim2. For each acknowledgment it keeps track of its name, status, packet number and the timestamp at which it was received. An attribute is also managed which works as a flag and which is made available to other objects to signal if an error has occurred during stimulation, in order to be able to manage its interruption within the application.

5.6 System Class

This class was created in order to integrate the methods of the BLE and FES classes. This class also represents the intermediary between the latter and the class that defines the GUI. The methods within this class represent the use cases of the system and are described below:

- BLE_and_FES_configuration: the purpose of this method is to instantiate the BLE and FES objects. First of all, it identifies which serial ports the RehaStim2 and the CC2540 dongle are connected to. In order to do that, the VID and PID of the two USB devices have been saved in the configuration file. Using these two identifiers, the method goes back to the respective serial ports. If the application is running on Raspberry Pi and from the configuration file it has been selected not to use the dongle, the BLE object that is instantiated is not the one that uses the CC2540 but the one that uses the bluepy library.

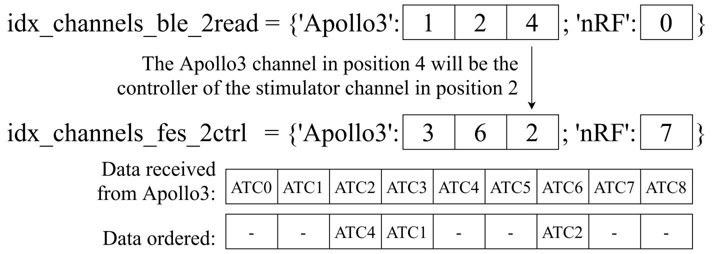
- connect: this method calls the open_serial methods of the BLE and FES classes to open the serial communication and then the scan method of the BLE class to search for available Bluetooth devices among those present in the white list. If bluepy is used instead of the dongle, the serial communication is not opened for the BLE object.

- refresh: this method terminates the Bluetooth connection with all connected devices and rescan for available ones. The methods invoked are respectively terminate_link and scan of the BLE class.

- devices_selection: this method receives as input the list of Bluetooth devices to connect to and calls the establish_link method of the BLE class.

- channels_selection: this method receives as input a list of 8 elements, respectively assigned to the 8 channels of RehaStim2. Each of these elements is a Python dictionary having two keys: *board* and *channel_idx*. The value of *channel_idx* is the number related to the acquisition channel to be used (referred to the board to which it is connected); the respective value of *board* instead is the name of the Bluetooth device to which the acquisition channel is connected (*nRF* or *Apollo3*). This list is useful for creating two attributes necessary for sorting the data received from BLE devices. These two attributes (idx_channels_ble_2read and idx_channels_fes_2ctrl) will be described in the data_ble_ordering method. Finally, if among the selected channels there are also channels connected to Apollo3, since it is necessary to enable them, this method calls enable_channels of the BLE class providing as input the list of channels to be enabled.

- data_ble_ordering: this method is in charge of sorting the data received from the sEMG channels according to the stimulator channels to which they have been assigned. To do this, the two attributes mentioned above are used. Both are dictionaries that have as keys the names of the boards used for the Bluetooth interface of the sEMG acquisition channels. In idx_channels_ble_2read, the values are lists containing the indices of the channels used for each board. In idx_channels_fes_2ctrl instead, the values are lists containing the indices of the stimulator channels to control. The indices position in the two lists is the same for the sEMG controller channel and the respective controlled stimulator channel. Figure 5.5 represents what has been described.



idx_channels_ble_2read = {'Apollo3': 1 | 2 | 4 ; 'nRF': 0 }

The Apollo3 channel in position 4 will be the controller of the stimulator channel in position 2

idx_channels_fes_2ctrl = {'Apollo3': 3 | 6 | 2 ; 'nRF': 7 }

Data received from Apollo3:	ATC0	ATC1	ATC2	ATC3	ATC4	ATC5	ATC6	ATC7	ATC8
Data ordered:	-	-	ATC4	ATC1	-	-	ATC2	-	-

Figure 5.5: Sorting mechanism of the data received by the sEMG channels.

- threshold_setting: this method is the node within which, based on the inputs provided, it is decided which method to call of the BLE class for the search and / or setting of the threshold to be applied for the ATC. If the method to call is auto_threshold_setting_Apollo3, it also calls check_channels_available in order to decide which command has to be sent.

- start_ATC_max_setting: in order to adapt to the variability among individuals who acts in the role of controllers, the application calibration phase involves the setting of the maximum ATC, i.e. the maximum ATC value that the subject is able to obtain by carrying out a voluntary contraction. Since the setting of the maximum ATC value is done with one channel at a time, this method first ensures that only one channel has been selected and then calls the start_action method of the BLE class and starts a thread with the ATC_max_setting method as target.

- ATC_max_setting: this method consists of a while loop that continues until the stop_ATC_max_setting method is called. Inside this loop, the ATC data are get from the queue where the BLE class inserts them when they are received. The data is then sorted with the data_ble_ordering method based on the channel selection. Next, the sorted data are put in the queue of this class to make them available to a higher level object in order to plot them. Finally, the data is processed. Whenever an ATC value greater than 1 is received for at least three consecutive times, it is considered that there is a movement in progress. The maximum ATC value obtained from each movement is appended to an array. The median of the values within this array constitutes the maximum ATC value for the selected channel.

- stop_ATC_max_setting: this method calls the stop_action method of the BLE class, ending the ATC_max_setting thread. Finally, the ATC queue of this class is emptied.

- start_IPW_max_setting: for the same inter-individual variability reasons mentioned in start_ATC_max_setting, the application calibration phase also involves the setting of the maximum stimulation intensity. The latter, depending on the modulation chosen, can be the current intensity (I) or the pulse width (PW). The purpose of this method is to identify the stimulation intensity needed to execute a complete movement. This method works on one channel at a time, so it first checks that only one channel has been selected. Then, it starts a thread that targets method IPW_max_setting.

- IPW_max_setting: first, this method initializes the stimulation channels calling the init_action method of the FES class. Then, the method enters a while loop which will be stopped by calling stop_IPW_max_setting. Within the cycle, the stimulation is commanded by calling the start_action method of the FES class. The modulation of the stimulation intensity is established by a stepped wave. Each step consists of an interval of 200 ms using increasing stimulation intensities of 2 mA (or 40 μs), up to a maximum value. Then, the stimulation intensity decreases with the same previous steps. At the end of each stepped wave, there is a rest phase lasting 4 seconds, before the start of a new stepped wave whose maximum value will be 2 mA (or 40 μs) higher than that of the previous stepped wave. When this method is terminated, the maximum stimulation intensity is set equal to the maximum value of the next stepped wave (i.e. the maximum of the last one, plus 2 mA (or 40 μs)).

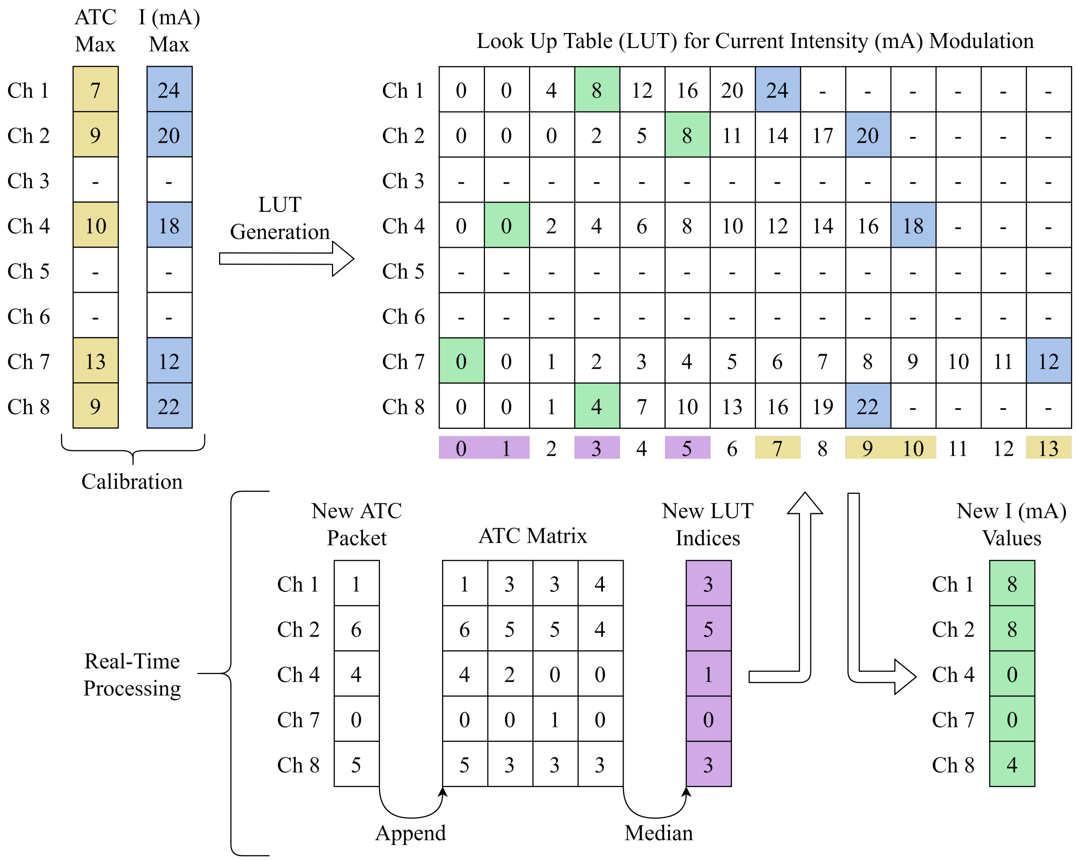
- stop_IPW_max_setting: this method ends thread IPW_max_setting and calls the stop_action method of the FES class.

- start_training: this method first ensures that at least one channel has been selected. Then, it calls the start_action method of the BLE class providing as input the names of the Bluetooth devices selected (i.e. the boards to interface with the sEMG channels and the electrogoniometers if they are used). Finally, it starts a thread with the main_stimulation method as target.

- main_stimulation: this method is the core of the application. The stimulation channels are firstly initialized with the init_action method of the FES class. Then, using the results of the calibration phase, the Look Up Table (LUT) needed for the processing phase is generated. Indeed, the maximum values of ATC and of the stimulation intensity are used to identify the extremes of each row of the LUT, as represented in Figure 5.6. At this point the while loop begins and will end when the stop_training method is called. Inside the loop, it checks whether there is any data waiting in the BLE class queue. If so, the following operations are performed (see Figure 5.6 as reference):

  - getting data from the queue;
  - sorting data using the ble_data_ordering method;
  - putting sorted data in the queue that will be used by the object that will plot the data;
  - appending the sorted data to the ATC matrix, keeping up to 4 values for each channel (i.e. the new value plus the three previous ones);
  - computing the median of the ATC matrix for each channel: the use of the median on 4 ATC values confers robustness to the modulation of the stimulation intensity;

  - reading the new stimulation intensity value from the LUT using the result from the previous step;
  - transmission of the command with the new stimulation parameters to the electrical stimulator calling the start_action method of the FES class.



	ATC Max	I (mA) Max
Ch 1	7	24
Ch 2	9	20
Ch 3	-	-
Ch 4	10	18
Ch 5	-	-
Ch 6	-	-
Ch 7	13	12
Ch 8	9	22

Look Up Table (LUT) for Current Intensity (mA) Modulation

Ch 1	0	0	4	8	12	16	20	24	-	-	-	-	-	-
Ch 2	0	0	0	2	5	8	11	14	17	20	-	-	-	-
Ch 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ch 4	0	0	2	4	6	8	10	12	14	16	18	-	-	-
Ch 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ch 6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ch 7	0	0	1	2	3	4	5	6	7	8	9	10	11	12
Ch 8	0	0	1	4	7	10	13	16	19	22	-	-	-	-
	0	1	2	3	4	5	6	7	8	9	10	11	12	13

	New ATC Packet	ATC Matrix				New LUT Indices
Ch 1	1	1	3	3	4	3
Ch 2	6	6	5	5	4	5
Ch 4	4	4	2	0	0	1
Ch 7	0	0	0	1	0	0
Ch 8	5	5	3	3	3	3

	New I (mA) Values
Ch 1	8
Ch 2	8
Ch 4	0
Ch 7	0
Ch 8	4

Figure 5.6: Look Up Table (LUT) generation and utilization. In the example shown, only 5 channels are in use and current intensity modulation has been chosen. The yellow and blue cells represent the results of the maximum ATC and maximum stimulation intensity calibration phases, respectively. These values are used for the generation of the LUT. The columns of the latter are indexed by the ATC values. The purple cells represent the median results and are used to locate the new stimulation values in the LUT, which are represented in the green cells.

- stop_training: this method ends the main_stimulation thread and calls the stop_action method of both the FES class and the BLE one. Finally, the ATC queue of this class is emptied.
- disconnect: this method terminates all the connections with the Bluetooth devices and closes all the serial communications opened.

5.7 GUI Class

This class is an *App* class belonging to the Kivy library. It is built using screens and popups, designed by inheriting from Kivy the *Screen* class and the *Popup* class, respectively. Once the application is started, the login screen represented in Figure 5.7 shows up. Here, the user is asked to enter its credentials. By clicking on *Login*, if username and password are correct, the *Screen Manager* goes to the next screen, which is represented in Figure 5.8a.

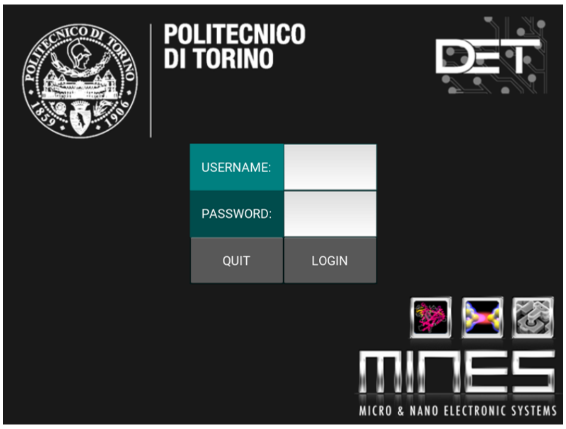
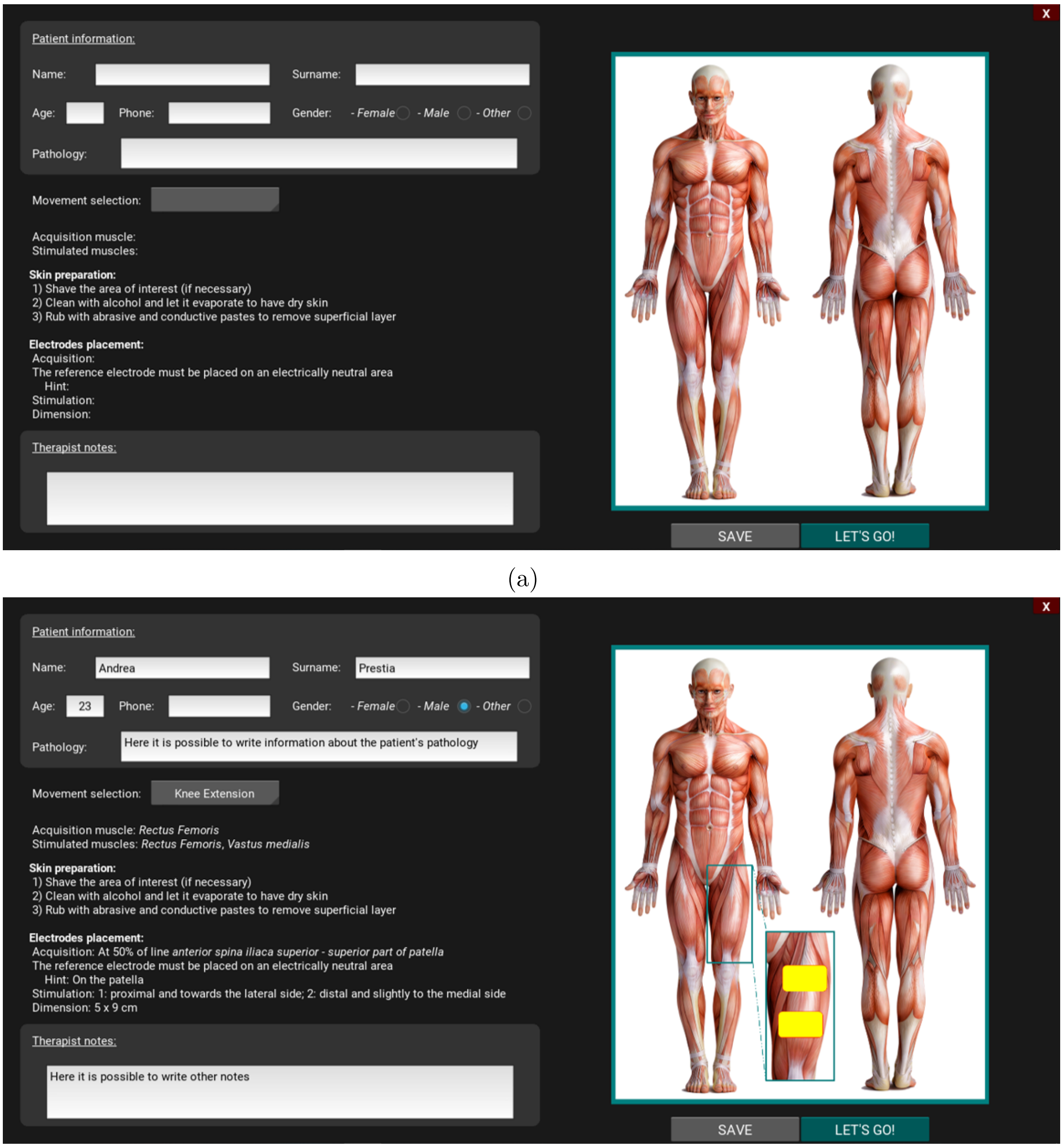


Figure 5.7: Login screen: Quit and Login are *Button* objects, while the white cells are *TextInput*. Both of these objects belong to the Kivy library.

5.7.1 Medical Support Screen

The screen in Figure 5.8 has been designed in order to offer support to the user. Indeed, in addition to allow the therapist to insert and save the patient's information, it offers support for the electrodes application, both for the acquisition of the sEMG signal and for the stimulation ones (see Figure 5.8b). The indications given for sEMG electrodes placement are those of the SENIAM project [91]. As regards the positioning of the stimulation electrodes instead, the indications reported are those provided by [77] and [92]. Finally, by clicking on *Let's Go!*, the screen manager goes to the main stimulation screen (see Figure 5.9a). The latter is the core of the application as it is the only screen class that interacts with the proposed system (i.e. the acquisition channels, the electrogoniometers and the electrical stimulator), working as control panel. The user can return to the medical support screen by clicking *Back*. This last step can be useful if, for example, the user needs last-minute clarifications about electrodes placement, so the control panel settings are not reset and it is possible to resume where it left off.



Patient information:

Name: ____ Surname: ____

Age: ____ Phone: ____ Gender: - Female ○ - Male ○ - Other ○

Pathology: ____

Movement selection: ____

Acquisition muscle:
Stimulated muscles:

Skin preparation:
1) Shave the area of interest (if necessary)
2) Clean with alcohol and let it evaporate to have dry skin
3) Rub with abrasive and conductive pastes to remove superficial layer

Electrodes placement:
Acquisition:
The reference electrode must be placed on an electrically neutral area
Hint:
Stimulation:
Dimension:

Therapist notes:

(a)

Patient information:

Name: Andrea Surname: Prestia

Age: 23 Phone: ____ Gender: - Female ○ - Male ● - Other ○

Pathology: Here it is possible to write information about the patient's pathology

Movement selection: Knee Extension

Acquisition muscle: *Rectus Femoris*
Stimulated muscles: *Rectus Femoris, Vastus medialis*

Skin preparation:
1) Shave the area of interest (if necessary)
2) Clean with alcohol and let it evaporate to have dry skin
3) Rub with abrasive and conductive pastes to remove superficial layer

Electrodes placement:
Acquisition: At 50% of line *anterior spina iliaca superior - superior part of patella*
The reference electrode must be placed on an electrically neutral area
Hint: On the patella
Stimulation: 1: proximal and towards the lateral side; 2: distal and slightly to the medial side
Dimension: 5 x 9 cm

Therapist notes:

Here it is possible to write other notes

(b)

Figure 5.8: Medical support screen: (a) is the basic screen; (b) is the result obtained by filling some fields and selecting a movement from the Kivy *Spinner*. The image on the right shows the positioning of the electrodes used for stimulation.

5.7.2 Main Screen

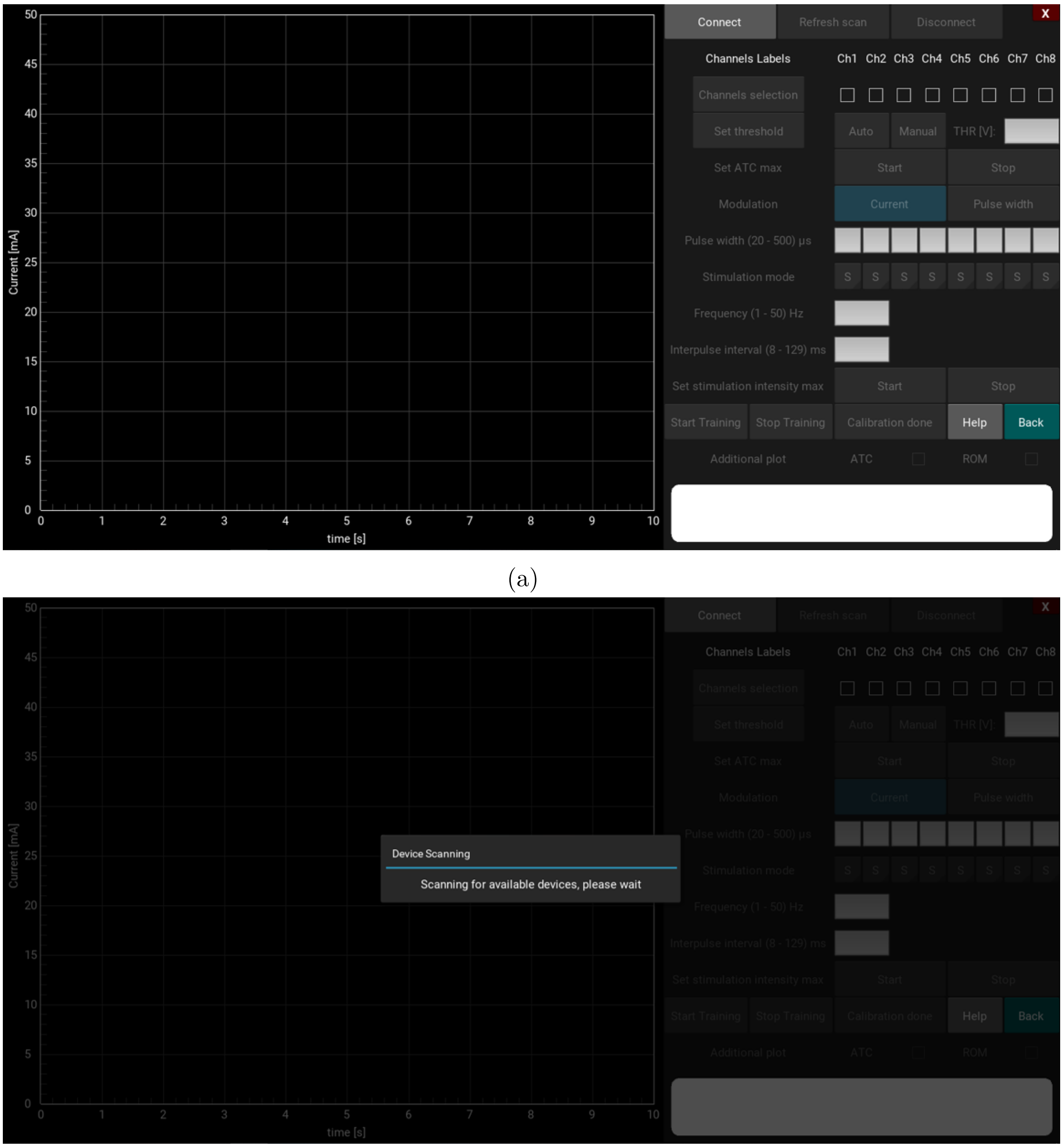
When this screen shows up, it appears as reported in Figure 5.9a. The enabling / disabling of the buttons serves to guide the user towards the operations to be performed and to prevent the carrying out of forbidden operations. Each time a button is pressed a method of this class is called, which will consequently call other methods to satisfy the requested operation. In order to simplify the understanding of this class, in this section it will not be described by listing all its methods but by following the application phases chronologically.

The first time the user presses the *Connect* button during the session, the BLE_and_FES_configuration method of the System class is called and the BLE and FES objects are therefore instantiated. Then, the *Scanning* popup shown in Figure 5.9b appears and a thread is instantiated. Inside this routine, the connect method of the System class is called and then the *Scanning* popup is dismissed. If the connection with the RehaStim2 and with the CC2540 dongle (if used) is not successful (e.g., because they are not connected), a warning message is displayed to the user. Otherwise, the *Connection* popup is opened for selecting the Bluetooth devices with which to establish the connection (see Figure 5.10a). The execution of this series of operations within a thread has served to interface in real-time with Kivy, allowing the opening of popups when required.

Before the *Connection* popup shows up, it is checked which Bluetooth devices were found during the scan and consequently the corresponding checkboxes inside the popup are enabled. Then, using the checkboxes, the user can select which devices to connect to among those found (see Figure 5.10a). By clicking on *Device selection done*, the devices_selection method of the System class is called. Then, if all the selected devices are connected, the spinners of the available channels are enabled in order to decide which ones to assign[1] to the stimulator channels to use. Otherwise, a warning is displayed to the user, to communicate which device has failed in establishing the Bluetooth connection. If the sEMG acquisition system with Apollo3 is among those selected, before enabling the spinners the application checks which channels are available (see Figure 5.10b). Furthermore, in this case the radio buttons are also disabled which allow the user to choose whether to command Apollo3 for the automatic threshold searching for the selected channels.

When the user press the *Channels selection done* (see Figure 5.10b), the channels_selection method of the System class is called and the checkboxes of the main screen relating to the assigned stimulation channels are enabled. Furthermore, if the auto threshold setting has been selected, the *Threshold Setting* popup shown in Figure 5.11a appears and a thread is instantiated that calls the method for the automatic threshold searching for all the selected Apollo3 channels. At the end of it, the popup is dismissed and the condition reported in Figure 5.11b is reached.

[1] If during the selection of the channels it was chosen to assign the second channel of Apollo3 to channel 3 of the stimulator (for example), in order to execute the application methods on that Apollo3 channel it is necessary to select the checkbox of channel 3



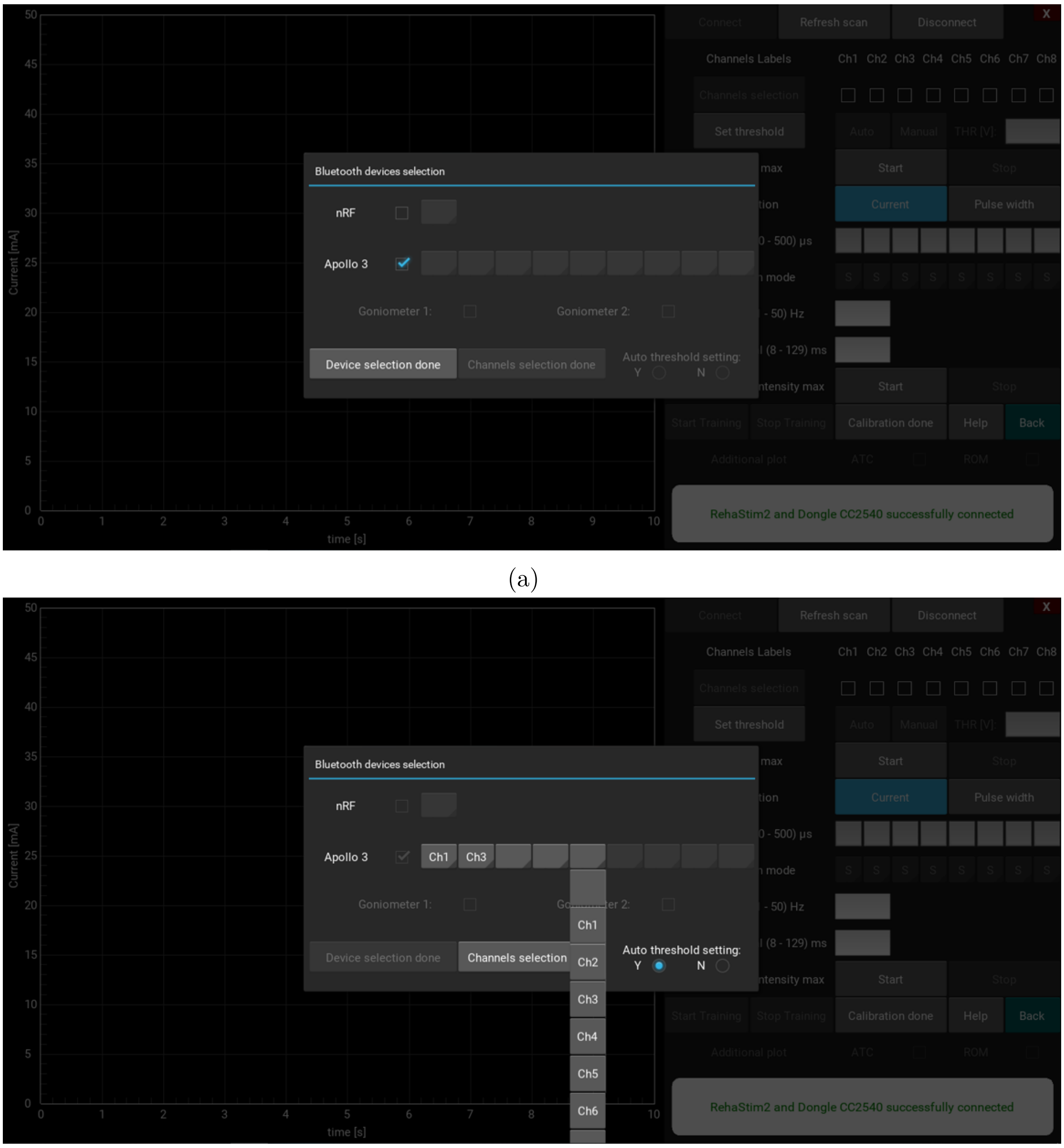
(a)

Device Scanning

Scanning for available devices, please wait

(b)

Figure 5.9: In (a) is reported what the interface looks like as soon as the main screen is reached. Clicking the Connect button, the system also scans for available Bluetooth devices and the scanning popup shown in (b) is then opened.



Bluetooth devices selection: nRF □, Apollo 3 ☑; Goniometer 1: □, Goniometer 2: □

RehaStim2 and Dongle CC2540 successfully connected

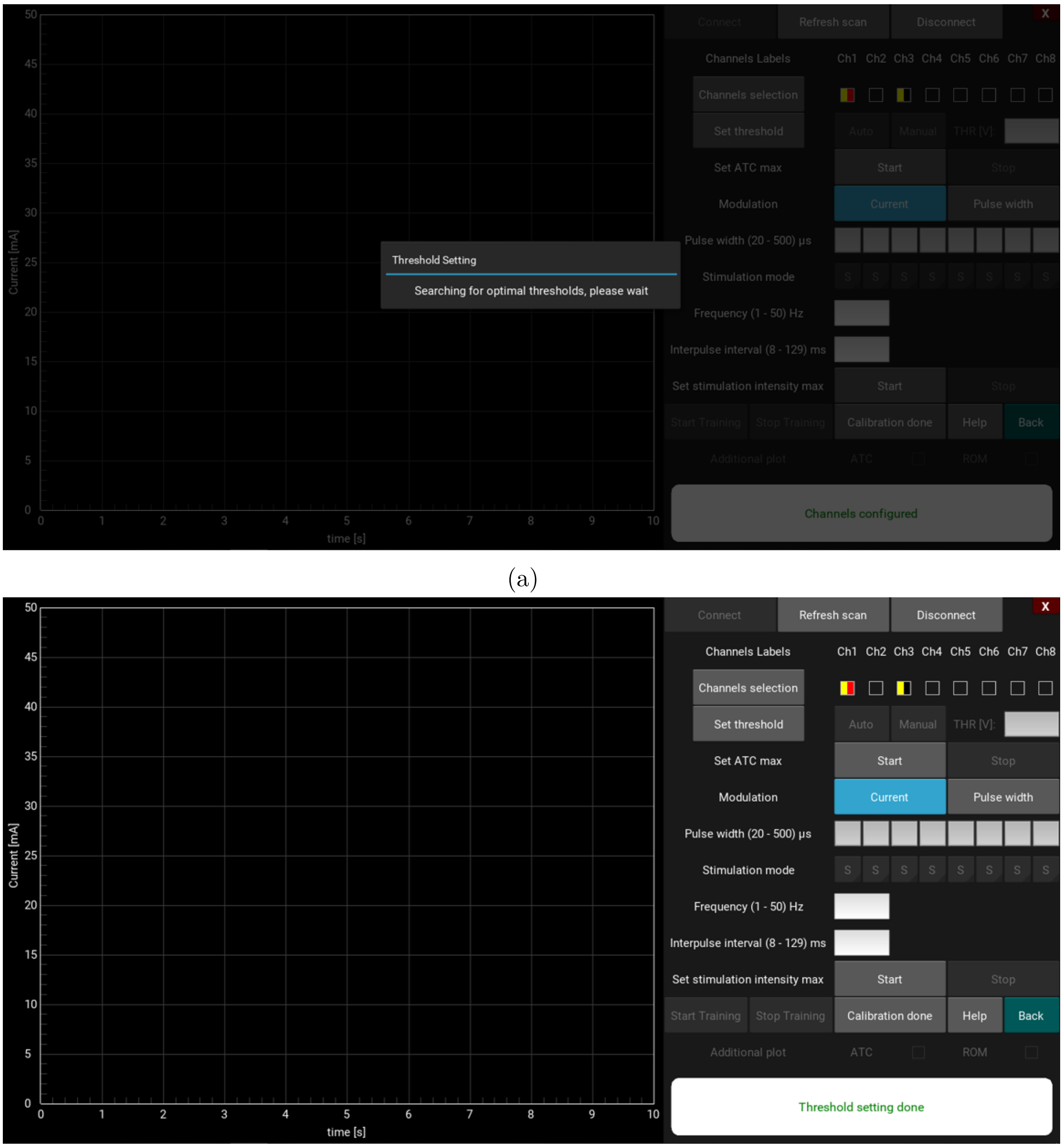
(a)

Apollo 3: Ch1, Ch3; Auto threshold setting: Y ● N ○

RehaStim2 and Dongle CC2540 successfully connected

(b)

Figure 5.10: At the end of the scanning of the available devices, the connection popup opens and the checkboxes of the found devices are enabled. In the example in (a), both acquisition systems and no goniometer were found. The user has decided to connect to Apollo3 and the spinners of the available channels are enabled, as shown in (b). At this point the user has decided that the first and second channels of Apollo3 will be used to control channels 1 and 3 of the electrical stimulator, respectively. Finally it was decided to proceed with the auto threshold setting for selected channels.



Threshold Setting

Searching for optimal thresholds, please wait

Channels configured

(a)

Threshold setting done

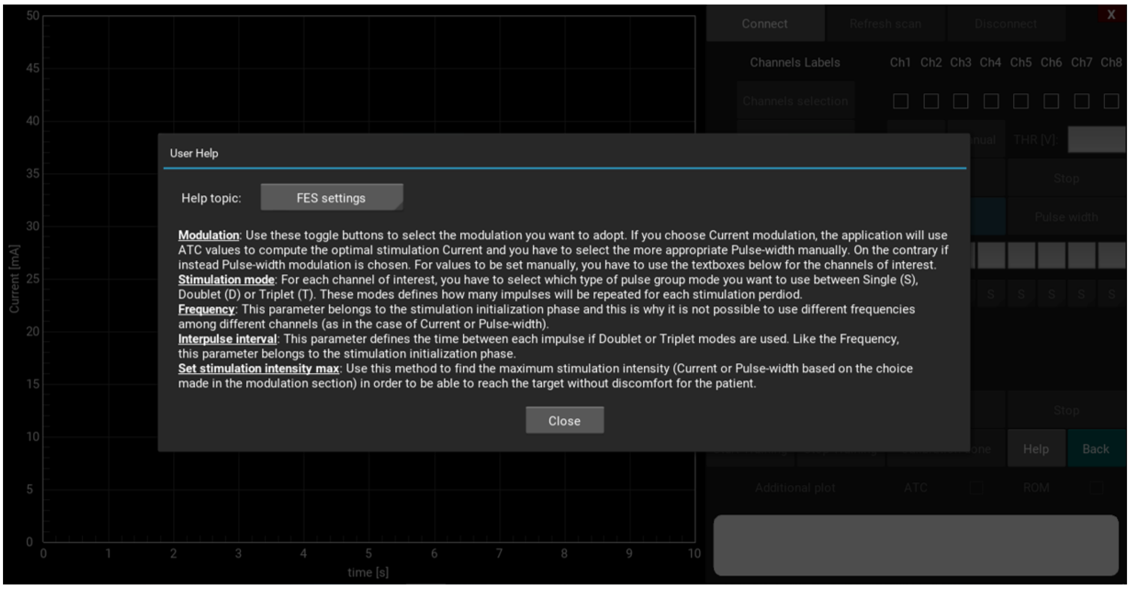
(b)

Figure 5.11: When the user selects the automatic threshold search, the threshold setting popup shown in (a) appears. At the end of the search, the application returns to the control panel, as shown in (b).

When the condition shown in Figure 5.11b is reached, it is possible to repeat the scan of the Bluetooth devices available by clicking on *Refresh scan*, which calls the refresh method of the System class and reopens the popups described above. If instead the user wishes to repeat the channel selection (or modify the stimulator channels assignment) without modifying the devices with which the connection has been established, it is possible to do so using the *Channel selection* button, which will reopen the connection popup in the channel selection condition (see Figure 5.10b).

If the user wishes to set the threshold for ATC, this is possible by pressing the *Set threshold* button, which will enable the *Auto* and *Manual* buttons and disable all the other application buttons (except *Channels selection* and *X*[2]). The selection of the channels to which set the threshold is done with the respective checkboxes in the control panel. By clicking *Auto*, the automatic search for the threshold will be carried out, while clicking *Manual*, the threshold written in the textinput on the right will be set. In both cases the threshold_setting method of the System class is called, supplying as input the setting mode (auto or manual), the selected channels and eventually the desired threshold.

Clicking the *Help* button, the user can get information about the activities that can be performed within the application. The activity of interest can be selected using a spinner, as shown in Figure 5.12.



Modulation: Use these toggle buttons to select the modulation you want to adopt. If you choose Current modulation, the application will use ATC values to compute the optimal stimulation Current and you have to select the more appropriate Pulse-width manually. On the contrary if instead Pulse-width modulation is chosen. For values to be set manually, you have to use the textboxes below for the channels of interest.
Stimulation mode: For each channel of interest, you have to select which type of pulse group mode you want to use between Single (S), Doublet (D) or Triplet (T). These modes defines how many impulses will be repeated for each stimulation perdiod.
Frequency: This parameter belongs to the stimulation initialization phase and this is why it is not possible to use different frequencies among different channels (as in the case of Current or Pulse-width).
Interpulse interval: This parameter defines the time between each impulse if Doublet or Triplet modes are used. Like the Frequency, this parameter belongs to the stimulation initialization phase.
Set stimulation intensity max: Use this method to find the maximum stimulation intensity (Current or Pulse-width based on the choice made in the modulation section) in order to be able to reach the target without discomfort for the patient.

Figure 5.12: User Help popup: using the spinner is possible to select the activity to which clarifications are needed.

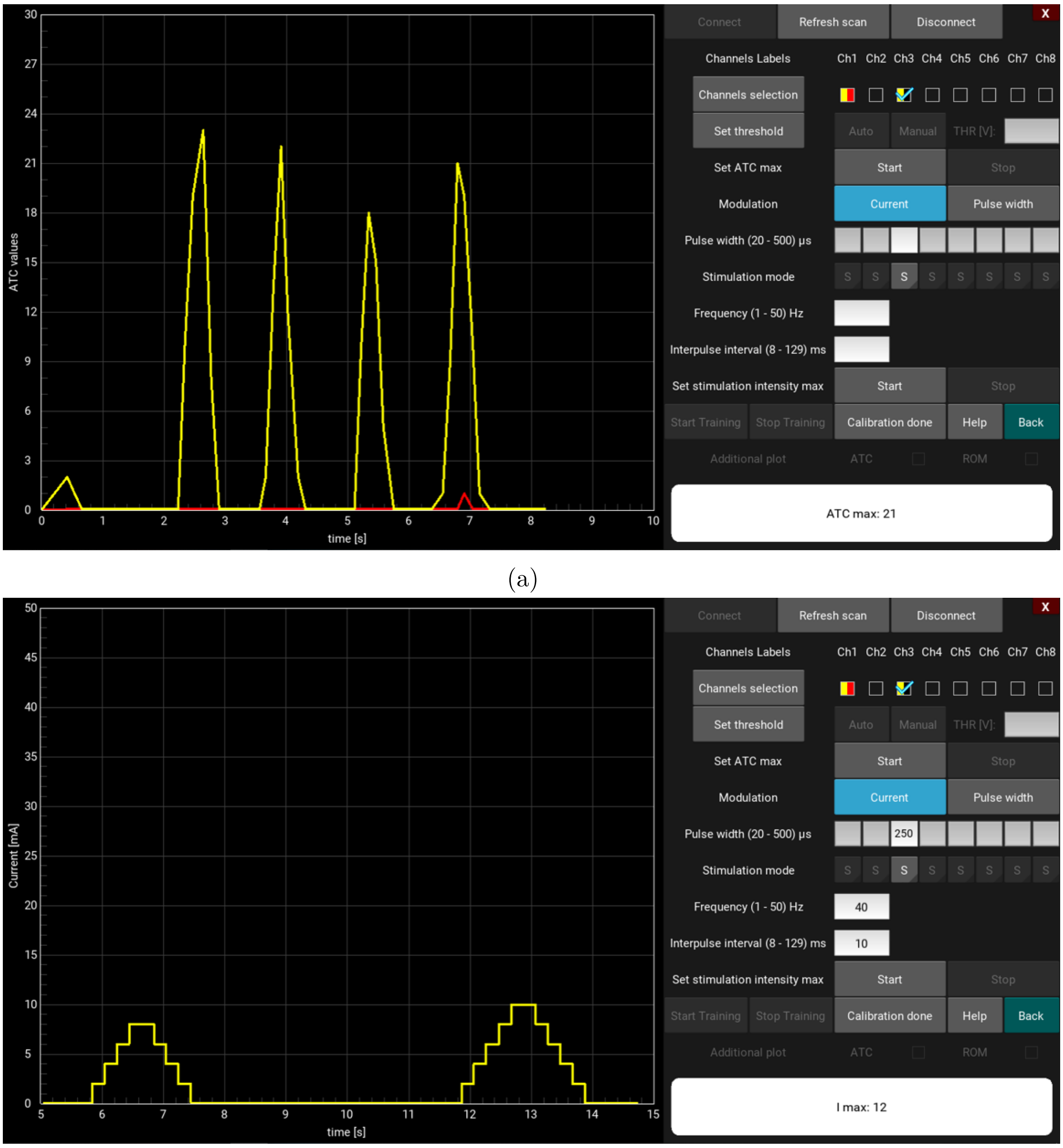
[2] The *X* button is always available for system disconnection and application closing

As regards the calibration phase of the maximum ATC value, by pressing the respective *Start* button the start_ATC_max_setting method of the System class is called and a thread is instantiated to update the plot of the ATC values received. Within this thread, data is obtained from the queue of the System class. Each time the timestamp of the last data reaches the end of the time axis, the latter shifts by 5 seconds. While the calibration is active, all the interface buttons are disabled except *Stop* and *X*. By clicking on *Stop*, the stop_ATC_max_setting method of the System class is called and the thread for plot updating is terminated. The interface buttons are then re-enabled and the result obtained from the calibration is displayed to the user, as reported in Figure 5.13a.

Before starting with the calibration of the maximum stimulation intensity to be modulated, it is necessary to set the other parameters. First of all it is necessary to choose whether to modulate the current or the pulse width, using the toggle button dedicated. The unmodulated parameter between the two is instead set manually by the user, using the appropriate textinputs which are enabled by selecting the respective channel from the checkboxes. The same goes for group mode spinners (Single, Doublet or Triplet). The stimulation frequency and the interpulse interval are the same for all channels, as these two parameters are set with the channel initialization command (see Section 3.2). If the entered parameters do not respect the range allowed (or some parameter is not inserted), a warning is displayed to the user. At this point, when the user presses *Start*, the calibration of the maximum stimulation intensity (current or pulse width, depending on the selection made) begins. The start_IPW_max_setting method of the System class is then called and, with the same logic used for the calibration of the maximum value of ATC, a thread for the plot is instantiated, which ends by pressing *Stop*. The result obtained is displayed to the user, as reported in Figure 5.13b. It is also possible to interrupt the calibration by pressing the emergency button of the RehaStim2, which can be used by the patient in case of discomfort.

By clicking *Calibration Done*, the *Start Training* button is enabled, together with the possibility to select which graphs to show. The presence of the stimulation intensity graph is mandatory, so that the therapist can always monitor the patient's stimulation. When the user presses *Start Training* the main stimulation phase begins, in which the therapist performs a movement from which the ATC signal is acquired, and it is used to modulate the stimulation. The method start_training of the System class is then called and a thread is instantiated for ATC, stimulation intensity, and Range Of Motion (ROM) plots. The latter is available only if at least one of the two electrogoniometers is connected. The stimulation can be terminated by clicking *Stop Training* or if the patient presses the emergency button of the RehaStim2. Figure 5.14 shows two possible examples of what the interface looks like at the end of the main stimulation phase.

Finally, pressing *Disconnect*, the disconnect method of the System class is called and all the value inserted in the control panel are cleared.



(a)

ATC max: 21

(b)

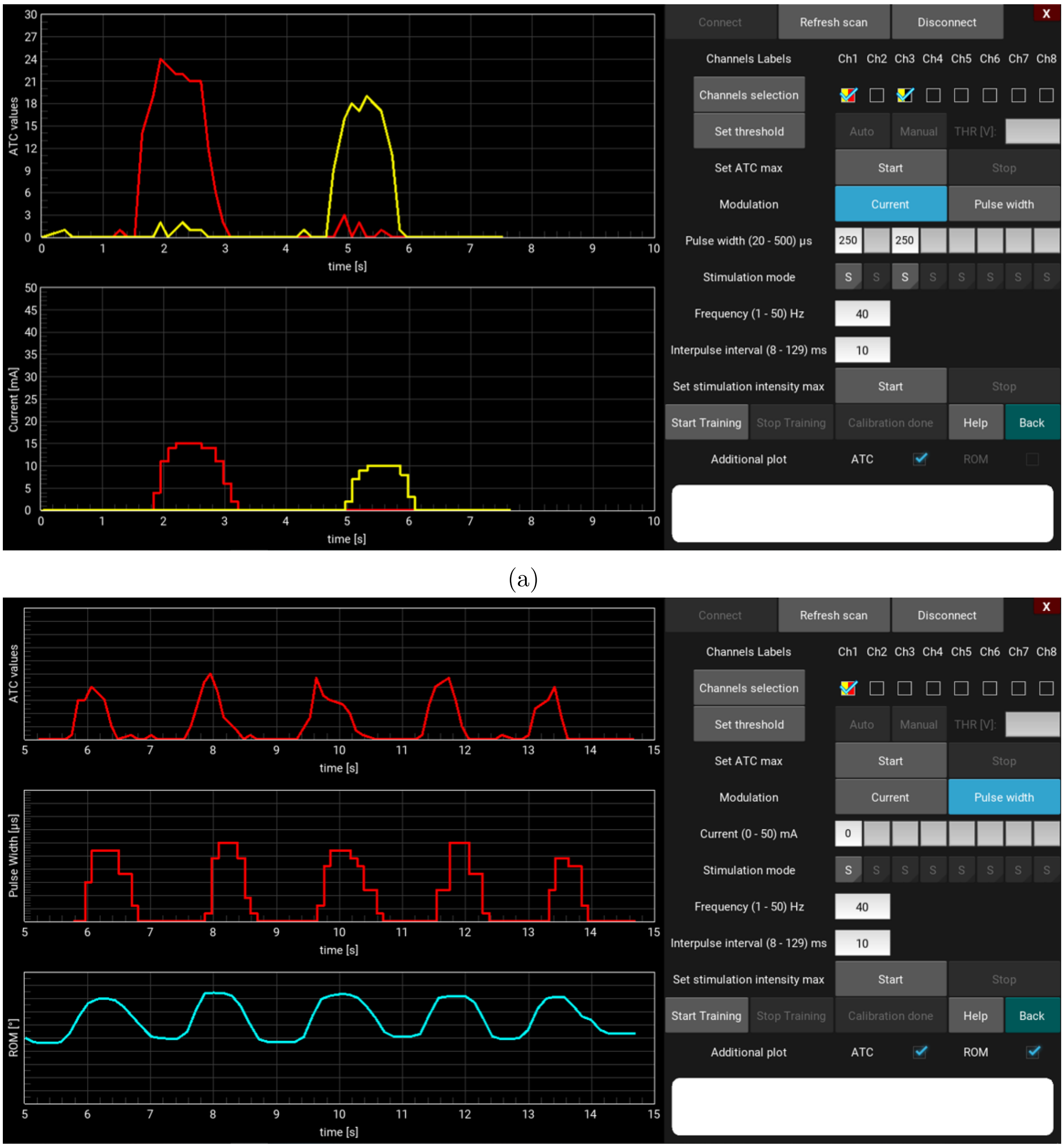
Pulse width (20 - 500) μs: 250

Frequency (1 - 50) Hz: 40

Interpulse interval (8 - 129) ms: 10

I max: 12

Figure 5.13: (a) shows the result of the calibration phase of the maximum ATC value. Although the calibration operates on one channel at a time (i.e. the selected one), the ATC data of all the channels present are still plotted, in order to monitor the crosstalk between the muscles involved. The result of the calibration is the median of the maximum values obtained by each movement (see the ATC_max_setting method in Section 5.6). (b) shows the result of the calibration phase of the maximum stimulation intensity (see the IPW_max_setting method in Section 5.6). Using the toggle button for modulation selection, the axis of the stimulation graph and the label corresponding to the parameter to be set manually are accordingly modified. The axes are also modified by pressing the Start of the two calibration phases, in order to adapt to the activity in progress.



Modulation: Current

Pulse width (20 - 500) μs: 250, 250

Frequency (1 - 50) Hz: 40

Interpulse interval (8 - 129) ms: 10

(a)

Modulation: Pulse width

Current (0 - 50) mA: 0

Frequency (1 - 50) Hz: 40

Interpulse interval (8 - 129) ms: 10

(b)

Figure 5.14: (a) shows an example of the graphs obtained during a wrist extension-flexion session. The red channel is active during wrist extension while the yellow channel is active during flexion. The slight presence of crosstalk should be noted, although it does not affect stimulation due to the robustness obtained from the median between consecutive ATC values. (b) shows an example involving the use of an electrogoniometer. Furthermore in this last Figure it is possible to see how the interface looks like if the pulse width modulation is selected.

Chapter 6

Software Latencies

In a rehabilitation system, minimal software response times are of fundamental relevance, in order to guarantee real-time interaction between therapist and patient during training. Therefore, a study on software latencies has been carried out, in which all the possible channels combinations have been tested. These measurements involved two different control units:

- An Asus PC with Intel® Core™ i7-5500 CPU, 2.4 GHz clock frequency, 8 GB RAM and Microsoft® Windows® 10 OS;
- A Raspberry Pi model 4B [93] with Cortex-A72 (ARM v8) 64 bit SoC, 1.5 GHz clock frequency, 4 GB RAM and Raspbian OS.

In the first case, the USB dongle CC2540 has been used as Bluetooth antenna, while in the second case, the tests were performed using both the CC2540 and the integrated Bluetooth antenna. The latencies are measured starting from the receiving of the ATC packet up to the completion of sending commands to the electrical stimulator. The total duration of each test was 300 seconds, thus obtaining more than 2300 samples for each Bluetooth device connected considering that the sending of notifications has a period of 130 ms. While running the tests, the application works under normal conditions, so with the GUI active while plotting the data exchanged with the rest of the system.

Table 6.1 shows the ID assigned to each combination of channels tested. Figures 6.1, 6.2, 6.3 report the obtained results using Microsoft® Windows® PC with dongle CC2540, Raspberry Pi with dongle CC540 and Raspberry Pi with bluepy library, respectively. Finally, Figure 6.4 shows a comparison between the median values of the measured data among all the possible configuration tested.

The median of the obtained latencies is always less than 10 ms, while the maximum latency assessed was less than 70 ms (see Figure 6.3). However, the latter outlier does not exceed the constraint imposed by this system, since being the notification period of 130 ms, this is also the value that must not be exceeded in order to finish the definition of the stimulation pattern always before receiving the new notification. The system, therefore, proves to be valid for real-time usage.

Table 6.1: ID assigned to each combination of channels tested.

Channels combination ID	**# Channels**	
	Ap3	**nRF**
C1	1	0
C2	2	0
C3	3	0
C4	4	0
C5	5	0
C6	0	1
C7	1	1
C8	2	1
C9	3	1
C10	4	1
C11	5	1

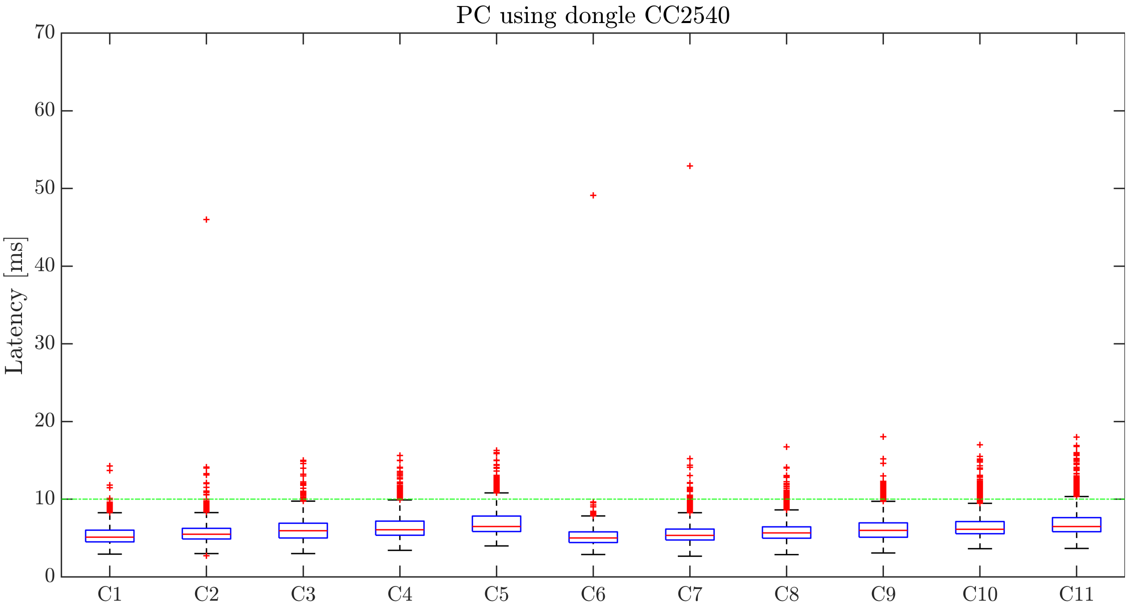


Figure 6.1: Latencies obtained using Windows® PC with dongle CC2540. The obtained median values from each combination are always less than 10 ms. The minimum median value is equal to 5.007 ms and has been obtained for C6 combination, while the maximum is equal to 6.483 ms and has been obtained for C5 combination. Instead, C11 combination (i.e., the one with all 6 channels) has a median latency equal to 6.478 ms, therefore not deviating particularly from C5.

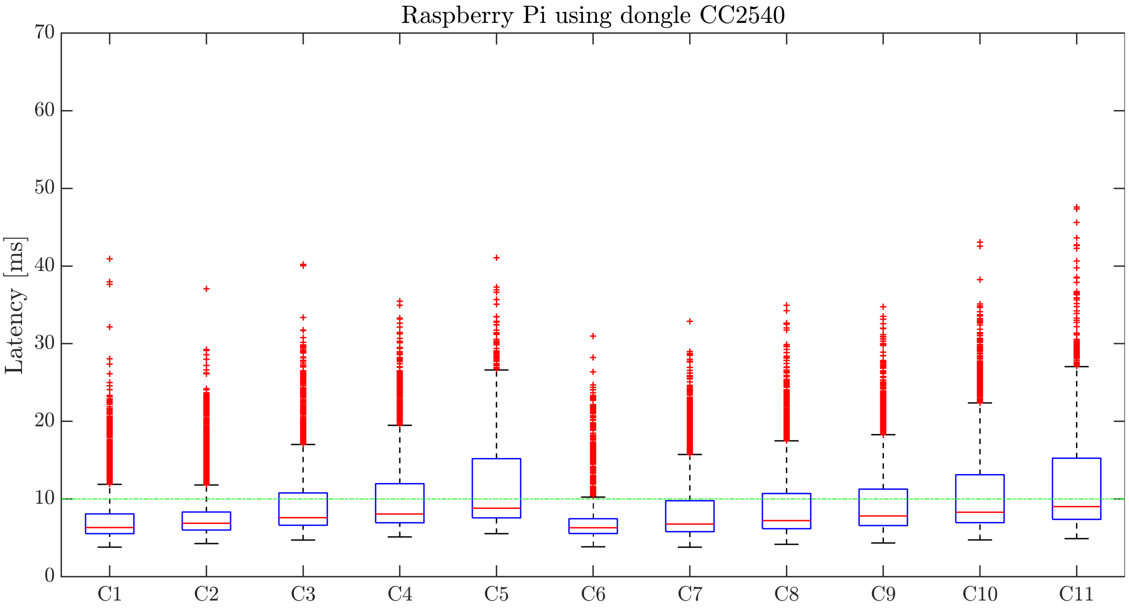


Figure 6.2: Latencies obtained using Raspberry Pi with dongle CC540. The obtained median values from each combination are always less than 10 ms. The minimum median value is equal to 6.289 ms and has been obtained for C6 combination, while the maximum is equal to 9.012 ms and has been obtained for C11 combination.

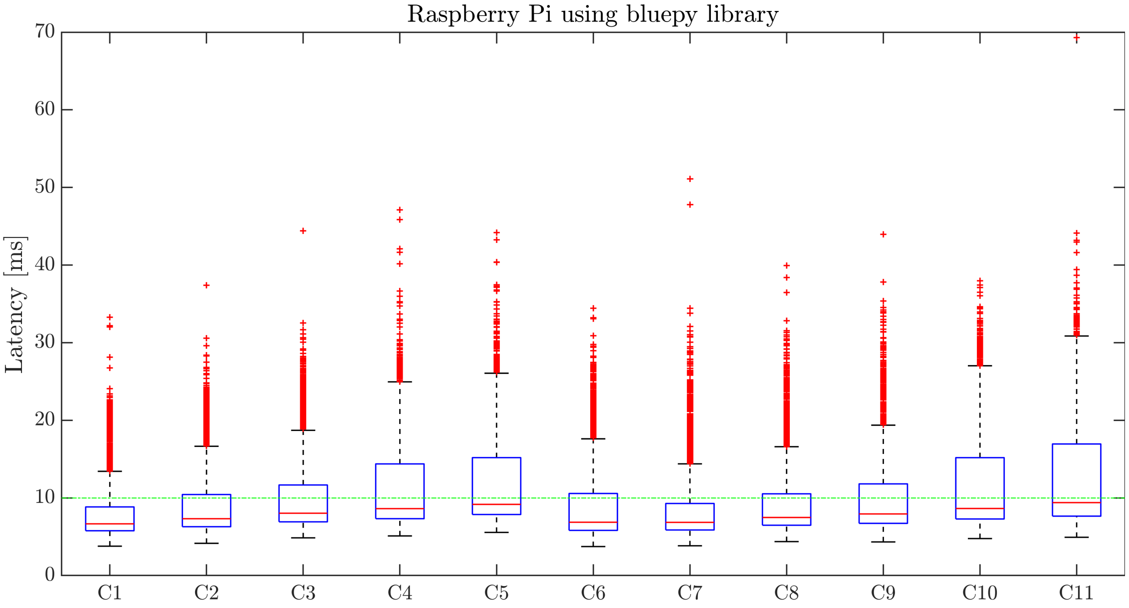


Figure 6.3: Latencies obtained using Raspberry Pi with bluepy library. The minimum median value is equal to 6.659 ms and has been obtained for C1 combination, while the maximum is equal to 9.394 ms and has been obtained for C11 combination. Here the maximum outlier is obtained, with a value equal to 69.318 ms, which is in any case less than 130 ms, thus not involving a constraint for the correct working of the system.

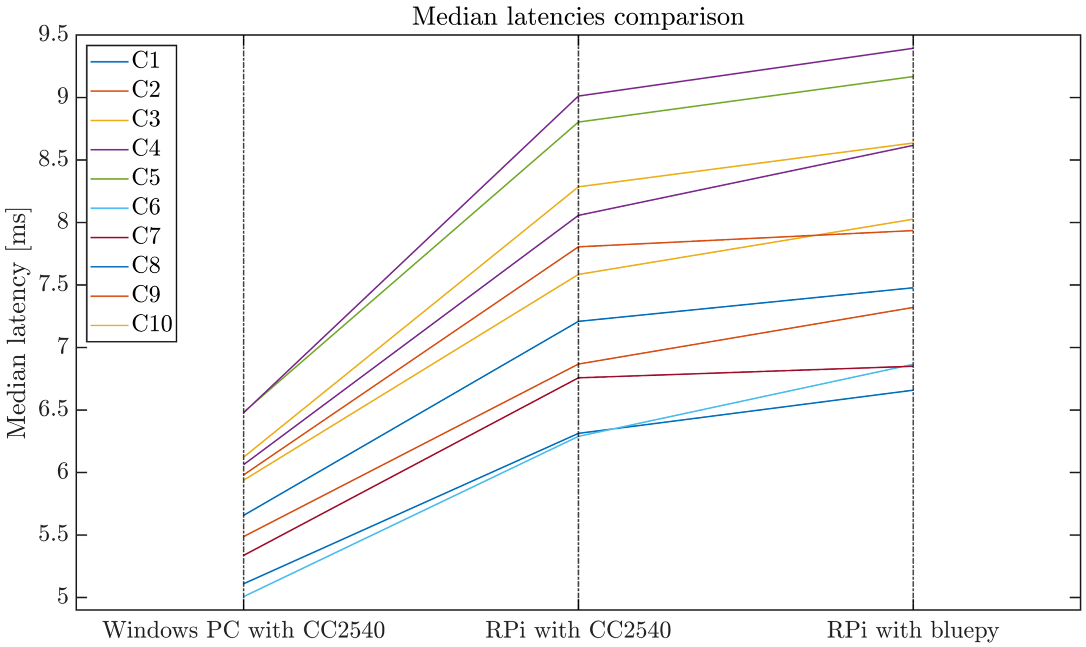


Figure 6.4: Latencies comparison among all the possible configuration tested. An increasing trend is evident among the three different hardware alternatives used. It is interesting to note that, as regards Raspberry Pi control unit, using the CC2540 dongle as an external Bluetooth antenna is more efficient than using a Python library that exploit the device's integrated one.

Chapter 7

Experimental Protocol Description

The experimental tests necessary to this thesis project were approved by the Bioethics Committee of the University of Turin [94]. In order to comply with COVID-19 regulations, all tests were conducted using face masks and hand sanitizer gels. The tests involved 6 subjects, 4 males and 2 females, aged between 24 and 30 years. As regards the validation of the ATC-FES system, the modality used is the therapist controlled one. Hence, the ATC signal is acquired from the therapist and is used to define a suitable FES pattern to be applied to the patient. The movements involved in this study are Elbow Flexion, Wrist Extension, Wrist Flexion, Knee Extension, Ankle Extension and Ankle Flexion.

7.1 Vicon Motion Capture System Introduction

In order to validate the system involved in this thesis project, experimental tests were carried out using the Vicon Motion Capture System [95]. Thanks to this powerful tool, it is possible to acquire the trajectories of a moving body in 3D space. The location where these tests were performed is the interdepartmental PolitoBIOMed Lab [96]. In this laboratory, 12 Vicon cameras [97] are provided. In order to acquire the 3D trajectory of a subject movement with the Vicon cameras, it is necessary to apply special reflective markers on the subject skin. The acquired trajectories can be then extracted using the Nexus software [98] and used to compare the movements involved in the validation tests.

7.2 Validation Tests Protocol Steps

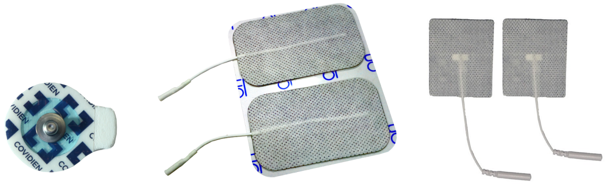
7.2.1 Vicon System Calibration

The Vicon system calibration was carried out using the Nexus software and following the Vicon Nexus User Guide [99] indications. After organizing the room setup by placing all the useful instruments in their intended location, the cameras are masked, in order to be able to ignore the reflections created by the instrumentation. Then, using the Vicon Active Wand [100], the cameras are calibrated so that they can reconstruct the 3D sampling volume.

Finally, the volume origin is set, using 3 reflective markers to indicate to the system which are the desired x and y axis.

7.2.2 Subjects Preparation

First, the reflective markers are applied using biocompatible adhesive tape, following the Plug-in Gait Reference Guide [101] indications. Then, acquisition or stimulation electrodes are applied, according to the role of each subject. As regards the application of the sEMG acquisition electrodes, in order to ensure good contact between them and the subject's skin, the indications of the SENIAM project [91] have been respected. Therefore, the subject's skin is first shaved (if necessary) and then cleared with alcohol. The subject in the therapist role is then asked to assume the reference posture and, through palpation, the location for a correct application of the electrodes is identified. The electrodes used for sEMG acquisition are the Kendall™ H124SG model (see Figure 7.1a), produced by Covidien [102]. The sensor is pre-gelled, Ag/AgCl coated and with a 24 mm diameter. The subject in the patient role undergoes the same type of skin treatment for the application of the stimulation electrodes. For their positioning the indications given in [77] and [92] were followed. The electrodes used are of two types, depending on the size of the muscle of interest (see Table 7.1): the larger ones are the RehaTrode electrodes, produced by HASOMED®, with a 5x9 cm size (see Figure 7.1b); the smaller ones are the PG470W model, produced by Fiab [103], with a 3.5x4.5 cm size (see Figure 7.1c). Since these electrodes are re-usable, conductive gel is applied to them before placement in order to ensure efficient stimulation.



(a) (b) (c)

Figure 7.1: Electrodes used for sEMG acquisition (a) and for electrical stimulation (b, c).

Table 7.1: Muscle of interest of each movement tested and respective size of the stimulation electrodes used.

Movement	**Muscle**	**Electrode size**
Elbow Flexion	*Biceps Brachii*	3.5x4.5 cm
Wrist Extension	*Extensor Carpi Ulnaris*	3.5x4.5 cm
Wrist Flexion	*Flexor Carpi Radialis*	3.5x4.5 cm
Knee Extension	*Rectus Femoris*	5x9 cm
Ankle Extension	*Gastrocnemius*	5x9 cm
Ankle Flexion	*Tibialis Anterior*	3.5x4.5 cm

7.2.3 ATC-FES System Calibration

Once the preparation of the subjects is completed, it is possible to proceed with the system calibration phase. First of all, the therapist is invited to maintain the rest condition, to be able to set the threshold needed for ATC evaluation. At the end of this step, in order to calibrate the maximum ATC value that can be expressed by the subject, the latter performs at least three repetitions of the movement of interest. The next step is the calibration of the maximum stimulation intensity. First, the user needs to set which is the modulated parameter: the alternatives are the current intensity and the pulse width. For these tests, only current intensity modulation has been used. Then, it is necessary to manually set the non-modulated parameters (i.e., the pulse width and the stimulation frequency). The values of these parameters are established following the indications provided in [104]. Finally, the calibration phase of the maximum stimulation intensity begins. The patient is stimulated with increasing intensity until reaching the motor threshold and performing the movement of interest. If during this phase the subject feels any discomfort, burning or pain, the stimulation can be interrupted either by the user or by pressing the emergency button of the RehaStim2. The stimulation parameters can then be modified to best adapt to the intrinsic variability of each subject. During this last calibration phase, the Range Of Motion (ROM) of the subject is acquired, in order to be able to normalize the data during the results analysis.

7.2.4 Test Execution

After completing the calibration phase, it is possible to proceed with the test. Each test consists of 3 sessions. Each session consists of 10 repetitions of the same movement, separated from each other by 10 seconds of rest, in order to minimize muscle fatigue. For the same reason, each session is separated from the next one by a resting phase of 5 minutes.

As regards the electrogoniometers validation, at least 25 repetitions were performed for each of the following movements: Elbow Flexion, Ankle Extension-Flexion, Knee Extension. During this phase the FES is not applied, therefore the presence of resting periods is not necessary.

Chapter 8

Experimental Results and Discussion

8.1 Trajectories Segmentation

The analysis of the trajectories acquired with the Vicon system and with the electrogoniometers was carried out in MATLAB® by applying a segmentation algorithm on the signal. This algorithm identifies four points within each movement: the beginning and the end of the rising phase, the beginning and the end of the falling phase. By obtaining these points it is possible to compute several parameters such as the Range of Motion (ROM), the execution times, the rising and falling velocities and the area under the curve (which can also be expressed separating the phases of rising, plateau and falling). Due to the trajectories variability among the movements carried out during the tests, the algorithm contains up to seven parameters that can be modified by the user:

- var_min: minimum degree of variation to detect rising and falling points;
- $Npoints2check\_start$: number of subsequent points to check in order to choose if the current point is the start of the rise or the fall;
- $NpointsMin\_agree$: minimum number of points in favour of the start rise/fall choice;
- $NpointsMax\_disagree$: maximum number of points against the start rise/fall choice;
- $Npoints2check\_end$: number of subsequent point so check to establish the end of the rise or the fall;
- min_dist: minimum number of points between two subsequent movements;
- $min\_excursion$: minimum movement excursion.

These parameters have a clear dependence on the signal sampling frequency, as increasing the latter increases the density of the points and decreases the variation between adjacent ones. In Figure 8.1, a segmented trajectory example is reported.

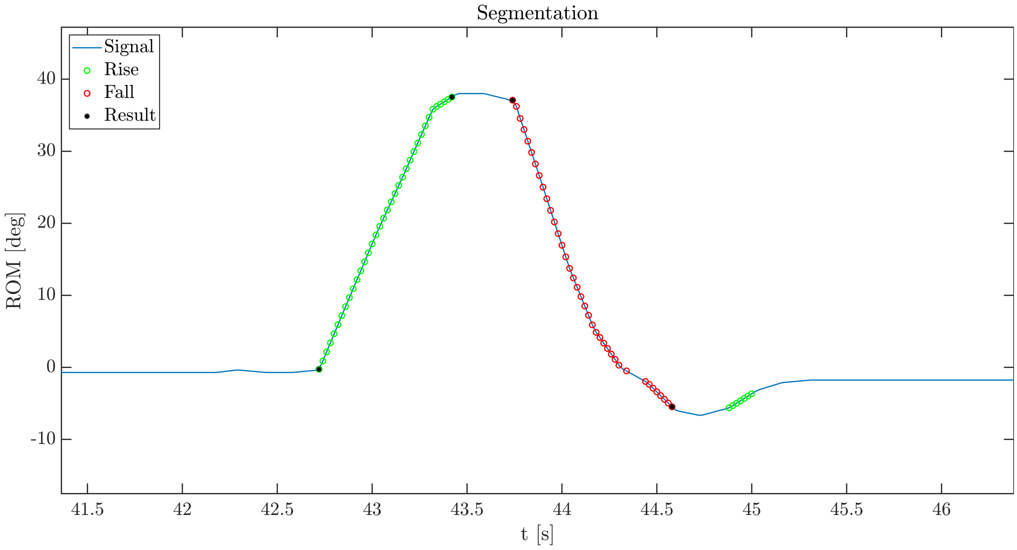


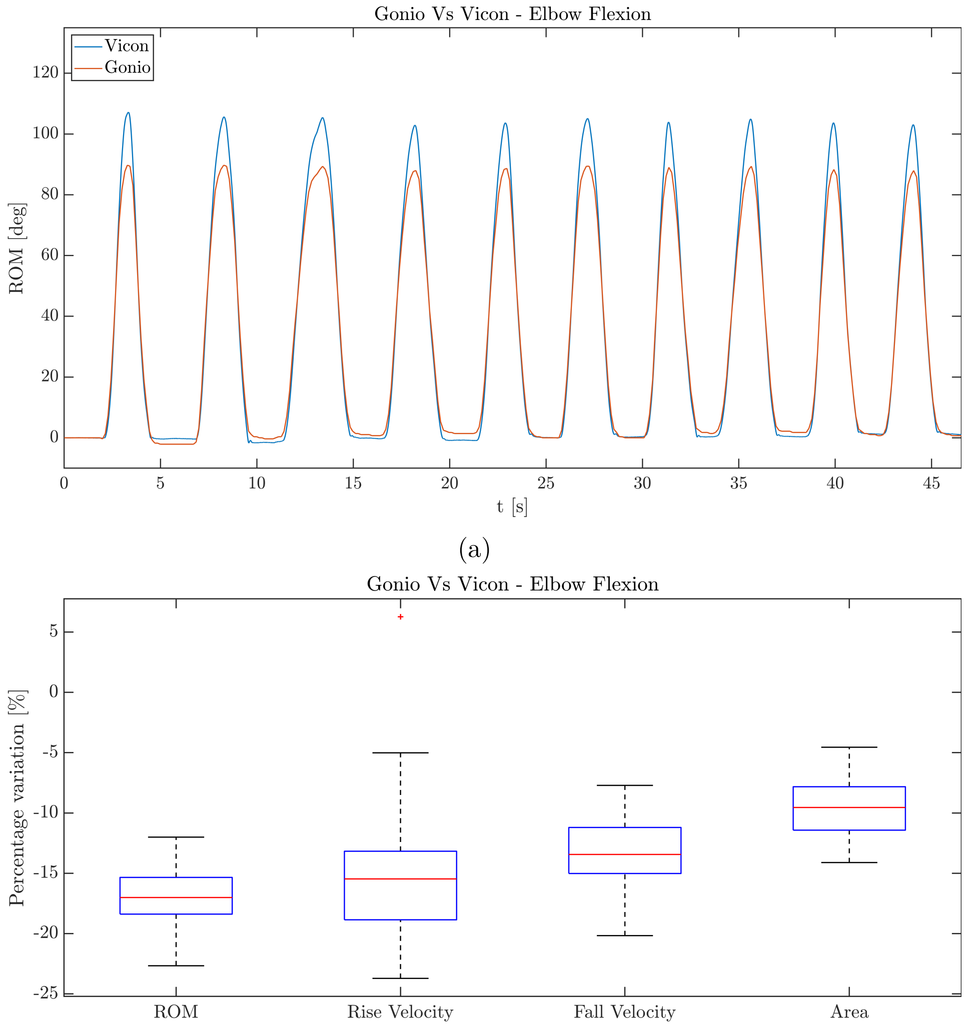
Figure 8.1: Trajectory segmentation: the green points are those identified as rising points, while the red points are the falling ones. The purpose of the segmentation is to identify the four points indicated in black (reported as results in the legend), in order to be able to extract the useful parameters from the trajectory. When, on the other hand, the rising and falling points do not satisfy the established acceptability criteria, they are not considered. For example, the rising points at the end of the trajectory are not considered part of a movement for the following reasons: they are to few, they do not reach the minimum excursion; too little time has passed since the end of the last movement.

8.2 Goniometer Validation

The developed prototype of electrogoniometer has been validated by comparing the trajectories acquired with the Vicon system. The compared parameters are the ROM, the rise and fall velocities and the area under the trajectory of each movement. In order to extract these parameters, each repetition has been segmented as in the example shown in Figure 8.1. The comparison was carried out by computing the difference between the parameters obtained with the two acquisition systems and normalizing to the reference one (i.e., the value obtained with the Vicon system). This variation has been expressed as a percentage, as indicated in Equation 8.1:

$$PercentageVariation = \frac{X_{gonio} - X_{vicon}}{X_{vicon}} \times 100 \tag{8.1}$$

The obtained results are shown in Figures 8.2, 8.3, 8.4 and summarized in Table 8.1, respectively divided between the movements performed: Elbow Flexion, Ankle Extension-Flexion, Knee Extension. The best results were obtained for Ankle Extension-Flexion, whose percentage variations in the parameters studied were the lowest. It should be noted that as the joint ROM increases, the results worsen considerably. The validation of the designed prototype, therefore, did not get a positive outcome, as improvements will certainly be necessary to allow this tool to be used for all the joints.



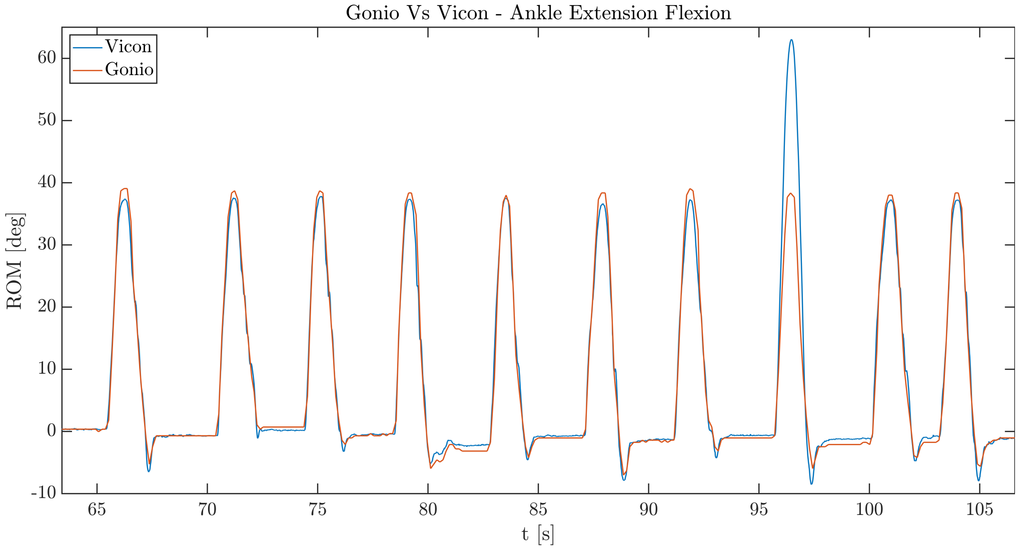
(a)

(b)

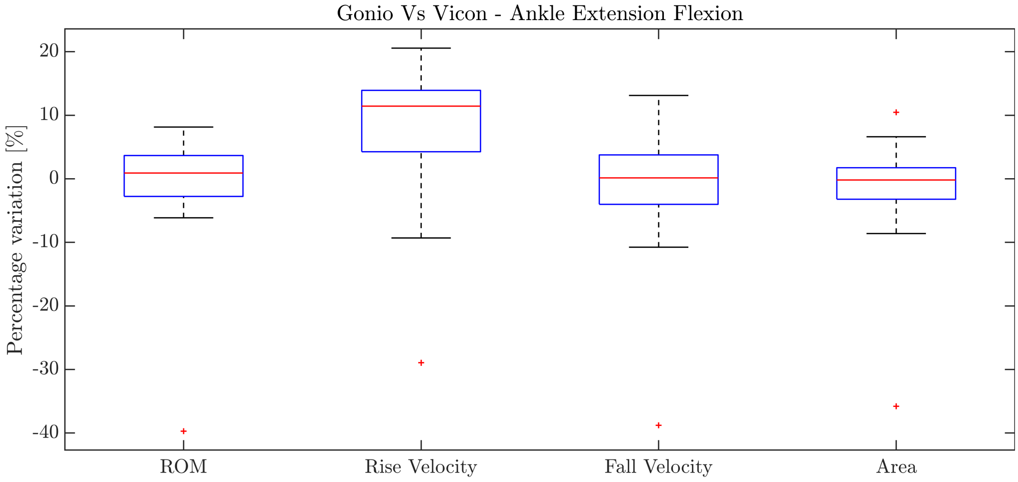
Figure 8.2: Goniometer and Vicon comparison during Elbow Flexion. (a): Example with 10 repetitions; (b): Boxplot with the entire session. As observable in (a), the trajectory acquired with the electrogoniometer does not reach the same amplitude as the one obtained with the Vicon system, resulting in a ROM variation of -17%.

Table 8.1: Goniometer validation results. Q_1 and Q_3 are the 25th and 75th percentile, respectively.

Gonio Vs Vicon	**Elbow F**			**Ankle EF**			**Knee E**		
Variation %	**Q_1**	**median**	**Q_3**	**Q_1**	**median**	**Q_3**	**Q_1**	**median**	**Q_3**
ROM	-18.39	-17.01	-15.34	-2.77	0.91	3.66	-22.22	-18.52	-14.67
Rise Velocity	-18.86	-15.47	-13.17	4.26	11.43	13.92	-40.52	-32.04	-25.33
Fall Velocity	-15.02	-13.44	-11.20	-4.02	0.14	3.76	-14.42	-3.65	5.52
Area	-11.43	-9.55	-7.83	-3.21	-0.19	1.74	-27.52	-23.99	-19.01

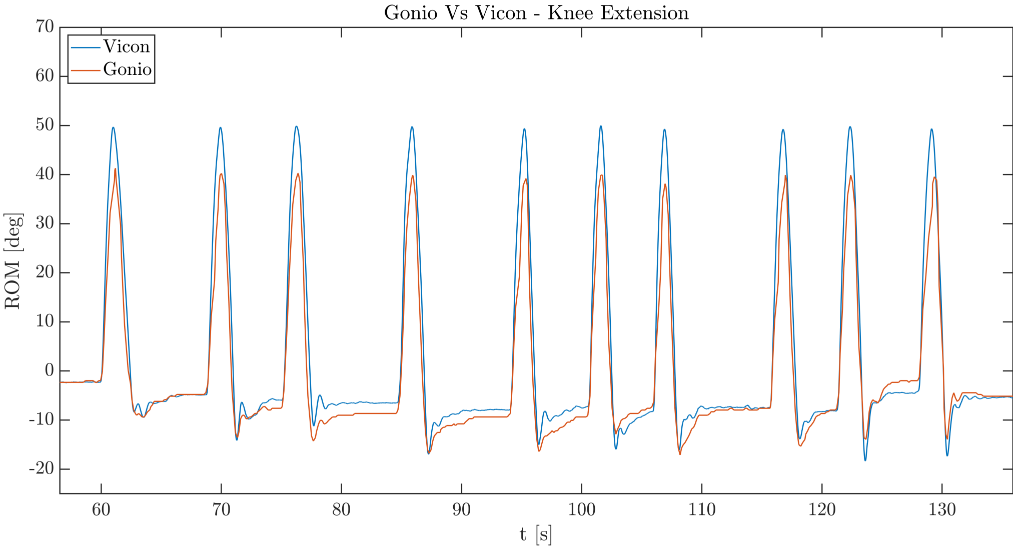


(a)

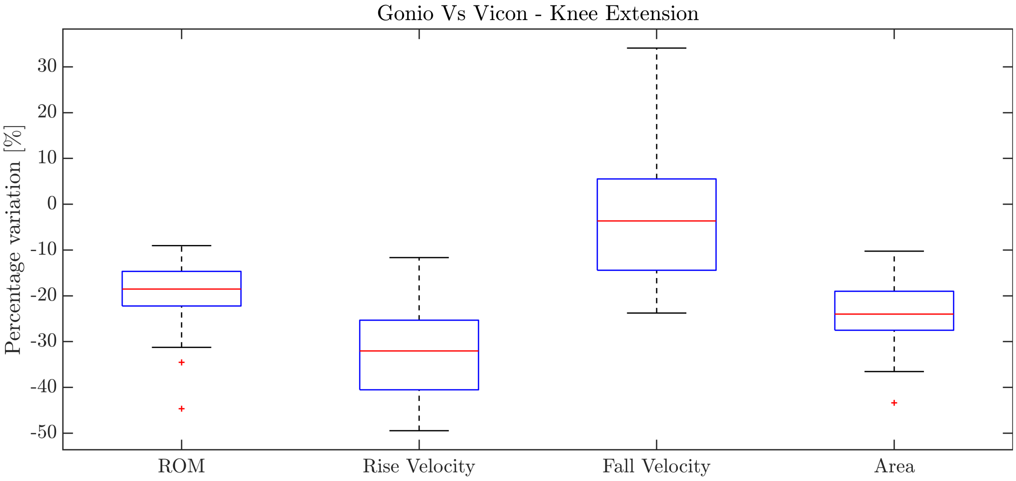


(b)

Figure 8.3: Goniometer and Vicon comparison during Ankle Extension Flexion. (a): Example with 10 repetitions; (b): Boxplot with the entire session. The obtained results with this movement are the best ones, with median percentages of variation close to zero for ROM, fall velocity and area. Instead, for the rise velocity a median of 11.43% was obtained. From the graph in (a) it is interesting to note that in the 10 excursions represented, the goniometer has always generated valid trajectories compared with Vicon, except for the seventh, in which the ROM has reached quite high values. This result suggests that the designed structure has a limit beyond which it is no longer able to follow the movement correctly.



(a)



(b)

Figure 8.4: Goniometer and Vicon comparison during Knee Extension. (a): Example with 10 repetitions; (b): Boxplot with all the repetitions made. The results obtained with this movement are the worst, with a rise velocity variation of -32.04%. Looking at the variation of the baselines recorded by the two acquisition systems in (a), it is probable that the goniometer structure needs improvements as regards the positioning stability on the joint.

8.3 Therapist Controlled FES Tests

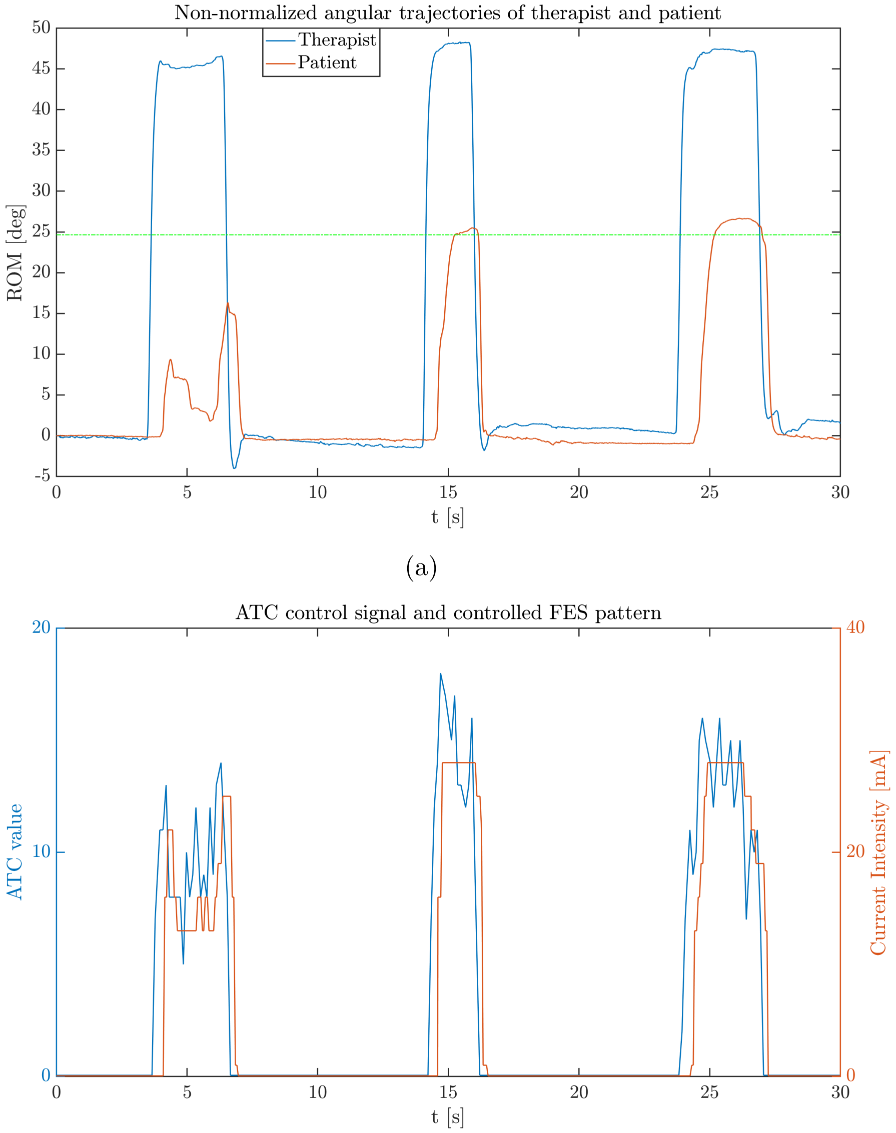
The trajectories of the two subjects in the respective roles of therapist and patient were all acquired using the Vicon system, in order to exploit a reliable tool to obtain the angular data. The latter have been processed in MATLAB® environment, segmenting the angular trajectories as in the example shown in Figure 8.1. The patient data are normalized to the maximum ROM obtained in the calibration phase (therefore referring to stimulated movement), while the therapist data are normalized to the maximum ROM obtained during the testing phase. All the parameters described has been extracted for each movement repetition. The first parameter of interest is the maximum of the normalized cross correlation coefficient (ρ) between the movement performed by the therapist and the one performed by the subject. This coefficient is computed as reported in Equation 8.2:

$$\rho = \hat{R}_{x,y,coeff}(m) = \frac{1}{\sqrt{\hat{R}_{x,x}(0)\hat{R}_{y,y}(0)}}\hat{R}_{x,y}(m) \tag{8.2}$$

The second parameter extracted is the delay between the therapist movement and the patient one. This is achieved by subtracting the timestamps of the points where the movements begin. Finally, three other parameters were extracted such as the ROM, the area under the curve and the rise velocity of each angular trajectory. These three parameters were analyzed by comparing them between patient and therapist and expressing their percentage variation, as reported in Equation 8.3:

$$X_{pt-th} = \frac{X_{pt} - X_{th}}{X_{th}} \times 100 \tag{8.3}$$

The reason why the fall velocity has not been considered is because the final stage of the movement (i.e., when the limb returns to its resting position) is not controlled by the FES, since the ATC value during the eccentric phase of the movement is very low (or zero).
In Figure 8.5a is reported an example which highlights how the use of an inappropriate FES pattern affects the resulting movement performed. Figures 8.6, 8.7, 8.8, 8.9 and 8.10 show the results obtained for each of the six movements under consideration. Finally, in Tables 8.2 and 8.3 an overview of the all obtained results is reported for upper limbs and lower limbs, respectively. The movements whose performance were the best are Elbow Flexion and Ankle Flexion, for which the highest median ρ has been obtained, equal to 0.96 and 0.94, respectively. As regards execution delays, the obtained median values are 0.65 and 0.6 seconds. The worst results were obtained with Wrist Flexion, with median ρ equal to 0.76 and median delay equal to 1.28 seconds. The similarity analysis carried out for the other tested movements (i.e., Wrist Extension, Knee Extension, Ankle Extension), reports the median ρ equal to at least 0.85, which was chosen as the acceptability threshold value, based on previous literature works.



(a)

(b)

Figure 8.5: In (a) is reported an example of obtained angular trajectories in which, in the first repetition shown, the patient was not able to perform a complete movement. Looking at (b), it is indeed possible to notice how the ATC profile was not high enough to achieve an appropriate FES pattern. The next two repetitions instead allowed the patient to reach the target ROM. Indeed, the green line represents the maximum ROM obtained by the patient during the stimulation intensity calibration phase.

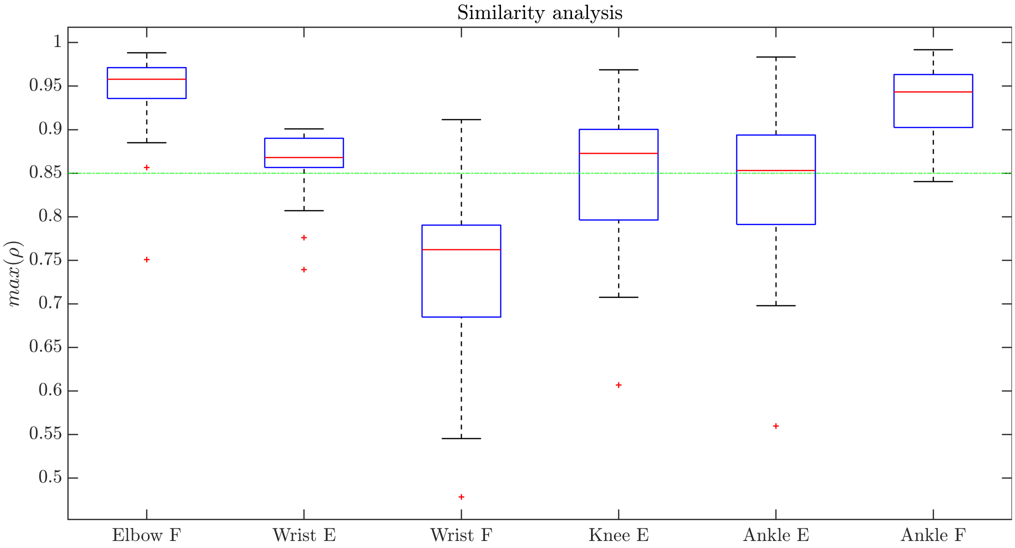


Figure 8.6: The green line represents the ρ acceptability threshold, which is set equal to 0.85. While for Elbow Flexion and Ankle Flexion the correlation coefficient is very high, with a median value of 0.96 and 0.94 respectively, as regards Wrist Flexion the obtained result is rather low, with a median value equal to 0.76.

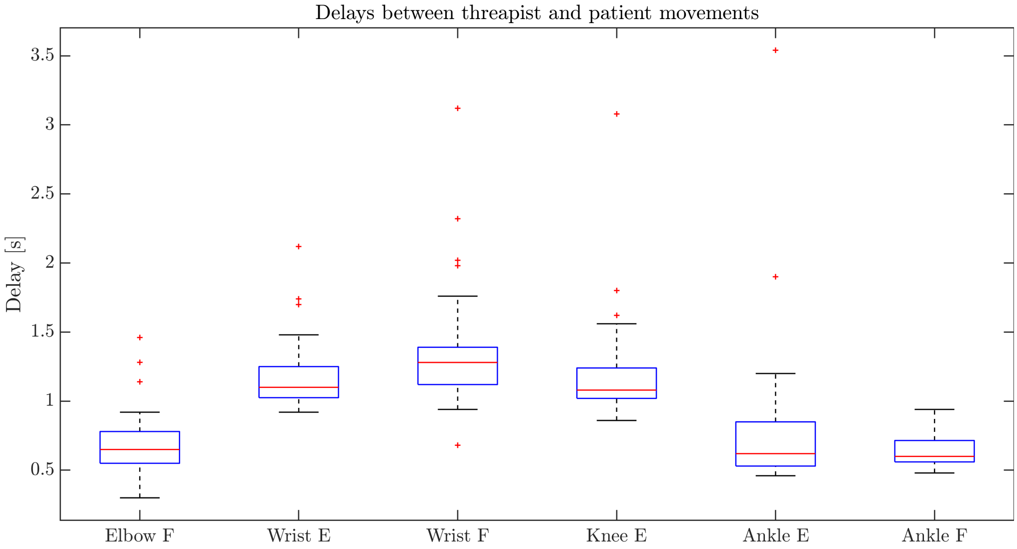


Figure 8.7: The movement with the worst performance in terms of execution delays between the movement of the therapist and the patient one is Wirst Flexion, with a median of 1.28 seconds. The best result instead is obtained with Ankle Flexion, with a median value of 0.60 seconds.

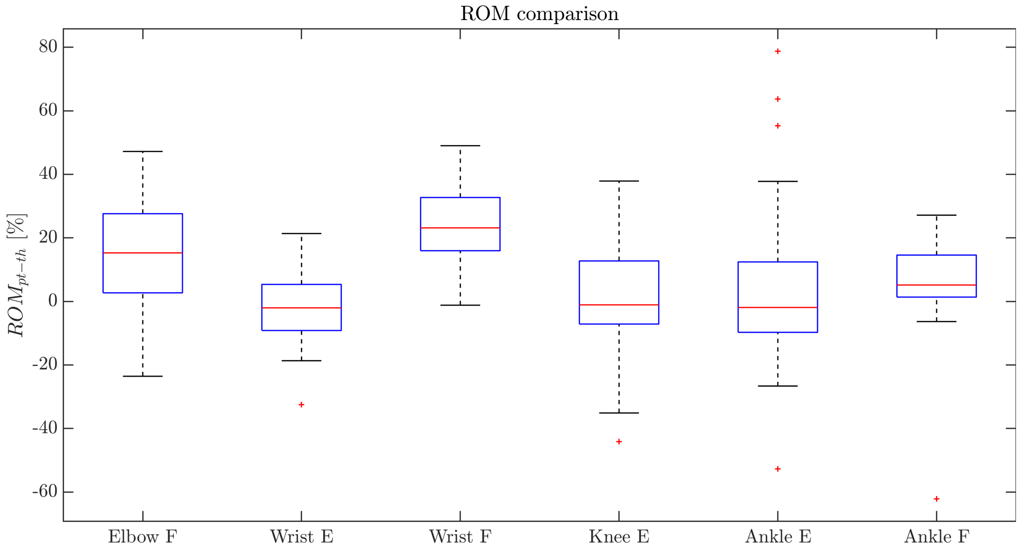


Figure 8.8: As regards Elbow Flexion, Wrist Flexion and Ankle Flexion, the variation percentage of ROM is positive. This is caused by the fact that, for these movements, the ROM reached by the patient during the calibration phase is lower than that obtained during testing. In these cases, therefore, normalization has led to values greater than 1, resulting in positive ROM variations.

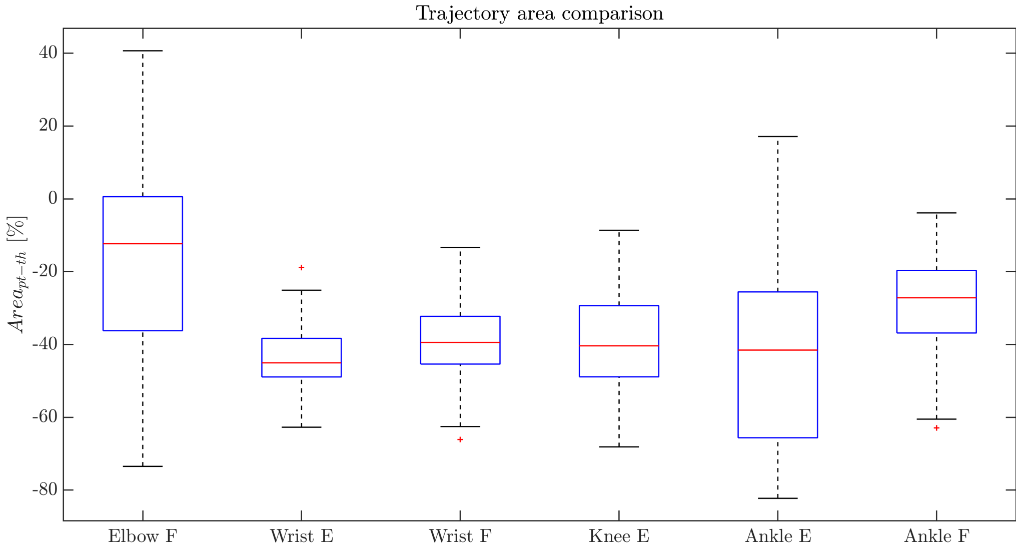


Figure 8.9: Wrist Extension and Ankle Extension are the movements with the greatest variation of area under the angular trajectory. The obtained median values for these two movements are -45.04% and -41.53%, respectively. Anyway, this parameter variation is quite high also in the other movements. The best result is obtained with Elbow Flexion, with a percentage variation of -12.32%.

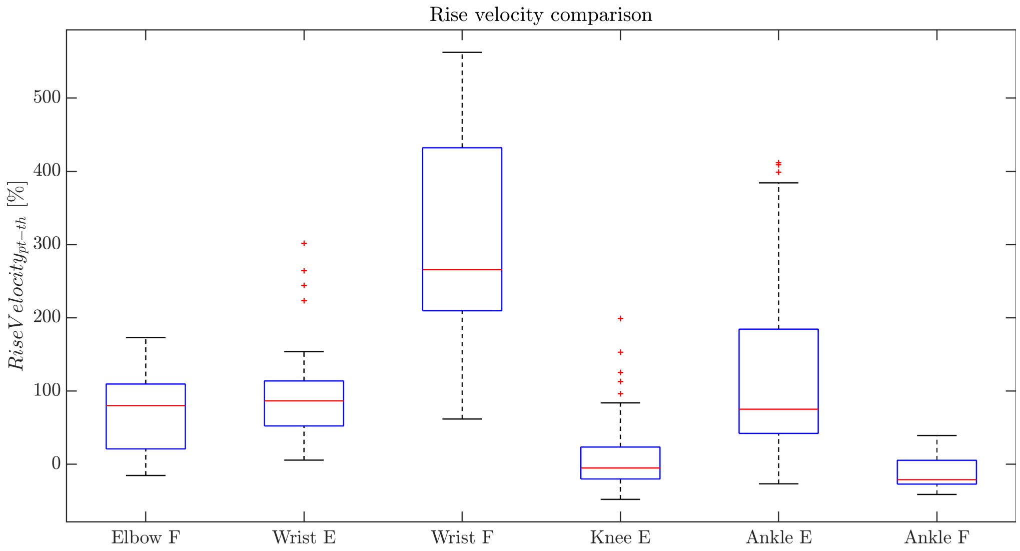


Figure 8.10: The rise velocity from the patient side is almost always greater than the one resulting from the therapist side. This is because in the latter case, before being able to obtain an ATC value high enough to trigger the stimulation, the subject has already executed part of the ROM. The highest rise velocity percentage variation has been obtained for Wrist Flexion, with a median equal to +265.81%.

Table 8.2: Overview of upper limbs tests results. Q_1 and Q_3 are the 25th and 75th percentile, respectively

Comparison	**Elbow F**			**Wrist E**			**Wrist F**		
parameters	**Q1**	**median**	**Q3**	**Q1**	**median**	**Q3**	**Q1**	**median**	**Q3**
$max(\rho)$	0.94	0.96	0.97	0.86	0.87	0.89	0.68	0.76	0.79
Delay [s]	0.55	0.65	0.78	1.03	1.10	1.25	1.12	1.28	1.39
ROM_{pt-th} [%]	2.69	15.27	27.61	-9.16	-2.03	5.35	15.96	23.14	32.71
$Area_{pt-th}$ [%]	-36.21	-12.32	0.60	-48.92	-45.04	-38.32	-45.38	-39.45	-32.26
$RiseVelocity_{pt-th}$ [%]	21.03	80.12	109.59	52.40	86.61	113.81	209.60	265.81	432.14

Table 8.3: Overview of lower limbs tests results. Q_1 and Q_3 are the 25th and 75th percentile, respectively

Comparison	**Knee E**			**Ankle E**			**Ankle F**		
parameters	**Q1**	**median**	**Q3**	**Q1**	**median**	**Q3**	**Q1**	**median**	**Q3**
$max(\rho)$	0.80	0.87	0.90	0.79	0.85	0.89	0.90	0.94	0.96
Delay [s]	1.02	1.08	1.24	0.53	0.62	0.85	0.56	0.60	0.72
ROM_{pt-th} [%]	-7.13	-1.09	12.73	-9.74	-1.91	12.43	1.35	5.15	14.58
$Area_{pt-th}$ [%]	-48.89	-40.37	-29.36	-65.63	-41.53	-25.55	-36.84	-27.17	-19.70
$RiseVelocity_{pt-th}$ [%]	-20.03	-5.04	23.58	42.20	75.27	184.53	-27.01	-21.00	5.50

Chapter 9

Conclusion and Future Perspective

In this thesis project, the evolution of an ATC-FES system to be used in the rehabilitation field is presented. The proposed system exploit and event-driven approach applying the Average Threshold Crossing (ATC) technique to the surface ElectroMyoGraphic (sEMG) signal and uses the obtained result to define a suitable pattern for Functional Electrical Stimulation (FES). Essentially, in the standard usage mode of the system, two subjects are involved: the ATC signal is acquired from the first subject during the execution of a movement; the FES is applied to the second subject in order to replicate the same movement. The two subjects can be therapist and patient, respectively. This system can provide great support for people suffering from muscle paralysis, as it allows the therapist and patient to actively participate in the rehabilitation session. For this reason, an evolution of the system control software has been presented in this thesis project. This application is able to run both on a personal computer and to be integrated into an embedded system thanks to the exploited multi-platform approach. Furthermore, a new prototype of articular electrogoniometers was developed, in order to be used for monitoring and studying the limbs trajectories carried out during the rehabilitation session.

In order to validate the software, tests were performed to measure computational latencies. The measures distribution always have median value lower than 10 ms, both for the configuration that involves the use of a personal computer, and for that integrated into an embedded system. The software was therefore successfully validated, as it allows a real-time control of the system.

Experimental tests were also accomplished to validate the proposed system. In order to have a reliable tool to acquire the limbs trajectories carried out during the trials, the Vicon Motion Capture System has been used. Thanks to the latter, it was possible to have a reference measurement for the validation of the designed electrogoniometers, and it was possible to acquire the movements during the FES sessions.

As regards the electrogoniometers validation, the limbs trajectories acquired with both the designed prototype and the Vicon system has been compared. The trials involved the following movements: Elbow Flexion, Ankle Extension Flexion and Knee Extension. The obtained results during Ankle Extension Flexion are the best ones: the median of the areas under the trajectories acquired by the electrogoniometer is lower than that of the Vicon by only 0.19%. Regarding the other two movements, however, the same parameter was 9.55% and 23.99% lower, respectively for Elbow Flexion and Knee extension. These values suggest that the designed prototype is unable to follow the trajectory beyond a certain Range Of Motion (ROM) of the involved joint.

The ATC-FES system validation has been carried out by using the Vicon system and comparing the trajectories acquired from the two subjects in their respective roles as therapist and patient. The trials involved a total of six subjects, and the movements analyzed were: Elbow Flexion, Wrist Extension, Wrist Flexion, Knee Extension, Ankle Extension and Ankle Flexion. The best results were obtained during the Elbow Flexion and Ankle Flexion sessions. The median of the cross-correlations obtained from each movement repetition is equal to 0.96 and 0.94, respectively. As for the delay between the movement performed by the therapist and the patient one, the median of the measured values is equal to 0.65 and 0.6 seconds, respectively. The worst results were obtained during the Wrist Flexion sessions, for which the obtained cross-correlation median is equal to 0.76, while that of delays was equal to 1.28 seconds. Despite these last values, the system proves to be a valid aid for carrying out movements using FES.

Compared to the current version, the presented system will evolve further in the near future, aiming for the following achievements:

- Control software: its future version should also work on tablets, in order to improve the portability of the system and simplify its use in the rehabilitation field;
- Electrogoniometer: the designed prototype needs to be improved and further comparison tests will be carried out with a reliable tracking system, in order to guarantee the possibility of using a portable tool for studying the angular trajectories performed during the rehabilitation sessions;
- Experimental tests with FES: it is necessary to expand the ambitions of the system by studying how to deal with muscle fatigue, as it is the main obstacle of this technique according to the literature. For example, considering the results obtained in this thesis project, given the good performances obtained during the elbow and ankle flexion sessions, these two movements can be used as a starting point for an in-depth study of the stimulation parameters. The first step will therefore be to carry out tests consisting of sessions in which the number of repetitions of the movement is high enough to allow the onset of muscle fatigue. During these tests, different combinations of stimulation parameters should be compared, in order to estimate which of them result in less fatigue of the subjects involved. Based on the data collected, a machine learning model could then be developed for managing the real-time definition of all the stimulation parameters.

Bibliography

[1] Walter R Frontera and Julien Ochala. “Skeletal muscle: a brief review of structure and function”. In: *Calcified tissue international* 96.3 (2015), pp. 183–195. DOI: 10.1007/s00223-014-9915-y.

[2] Breaking Muscle. *Biceps - Anatonomy and Kinesiology.* Retrieved online on 19/08/2020. URL: https://breakingmuscle.com/fitness/two-jointed-muscles-of-the-arms-how-to-train-them.

[3] S.K. Gollapudi, J.J. Michael, and M. Chandra. “Striated Muscle Dynamics”. In: *Reference Module in Biomedical Sciences.* Elsevier, 2014. ISBN: 978-0-12-801238-3. DOI: 10.1016/B978-0-12-801238-3.00251-8.

[4] Oregon State University. *Skeletal Muscle.* Retrieved online on 19/08/2020. URL: https://open.oregonstate.education/aandp/chapter/10-2-skeletal-muscle/.

[5] B. Cadot and E.R. Gomes. “Skeletal Muscle”. In: *Encyclopedia of Cell Biology.* Ed. by Ralph A. Bradshaw and Philip D. Stahl. Waltham: Academic Press, 2016, pp. 677–682. ISBN: 978-0-12-394796-3. DOI: 10.1016/B978-0-12-394447-4.20065-5.

[6] Brian R. Berridge, Brad Bolon, and Eugene Herman. “Chapter 10 - Skeletal Muscle System”. In: *Fundamentals of Toxicologic Pathology (Third Edition).* Ed. by Matthew A. Wallig et al. Third Edition. Academic Press, 2018, pp. 195–212. ISBN: 978-0-12-809841-7. DOI: 10.1016/B978-0-12-809841-7.00010-1.

[7] Cindy L. Stanfield. *Fisiologia IV edizione.* EdiSES.

[8] Hoffman Hearts HBS. *Muscle contraction.* Retrieved online on 19/08/2020. URL: https://www.hoffmanheartshbs.com/uploads/2/6/7/6/26760401/952987223_orig.jpg.

[9] *Book: Anatomy and Physiology (Boundless).* LibreTexts.

[10] Stefano Schiaffino and Carlo Reggiani. “Chapter 60 - Skeletal Muscle Fiber Types”. In: *Muscle.* Ed. by Joseph A. Hill and Eric N. Olson. Boston/Waltham: Academic Press, 2012, pp. 855–867. ISBN: 978-0-12-381510-1. DOI: 10.1016/B978-0-12-381510-1.00060-0.

[11] Purves D, Augustine GJ, Fitzpatrick D, et al., editors. Neuroscience. 2nd edition. Sunderland (MA): Sinauer Associates; 2001. The Regulation of Muscle Force. Available from: https://www.ncbi.nlm.nih.gov/books/NBK11021/.

[12] Chris Gregory and C Bickel. “Recruitment Patterns in Human Skeletal Muscle During Electrical Stimulation”. In: *Physical therapy* 85 (May 2005), pp. 358–64. DOI: 10.1093/ptj/85.4.358.

[13] Gad Alon. “Functional Electrical Stimulation (FES): Clinical successes and failures to date”. In: *Journal of Novel Physiotherapy and Rehabilitation* 2 (Nov. 2018), pp. 080–086. DOI: 10.29328/journal.jnpr.1001022.

[14] Nawadita Parajulli et al. “Real-Time EMG Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation”. In: *Sensors* 19 (Oct. 2019), p. 4596. DOI: 10.3390/s19204596.

[15] Samer Chantaf, Lobna Makni, and Amine Nait-ali. “Single Channel Surface EMG Based Biometrics”. In: *Hidden Biometrics: When Biometric Security Meets Biomedical Engineering*. Ed. by Amine Nait-ali. Singapore: Springer Singapore, 2020, pp. 71–90. ISBN: 9789811309564. DOI: 10.1007/978-981-13-0956-4_4. URL: https://doi.org/10.1007/978-981-13-0956-4_4.

[16] Laxmi Shaw and Sangeeta Bhaga. “Online EMG Signal Analysis for diagnosis of Neuromuscular diseases by using PCA and PNN.” In: *International Journal Of Engineering Science and Technology 0975-5462* 4 (Oct. 2012), pp. 4453–4459.

[17] Delsys Carlo J. De Luca. *Surface Electromyography: Detection and Recording.*

[18] Stephen Lee and John Kruse. “Biopotential electrode sensors in ECG/EEG/EMG systems”. In: *Analog Devices* (Jan. 2008).

[19] José Guerreiro, Andre Lourenco, and Ana Fred. “A Biosignal Embedded System for Physiological Computing”. PhD thesis. Dec. 2013. DOI: 10.13140/2.1.4596.0481.

[20] M.R. Neuman. “Biopotential electrodes”. In: *Medical Devices and Human Engineering*. Jan. 2014, pp. 4–1.

[21] Luca Mesin. *Introduction to biomedical signal processing.* 2017. ISBN: 9788892332485.

[22] Roberto Merletti and Silvia Muceli. “Tutorial. Surface EMG detection in space and time: Best practices”. In: *Journal of Electromyography and Kinesiology* 49 (Oct. 2019), p. 102363. DOI: 10.1016/j.jelekin.2019.102363.

[23] Muhammad Jamal. “Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis”. In: *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*. Oct. 2012. ISBN: 978-953-51-0805-4. DOI: 10.5772/52556.

[24] Rubana Chowdhury et al. “Surface Electromyography Signal Processing and Classification Techniques”. In: *Sensors (Basel, Switzerland)* 13 (Sept. 2013), pp. 12431–66. DOI: 10.3390/s130912431.

[25] Derek O’Keeffe et al. “Stimulus artifact removal using a software-based two-stage peak detection algorithm”. In: *Journal of Neuroscience Methods* 109 (Sept. 2001), pp. 137–145. DOI: 10.1016/S0165-0270(01)00407-1.

[26] Yurong Li, Jun Chen, and Yuan Yang. “A Method for Suppressing Electrical Stimulation Artifacts from Electromyography”. In: *International Journal of Neural Systems* 29 (Nov. 2018). DOI: 10.1142/S0129065718500545.

[27] Christopher Spiewak et al. “A Comprehensive Study on EMG Feature Extraction and Classifiers”. In: *Open Access Journal of Biomedical Engineering and its Applications* 1 (Feb. 2018). DOI: 10.32474/OAJBEB.2018.01.000104.

[28] George Dimitrov et al. “Muscle Fatigue During Dynamic Contractions Assessed by New Spectral Indices”. In: *Medicine and science in sports and exercise* 38 (Dec. 2006), pp. 1971–9. DOI: 10.1249/01.mss.0000233794.31659.6d.

[29] Delsys Carlo J. De Luca. *The Use of Surface Electromyography in Biomechanics.*

[30] Dennis Tkach, He Huang, and Todd Kuiken. “Study of stability of time-domain features for electromyographic pattern recognition”. In: *Journal of neuroengineering and rehabilitation* 7 (May 2010), p. 21. DOI: 10.1186/1743-0003-7-21.

[31] Joshua Richman and Joseph Moorman. “Physiological Time-Series Analysis Using Approximate Entropy and Sample Entropy”. In: *American journal of physiology. Heart and circulatory physiology* 278 (July 2000), H2039–49. DOI: 10.1152/ajpheart.2000.278.6.H2039.

[32] Danilo Demarchi. *Bio-Inspired Electronics.* URL: https://prezi.com/okmagebhegav/bio-inspired-electronics-danilo-demarchi/ (visited on 11/10/2020).

[33] Marco Crepaldi et al. “A quasi-digital radio system for muscle force transmission based on event-driven IR-UWB”. In: *2012 IEEE Biomedical Circuits and Systems Conference (BioCAS).* IEEE. 2012, pp. 116–119.

[34] Paolo Motto Ros et al. “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.* IEEE. 2013, pp. 51–56.

[35] Stefano Sapienza et al. “On integration and validation of a very low complexity ATC UWB system for muscle force transmission”. In: *IEEE transactions on biomedical circuits and systems* 10.2 (2015), pp. 497–506.

[36] Amirhossein Shahshahani et al. “An all-digital spike-based ultra-low-power IR-UWB dynamic average threshold crossing scheme for muscle force wireless transmission”. In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE. 2015, pp. 1479–1484.

[37] National Spinal Cord Injury Statistical Center. *Spinal cord injury.* URL: https://www.sci-info-pages.com/wp-content/media/NSCISC-2019-Spinal-Cord-Injury-Facts-and-Figures-at-a-Glance.pdf (visited on 09/25/2020).

[38] Salim Virani et al. “Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association”. In: *Circulation* 141 (Jan. 2020). DOI: 10.1161/CIR.0000000000000757.

[39] Kei Masani and Milos Popovic. “Functional Electrical Stimulation in Rehabilitation and Neurorehabilitation”. In: *Springer Handbook of Medical Technology (pp.877-896).* Jan. 2011, pp. 877–896. ISBN: 978-3-540-74657-7. DOI: 10.1007/978-3-540-74658-4_44.

[40] Milos Popovic, Kei Masani, and Silvestro Micera. “Functional Electrical Stimulation Therapy: Recovery of Function Following Spinal Cord Injury and Stroke”. In: *Neurorehabilitation Technology.* Dec. 2012, pp. 105–121. DOI: 10.1007/978-1-4471-2277-7_7.

[41] D. Scott W. T. Liberson H. J. Holmquest and M. Dow. “Functional electrotherapy: stimulation of the peroneal nerve synchronized with the swing phase of the gait of hemiplegic patients”. In: *Arch. Phys. Med. Re-habil.* 42 (1961), pp. 101–105.

[42] Dinesh Bhatia et al. “State of art: Functional Electrical Stimulation (FES)”. In: *Int. J. of Biomedical Engineering and Technology* 5 (Feb. 2011), pp. 77–99. DOI: 10.1504/IJBET.2011.038474.

[43] Milos R. Popovic. *Functional Electrical Stimulation.* URL: https://en.wikipedia.org/wiki/Functional_electrical_stimulation#/media/File:Functional_Electrical_Stimulation_.tif (visited on 10/24/2020).

[44] Johnston TE Chou LW Lee SC and Binder-Macleod SA. “The effectiveness of progressively increasing stimulation frequency and intensity to maintain paralyzed muscle force during repetitive activation in persons with spinal cord injury”. In: *Arch. Phys. Med. Re-habil.* 89 (May 2008), pp. 856–864. DOI: 10.1016/j.apmr.2007.10.027.

[45] *On Transcutaneous Functional Electrical Stimulation.* URL: https://neupsykey.com/on-transcutaneous-functional-electrical-stimulation/ (visited on 10/24/2020).

[46] Carles Gomez, Joaquim Oller Bosch, and Josep Paradells. “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology”. In: *Sensors (Basel, Switzerland)* 12 (Dec. 2012), pp. 11734–53. DOI: 10.3390/s120911734.

[47] Bluetooth Special Interest Group (SIG). *Bluetooth Market Update 2020.* URL: https://www.bluetooth.com/wp-content/uploads/2020/03/2020_Market_Update-EN.pdf (visited on 10/24/2020).

[48] Jacopo Tosi et al. “Performance Evaluation of Bluetooth Low Energy: A Systematic Review”. In: *Sensors* 17 (Dec. 2017), p. 2898. DOI: 10.3390/s17122898.

[49] Robin Heydon. *Bluetooth Low Energy.* URL: https://www.cl.cam.ac.uk/teaching/1819/MobSensSys/mobile-10.pdf (visited on 10/24/2020).

[50] Delsys Carlo J. De Luca. *Fundamental Concepts in EMG Signal Acquisition.*

[51] *Trigno EMG sensors.* URL: https://delsys.com/trigno/research/#trigno-avanti-sensor (visited on 11/07/2020).

[52] *pico EMG Cometa Systems.* URL: https://www.cometasystems.com/products/picoemg (visited on 11/07/2020).

[53] D. A. F. Guzman et al. “Very low power event-based surface EMG acquisition system with off-the-shelf components”. In: *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS).* 2017, pp. 1–4. DOI: 10.1109/BIOCAS.2017.8325152.

[54] Stefano Sapienza et al. “On-Line Event-Driven Hand Gesture Recognition Based on Surface Electromyographic Signals”. In: May 2018, pp. 1–5. DOI: 10.1109/ISCAS.2018.8351065.

[55] Andrea Mongardi et al. “A Low-Power Embedded System for Real-Time sEMG based Event-Driven Gesture Recognition”. In: Nov. 2019, pp. 65–68. DOI: 10.1109/ICECS46596.2019.8964944.

[56] Fabio Rossi et al. "Wireless Low Energy System Architecture for Event-Driven Surface Electromyography". In: May 2019, pp. 179–185. ISBN: 978-3-030-11972-0. DOI: 10.1007/978-3-030-11973-7_21.

[57] Fabio Rossi et al. "An Event-Driven Closed-Loop System for Real-Time FES Control". In: Nov. 2019, pp. 867–870. DOI: 10.1109/ICECS46596.2019.8965153.

[58] Fabio Rossi et al. "Embedded Bio-Mimetic System for Functional Electrical Stimulation Controlled by Event-Driven sEMG". In: *Sensors* 20 (Mar. 2020), p. 1535. DOI: 10.3390/s20051535.

[59] Zuozheng Lou, Peng Yao, and Dingguo Zhang. "Wireless Master-Slave FES Rehabilitation System Using sEMG Control". In: vol. 7507. Oct. 2012, pp. 1–10. ISBN: 978-3-642-33514-3. DOI: 10.1007/978-3-642-33515-0_1.

[60] Yukihiro Hara. "Rehabilitation with Functional Electrical Stimulation in Stroke Patients". In: *International Journal of Physical Medicine and Rehabilitation* 01 (Jan. 2013). DOI: 10.4172/2329-9096.1000147.

[61] Yuyang Chen, Chenyun Dai, and Wei Chen. "A Real-time EMG-controlled Functional Electrical Stimulation System for Mirror Therapy". In: Oct. 2019, pp. 1–4. DOI: 10.1109/BIOCAS.2019.8919069.

[62] Yuxuan Zhou et al. "Electromyographic Bridge—a multi-movement volitional control method for functional electrical stimulation: prototype system design and experimental validation". In: vol. 2017. July 2017, pp. 205–208. DOI: 10.1109/EMBC.2017.8036798.

[63] Yuxuan Zhou et al. "A frequency and pulse-width co-modulation strategy for transcutaneous neuromuscular electrical stimulation based on sEMG time-domain features". In: *Journal of neural engineering* 13 (Dec. 2015), p. 016004. DOI: 10.1088/1741-2560/13/1/016004.

[64] Zheng-Yang Bi et al. "Prototype System Design and Experimental Validation of Gait-Oriented EMG Bridge for Volitional Motion Function Rebuilding of Paralyzed Leg". In: Mar. 2019, pp. 79–82. DOI: 10.1109/ICBCB.2019.8854632.

[65] Maikutlo Kebaetse et al. "Strategies That Improve Paralyzed Human Quadriceps Femoris Muscle Performance During Repetitive, Nonisometric Contractions". In: *Archives of physical medicine and rehabilitation* 86 (Nov. 2005), pp. 2157–64. DOI: 10.1016/j.apmr.2005.06.011.

[66] Trisha Kesar and Stuart Binder-Macleod. "Effect of frequency and pulse duration on human muscle fatigue during repetitive electrical stimulation". In: *Experimental physiology* 91 (Dec. 2006), pp. 967–76. DOI: 10.1113/expphysiol.2006.033886.

[67] Woohyoung Jeon and Lisa Griffin. "Effects of pulse duration on muscle fatigue during electrical stimulation inducing moderate-level contraction: Pulse Duration Effects during NMES". In: *Muscle and Nerve* 57 (Sept. 2017). DOI: 10.1002/mus.25951.

[68] Hallett M Panizza M Nilsson J. "Optimal stimulus duration for the H reflex". In: *Muscle Nerve* 12 (July 1989), pp. 576–9. DOI: 10.1002/mus.880120708.

[69] Gad Alon et al. “Comparing four electrical stimulators with different pulses properties and their effect on the discomfort and elicited dorsiflexion”. In: (Oct. 2013).

[70] Fabio Rossi. *Low power system for event-driven control of functional electrical stimulation.* Torino, 2017.

[71] Sofia Cecchini. *Average Threshold Crossing Validation for Functional Electrical Stimulation applied to surface ElectroMyoGraphic signals.* 2018.

[72] Ricardo Maximiliano Rosales. *Real-Time Embedded System for Event-Driven sEMG Acquisition and FES Control.* 2019.

[73] Nordic Semiconductor. *nRF52840.* URL: `https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840` (visited on 11/20/2020).

[74] Ambiq Micro. *Apollo3 Blue.* URL: `https://ambiq.com/apollo3-blue/` (visited on 11/20/2020).

[75] Enrique Spinelli, Nayghet Martínez, and Miguel Mayosky. “A transconductance driven-right-leg circuit”. In: *Biomedical Engineering, IEEE Transactions on* 46 (Jan. 2000), pp. 1466–1470. DOI: `10.1109/10.804574`.

[76] HASOMED. *RehaStim2.* URL: `https://hasomed.de/en/products/rehamove/` (visited on 10/24/2020).

[77] HASOMED GmbH. *Operation Manual RehaStim2, RehaMove2.*

[78] Bjoern Kuberski. *ScienceMode2, RehaStim2 Stimulation Device, Description and Protocol.*

[79] S. Affatato. “2 - Biomechanics of the knee”. In: *Surgical Techniques in Total Knee Arthroplasty and Alternative Procedures.* Ed. by Saverio Affatato. Oxford: Woodhead Publishing, 2015, pp. 17–35. ISBN: 978-1-78242-030-9. DOI: `https://doi.org/10.1533/9781782420385.1.17`. URL: `http://www.sciencedirect.com/science/article/pii/B9781782420309500026`.

[80] Lucie Brosseau et al. “Intra- and intertester reliability and criterion validity of parallelogram and universal goniometers for measuring maximum active knee flexion and extension of patients with knee restrictions”. In: *Archives of physical medicine and rehabilitation* 82 (Apr. 2001), pp. 396–402. DOI: `10.1053/apmr.2001.19250`.

[81] Valentina Agostini et al. “A Wearable Magneto-Inertial System for Gait Analysis (H-Gait): Validation on Normal Weight and Overweight/Obese Young Healthy Adults”. In: *Sensors* 17 (Oct. 2017), p. 2406. DOI: `10.3390/s17102406`.

[82] SparkFun Electronics. *Artemis Thing Plus.* URL: `https://www.sparkfun.com/products/15574` (visited on 11/18/2020).

[83] Bourns. *Encoder EMS22A.* URL: `https://www.bourns.com/products/encoders/magnetic-encoders/product/EMS22A` (visited on 11/18/2020).

[84] formlabs. *Form 3.* URL: `https://formlabs.com/3d-printers/form-3/` (visited on 11/18/2020).

[85] *Kivy’s documentation.* URL: `https://kivy.org/doc/stable/` (visited on 11/10/2020).

[86] Texas Instrument. *Dongle CC2540*. URL: https://www.ti.com/tool/CC2540EMK-USB#:~:text=The%20CC2540%20USB%20Evaluation%20Module,free%20tool%20SmartRF%20Packet%20Sniffer). (visited on 10/24/2020).

[87] *KV language*. URL: https://kivy.org/doc/stable/guide/lang.html (visited on 11/10/2020).

[88] *pySerial's documentation*. URL: https://pyserial.readthedocs.io/en/latest/index.html (visited on 11/10/2020).

[89] *BlueZ: Official Linux Bluetooth protocol stack*. URL: http://www.bluez.org/about/ (visited on 11/10/2020).

[90] *bluepy: a Bluetooth LE interface for Python*. URL: https://ianharvey.github.io/bluepy-doc/ (visited on 11/10/2020).

[91] *SENIAM project*. URL: http://www.seniam.org/ (visited on 11/12/2020).

[92] *Electrode Placement and Functional Movement*. URL: https://www.axelgaard.com/Education/Electrode-Placement-and-Functional-Movement (visited on 11/12/2020).

[93] RaspberryPi. *Raspberry Pi 4 Tech Specs*. URL: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home (visited on 11/20/2020).

[94] Università degli Studi di Torino. *Comitato di Bioetica dell'Ateneo*. URL: https://www.unito.it/ricerca/strutture-e-organi-la-ricerca/comitato-di-bioetica-dellateneo (visited on 11/24/2020).

[95] Vicon Motion Systems. *Vicon*. URL: https://www.vicon.com/ (visited on 11/24/2020).

[96] Politecnico di Toino. *PolitoBIOMed Lab*. URL: http://www.biomedlab.polito.it/en/ (visited on 11/24/2020).

[97] Vicon Motion Systems. *Vicon cameras*. URL: https://www.vicon.com/hardware/cameras/ (visited on 11/24/2020).

[98] Vicon Motion Systems. *Nexus*. URL: https://www.vicon.com/software/nexus/ (visited on 11/24/2020).

[99] Vicon Motion Systems. *Vicon Nexus User Guide*. URL: https://docs.vicon.com/display/Nexus211/Vicon+Nexus+User+Guide (visited on 11/24/2020).

[100] Vicon Motion Systems. *Vicon calibration*. URL: https://www.vicon.com/hardware/devices/calibration/ (visited on 11/24/2020).

[101] Vicon Motion Systems. *Plug-in Gait Reference Guide*. URL: https://docs.vicon.com/display/Nexus211/Plug-in+Gait+Reference+Guide (visited on 11/24/2020).

[102] Medtronic. *Covidien products*. URL: https://www.medtronic.com/covidien/en-us/products.html (visited on 11/24/2020).

[103] FIAB. *FIAB products*. URL: https://www.fiab.it/prodotti.php (visited on 11/24/2020).

[104] HASOMED GmbH. *RehaMove, Functional electrical stimulation*.