Device control system based on classified EMG signals: a machine learning approach

Ana Carolina Barbosa

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering.

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Laercio Simas Mattos

Bragança

2023

Device control system based on classified EMG signals: a machine learning approach

Ana Carolina Barbosa

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering. Work developed under the Dual Degree Program between the Polytechnic Institute of Bragança (IPB) and the Federal Center for Technological Education of Minas Gerais (CEFET-MG).

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Laercio Simas Mattos

Bragança

2023

Dedication

I dedicate this work to my grandparents, the ones who taught me the true meaning of love.

Acknowledgment

There are countless people without whom I would never be able to conclude this work, like all those who came before me and dedicated their lives to knowledge, making possible the tools used to elaborate this project, because as Isaac Newton said "If I have seen further it is by standing on the shoulders of Giants".

To all my family, especially my parents Tânia and José, who together built the whole foundation for my growth, teaching me values that I will take with me throughout my life. And to my brother and sister, Aline and William who always showed support in any way they could

I also thank my second family Joana, Juliana and Luiza who believed in me when I didn't believe in myself, who took care of me many times better than I did, never letting me give up. To my friends Jesus, Melo, Edilson and Vinícius who always showed themselves willing to help me in any crazy project that came up. They made every day of my academic trajectory better.

I am also immensely grateful to the friends that the Exchange Program gave me, Ana, Danilo, Debora, Jefferson, João, Julia, Nathália, Renata, Shinti, Vitor, Willian, Giovanna, Sofia e Gabriela with them I was able to live experiences that I never believed possible, and through this I was able to become a better professional as well as a better person. Special thanks to Paloma who, countless times, turned over nights giving me her support.

To the Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)'s contract workers, I will be eternally grateful for the caffeine provided and for always being willing to help whatever the problem was, to the employees of the coordinations and directorates who worked hard every time we needed them and to the professors who

shared their knowledge the best way they knew how. To all the employees of CEFET-MG campus Leopoldina, my gratitude.

To the professors Laercio, Deilton and Lima, who believed in this project from the beginning, especially to professors Laercio and Lima, who oriented me at every step, besides giving all the necessary support, without them this project would never get off the paper.

Lastly I thank myself for having tried.

Abstract

In contemporary society, certain physical attributes are celebrated, while others are stigmatized, leading to barriers in the social inclusion of individuals who do not conform to these idealized standards. People with disabilities (PwD) often face societal prejudices, further exacerbated by the absence of adaptive tools, pushing them away from a conventional life. Approximately 15% of the global population has some form of disability, with a significant portion experiencing physical disabilities related to upper limbs. Among these, many undergo amputation, a transformative process that affects both their physical and psychological well-being. Prostheses, while beneficial, have limitations in replicating the full range of limb movements and are often financially inaccessible to many. This research proposes an innovative system that leverages the retained ability of amputees to generate electromyographic (EMG) signals post-amputation. The system aims to control electronic devices directly through these signals, bypassing the need for prosthetics. Potential applications include replacing traditional computer mice and controlling gaming platforms. The core design is a compact bracelet equipped with non-invasive EMG sensors, an accelerometer, and a gyroscope. Data from these sensors are processed using artificial intelligence techniques to generate device-specific commands. The overarching goal is to enhance the autonomy and social integration of amputees, while also contributing to technological advancements in the field.

Keywords: Electromyographic (EMG) signals, Artificial intelligence, People with Disabilities (PwD)

Resumo

Na sociedade contemporânea, certos atributos físicos são celebrados, enquanto outros são estigmatizados, levando a barreiras na inclusão social de indivíduos que não se enquadram nesses padrões idealizados. Pessoas com deficiências (PcD) frequentemente enfrentam preconceitos sociais, ainda mais exacerbados pela ausência de ferramentas adaptativas, afastando-as de uma vida convencional. Aproximadamente 15% da população global possui algum tipo de deficiência, com uma parcela significativa experimentando deficiências físicas relacionadas aos membros superiores. Dentre estes, muitos passam por amputações, um processo transformador que afeta tanto o bem-estar físico quanto o psicológico. Próteses, embora benéficas, têm limitações em replicar a totalidade dos movimentos dos membros e muitas vezes são financeiramente inacessíveis para muitos. Esta pesquisa propõe um sistema inovador que aproveita a capacidade retida dos amputados de gerar sinais eletromiográficos (EMG) pós-amputação. O sistema visa controlar dispositivos eletrônicos diretamente por meio desses sinais, eliminando a necessidade de próteses. As aplicações potenciais incluem a substituição de mouses de computador tradicionais e o controle de plataformas de jogos. O design central é uma pulseira compacta equipada com sensores EMG não invasivos, um acelerômetro e um giroscópio. Os dados desses sensores são processados usando técnicas de inteligência artificial para gerar comandos específicos para dispositivos. O objetivo principal é melhorar a autonomia e integração social dos amputados, contribuindo também para avanços tecnológicos no campo.

Palavras-chave: Sinais Eletromiográficos (EMG), Inteligência Artificial, Pessoas com Deficiência (PcD)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In our society while some body marks represent beauty, health, and perfection, others carry the burden of being seen as a defect. In the world, different bodies are often seen as disadvantageous, which makes the integration of people who carry these marks a difficult and painful process. In this context, Person with disabilities (PwD) are commonly seen as imperfect and incapable, and this is aggravated by the lack of adapted means and tools, making these people experience countless difficulties in performing daily tasks, pushing them further away from what is called a "normal" life.

According to [1], about 16% of the world's population lives with some form of disability. The global disability prevalence is higher than previous estimates, which date from the 2011 and suggested a figure of around 15% [2].

In a less comprehensive scenario such as the Brazilian one it is estimated that 8.4% of the population has some of the four classifications of disability, which are intellectual, physical, hearing and visual, which in a population larger than 200 million individuals is a very significant number. Specifically, 2.7% of people declare to have a physical disability related to the use of the upper limbs [3]. Within this category are people who have undergone amputations, a process where it is estimated that the worldwide incidence is approximately 1 million people per year [4]. 30% of which corresponds to amputations of the arm region [5].

The amputee goes through a long rehabilitation process, often accompanied by a set of

ambiguous feelings in relation to their own body, since when they see their new condition, in which they no longer fit into the standards of normality valued in the environment where they are inserted, they end up losing not only a member but also the life they used to live [6].

This experience reflects the wider challenges encountered by people with disabilities, similar to those observed in the European Union. Approximately 100 million EU citizens live with some form of disability [7], facing notable socio-economic disparities. For instance, the employment rate for disabled individuals aged 20-64 is just 50.8%, compared to 75% for those without disabilities, and 28.4% are at risk of poverty or social exclusion, a concerning figure against the general population's 17.8%.

In this context, the possibility of using assistive technologies brings with it a unique meaning, since they not only rescue the body's functionality, but also rebuild the image of a "complete" being, making the person to be seen again as someone "healthy" [6].

Consequently, prostheses, which are commonly perceived as the most comprehensive solution for amputations, significantly enhance the quality of life for their users. However, despite the advancements in related studies, they are incapable of replicating all the movements of a biological limb. This limitation, compounded by the inherent delay between command and movement execution — typically due to factors such as sensor processing speed and data transmission time to the controller — hinders and complicates their use. Furthermore, they carry a high market value and remain inaccessible to a substantial segment of the population.

Within the outlined scope, the proposed project is presented. Given that amputees retain the ability to generate corresponding EMG signals to movements, even after limb amputation, the technological innovation delineated here aims at the development of a system that directly controls electronic devices through these EMG signals, eliminating the need for a prosthesis as an intermediary. Such an approach has the potential to minimize the response time of devices and overcome previously described limitations.

This apparatus offers a diverse range of potential applications. One of them is the replacement of conventional pointing devices, such as mice, providing amputees with

a more fluid interaction with computers. Additionally, there is a vision for its use in controlling touch interfaces on mobile devices and gaming platforms, where user response agility is imperative. In the long run, such technology could supplant conventional gaming consoles, often scarce in adaptive versions for amputees.

The initial conception of the project focuses on the implementation of a compact-sized device, structured as a bracelet, equipped with non-invasive EMG sensors, an accelerometer, and a gyroscope. The data collected by these sensors will be transmitted via Bluetooth connection to the target device, in this context, a computer. Subsequently, data from the accelerometer and gyroscope will assist in determining the spatial orientation of the bracelet, while the EMG signal, processed by the user, will be interpreted using artificial intelligence techniques. Based on these analyses, specific commands will be generated and directed to the device in question.

1.1 General Objectives

Given the difficulties faced by PwD the main goal of this project is to create a device that relies on EMG signals to enable amputees to control devices. The aim of this device is to enhance users independence and social integration while also contributing to the field of technologies.

Despite the increasing attention being paid to technologies finding practical and effective solutions for amputees remains a challenge. The complex nature of interpreting EMG signals and converting them into commands for devices makes developing such solutions an intimidating task. However creating a version that's truly usable by these individuals requires an amount of time. As a result this project primarily focuses on conducting a study to understand how different approaches impact the outcome of the technology. Additionally it aims to establish a foundation, for investigations and advancements in this field.

1.2 Specific Objectives

This section presents the specific objectives of this work, highlighting the importance of each topic to achieve the general objective.

1.2.1 Identification of myoelectric signals

At this stage, the goal was to properly capture the myoelectric signal from the forearm muscles. This involved filtering the signal in a way that would minimize noise interference in the AI training, while preserving its characteristics. With the data collected in a feasible manner, the intention is to create a database that relates the signals to the movements of different individuals, allowing this information to be used in future studies.

1.2.2 Artificial Intelligence

The training phase in artificial intelligence systems is a foundational pillar for the optimal performance of the device under discussion. During this process, the aim is to explore various training strategies and apply distinct filters to the myoelectric signal, with the intent of discerning the configuration that maximizes accuracy in pattern recognition.

Such an investigation requires a meticulous analysis of the performances of various machine learning models, as well as a profound understanding of the impacts of each filter used. The paramount goal is to develop an artificial intelligence model that interprets the EMG signals with notable precision and swiftness, consequently reducing the interval between signal capture and the activation of the corresponding command.

Through this rigorous process, the aspiration is to solidify the efficiency of the device in question, thereby expanding the autonomy and interaction capabilities of amputee individuals

1.2.3 Identification of the Bracelet Movement

The purpose of this step is to analyze the data collected by the accelerometer and gyroscope in the bracelet, in order to map the movements of the device in space. With the mapping done, the data generated will be used to control the cursor on the devices to be commanded.

1.2.4 Control of the Devices

At this point, the aim is to convert the processed data from the previous stages into commands that can be applied to the devices to be controlled. As mentioned earlier, the information obtained through the three-dimensional movement of the bracelet will be used to control the cursor on the device, while the EMG signal patterns will be transformed into specific commands, adapted according to the needs of each scenario or device in question.

1.3 Structure of the paper

This document has been structured with the purpose of guiding the reader through the reading process, aiming to facilitate a comprehensive understanding of the research project discussed. For this reason, the document is divided into distinct chapters, each one focusing on elucidating a specific aspect of the overall narrative of the study. In the subsequent sections, a brief introduction to each of these chapters will be presented.

Chapter 1 introduces the project to the reader, outlining the problem in question and the proposed solution. In this section, the general objectives are established, providing a comprehensive view of the project's ambitions. Subsequently, the focus shifts to specific objectives, detailing the importance of each stage of the process. The chapter concludes with the presentation of the document's structure, offering the reader a guide to anticipate the flow of subsequent information.

In Chapter 2, the document delves into the existing body of knowledge on the subject.

This literature review is fundamental for situating the current research within the broader academic dialogue. It also examines the tools and methodologies that were previously employed, laying the groundwork for the project's unique approach.

Chapter 3 is dedicated to methodology, revealing the methods employed in the development of the research and highlighting their relevance. This chapter aims to familiarize the reader with the development process, frankly addressing the challenges faced and the solutions adopted. It provides a transparent view of the research trajectory, offering insights into the practical aspects of conducting the study.

In Chapter 4, the obtained results are presented and analyzed in detail. This chapter focuses on the in-depth exploration of the findings, interpreting them to provide a clear understanding of the lessons learned. The presentation of data is enriched with visual aids, thus facilitating the reader's interpretation. Furthermore, the chapter honestly addresses any discrepancies or unexpected results, integrating them into the research as valuable elements that enhance the overall understanding of the study.

Finally, Chapter 5 concludes the research. It revisits the objectives established in the first chapter and reflects on the extent to which they have been achieved. This chapter is contemplative, considering the implications of the findings and their contributions to the field. It recognizes the limitations of the study and suggests paths for future research, ensuring that the end of this document is not the conclusion of the investigation, but an invitation to further explorations.

Chapter 2

State of art and tools presentation

This chapter aims to provide an overview of the key theories, concepts, and technologies that are relevant to understanding this research, as well as to present the methods and tools addressed in this study. This literature review aims to identify the main issues and gaps in knowledge related to the topic of the work.

2.1 Myoelectric Signals

Understanding the aspects related to myoelectric signals is essential to the understanding of this research. This section presents an explanation from basic concepts of muscle physiology to acquisition techniques and characteristics of the signal in question.

2.1.1 Muscles

Movement in the human body is the result of alternation between contraction and relaxation of muscles, and muscle strength is a reflection of the primary function of muscles to convert chemical energy into mechanical energy [8].

In anatomy there are three types of muscle tissue, the smooth, the cardiac, both of which are activated involuntarily, and finally the skeletal muscle, which is mostly activated voluntarily. Here, only the skeletal muscle will be treated, since it is responsible for

the movement of the body. There are, according to [8], four properties that make the functioning of this tissue possible:

- Electrical excitability: The ability of muscle tissue to produce electrical signals, called action potentials, in response to stimuli, such as chemical stimulation generated by the release of neurotransmitters by neurons.
- Contractility: the ability of muscle tissue to contract strongly when stimulated by an action potential.
- Extensibility: the capacity of muscle tissue to extend without being injured.
- Elasticity: the ability of muscle tissue to return to its initial state after a contraction or stretching.

The property that interests us here is electrical excitability, specifically action potentials, as they are the basis of muscle electrical activity, and the EMG signal studied is the reading of this electrical activity. It is worth noting that the emergence of these action potentials occurs during muscle contraction.

With that in mind, now move on to understanding how muscle contraction and stretching work. To achieve this understanding, it is necessary to first examine the macro and microscopic anatomy. This requires a brief exploration of the organization from the muscle to the myosin and actin filaments. It is possible to analyze this trajectory through Figure 2.1.

Each muscle is considered a separate organ, containing within it thousands of muscle fibers, each of these fibers being formed by even smaller units called myofibrils, which in turn are made up of myosin and actin filaments that are responsible for the actual muscle contractions. Filling the spaces between the myofibrils is the intracellular fluid known as sarcoplasm, this fluid contains large amounts of potassium, magnesium, and phosphate, as well as multiple protein enzymes. In addition, the sarcoplasmic reticulum stores a large concentration of calcium ions (Ca^{++}) and has the function of releasing these ions during the process of muscle contraction [8].

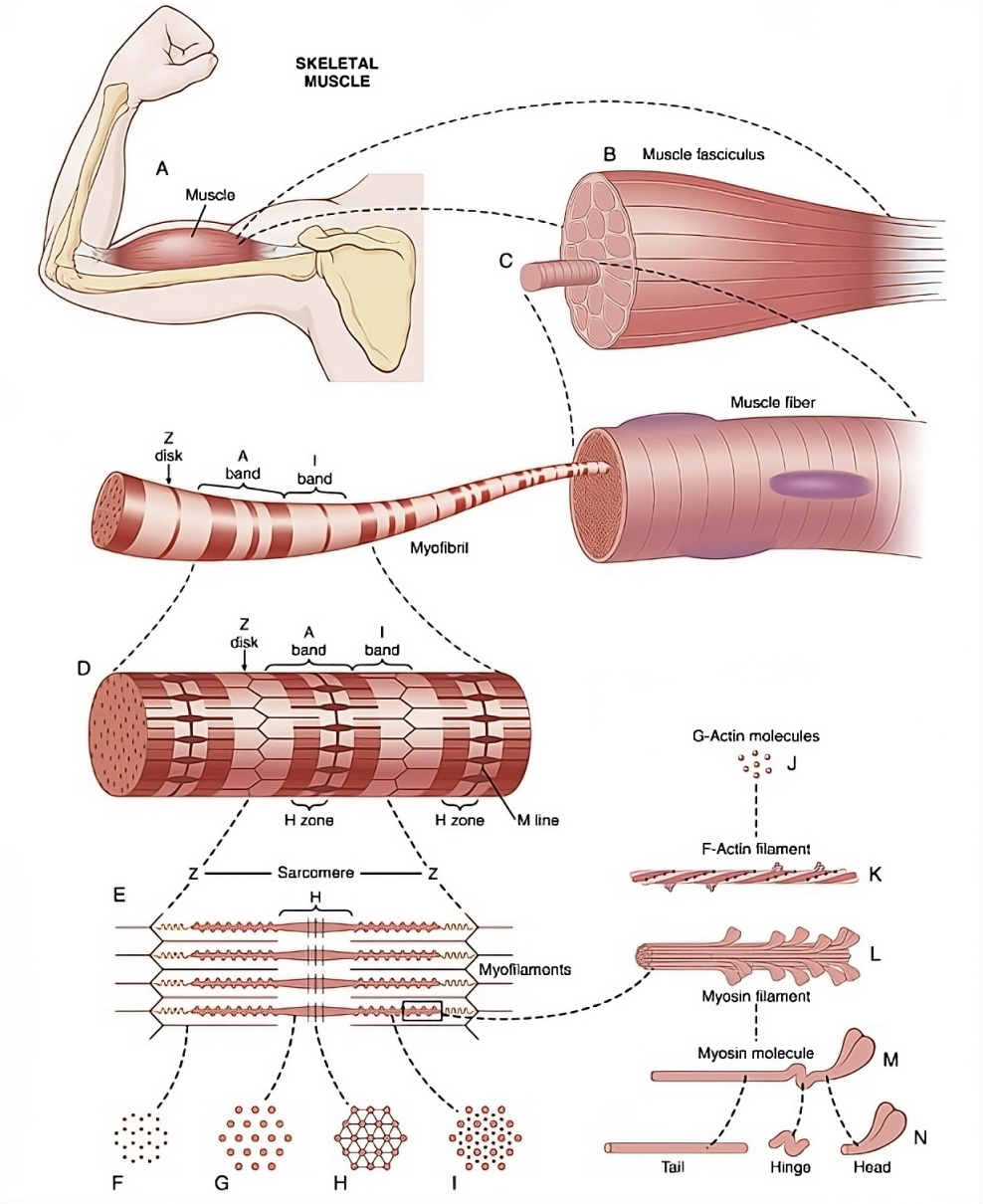


Figure 2.1: Organization of skeletal muscle from the macroscopic to the molecular level [9].

An important point to be emphasized is that there are other classifications in the organization of the muscle, both at the macroscopic level, such as connective tissues and blood vessels, and at the microscopic level, such as the molecules that compose the filaments, but for the applications of this work it is not necessary to study these topics in depth.

Therefore, based on the above, Proceeding with the analysis of the contraction movement of the skeletal muscle, which shortens during contraction, because the myosin and actin filaments slide over each other, causing this shrinking.

As mentioned earlier, skeletal muscle tissue functions primarily in a voluntary manner, and thus can be consciously controlled by neurons. The neurons that stimulate the muscle fibers are called somatic motor neurons, each of which has a filiform axon extending from the brain or spinal cord to a group of muscle fibers.

The action potentials, which as said before are the electrical signals that arise in response to stimulation, emerge at the neuro-muscular junction, which is nothing more than the synapse between the somatic motor neuron and the muscle fiber. The synapse is the region where communication occurs, and most of the time has a small gap, called synaptic cleft, which separates the cells, in this sense as the cells do not touch physically, and the action potential is not able to cross this gap communication is done by the release of a chemical messenger called neurotransmitter.

The action potential is invoked as follows, with the arrival of the nerve impulse the voltage-dependent channels are stimulated to open. Since calcium ions are more concentrated in this region, Ca^{++} flows through the open channels, and in turn stimulates the synaptic vesicles, which as a result fuse with the plasma membrane of the motor neuron and release the substance Acetilcolina (ACh) into the synaptic cleft, this substance diffuses through the synaptic cleft between the motor neuron and the motor plate. With the binding of two molecules of ACh to the receptor on the motor plate an ion channel opens in the receptor, and once this channel is open, small cations, especially Sodium cation (Na^{+}) cations can flow from the membrane to the motor neuron.

Then with that of Na^{+} the interior of the muscle fiber becomes more positively charged,

with this change in potential, the membrane triggers a muscle action potential. Each nerve impulse normally evokes a muscle action potential, which in turn propagates through the sarcoplasm to the T-tubule system, causing the sarcoplasmic reticulum to release its Ca^{++} stored in the sarcoplasm and the muscle fiber subsequently contracts [9]. This process can be seen through 2.2 of this paper.

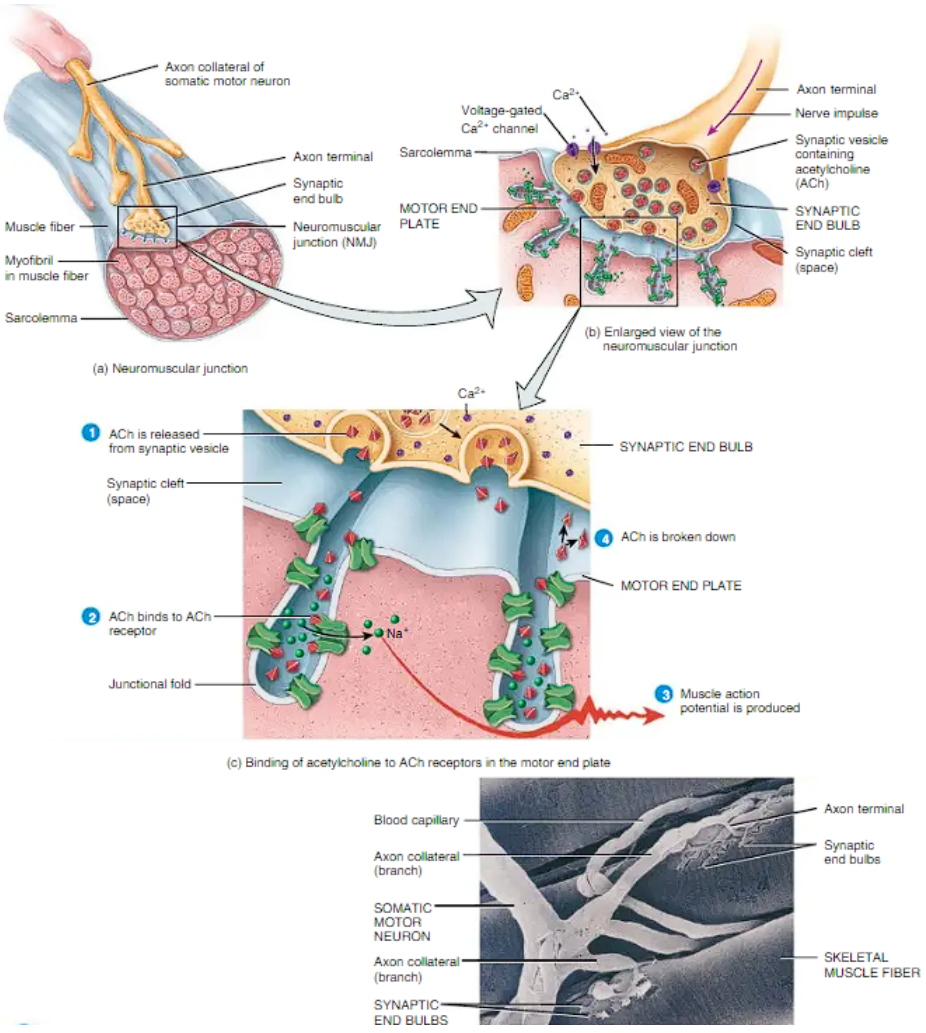


Figure 2.2: Emergence of the nerve action potential [10].

In general, the neuromuscular junction is close to the midpoint of the skeletal muscle fiber, and the action potentials propagate towards both ends of the fiber, this distribution makes possible the practically simultaneous activation, and consequently the contraction of all parts of the muscle fiber.

In summary, it can be defined that the EMG signals analyzed in this project are generated during muscle contractions by action potentials. These signals can be measured using electromyography, which is widely used in studies of muscle diseases and in the evaluation of muscle performance during physical exercises or other activities, such as the research presented here, which aims to apply the EMG signal in the development of assistive technologies.

2.1.2 Electromyography

Electromyography is a technique for recording and monitoring the electrical activity of muscles, resulting from the summation of action potentials generated in muscle fibers during their contraction [11].

The EMG signal is typified as a temporal function, delineated by its amplitude, frequency, and phase. In electromyography, there are both invasive techniques, which involve the insertion of needles or wires directly into the muscle, and non-invasive methods, where electrodes are placed in contact atop the skin to capture the signal.

This signal is inherently intricate, being modulated by the nervous system and the anatomical and physiological properties of muscles. As it traverses various tissues, the EMG signal is susceptible to noise interference. Moreover, EMG detectors, particularly those positioned on the skin's surface, simultaneously capture signals from multiple motor units, leading to the amalgamation of distinct signals. Hence, the advanced methodologies for EMG signal detection are paramount in biomedical engineering.

Upon detecting and recording the EMG signal, two primary concerns arise: the signal-to-noise ratio and signal distortion. The former pertains to the proportion between the EMG signal's energy and that of the noise, while the latter signifies alterations in the relative contribution of each frequency component in the EMG signal. For muscular signal acquisition, both invasive and non-invasive electrodes are employed. Surface electrodes capture a composite signal of all action potentials from muscle fibers beneath the skin, whereas needle or wire electrodes, directly inserted into the muscle, record individual

muscle fiber action potentials.

Post electrode capture, the signal undergoes amplification, typically by a differential amplifier. Subsequent amplification stages may ensue. Prior to display or storage, the signal might undergo processing to eliminate low or high-frequency noises or other artifacts. Often, the signal's amplitude is of primary interest, leading to rectification and computation of the signal to indicate the EMG amplitude.

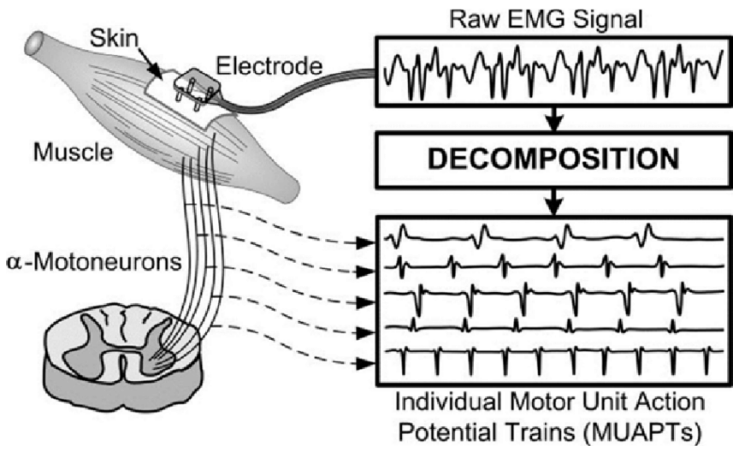


Figure 2.3: Acquisition process of the EMG signal and its decomposition to obtain the MUAPs [12].

EMG signals are the superimposition of multiple motor unit activities, as demonstrated by Figure 2.3. The amplitude of the EMG signal ranges from 0 to 10 mV (+5 to -5) pre-amplification. The EMG signal accrues noise as it traverses different tissues. It is imperative to understand the characteristics of electrical noise, which can be categorized as follows:

1. Equipment inherent noise: Inevitable, can only be minimized with high-quality components.

2. Ambient noise: Stemming from electromagnetic radiation, it may have an amplitude significantly larger than the EMG signal.

3. Movement artifact: Caused by data irregularities due to the electrode interface or electrode cable.

4. Signal inherent instability: The EMG's random amplitude is influenced by the firing rate of motor units.

Factors affecting the EMG signal can be classified into three categories:

1. Causal factors: Direct effects on signals, subdivided into extrinsic and intrinsic.

2. Intermediate factors: Physical and physiological phenomena influenced by causal factors.

3. Deterministic factors: Influenced by Intermediate factors.

To maximize the quality of the EMG signal, it's essential to maintain a high signal-to-noise ratio and minimize signal distortion. During processing, only positive values are analyzed, with a preference for full-wave rectification [13].

2.2 Signal filters: definition and types

In the present context, the effects of filters on signal processing will be explored, as they play a role in modifying the spectral content of these signals. The various types of filters employed in this research will be discussed, with the purpose of describing their characteristics and specific applications. The main objective is to provide an in-depth understanding of these filters, thus establishing a solid foundation for comprehending the present study.

- **Bandpass Filter** A bandpass filter allows a specific range of frequencies to pass while attenuating or rejecting frequencies outside that range. It can be created by combining a highpass and a lowpass filter [14], [15].

  The gain of a bandpass filter is ideally maximum within the passband and minimum outside of it, and the attenuation rate is determined by the design requirements [14]–[16].

Designing a bandpass filter involves determining the center frequency, the width of the passband, and the attenuation rate for frequencies outside the passband. There are different configurations of bandpass filters, such as Butterworth and Chebyshev, each with its own frequency response and implementation characteristics [14]–[16].

- **Kalman Filter** The Kalman Filter is a mathematical estimator used to solve the linear-quadratic problem by estimating the state of a linear dynamic system affected by random noise. It provides optimal statistical estimation of the state based on measurements that are linearly related to the state but corrupted by white noise [17].

  From a practical perspective, the Kalman Filter has found wide applications in controlling complex dynamic systems such as manufacturing processes, aircraft, ships, and spacecraft. It allows for inferring missing information from indirect and noisy measurements, enabling a better understanding of system behavior. Furthermore, the Kalman Filter is useful for predicting future behavior in uncontrollable dynamic systems like river flows, celestial body trajectories, or commodity prices [17].

- **Moving Average Filter** Moving Average Filters (MAFs) are cost-effective and easy to implement linear-phase FIR filters. They operate by continuously calculating the average value of input signals within a sliding time window T_W. This helps reduce undesired harmonics and distortion. MAFs exhibit a low-pass filtering behavior, providing unity gain at zero frequency and attenuating frequencies at regular intervals determined by the window size. They are commonly used as ideal low-pass filters under specific conditions [18], [19].

  Due its characteristics, the MAF suffers a response delay on T_W. Typically, a smaller T_W means a shorter MAF response delay; however, the filtering performance is also dependent on T_W [19].

- **Notch Filter** A notch filter is a linear filter that selectively eliminates a specific frequency component, known as the notch frequency, while maintaining unit gain at other frequencies. These filters are widely used in various applications where

targeted frequency components need to be removed [15].

However, FIR notch filters often have a wide bandwidth, resulting in significant attenuation of nearby frequencies. To address this, poles can be introduced into the filter to create resonance and reduce the notch bandwidth. However, this may introduce a slight ripple in the passband, which can be minimized by adding more poles and/or zeros. Nevertheless, this approach is considered ad hoc and relies on trial-and-error, presenting a notable challenge [15].

2.3 Machine Learning

In the context of machine learning, comprehending the intricacies of data-driven algorithms is fundamental for a full grasp of this research. This section aims to elucidate the fundamental principles of computational learning, detailing the methodologies and nuances of algorithmic design, as well as data interpretation. These elements are essential for a comprehensive understanding of the research.

Machine learning is a dynamic and multifaceted field, at the heart of significant advances in technology and data analysis. This field is distinguished by its diversity of systems, which include supervised, unsupervised, reinforcement, and others, each providing unique applications and specific approaches. It is highly valued for its ability to simplify complex problems, adapt to constantly changing environments, and extract valuable insights from large volumes of data, particularly through data mining. Ranging from simple algorithms like linear and logistic regressions to more advanced techniques such as neural networks and deep learning, machine learning finds applications in a wide range of fields, including image recognition and natural language processing [20] [21] [22].

Furthermore, machine learning is crucial in pattern recognition, relying on probabilistic models and decision theory. A deep understanding of probabilities, decision theory, and information theory is essential for the effective application of machine learning techniques, especially in tasks of pattern recognition. This ability to tackle complex problems where traditional methods are not efficient is one of the highlights of machine learning, along with

its versatility in classification, regression, and outlier detection. Its increasing integration into various technological areas demonstrates its importance and practical applicability, highlighting how it adapts and solves complex tasks in different technological contexts [20] [21] [22].

For Machine Learning concepts to be applied, a few core elements are required. First of all, is needed to have a relevant amount of data. This data is used to train the model and needs to be representative and contain meaningful information for the problem. The preparation and processing of data are also extremely important steps in building a good machine learning model.

The learning techniques of a machine learning model can be divided into three types:

- **Supervised Method:** The supervised method in machine learning is characterized by the use of training data that include both the input vectors and the corresponding target vectors. Each example in the training dataset consists of an input and output pair, where the output is already known. This method is used in classification and regression tasks, being effective in making accurate predictions with well-labeled and representative data. Classification involves categorizing input vectors into discrete classes, such as identifying emails as 'spam' or 'not spam'. Regression focuses on predicting a continuous variable based on inputs, like the price of a house from its characteristics [21].

- **Unsupervised Method:** The unsupervised method works with training data composed only of input vectors, without target vectors. It seeks to explore the intrinsic structure and patterns of the data without guidance from predefined outputs. It is applied in clustering, to discover natural groups in the data, like segmenting customers by purchasing preferences, and in density estimation, useful in tasks such as anomaly detection. This method is valuable when labeled data are scarce or for exploratory insights into the data [21].

- **Reinforcement:** Is a dynamic method to learn optimal actions through trial and

error in an interactive environment. Unlike other approaches that rely on pre-existing labeled data, RL operates in an environment where the agent learns by taking actions and observing the resulting feedback or rewards. This feedback serves as a signal for the algorithm to adjust its decision-making process and maximize its long-term objective, known as the reward function [23].

2.3.1 Machine learning methods

Some examples of methods are:

Decision tree

Decision tree emerged as versatile tools for prediction and classification, and were one of the pioneering statistical algorithms to be implemented in electronic form during the widespread adoption of digital circuits for computational purposes in the last decades of the 20th century. Over time, they have evolved into highly adaptive and computationally intensive methods that find applications in a variety of disciplines. Decision trees now serve as general-purpose mechanisms for prediction and classification, as well as indispensable components in artificial intelligence, machine learning, knowledge discovery, and inductive rule construction. Their use extends to a diverse range of tasks in data mining, knowledge discovery, machine learning, and artificial intelligence [24].

A key feature of decision trees lies in their recursive subset approach, employed to analyze a target data field based on the values of associated input fields or predictors. This methodology facilitates the creation of partitions within the data set, resulting in descending subsets of data called leaves or nodes. Notably, these leaves or nodes contain progressively similar intra-leaf (or intra-node) target values, while exhibiting progressively different inter-leaf (or inter-node) values at each level of the tree. This hierarchical organization allows decision trees to effectively discern patterns and relationships within the data, enabling robust predictive and classification capabilities. This unique feature, combined with the computational intensity and interdisciplinary applicability of decision

trees, contributes to their broad utility in a variety of research and practice domains [24].

Random forest

Random forest is an extension of decision tree models and consists of creating multiple individual decision trees through a process called "ensemble learning". Each tree in a random forest is built and trained using variables or a set of random information in the dataset, allowing each tree to learn different characteristics of the data set decision trees [25].

When a prediction is required, after the model is trained, each tree performs its individual prediction and then a final decision is made by the model. This collective process reduces overfitting and consequently improves the generalizability of the model.

One of the main advantages of Random Forest is their ability to handle large data sets while maintaining robustness, since in a sense the problem is divided into several decision trees. Furthermore, they are less sensitive to outliers and noise when compared to individual decision trees [26].

Support vector machine - SVM

Support Vector Machine (SVM) is a widely used machine learning algorithm for classification and regression. Its main goal is to find the optimal hyperplane that separates the input data into distinct classes, seeking the largest margin between the closest training samples. Such samples are called support vectors and play a crucial role in the definition of the hyperplane [27].

For non-linearly separable data sets, mapping tricks are employed to project them into a higher dimensional space. In this space, linear separation through a hyperplane becomes possible [27].

This machine learning technique has good generalization capabilities, since the optimization process is based on margin maximization, which tends to avoid unwanted overfittings. The possibility of application to non-linearly separable problems makes SVMs

extremely flexible and able to handle a wide variety of classification and regression problems [27].

However, the training process of an SVM can become computationally demanding for large data sets, so it is necessary to evaluate the size of the data set when using the SVM as a learning algorithm [28].

2.3.2 Clustering

Clustering is a technique used to group similar objects or data points based on their intrinsic characteristics, aiming to discover patterns, structures, or relationships within a dataset without prior knowledge of the specific groups or classes [29].

The challenge lies in dividing the data into groups in a way that maximizes their similarity. Clustering has been extensively studied due to its diverse applications in data mining and machine learning, including summarization, segmentation, and targeted marketing [30].

There is a wide variety of problems that can be solved with clustering algorithms, such as Collaborative Filtering, Data Summarization, Biological Data Analysis, and Social Network Analysis [30].

K-means

The K-means is a clustering algorithm used to divide N objects into K groups. Initially, the centroids of the groups are selected, which can be random or strategically chosen. Each object is then assigned to the nearest centroid based on Euclidean distance. The process iterates, recalculating the centroids as the average of the objects in each group until there is little or no change in the centroids or until a limit of iterations is reached. K-means is effective and simple, ideal for multi-dimensional data analysis, but it has limitations such as the need to define the number of groups beforehand and sensitivity to the initial choice of centroids, which can produce variable results. It works best with spherical groups of similar size. [31]

2.3.3 Confusion matrix

A confusion matrix is a fundamental tool for evaluating classification algorithms in machine learning. It is a two-dimensional matrix where rows represent true labels (or actual classes) and columns represent labels predicted by the classifier. For each class, the confusion matrix shows the number of correct predictions (true positives, TP) and incorrect predictions (false positives, FP, false negatives, FN, and true negatives, TN). [32]

Let's use a simple example to illustrate how a confusion matrix is calculated:

Suppose we have a classification problem with three classes (A, B, C) and a test dataset with the following predictions and true labels:

Instance	True Label	Prediction
1	A	A
2	B	C
3	C	B
4	A	A
5	C	C

Table 2.1: Test dataset with predictions and true labels

The confusion matrix for this example would be:

Confusion	Predicted A	Predicted B	Predicted C
Actual A	2 (TP A)	0 (FN A)	0 (FN A)
Actual B	0 (FP B)	0 (TP B)	1 (FN B)
Actual C	0 (FP C)	1 (FN C)	1 (TP C)

Table 2.2: Confusion matrix

- True Positives (TP): The number of instances correctly classified as a specific class.
- False Positives (FP): The number of instances incorrectly classified as a specific class but actually belong to another class.
- False Negatives (FN): The number of instances that belong to a specific class but were incorrectly classified as another class.

- True Negatives (TN): The number of instances that do not belong to a specific class and were correctly classified as not belonging to that class.

The confusion matrix is a powerful tool for understanding how a classifier is performing for each individual class and also for the dataset as a whole. It provides valuable insights into classification errors, allowing developers to adjust their models to improve performance on specific classes. In the above example, for instance, it is evident that the classifier struggles to correctly classify classes B and C, indicating an area for improvement.

Performance measures

In the field of multi-label classification, evaluating the performance of models is essential and is primarily conducted using three metrics: accuracy, precision, and recall. These metrics provide valuable insights into different aspects of model performance, including its ability to correctly identify relevant labels (recall), the accuracy in predicting labels (precision), and the overall effectiveness in predicting correct labels (accuracy) [33].

- **Accuracy**

  Accuracy is calculated by considering the proportion of correctly predicted labels relative to the total number of labels. For each data instance, accuracy is derived from the intersection of the sets of true and predicted labels divided by the union of these sets. The average of these values for all instances constitutes the model's overall accuracy. The formula for accuracy is expressed as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{2.1}$$

  Here, N represents the number of instances, Y_i the set of true labels, and Z_i the set of predicted labels for the i-th instance [33].

- **Precision**

  Precision measures the proportion of correctly predicted labels relative to the total

number of predicted labels. For each data instance, it is calculated by the intersection of the sets of true and predicted labels divided by the number of predicted labels. The overall precision is the average of these values for all instances. Its formula is given by:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{2.2}$$

In this formula, $|Y_i \cap Z_i|$ indicates the number of correct labels in the i-th instance, while $|Z_i|$ represents the total number of predicted labels in the same instance [33].

- **Recall**

  Recall assesses the proportion of true labels that have been correctly predicted. For each data instance, it is the intersection of the sets of true and predicted labels divided by the number of true labels. The overall recall is the average of these values for all instances. The formula for recall is:

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{2.3}$$

  Here, $|Y_i \cap Z_i|$ represents the number of correct labels in the i-th instance, and $|Y_i|$ the total number of true labels in the same instance [33].

These metrics are crucial for understanding and quantitatively assessing the performance of classifiers in multi-label scenarios, allowing a detailed analysis of a model's accuracy, precision, and recall.

2.4 Identification of the Bracelet Movement

This section discusses and elucidates the technologies employed to detect the movement of the bracelet in space.

2.4.1 Analysis of Gyroscope and Accelerometer Devices

Accelerometer

The accelerometer, an automatic instrument essential for measuring accelerations, is capable of identifying and quantifying vibrations and accelerations experienced by an object, such as inclinations. Historically, accelerometers have been widely used in various applications. They are fundamental in the analysis of vibrations in vehicles, machinery, architectural structures, and security systems. Moreover, their applicability extends to the realm of electronic devices, being employed in three-dimensional games, computer pointing devices, and mobile telephony. These instruments also play a crucial role in monitoring seismic activities and in multimedia applications, such as in Video on Demand (VOD) with three-dimensional features [34] [35].

The basic operation of a Micro-Electro-Mechanical Systems (MEMS) accelerometer, which integrates mechanical and electrical components on a microscopic scale, is elucidated by Newton's second law. In this system, a mass coupled to a spring is positioned within a frame of reference. MEMS accelerometers function based on two principles: the measurement of the mass displacement and the quantification of the frequency of a vibratory element, whose variations are indicative of changes in tension [34] [35].

Gyroscope

The gyroscope is a device affixed to a structure, capable of determining the angular velocity of that structure during its rotation. There are several categories of gyroscopes, which are differentiated based on their operational physics and incorporated technology. These devices can operate autonomously or integrated into more elaborate systems, such as the Inertial Measurement Unit (IMU), gyrocompass, attitude and direction reference systems, as well as navigation systems. Figure 2.4 illustrates the rotation axes of a gyroscope [34] [35].

The operational principle of the MEMS gyroscope is based on the Coriolis effect. According to this effect, in a rotating reference frame with a certain angular velocity, a

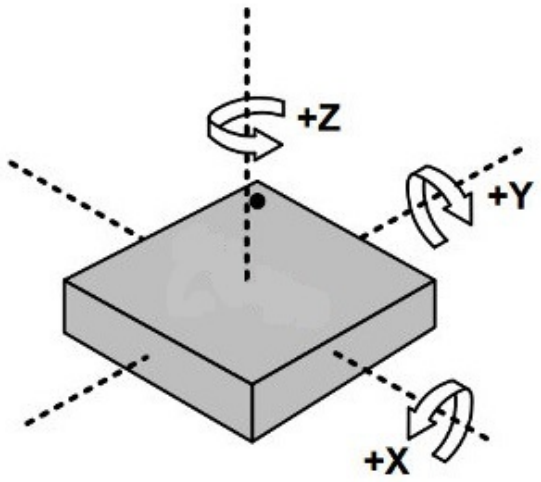


Figure 2.4: Rotation axes of a gyroscope [36].

moving mass at a given speed will be subject to a force. To quantify the Coriolis effect, the MEMS gyroscope has vibratory masses that oscillate along a drive axis. A secondary oscillation is induced in the perpendicular sensory axis, resulting in the displacement of the mass from its original trajectory when the gyroscope is rotated [34] [35].

Applications and Implications

Accelerometers and gyroscopes have been extensively employed in various areas, including navigation, aerospace, industrial robotics, and automotive. Recently, their applicability has been expanded to smartphones, providing an innovative interface for motion detection and gestures in smart devices. A deep understanding of the behavior of MEMS and the intrinsic characteristics of accelerometers and gyroscopes allows engineers and designers to develop optimized, efficient, and economical products for large-scale applications [34] [35].

2.4.2 Euler rotation

According to Euler's Rotation Theorem, any rotation in three-dimensional space can be described using three distinct angles. These angles, known as Euler angles, are used to

represent three separate rotation matrices. The overall rotation can then be expressed as the product of these matrices [37].

Fundamentally, this theorem allows for any rotation or orientation of a rigid body in three-dimensional space to be characterized by a sequence of three rotations. These rotations occur successively around three fixed orthogonal axes, specifically: 1. First Rotation (ϕ) (Yaw): Rotation around the z-axis. 2. Second Rotation (θ) (Pitch): Rotation around the y-axis.

For a clearer visual representation of these rotations, one can refer to Figure 2.5. In this illustration, the first rotation is indicated in green, the second in red, and the third in blue.

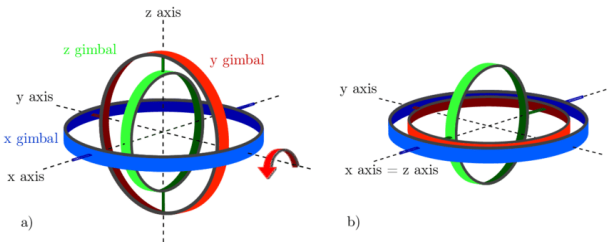


Figure 2.5: Representation of the rotation of the euler angles (a) and the "gimbal lock" effect (b) [38].

Yaw, Roll, and Pitch angles are crucial in determining and modulating the orientation of vehicles, including aircraft, ships, and autonomous cars. To illustrate:

- In aviation, meticulous control of these angles is essential for executing aerial maneuvers, maintaining flight stability, and ensuring safe landings.
- For ships, understanding and adjusting rotation and tilt are fundamental to maintaining stability in rough sea conditions.
- In vehicles, accurately perceiving the vehicle's orientation in relation to the surrounding environment is vital for safe navigation and precise maneuvering. The referenced article highlights the importance of quaternions in robotics, where they are used to represent the orientation of robots and drones. The utility of quaternions is particularly evident in controlling the orientation of mobile entities in a

three-dimensional context [39].

Gimbal Lock

Gimbal lock is a phenomenon that occurs in angular coordinate systems, commonly found in devices such as gyroscopes and accelerometers, essential for navigation and orientation. This effect is particularly notable in systems that use three gimbals (ring-shaped mechanisms) to allow rotation along three axes, a typical arrangement in many gyroscopes. 'Gimbal lock' occurs when two of the three axes of rotation align, resulting in the loss of one degree of freedom. This means that the system becomes incapable of differentiating rotation around two of its axes, as illustrated in Figure 2.5 'b'. In this situation, the system loses its accuracy in measuring rotation on one of the axes, which can lead to significant errors in navigation or orientation, as discussed in [40]. This phenomenon represents a singularity in the parametrization of three-dimensional rotations and is a technical challenge in various engineering applications, including navigation systems.

A practical example of this effect can be observed in airplanes that use gyroscopic-based navigation systems. When the airplane performs a 'pitch' movement (tilting up or down) at a 90-degree angle, the 'roll' and 'yaw' axes may align. This results in 'gimbal lock', causing the system to no longer differentiate between roll and yaw, thereby compromising the aircraft's ability to navigate accurately.

2.4.3 An Examination of Quaternions

Sensory devices, such as accelerometers and gyroscopes, possess the capability to provide data in various formats, which are determined based on the specificity of the device and the nature of the application at hand. One such representation is termed "quaternion" [41].

Quaternions, an extension of complex numbers, are extensively employed in domains like computer graphics, robotics, and navigation systems, serving as tools to represent rotations and orientations in three-dimensional space. One of their most notable advantages

is the ability to depict rotations without incurring the issue of "gimbal lock", a constraint associated with Euler angles [41].

In the context of Inertial Measurement Units (IMUs) and sensor fusion systems, quaternions play a pivotal role, often being used to integrate data from accelerometers, gyroscopes, and magnetometers. This integration enables the calculation of a device's orientation relative to a global reference system [41].

Thus, it is imperative to recognize that while accelerometers and gyroscopes in isolation quantify linear acceleration and angular velocity, respectively, the combination and processing of this data in more sophisticated systems allow the resulting orientation to be represented in the form of a quaternion [41].

2.4.4 Conversion of Quaternions to Yaw, Roll, and Pitch

As previously explained, quaternions represent a robust mathematical approach to representing orientations and rotations in three-dimensional space. They extend complex numbers and are distinguished by four unique components. One of the most significant applications of quaternions is the accurate representation of spatial rotations, addressing challenges such as "gimbal lock" [42]. A quaternion consists of four components: one scalar and three vector components, generally represented as $q = (w, x, y, z)$. The conversion of a quaternion to Euler angles (yaw, pitch, roll) can be performed using the standard formulas:

1. **Roll (rotation around the X-axis)**:

$$\text{roll} = \arctan 2(2(wx + yz), 1 - 2(x^2 + y^2)) \tag{2.4}$$

2. **Pitch (rotation around the Y-axis)**:

$$\text{pitch} = \arcsin(2(wy - zx)) \tag{2.5}$$

3. **Yaw (rotation around the Z-axis)**:

$$\text{yaw} = \arctan 2(2(wz + xy), 1 - 2(y^2 + z^2)) \tag{2.6}$$

In the equations presented above, we encounter specific trigonometric functions that are crucial for the conversion process: arctan2 and arcsin. Arctan2, or arc tangent, is a variant of the inverse tangent function that takes into account the signs of both arguments to provide the angle in the correct quadrant. It is particularly useful in converting Cartesian coordinates to polar coordinates, which is essential in calculating the roll and yaw. On the other hand, arcsin, or arc sine, is the inverse sine function, pivotal in determining the pitch. This function returns the angle whose sine is the given number, thus playing a key role in translating the quaternion's components into the pitch angle in Euler form.

These formulas are based on the mathematical relationships between quaternions and Euler angles, as described in [43] and in [44]. They provide an intuitive way to convert quaternion rotations into Euler angles, despite the susceptibility to Gimbal Lock. It is important to note that this conversion may not be unique, especially when the pitch approaches $\pm\frac{\pi}{2}$ (90 degrees), which can lead to ambiguities in roll and yaw values. The feasibility of this conversion can also be seen in the article [42], which emphasizes the resistance of quaternions against numerical errors compared to other representations, such as rotation matrices. Furthermore, they are predominantly applied in aeronautical simulations due to their accuracy and efficiency.

2.5 Related Work

This section presents a selection of recent research that is directly relevant to the proposed project. These studies have been carefully grouped by topics to clearly illustrate how each relates to and contributes to the objectives and methodologies of our project.

2.5.1 Inclusion and Accessibility in Rehabilitation Technology

The study of accessible rehabilitation technology has progressed significantly, focusing on addressing the rising prevalence of disabling conditions globally [45]. Rehabilitation improves function and quality of life in many conditions, especially when applied intensively. However, limited workforce often fails to deliver evidence-based intensive rehabilitation. Technology offers a solution by providing self-rehabilitation tools, bridging the gap between evidence and practice. Yet, many face barriers accessing rehabilitation technology due to costs, training, education, portability, and inadequate design. To address this, a center was established to develop accessible technology through a co-creation model, involving engineers, researchers, users, and industry [45]. This center employs a co-creation model, combining engineering and science with user experience and industrial partnerships. The initial focus is on stroke, with participants being recruited through a medical charity to participate in an 8-week rehabilitation program supervised by healthcare professionals [46].

The study on access to assistive technology for people with disabilities has also advanced significantly, focusing on countries like Nepal, India, and Bangladesh [46]. The research revealed that AT service provisions are underdeveloped in these countries. AT users have limited awareness of their rights to these services and the availability of AT services. Lack of accessibility, eligibility, reach, and affordability are major barriers to accessing AT services for PWD in these countries. The research concludes that increasing community-level awareness, increasing government funding, and a flexible community-based AT service model are ways forward to ensure access to AT services for PWD in these countries [46].

Furthermore, the provision of assistive technology has seen significant advancements, focusing on an international framework to ensure the availability and accessibility of high-quality and affordable assistive technology [47]. The United Nations Convention on the Rights of Persons with Disabilities (CRPD) created an international legal obligation for countries, committing ratifying states to implement measures to facilitate access to AT

solutions for those in need. The paper emphasizes the importance of effective information systems to ensure people are aware of available AT solutions and their quality, usability, and effectiveness [47].

The study on assistive technologies (ATs) has also advanced significantly, highlighting their centrality in realizing the Convention on the Rights of Persons with Disabilities [48]. Assistive technologies promote participation and inclusion in society, supporting access to health, social services, education, work, and other vital experiences for people with disabilities, the elderly, and those with chronic conditions. The Global Report on Assistive Technology, released in May 2022 by WHO and UNICEF, calls for concrete actions to improve global access to AT and recognizes AT as both a means and an end in itself for realizing the rights of people with disabilities. The United Nations Convention on the Rights of Persons with Disabilities (UNCRPD, 2006) is the most widely ratified human rights convention, affirming the right to participate in society on an equal basis with others. This paper highlights examples of how AT can play a role in realizing each of the fundamental rights affirmed in the UNCRPD [48].

In conclusion, the state of the art in EMG signals, pattern recognition, and inclusion and accessibility in rehabilitation technology has shown significant advancements. These advancements have the potential to improve the quality of life for many individuals, especially those with disabilities. The integration of these technologies and methodologies can pave the way for more inclusive and accessible solutions in the future.

2.5.2 Electromyographic (EMG) Signals and Their Applications

Electromyographic (EMG) signals have been a focal point of research, with significant advancements in their applications across various domains. In the medical field, the study of biofeedback eletromiográfico (EMG) has seen substantial progress, especially in its applications for neurological rehabilitation and post-stroke treatment [49]. This advancement has enabled a deeper understanding of the motor function of the upper limb and pain relief in patients with post-stroke shoulder-hand syndrome [49]. Physical

therapy techniques, particularly when combined with EMG biofeedback and rehabilitation training, have been identified as potentially the most effective approach for treating these patients [49].

In addition to medical applications, EMG signals have found relevance in biomedical engineering, biomechanics, and neuroscience [50]. The field has witnessed improvements in the acquisition, processing, and interpretation of EMG signals, which has paved the way for a better understanding of muscle functions, diagnosis of neuromuscular disorders, and control of assistive devices [51]. Techniques such as decomposition into individual motor units, machine learning, and brain-machine interfaces have been explored to enhance the utility of EMG signals [52].

However, the integrity of EMG signals can be compromised by noise and artifacts. Efforts have been made to address these challenges. For instance, a study by [53] evaluated four filtering procedures and found the wavelet method to be superior in preserving information and ensuring accuracy.

Machine learning has also been integrated into EMG signal analysis. A study developed a control system for a robotic hand using an Artificial Neural Network (ANN), achieving an accuracy of 84.78% in identifying hand movements [54]. Another research utilized the Q-Tuned Wavelet Transform (TQWT) for feature extraction and applied various machine learning classifiers, with the Random Forest (RF) classifier achieving the highest success rate of 98.64% in classifying neuromuscular disorders [55].

Furthermore, EMG features have been explored for early detection of medical conditions, such as hand osteoarthritis [56]. In the realm of assistive technology, EMG signal processing combined with machine learning techniques has been employed to enhance prosthesis functionality, leading to increased accuracy and response speed [57]. Studies like [58] have also showcased significant advancements in motion estimation and EMG signal classification to further improve prosthesis functionality.

2.5.3 Pattern Recognition and Adaptive Classification

The study of adaptive incremental gesture classification for upper limb prostheses has seen significant advancements, particularly in myoelectric pattern recognition for controlling upper limb prostheses [59]. However, the non-stationary characteristics of EMG signals, caused by physiological changes (e.g., muscle fatigue) or non-physiological changes (e.g., electrode-skin impedance), pose challenges in prosthesis usage and degrade the performance of the myoelectric control system. In this context, a robust set of features has been proposed to be integrated with adaptive learning techniques to enhance myoelectric performance. The results indicate that using the proposed features with an adaptive classifier improves classification accuracy from 91.58% to 97.2% in online classification settings [59].

2.5.4 Machine Learning in Assistive Technology

The field of assistive technology has been experiencing significant advancements, particularly with the integration of machine learning to enhance the efficacy and personalization of devices. Recent studies have delved into the application of assistive physiotherapy devices (APT), which are essential in maintaining or improving the functionality and independence of individuals, especially in the face of a growing population of elderly and people with disabilities [60]. With technological evolution, components such as self-powered sensors and optimized machine learning algorithms are becoming pivotal for the provision of effective assistive services.

Furthermore, machine learning-aided assistive technology is being explored to provide solutions for people with dysarthria, a collective term for neuromotor speech disorders [60]. Current research is focused on developing a model based on Convolutional Neural Networks (CNNs) that can recognize specific keywords uttered by users with spastic dysarthria, with initial experiments showing promising results [61].

Another interdisciplinary approach combines ethnography, cognitive science, and machine learning to inform the development of context-sensitive personal computing technologies [62]. This "hybrid methodology" suggests that the assistive technology of the future will be more seamlessly integrated into people's daily lives and, to be truly effective, must be deeply aware of the human context.

Lastly, assistive vision, employing computer vision and machine learning, is advancing rapidly, especially in computer-mediated assistive vision systems [63]. One area of focus is the development of retinal implants that can partially restore vision for those with retinal dystrophies. Current research is exploring how deep reinforcement learning (DRL) can be used to learn visual filters that can be applied in prosthetic vision devices, allowing for richer, task-oriented scene representations [63].

Chapter 3

Methodology and Development

The purpose of this chapter is to present the research's methodology and its alignment with the project's goals. It offers insights into the rationale behind methodological choices and their expected advantages. The chapter also chronicles the development stages, challenges faced, and the solutions implemented. A comprehensive overview of the tools and technologies used is provided, emphasizing their significance in the research context.

3.1 Myoelectric Signal Capture

This section describes the methods and procedures used in this study to collect and analyze the data. It provides a clear overview of how the research was conducted.

3.1.1 Data Acquisition

The Myo Gesture Armband, an innovative product crafted by THALMCLABS, stands as a wearable technology designed to capture electromyographic (EMG) signals from the forearm muscles. Beyond its EMG capabilities, the armband is also outfitted with accelerometers and gyroscopes, providing a comprehensive perspective on arm movement and orientation.

With its 8 dedicated channels, the Myo Gesture Armband can pick up signals from

a variety of forearm muscles, offering a deep insight into muscle activities during diverse movements.

Although the Myo Gesture Armband is no longer commercially available, its selection for this project was strategic. The armband's design simplifies data acquisition, allowing the research to lean more towards data analysis and applications rather than getting entangled in hardware development and adjustments.

As depicted in Figure 3.1, the design and placement of the Myo Gesture Armband on the forearm can be visually observed. This positioning ensures optimal data capture, highlighting the specific characteristics of muscle activity in different movements.

For the scope of this project, our main emphasis will be on data from electromyographic sensors and combined data from accelerometers and gyroscopes. This targeted approach ensures that the most pertinent and critical data is harnessed for training and analysis, improving the results of the machine learning methodologies in question.

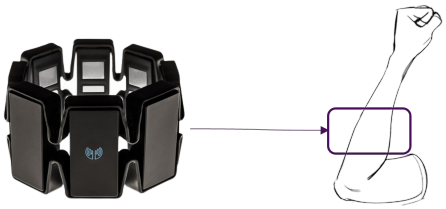


Figure 3.1: Myo Gesture Armband and its position on the forearm.

The primary objective of this phase was to build a robust database tailored for the training of previously introduced machine learning techniques. Initially, data were collected from a group of 12 volunteers, consisting of 5 women and 7 men, all aged between 20 and 30 years. This selection ensured they were in good health and without any amputations. Each participant had the Myo Gesture Armband securely positioned on their right forearm.

During the initial data collection process, 60 distinct files were recorded for each volunteer. Of these files, 30 recordings captured 'Movement 0', which involves keeping the hand at rest. These recordings were evenly divided among positions '1', '2', and '3',

as illustrated in the first column of Figure 3.2. In the other 30 recordings, the volunteers performed 'Movement 1', involving closing and opening their hands, also varying the arm position every 10 recordings, as shown in the second column of Figure 3.2. 'Movement 1' included variations in duration and a greater focus on activating the forearm muscles when closing the hands, thereby intensifying the EMG signal. This aspect is highlighted in the analysis of Figure 3.9, which compares the signal from this initial database with a signal of the same movement, but executed more naturally, that is, without applying much force during the action. Each recording lasted 3 seconds, with stored samples ranging between 600 and 800.

A crucial aspect of data collection was the intentional variation of arm positions during recordings. As illustrated in Figure 3.2, three positions are observed: 'position 1', where the forearm is at a 90º angle relative to the rest of the body; 'position 2', with the arm at 90º from the body and the forearm at 90º from the arm; and 'position 3', where the entire arm is relaxed. This variation in positions was strategic, given the nature of surface EMG sensors, which are susceptible to noise from neighboring motor units. Capturing data in these different positions allowed for the intentional introduction of noise. The hypothesis guiding this methodology was that training machine learning models on data with these varied levels of noise would increase their adaptability and accuracy, enabling them to effectively discern genuine EMG signals, even in scenarios with unavoidable noise.

Subsequently, the collected data was saved in files, where each column represents an EMG channel, facilitating data processing. In Figure 3.3, it is possible to observe at the top in black the 8 channels during the execution of the movement. At the bottom of the Figure, in blue, are the same 8 channels during the absence of movement. These collected and organized data files provide the foundation for the subsequent steps of analysis and training of the machine learning algorithms.

Upon observing Figure 3.3, it is evident that the movement is characterized by signals with higher amplitudes. Specifically, when the volunteer performs the movement, the EMG signals exhibit larger amplitudes, whereas in the absence of movement, only small amplitudes are observed. Furthermore, it can be noted that, for the hand-closing

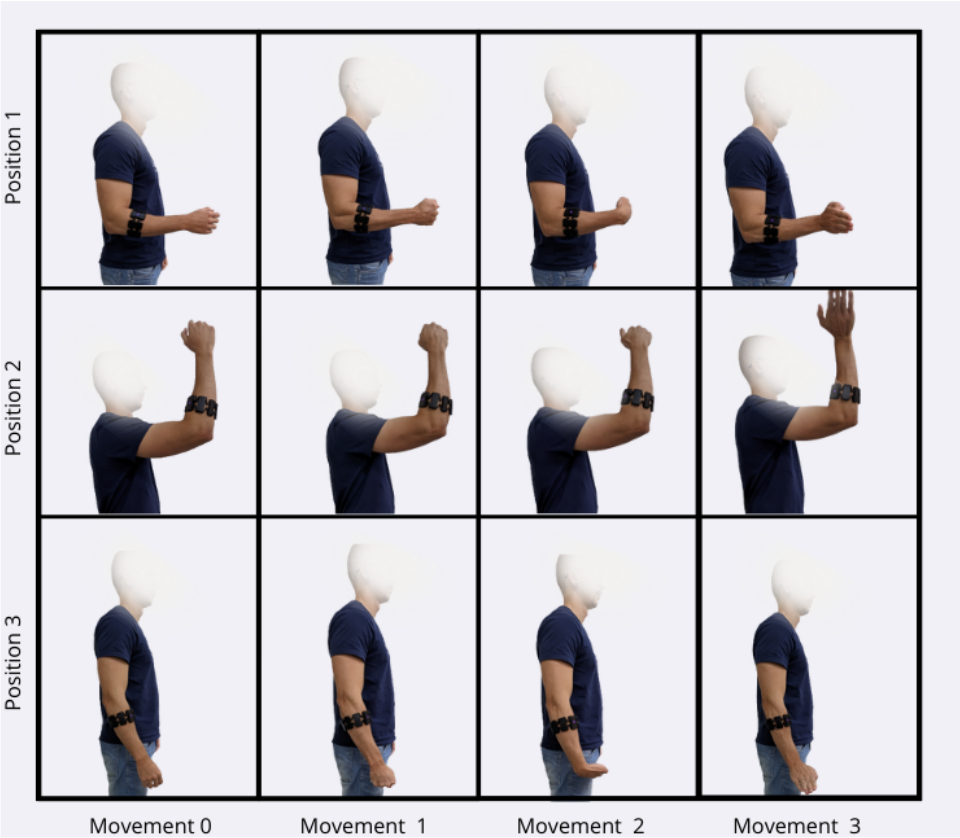


Figure 3.2: Movements and capture positions.

movement, the signal is not uniformly reproduced across all channels, indicating varied muscular activation intensities based on their respective contributions to the execution of the movement.

After conducting the initial tests with the first database, it became necessary to expand the range of movements to be classified. In this context, movements '2' and '3' were added. 'Movement 2' was defined as rotating the wrist to tilt the hand towards the body, as illustrated in the third column of Figure 3.2, while 'movement 3' involves rotating the wrist in the opposite direction, as shown in the fourth column of the same Figure.

This second database, therefore, comprises four distinct classes of movements. The construction of this new database followed the patterns established by the first, with the addition of the new movements. It is important to note that 'movements 0' and '1' were recorded again, given that the volunteers in this phase were different from the previous

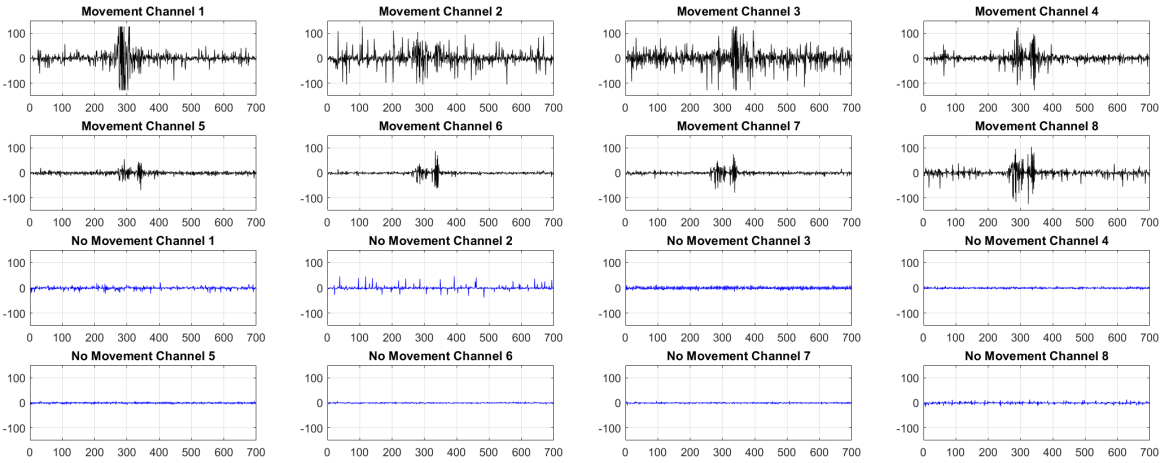


Figure 3.3: Comparison of 8-channel EMG signals during movement execution (Black) and absence of movement (Blue).

ones. Physiological variables such as skin thickness and fat content, as well as factors like movement intensity, can influence both the quality of the recording and the patterns of the EMG signals, including the signal amplitude. Thus, the decision was made to rebuild the database with 12 new volunteers, this time evenly split between 6 men and 6 women.

Another implemented change was instructing the volunteers not to exert too much force during movements. Considering that the final application of this classification is to use movement identification as a command for a system, performing movements with excessive intensity could become tiresome over time. Additionally, as the device is primarily intended for individuals who have suffered amputations, there is a possibility of atrophy in the stump area due to lack of muscle use, which could affect the amplitudes of the EMG signals, making them smaller. By creating the database with less intense movements, it is expected to achieve results more aligned with real-world use. In Figure 3.4, it is possible to compare the signals from the initial database with those from the new database. Observing that the signals from the first two lines correspond to 'Movement 0', or the absence of movement, while the last two lines represent the signals for 'Movement 1'. It can be noted that the signal in black, pertaining to the first database, has a higher amplitude compared to the signal in red, which corresponds to the movements recorded in the second database.

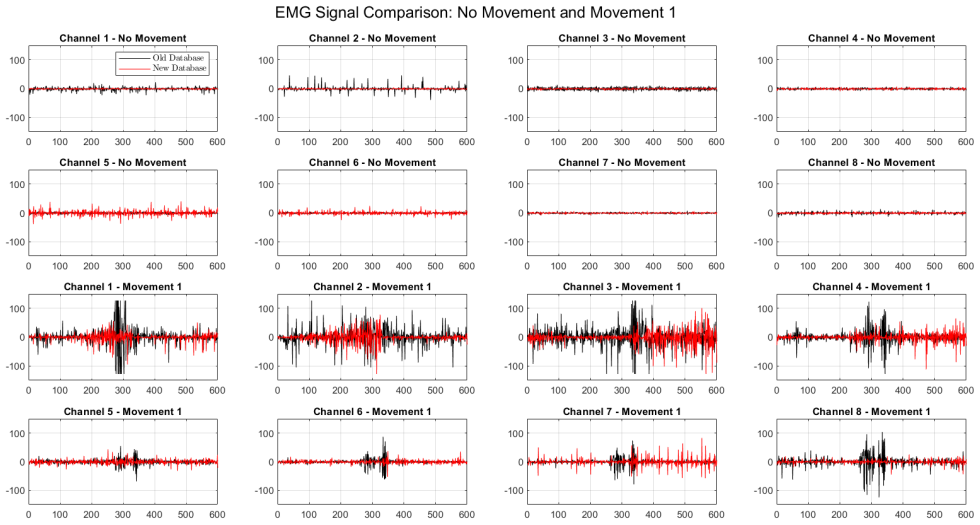


Figure 3.4: Comparison of Movement 0 (Upper part) and Movement 1(Lower part) of the initial database (Black) and the second version (Red).

In Figure 3.5, it is evident that the signals corresponding to movements 2 and 3, with movement 2 in blue and movement 3 in black. A notable difference between these two movements is the inversion of the channels that exhibit higher amplitude in the EMG signal. For instance, while in movement 2, channel 7 displays a more intense signal with higher amplitudes compared to channel 3, in movement 3, the most intense channel is channel 3. This demonstrates that different muscle groups are activated for each specific movement.

In summary, up to this point, we have a total of 360 distinct records for each movement. Considering that six movements were recorded, these are: Movement 0 with the Initial Database and the Second Database; Movement 1 also divided between the Initial Database and the Second Database; and Movement 2 and Movement 3, both belonging to the Second Database. This results in a total of 2,160 records, which form the foundation for the databases used in training machine learning models. Each record includes data from eight different channels, named from Channel_1 to Channel_8. A line in the file represents a set of readings from these channels at a specific moment. The values in each column correspond to the amplitudes of the myoelectric signals captured, reflecting the

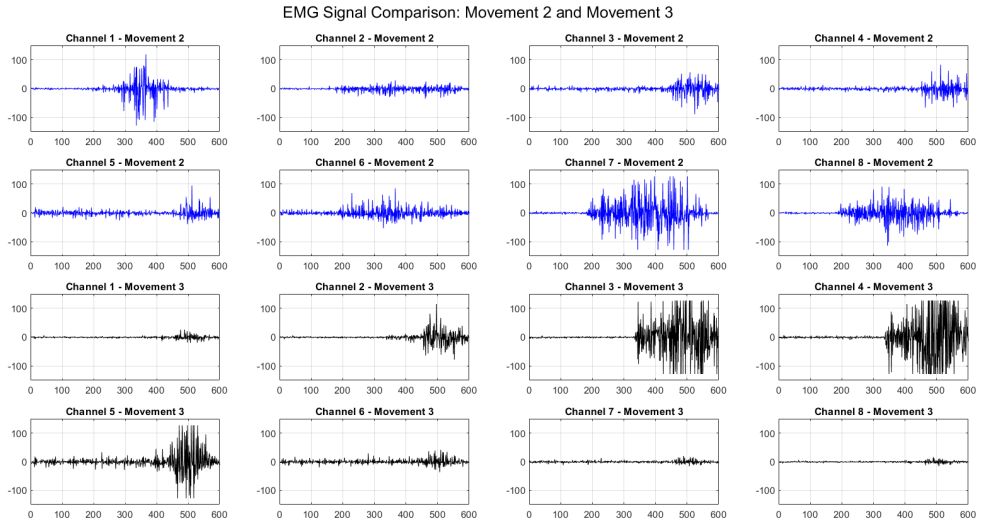


Figure 3.5: Comparison of Movement 2 (Blue) and Movement 3(Black).

muscle's electrical activity during the measurement.

3.1.2 Data Filtering

After properly storing the data, the next step was to process it. In order to gain a better understanding of the filters that operate in the frequency domain, the Fourier transform was applied tovisualize the frequencies present in the signal, as well as their magnitude. Figure 3.6 displays the applied transform for each channel, represented in black for the movement data and in blue for the non-movement data.

The first filter applied was the bandpass filter, with a passband ranging from 5 Hz to 500 Hz. This range was chosen based on the frequency range typically present in healthy skeletal muscle action potentials.

Next, with the cutoff frequencies defined, the bandpass filter was applied to all files in the database. The filter was applied to each column of the files, using the Fourier transform to perform the filtering. The result of this filtering can be observed in Figure 3.7, where the original signal of the first database is represented in black and the filtered signal is represented in red. For better visualization, only channel one is being presented.

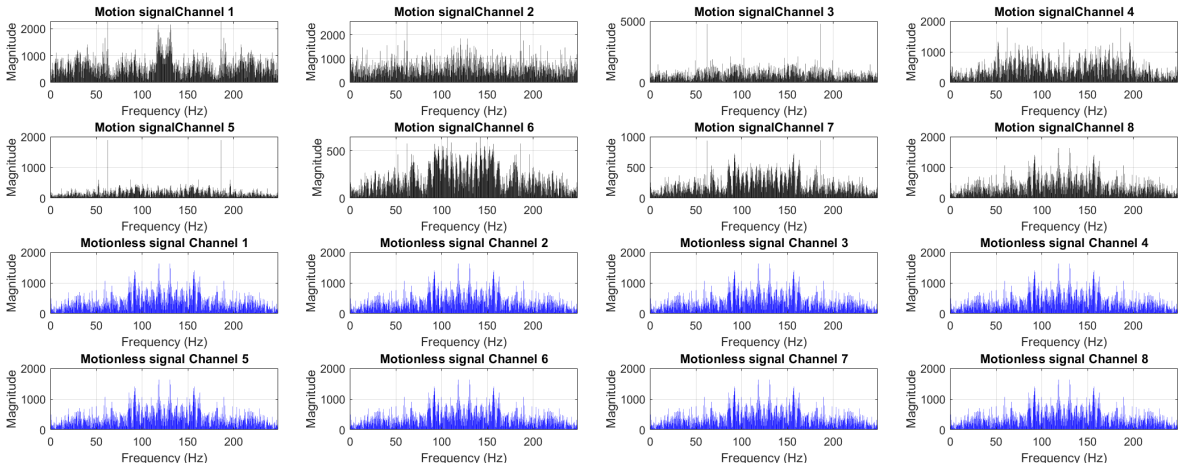


Figure 3.6: Fourier Transform of Raw Signal during movement execution (Black) and absence of movement (Blue).

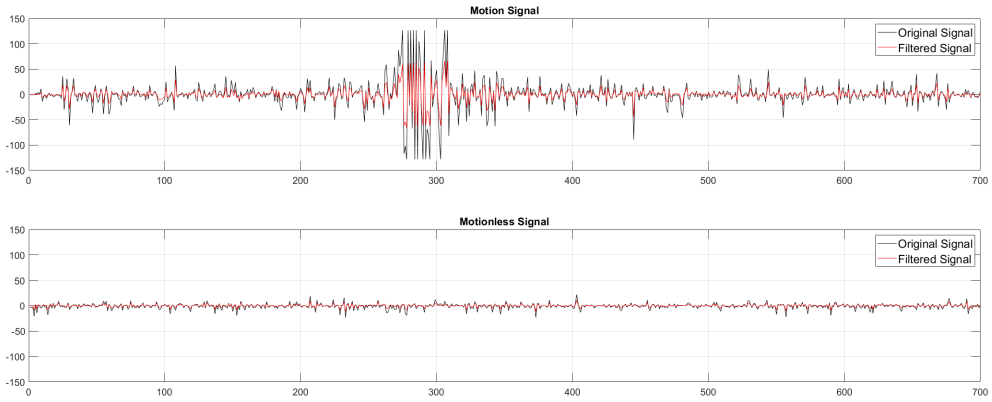


Figure 3.7: Bandpass Filtering.Signal during the execution of the movement (Upper part) and absence of movement (Lower part).

It is evident that the filter performed well in attenuating the signal, even with the removal of a relatively narrow frequency range. It is important to note that although the filter was designed to eliminate all frequencies below 5 Hz and above 500 Hz, only frequencies below 5 Hz were effectively eliminated. This is because in the original signal, the maximum frequency present is around 250 Hz, so there were no frequencies above 500 Hz to be removed. The application of the Fourier transform to the filtered signal allows visualizing the filter's effect. As shown in Figure 3.8, frequencies below 5 Hz were efficiently suppressed after the filter application.

Another filter used for frequency elimination is the notch filter, which aims to remove a

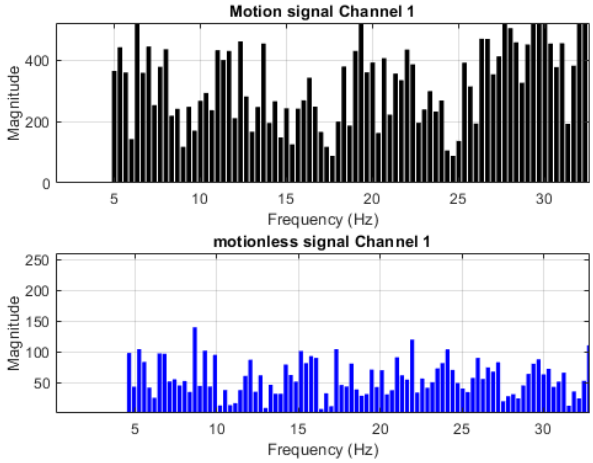


Figure 3.8: Fourier Transform of Bandpass Filtering. Signal during the execution of the movement (Upper part) and absence of movement (Lower part).

specific frequency. In this case, the filtering process was performed to eliminate the 50 Hz frequency, which represents the interference caused by the power grid in Portugal, where the data was captured. Similar to the bandpass filter, the notch filter is also applied to each column of the files. The effectiveness of the filtering can be observed in Figure 3.9, where the original signal is represented in black and the filtered signal is represented in red. It is important to note that only channel one is presented for better visualization of the results.

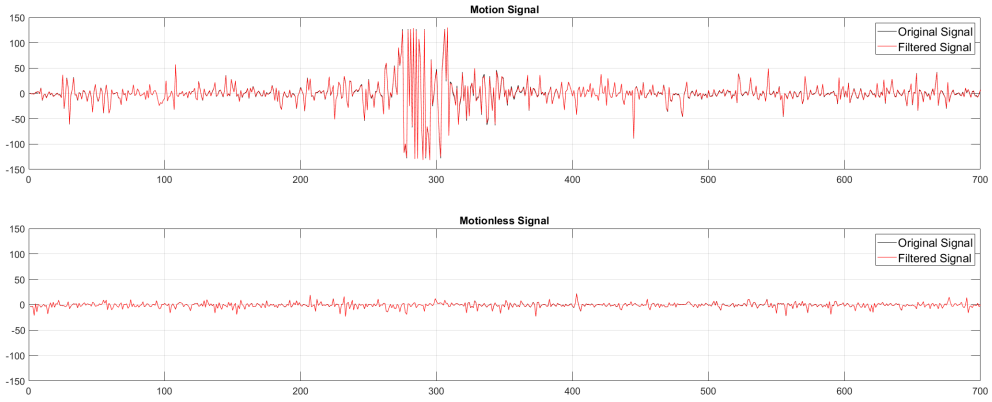


Figure 3.9: Notch Filtering. Signal during the execution of the movement (Upper part) and absence of movement (Lower part).

By performing the Fourier transform of the filtered signal, as illustrated in Figure

3.10, it is evident that the filter was effective in attenuating the frequencies near 50 Hz. However, it is notable that the elimination of this single frequency did not have a significant impact on the signal, as the filtered signal overlaps almost completely with the original signal. This indicates that the 50 Hz interference caused by the power grid was not a dominant source of distortion in the captured signal.

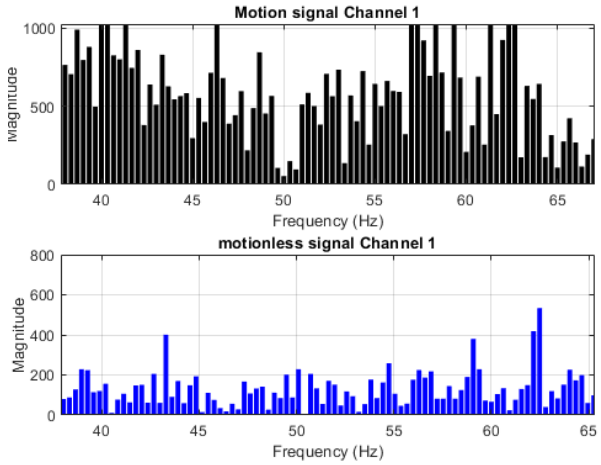


Figure 3.10: Fourier Transform of Notch Filtering. Signal during the execution of the movement (Upper part) and absence of movement (Lower part).

The next step involved applying the moving average filter, testing different window sizes, including 3, 5, 10, 50, and 100 samples. The result of this filtering can be observed in Figure 3.11, where the left side shows the filtering of the signal with movement 1, and the right side shows the filtering of the signal without movement. It is noticeable that as the window size increases, the signal is more attenuated, meaning there is a greater smoothing of the signal.

Finally, the last filter applied was the Kalman filter, which is implemented in the prediction and update stages. In the prediction stage, the filter estimates the next state of the system based on the previous state, while in the update stage, the filter corrects the estimated state based on the actual measurement. This filter is effective at reducing random noise. The result of this filtering can be seen in Figure 3.12, where a significant attenuation of the signal is evident.

It is important to note that the figures previously shown correspond to the initial

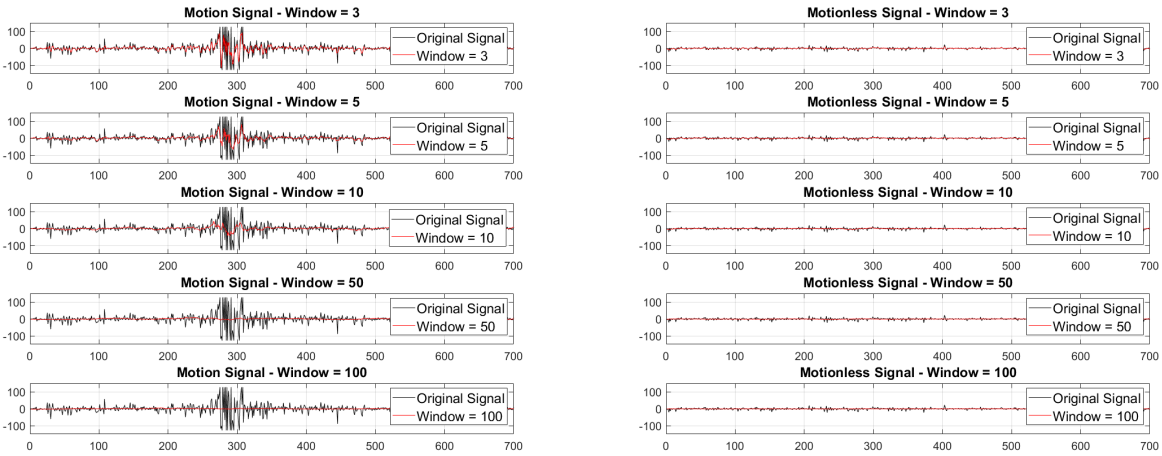


Figure 3.11: Moving Average Filtering. Signal during movement (Right) and absence of movement (Left).

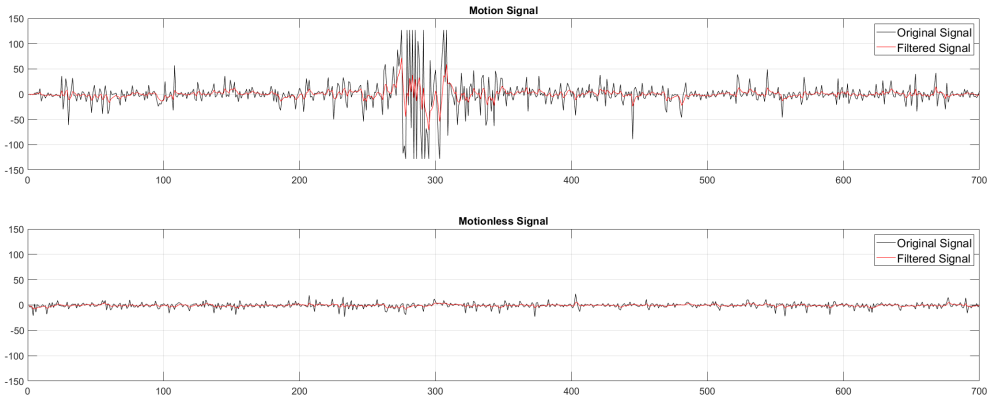


Figure 3.12: Kalman Filtering. Signal during the execution of the movement (Upper part) and absence of movement (Lower part).

database. These were selected because, as mentioned earlier, the data from the initial version exhibit more intense signals. With this in mind, the visualization of the applied filters becomes clearer. However, the filters have been applied to all records.

The purpose of testing a variety of filters is to identify the ideal combination that enhances the performance of machine learning models, particularly in terms of precision and recall. Each filter has the potential to extract unique characteristics from EMG signals, significantly contributing to the accuracy of their interpretation. This precision is crucial in applications that require meticulous control, such as in the manipulation of electronic

devices. Since EMG signals are subject to interference from external and internal noises, using a diverse range of filters can be decisive in mitigating these disturbances, thereby strengthening the system's reliability.

3.1.3 Databases

In this research, training and evaluating machine learning methods with filters applied to the data was essential in investigating the possible relationship between filtered EMG signals and the accuracy of the trained models. To facilitate this, four distinct database versions were developed, each aimed at understanding different aspects. The separation of the base datasets is illustrated in Figure 3.13. Versions V1 and V2 focused on two classes - movement and non-movement - to assess the impact of movement intensity on the models. Versions V3 and V4 were expanded to progressively include more classes of movements (three and four classes, respectively), with the goal of understanding how increased complexity affects model performance.

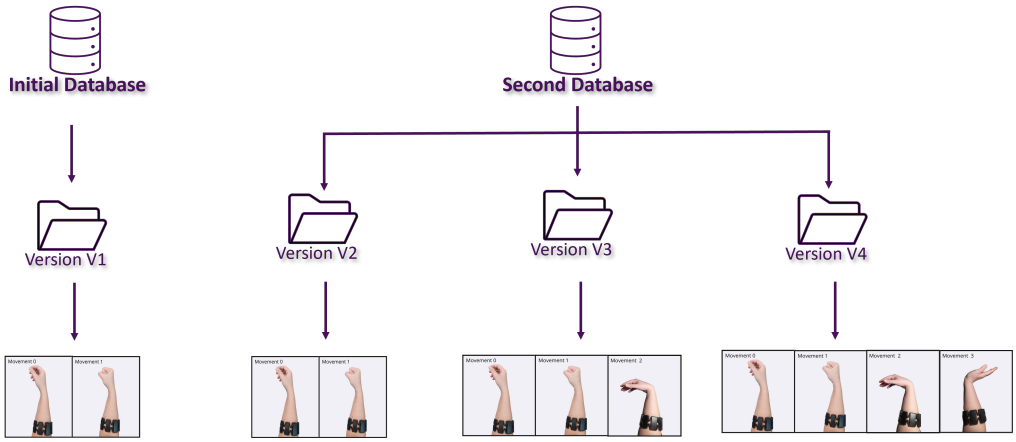


Figure 3.13: Illustration on how the base datasets were divided.

Following this, we applied various filters to each of these base versions. The process of filtering and the resultant datasets are detailed in Figure 3.14. Each base version (V1, V2, V3, and V4) was subjected to different filtering processes, resulting in nine unique datasets per version. This led to a total of 36 distinct datasets, which were then used to train the machine learning methods.

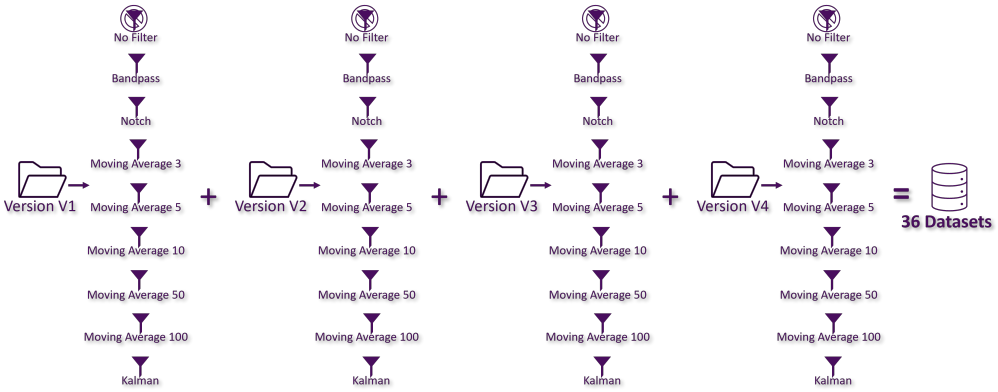


Figure 3.14: Final datasets creation process.

For clarity, each of these datasets was named systematically to reflect the version and the type of filter used. For example, datasets originating from V1 include 'V1-no filters', 'V1-Bandpass', and 'V1-Notch', and this naming pattern is consistent up to the V4 datasets, with names like 'V4-Kalman'.

3.2 Machine Learning Methods

After the proper data prepossessing, it's time to start training the methods. Three supervised methods were chosen and one unsupervised methods, which will be explained below.

In the context of supervised learning models, an approach was adopted that incorporates algorithms such as Decision Trees, Random Forest, and Support Vector Machines (SVM). The process of implementing these models was characterized by a series of methodologically consistent steps.

Initially, the data preparation involved reading and merging information related to movement and non-movement from CSV files, resulting in a unified DataFrame. Subsequently, these data were segmented into three distinct sets: training (70%), validation (20%), and testing (10%).

The training set, which comprises the largest portion, primarily aims at the model's learning. In this phase, the model is exposed to a wide range of data, allowing the

adjustment of its internal parameters – for example, the weights in a neural network – to make accurate predictions or classifications. The choice of approximately 70% of the data for this phase is justified by the need to provide a comprehensive information base for the model's effective learning.

The validation set is then used after training. It serves a dual purpose: adjusting the model's hyperparameters and providing a preliminary assessment of the model's generalization capability to unseen data. This step is crucial to prevent overfitting, which occurs when a model is excessively specialized in training data, losing its ability to accurately predict new information.

Finally, the test set is used to measure the final performance of the model. Unlike the training and validation data, this set is used exclusively after the completion of training and validation, serving as an impartial indicator of the model's ability to generalize to new data. Performance on the test set is considered the definitive metric of the model's effectiveness in real-world application conditions.

It is imperative to differentiate the purposes of the validation and test sets. The former is used iteratively during the model's development, assisting in the adjustment of hyperparameters and the detection of overfitting. The latter, however, is used only once, after the completion of training and validation phases, and provides an objective measure of the model's final performance.

The segregation of data into these three categories is essential to ensure that the developed model is robust, accurate, and efficient in generalizing to new data, indispensable characteristics for high-performance machine learning models.

In a subsequent step, the treatment of missing data was addressed. To ensure the integrity of the dataset intended for training, the missing values were filled with the mean. This strategy was adopted in response to a particular observation: certain filters, such as the moving average, tend to generate null values at the data edges, where insufficient information is available for the effective calculation of the mean. Filling the null values with the mean helps to preserve the overall statistical structure of the dataset, a crucial aspect in machine learning, as the data distribution can significantly impact the model's

performance. Moreover, if the null values are not random and are concentrated in specific areas of the dataset — such as at the signal edges —, using the mean to fill them can mitigate the bias potentially introduced by these gaps.

For the unsupervised model, the K-Means algorithm was chose. The number of clusters varied according to the number of classes in each model version. The process was similar to that of supervised models, with a particular emphasis on normalizing the data before applying K-Means, ensuring a fair comparison among attributes.

The scikit-learn library was utilized, employing standard functions like `train_test_split` for data division and `SimpleImputer` for handling missing values. Each model was configured with specific parameters. For instance, in K-Means, the `n_clusters` parameter was adjusted as needed, while in Decision Trees, parameters like `min_samples_split` were used to optimize performance.

The default parameters in scikit-learn for each model were crucial for our analysis. In the Decision Tree, the 'gini' criterion was used and did not set a limit for maximum depth, allowing the tree to grow freely. This could lead to more complex models but also to the risk of overfitting. For the Random Forest algorithm, utilizing 100 trees (estimators) and the 'gini' criterion aimed to balance computational efficiency and model accuracy.

In the SVM, the default value of C=1.0 and the 'rbf' kernel with gamma='scale' were selected to balance complexity and generalization, making them ideal for nonlinear problems. Finally, in K-Means, the standard number of clusters was adjusted according to the project's needs, using the k-means++ method for initializing the cluster centers.

Differing from conventional methods of random initialization, k-means++ strategically selects initial centroids to improve the clustering process. This method begins by randomly choosing one data point as the first centroid. Subsequent centroids are selected from the remaining data points, with a preference for those farther from existing centroids. This methodology significantly enhances the probability of effective and efficient data partitioning. It reduces the risk of poor initial centroid placement, which can adversely affect the final clustering results.

This methodological approach allowed for a detailed analysis of the impact of EMG

filters and the number of classes on the accuracy of machine learning models. By maintaining the default model parameters, the analysis focused more on the influence of the data and less on the intricacies of fine-tuning the models, facilitating the interpretation of the results in the context of the EMG signal study, and providing valuable insights into the characteristics of EMG signals and the efficacy of the chosen models.

Furthermore, a specific test was conducted on a single machine learning model from this project, notably the one that demonstrated higher accuracy compared to the others. The purpose of this test was to statistically evaluate the effectiveness of the model across its different versions (V1, V2, V3, and V4), by quantifying its rates of correct and incorrect classifications. The aim was to verify whether the accuracy derived from the stored data corresponds to the classification performed in real-time, essentially testing the practical effectiveness of the models. For this purpose, a simple script was developed, capable of recording the frequency of detection of each movement (M0, M1, M2, and M3) over a 10-second interval, as well as tallying the total number of classifications made. This approach was chose due to the high frequency with which the movements are classified by the model, making manual classification impractical, especially for versions with a greater number of classes. To gather the data, the script was executed individually for each movement in each version of the model. For instance, when analyzing movement 0 (M0) in version 1 (V1), the script recorded both the frequency of correct identification of M0 and the incidence of erroneous identifications, such as M1, in addition to the total number of classifications made during that period. This same procedure was repeated for movement 1 (M1) in V1, and subsequently, for the other movements in the other versions of the model.

3.3 Identification of the Bracelet Movement

After completing the machine learning component, the research progressed to spatial movement identification using the Myo Armband. A Python script was employed to interface with the Myo Armband device, with the primary objective of translating the

bracelet's movements into mouse cursor movements on the computer screen.

The script begins by importing the necessary libraries. The `pyomyo` library facilitates direct communication with the Myo Armband device. Libraries such as `math`, `numpy`, and `scipy.spatial.transform` are used for mathematical computations within the program, while `pyautogui` enables user interface automation, including mouse cursor control.

In the developed system, the speed and direction of the cursor are controlled by the Myo Armband bracelet. The cursor's speed is directly proportional to the movement speed of the bracelet. As for the direction, it is determined by Euler angles: yaw (horizontal rotation) and pitch (vertical rotation). The choice of Euler angles, despite their susceptibility to 'gimbal lock', is due to the limited range of motion of the bracelet. In the context of wearing the bracelet on the forearm, large amplitude rotations are impractical. Moreover, the roll angle is not used during the application, which significantly reduces the likelihood of 'gimbal lock' occurring. This choice is further reinforced by the simplicity and intuitiveness of Euler angles, making them the preferred approach in this research. The formulas used are:

$$\text{velocity\_x} = \text{speed} \times \sin(\text{yaw}) \tag{3.1}$$

$$\text{velocity\_y} = -\text{speed} \times \sin(\text{pitch}) \tag{3.2}$$

To transform a quaternion into Euler angles (yaw, pitch, and roll), we apply the formulas delineated in 2.4, 2.5, and 2.6.

Following conversion, angles are adjusted by subtracting the initial orientation. This calibrates the system, ensuring movements are relative to the Myo Armband's initial position:

$$\text{yaw\_adjusted} = \text{yaw} - \text{yaw\_initial} \tag{3.3}$$

$$\text{pitch\_adjusted} = \text{pitch} - \text{pitch\_initial} \tag{3.4}$$

$$\text{roll\_adjusted} = \text{roll} - \text{roll\_initial} \tag{3.5}$$

The orientation and rotation sense of the bracelet's yaw, pitch, and roll can be observed in 3.15.

The script employs `pyautogui`, a user interface automation library, to move the cursor. The `moveRel` function of `pyautogui` is used to move the cursor on the screen, receiving displacement values on the x and y axes and moving the cursor accordingly. The logic was implemented to smooth the cursor movement and adjust sensitivity, using weighted average calculations and linear interpolations to refine cursor control.

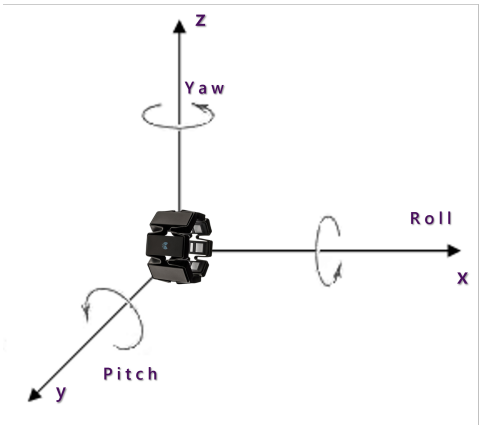


Figure 3.15: Yaw, Pitch, and Roll Diagram.

The accelerometer plays a crucial role in interpreting user gestures for mouse cursor control. Initially, the acceleration magnitude is calculated to determine whether the device is in motion. This magnitude is derived from the square root of the sum of the squares of the accelerations on each of the three axes. This information is vital for controlling the cursor's stillness: if the acceleration magnitude falls below a predefined threshold, the system interprets that the device is stationary, resulting in no movement of the cursor. Interestingly, while the accelerometer provides data about acceleration, its primary function in this context is to detect the absence of movement. Conversely, data from the gyroscope and orientation are used to determine the cursor's speed and direction.

This process was intended to result in a fluid and responsive translation of the Myo Armband's movements into cursor actions on the computer screen, creating an intuitive and efficient control interface.

3.4 Control of the Devices

Following the completion of the classification stages for movements 0, 1, 2, and 3, and the spatial movement identification of the Myo Armband, our study progressed to the integration phase of these modules. This stage was crucial for assessing the functionality of the device proposed by this research.

Initially, the V1 and V2 versions of the classification methods were limited to discerning only two movements: movement 0 (or absence of movement) and movement 1 (the motion of opening and closing the hand). This limitation restricted the potential applications of the system.

With this in mind, an initial logic was adopted where, upon activating the program, the real-time categorization of the user's movements commenced. Then, by maintaining movement 1 for two seconds, the computer cursor becomes controlled by the bracelet, enabling the user to manipulate the bracelet in space, correspondingly moving the cursor on the screen. An important aspect was mirroring the horizontal displacements of the cursor, considering that the user is positioned facing the equipment, thereby enhancing the system's usability. If movement 1 is detected for an additional duration of two seconds, the cursor is deactivated.

With the implementations of the V2 and V4 movement classifiers, capable of identifying more movements, new commands became possible to integrate into the system. V3 is capable of identifying movements 0, 1, and 2, while V4 adds movement 3. These additional movements were associated with specific commands, such as a left mouse button click for movement 2 and a right mouse button click for movement 3.

The "pyautogui" library used in this project offers a wide range of commands, including cursor manipulation, keyboard interactions, messages and alerts, visual interface capture, among other functionalities. This variety of commands allows for considerable customization according to the user's specific needs, expanding the interaction possibilities with the system.

In this phase of the project, two time counters were also added to the code. The

first counter is intended to calculate the time required to process the EMG signal before categorization, in cases where filters are applied. The second counter assesses the total time needed for complete categorization. These counters are fundamental in evaluating all combinations of filters and algorithms, identifying the most rapid and effective combinations.

In summary, the fusion of hand movement categorization and spatial movement detection of the bracelet resulted in an integrated solution, developed with the aim of ensuring a fluid and efficient interaction between the participant and the device.

Chapter 4

Results

In this chapter, the results obtained from the research are presented. The implications of these findings are discussed in detail, with a focus on their relevance and significance. Additionally, any unexpected results or anomalies that emerged during the research are highlighted. To assist in the clear understanding of the main results, visual aids such as graphs and tables are included.

4.1 Filtering and Machine Learning

The results presented in Table 4.1 showcase the performance dynamics of various machine learning models under the influence of different applied filters compared to models trained with the pure signal. These results correspond to V1, involving the initial dataset classifying only two movements. In this version, the Random Forest model emerges as a consistent leader, demonstrating superior accuracy across all filter scenarios. Notably, employing the Moving Average filter with a window size of 100 significantly boosts its performance, achieving an accuracy of 0.901 in the test set and 0.900 in the validation set.

The accuracy of the supervised models presented was calculated following established methodologies in machine learning evaluation, as detailed in the 'State of the Art and Tool Presentation' section 2.3.3. This calculation involved comparing the predicted labels

from the models with the actual labels of the dataset. Accuracy was determined by the proportion of correct predictions relative to the total number of predictions made. Although this process is not elaborated here due to its standard nature in the field of machine learning, it is based on the principles of confusion matrix analysis, as previously discussed. The precise classification of each instance—whether a movement was correctly identified or misclassified—contributed to the overall accuracy metric, providing a clear measure of the model's performance across different filtering scenarios. This metric serves as a comprehensive indicator of model efficacy.

Compared to the Decision Tree model, the SVM (Support Vector Machine) model shows better performance in most scenarios. The exception occurs in two specific cases where both models applied the Moving Average filter with window sizes of 50 and 100 samples. This suggests that filters with greater attenuation have a positive impact on the Decision Tree model.

On the other hand, the clustering model demonstrated unsatisfactory results when compared to classification methods in all scenarios, showing accuracies around 50%, which is comparable to a random choice in a two-class context.

Although clustering and classification are fundamental methods in machine learning, they serve different purposes. Clustering aims to identify inherent structures in data without the use of labels, while classification seeks to assign known labels to new data based on a pre-trained model. These differences make each method more suitable for different types of problems and analyses.

In the K-Means method used in this study, accuracy is assessed using the `accuracy_score` function from scikit-learn. This function compares the actual labels with those predicted by the model. For the validation set (`val_accuracy`), accuracy is calculated by comparing the true labels of the set (`true_labels_val`) with those predicted by the K-Means model (`predicted_labels_val`). Similarly, for the test set (`test_accuracy`), accuracy is obtained by comparing the real labels of the set (`true_labels_test`) with those predicted by the model. This method provides a quantitative measure of the K-Means model's ability to correctly cluster the data.

Clustering was employed with the aim of discovering non-obvious subgroups, where standard classification might not capture the full complexity of the data. Through clustering, additional patterns or nuances can be revealed. Comparing the accuracies of the models can help explore how clustering complements or provides insights beyond conventional classification. However, it is crucial not to draw direct or simplistic conclusions from this comparison, as the methods serve different purposes. The comparison should be made within the limitations and specific contexts of each approach.

These findings underscore the robustness of the Random Forest model for this specific dataset and suggest that the clustering model may not be the most suitable for the analysis performed.

Table 4.1: Accuracy of Trained Models - V1- Initial Database with 2 Classes

		Moving Average						
	No Filter	**BandPass**	**3**	**5**	**10**	**50**	**100**	**Kalman**	**Notch**
Decision Tree									
Accuracy Test	0,798	0,811	0,775	0,762	0,769	0,825	0,866	0,771	0,805
Accuracy Validation	0,800	0,809	0,775	0,759	0,772	0,823	0,865	0,770	0,802
Random Forest									
Accuracy Test	0,867	0,876	0,850	0,838	0,845	0,882	0,901	0,843	0,871
Accuracy Validation	0,866	0,874	0,850	0,836	0,844	0,881	0,900	0,843	0,869
SVM									
Accuracy Test	0,815	0,819	0,812	0,805	0,807	0,799	0,788	0,806	0,812
Accuracy Validation	0,814	0,820	0,812	0,803	0,806	0,798	0,788	0,806	0,811
Clustering									
Accuracy Test	0,527	0,530	0,505	0,542	0,551	0,552	0,528	0,539	0,536
Accuracy Validation	0,528	0,535	0,508	0,545	0,559	0,553	0,527	0,537	0,532

When initially focusing on supervised models, it is observed that the Moving Average filter with a window of 100 proved to be the most efficient in the Decision Tree and Random Forest methods. It is interesting to note that the same filter, with a window of 50, also yielded positive results, being the second most effective in these training methods.

However, it is important to highlight that these filters, despite being successful in the mentioned models, performed poorly in the SVM model, even when compared to the SVM trained with the original, unfiltered signal. This suggests that excessive signal attenuation may not be ideal for this specific machine learning method. The SVM aims

to identify the optimal hyperplane to maximize the margin between classes, and excessive data smoothing, a potential effect of a large Moving Average filter window, might reduce the distinction of essential features for the SVM decision boundary construction.

In the case of the SVM, the Bandpass filter proved to be the most effective, albeit with a slight difference compared to the model trained with the original signal, showing an improvement of approximately 0.006, or less than 1%. Except for Bandpass, other filters did not offer significant advantages, suggesting that for the proposed application, training the model without a filter might be more advantageous, especially to ensure a faster real-time system response without the need for signal preprocessing. Moreover, the SVM showed less performance variation with different filters, indicating a possible insensitivity to these and a limited benefit from their application.

Validation results reveal that the BandPass filter is consistently effective across all models, particularly in the SVM. In the Random Forest and Decision Tree models, this filter is only surpassed by models with the Moving Average filter with 50 and 100 windows. On the other hand, the Notch filter showed very similar results to the models trained without a filter, indicating little to no improvement, which was also expected since this filter keeps the signal almost unchanged. The Moving Average filters with windows of 3, 5, 10, and the Kalman filter proved to be unsatisfactory, performing worse than the unfiltered models. Therefore, these filters do not justify the preprocessing cost for the final application, indicating a negative impact on accuracy compared to other filters.

It can then be concluded that the performance of the Moving Average filters can vary significantly among different machine learning models. This variation depends on the chosen window size, which can sometimes significantly improve performance or even result in negative impacts.

These observations can be verified in Figure 4.1, which contains a graph illustrating the validation accuracies of each trained model. It is worth noting that the similarity in the results of the Decision Tree and Random Forest models is expected, considering that Random Forest is composed of multiple decision trees.

To facilitate the understanding of the graphs presented in this chapter, it's important

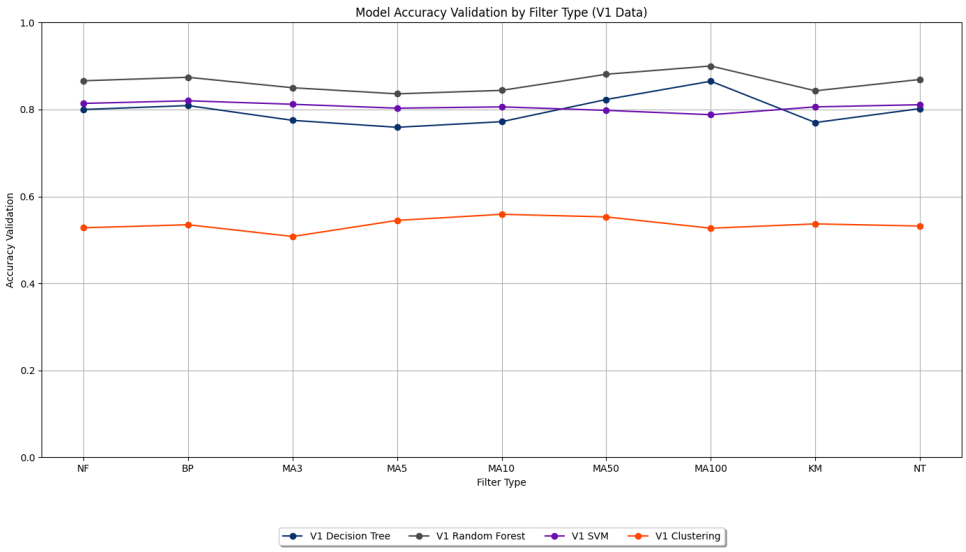


Figure 4.1: V1- Validation accuracy of each of the machine learning methods for each one of the filters.

to note that the names of the applied filters have been abbreviated as follows: NF for No Filter, BP for BandPass, MA3, MA5, MA10, MA50, MA100 for Moving Average with window sizes of 3, 5, 10, 50, and 100 samples respectively, KM for Kalman, and NT for Notch. It is also pertinent to mention that the numbers associated with the Moving Average filters indicate the size of the window used in each of them.

Turning the focus solely to the Clustering model, a striking contrast is observed compared to the supervised methods. The accuracies achieved were substantially lower. The best result was a validation accuracy of just 0.559, achieved with the Moving Average filter and a window of 10. This performance is significantly lower than that of the supervised methods, suggesting that Clustering might not be the most effective approach for this particular dataset.

Moving forward with our analysis, after the initial assessments on the V1 database, it was time to examine how the modifications made during data collection impacted the training results with the V2 database. As previously described in more detail, V2 was designed to make the movements to be classified more natural. The validation and test accuracies obtained from the new trainings are detailed in Table 4.2. An immediate

observation is a decline in the accuracies of all models.

Table 4.2: Accuracy of Trained Models - V2- Second Database with 2 Classes

		Moving Average						
	No Filter	**BandPass**	**3**	**5**	**10**	**50**	**100**	**Kalman**	**Notch**
Decision Tree									
Accuracy Test	0,718	0,723	0,689	0,664	0,675	0,727	0,804	0,684	0,730
Accuracy Validation	0,716	0,723	0,689	0,665	0,676	0,727	0,805	0,681	0,729
Random Forest									
Accuracy Test	0,782	0,799	0,760	0,739	0,749	0,802	0,870	0,758	0,810
Accuracy Validation	0,779	0,799	0,762	0,740	0,751	0,803	0,871	0,755	0,808
SVM									
Accuracy Test	0,738	0,746	0,735	0,721	0,727	0,723	0,734	0,732	0,738
Accuracy Validation	0,735	0,743	0,735	0,721	0,726	0,725	0,734	0,729	0,736
Clustering									
Accuracy Test	0,459	0,451	0,474	0,478	0,458	0,446	0,458	0,454	0,458
Accuracy Validation	0,460	0,454	0,475	0,478	0,459	0,443	0,459	0,451	0,461

Figure 4.2 provides a clearer view of this difference. On the left graph, the accuracies achieved with the V1 database are displayed, while on the right, the results obtained with the V2 database are shown. It is noteworthy that the behavior of the machine learning models was quite consistent across both databases. Despite a slight loss in precision from one database to the other, the performance of the models remained similar to each other.

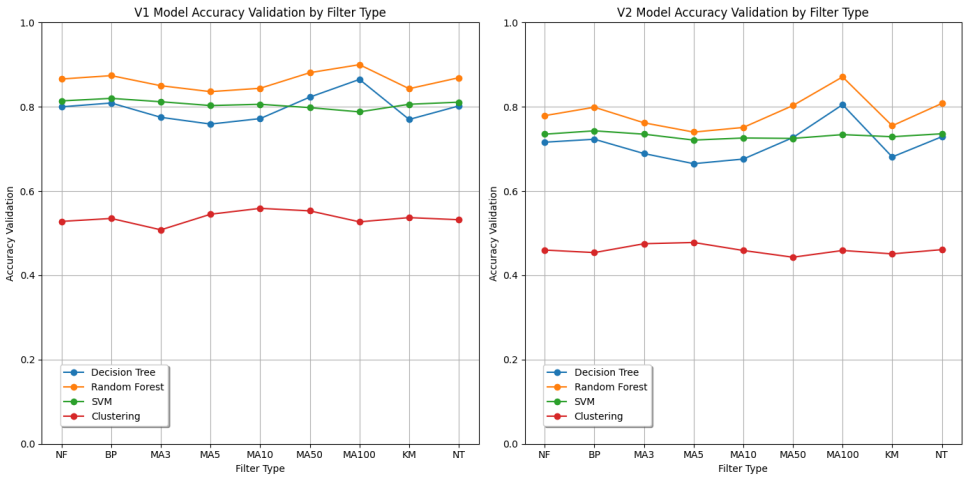


Figure 4.2: Comparison of the accuracys of V1 (left) and V2 (right).

Another interesting observation from Figure 4.2 is that the SVM models showed the least variation with the applied filters compared to the Random Forest and Decision Tree

models. This confirms an earlier observation that SVM exhibits the smallest performance variation with different filters, indicating a relative insensitivity to the filters used in this work. A similar trend seems to occur with the clustering method, although it exhibited slightly more noticeable variations than the SVM.

After completing these analyses, the training of the models with the V3 and V4 databases commenced. As previously mentioned, V3 involves the classification of three distinct movements, while V4 addresses the classification of four movements. With this approach, the aim is to observe the impact that increasing the number of classes to be identified has on the models. The accuracies achieved with the training of V3 can be seen in Table 4.3, and the results from the training of V4 are presented in Table 4.4.

Table 4.3: Accuracy of Trained Models - V3- Second Database with 3 Classes

		Moving Average						
	No Filter	**BandPass**	**3**	**5**	**10**	**50**	**100**	**Kalman**	**Notch**
Decision Tree									
Accuracy Test	0,525	0,525	0,497	0,480	0,492	0,568	0,678	0,491	0,552
Accuracy Validation	0,523	0,528	0,498	0,477	0,492	0,569	0,676	0,492	0,553
Random Forest									
Accuracy Test	0,595	0,612	0,573	0,559	0,575	0,685	0,794	0,572	0,678
Accuracy Validation	0,596	0,612	0,573	0,559	0,577	0,684	0,793	0,574	0,678
SVM									
Accuracy Test	0,550	0,558	0,547	0,538	0,543	0,551	0,572	0,545	0,553
Accuracy Validation	0,548	0,554	0,546	0,536	0,544	0,553	0,571	0,544	0,549
Clustering									
Accuracy Test	0,294	0,301	0,302	0,323	0,305	0,282	0,295	0,316	0,296
Accuracy Validation	0,296	0,303	0,304	0,322	0,305	0,282	0,293	0,314	0,296

Initially, a similar trend is observed between the V1 and V2 sets, where accuracies decreased. This decrease became even more pronounced in V3 and V4 as the number of classes increased. Taking the decision tree model without a filter as an example, a validation accuracy of 0.80 in V1 was observed. However, this accuracy fell to approximately 0.72 in V2. The drop was more significant in V3, where the accuracy reached 0.52, and in V4, where it decreased further to 0.42. This progression can be better understood through Figure 4.3, which presents detailed graphs of the accuracies for each version.

An interesting point to note is that not all models exhibited the decreasing trend

Table 4.4: Accuracy of Trained Models - V4- Second Database with 4 Classes

		Moving Average						
	No Filter	**BandPass**	**3**	**5**	**10**	**50**	**100**	**Kalman**	**Notch**
Decision Tree									
Accuracy Test	0,422	0,424	0,398	0,383	0,398	0,487	0,609	0,396	0,443
Accuracy Validation	0,422	0,425	0,398	0,383	0,398	0,488	0,610	0,394	0,444
Random Forest									
Accuracy Test	0,489	0,509	0,468	0,457	0,474	0,619	0,755	0,470	0,564
Accuracy Validation	0,492	0,509	0,467	0,458	0,475	0,620	0,755	0,469	0,566
SVM									
Accuracy Test	0,446	0,451	0,445	0,436	0,440	0,455	0,480	0,440	0,445
Accuracy Validation	0,447	0,451	0,445	0,436	0,442	0,457	0,482	0,440	0,447
Clustering									
Accuracy Test	0,220	0,216	0,227	0,224	0,223	0,217	0,232	0,227	0,228
Accuracy Validation	0,221	0,215	0,227	0,223	0,224	0,216	0,231	0,225	0,228

observed earlier. When analyzing the Random Forest model applied with the Moving Average filter with a window of 100 — which showed the best results in all tests — a different pattern emerges. In the V1 set, this model achieved a validation accuracy of 0.90, which slightly decreased to 0.87 in V2, a more gradual decline compared to the previous example. In V3, the accuracy fell to 0.79, and in V4, it reached 0.76, showing a less pronounced variation compared to other models. On the other hand, the decision tree model with the same filter exhibited a similar trend but with proportionally larger decreases.

In Figure 4.4, there is a clear visualization of how accuracy was impacted across different models. The figure presents four graphs, each corresponding to a machine learning method. In these graphs, the evolution of the accuracy of each filter across versions V1, V2, V3, and V4 is observed. It's immediately noticeable that the clustering and SVM models were minimally affected by the filters, with SVM showing a slightly lesser impact than clustering. In the Random Forest and Decision Tree models, similar impacts are observed on both. However, three filters — Moving Average 100 (MA100), Moving Average 50 (MA50), and BandPass (BP) — showed superior performance in Random Forest compared to Decision Tree, particularly as more classes were added. These filters were especially effective in maintaining accuracy amidst the addition of new classes.

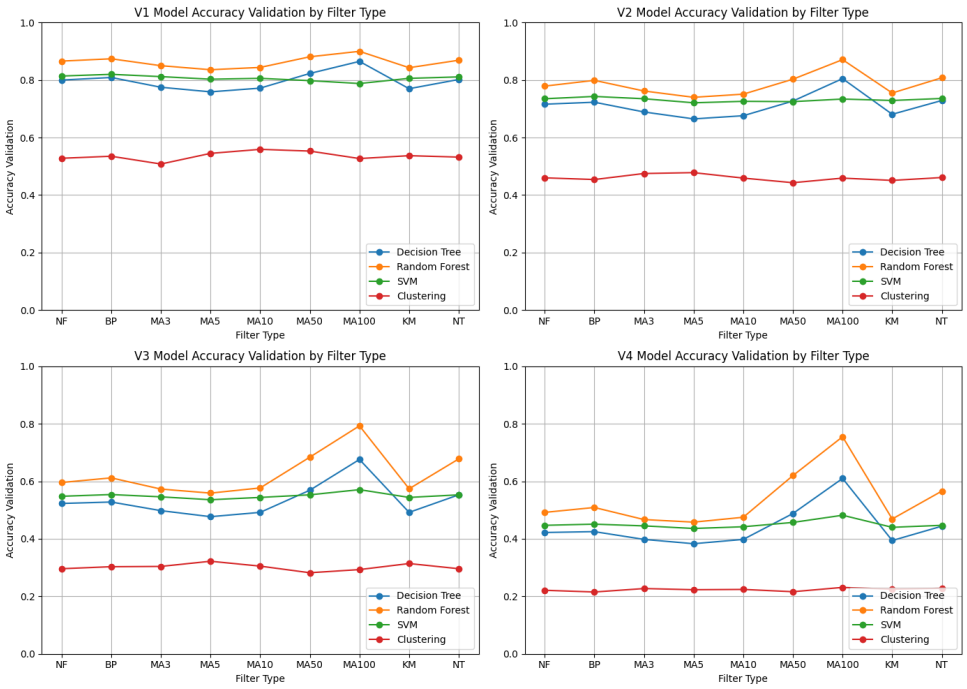


Figure 4.3: Comparison of the accuracies of V1 (Upper Left), V2 (Upper Right), V3 (Lower Left) and V4 (Lower Right).

After the accuracy evaluation, the trained models underwent processing time analysis. Two timers was implemented in the EMG signal real-time classification code: one to measure the signal filtering time and another for the total classification time, which includes both filtering and classification. These processing times are variable, affected by factors such as computer processing speed and available memory. To mitigate these variations and increase the reliability of the results, the code was set up to record and calculate the average of ten time measurements. This method aims to reduce random variability and ensure greater accuracy in the results. The tables with the average times for each model and version are available in Appendix A. This chapter focuses on the representative graphs of these data.

Figure 4.5 displays the processing times for each filter applied across different machine learning models in training versions V1, V2, V3, and V4. The analysis reveals that most filters have very similar processing times, with two notable exceptions. The first is the Kalman filter, which recorded the longest filtering times. This aspect, coupled with its

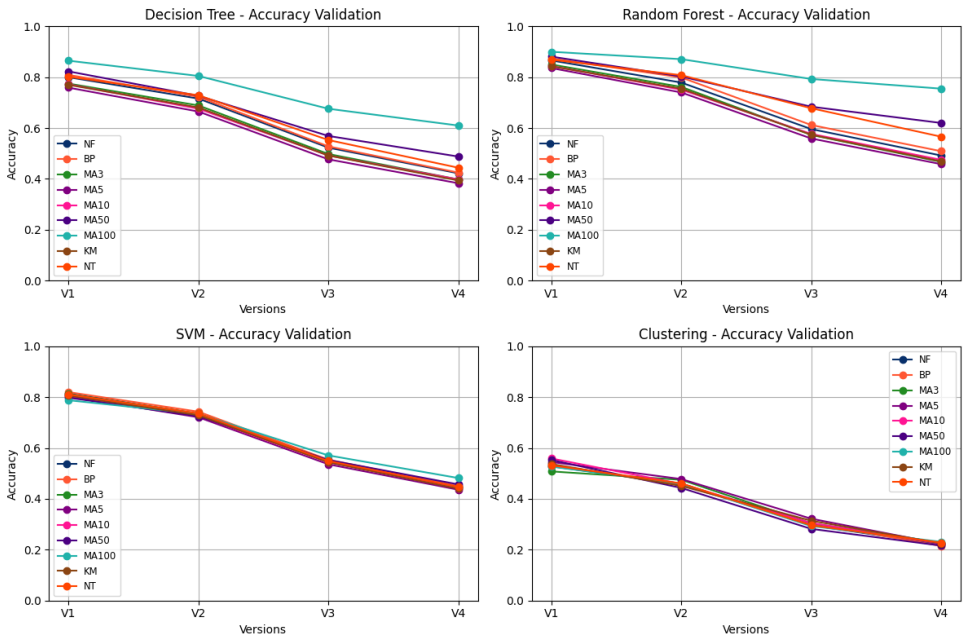


Figure 4.4: Accuracies of each of the machine learning models over the versions.

negative impact on the models' accuracy, suggests that it is not the most efficient choice for this dataset. In contrast, the Notch Filter showed only a slight variation in processing times, especially when compared to the Kalman filter.

Figure 4.6 provides a detailed comparison of the total processing times for movement classification. It is observed that the Kalman filter significantly influenced the total time, as indicated by the peak in the graph related to this filter. The clustering and decision tree models stood out for their shorter processing times, being approximately 10 milliseconds faster than the others. On the other hand, models using Random Forest and SVM recorded longer processing times. Notably, the SVM generally had the longest processing time, with the exception of the SVM model in the V1 dataset, which showed times close to those of the Random Forest models. However, with the introduction of more classes into the models, there was a proportional increase in processing time.

Based on the gathered information, it can be concluded that althoughclustering models exhibit good processing times, they are not the most viable option due to their significantly lower accuracy compared to other models. Moreover, their application in real-time is constrained, as they fail to classify data correctly. On the other hand, the decision tree

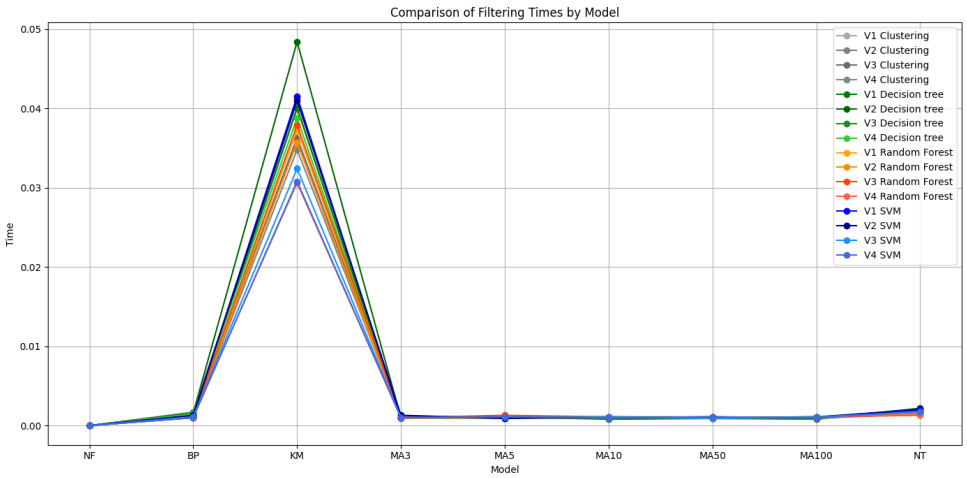


Figure 4.5: Graph of filtering time.

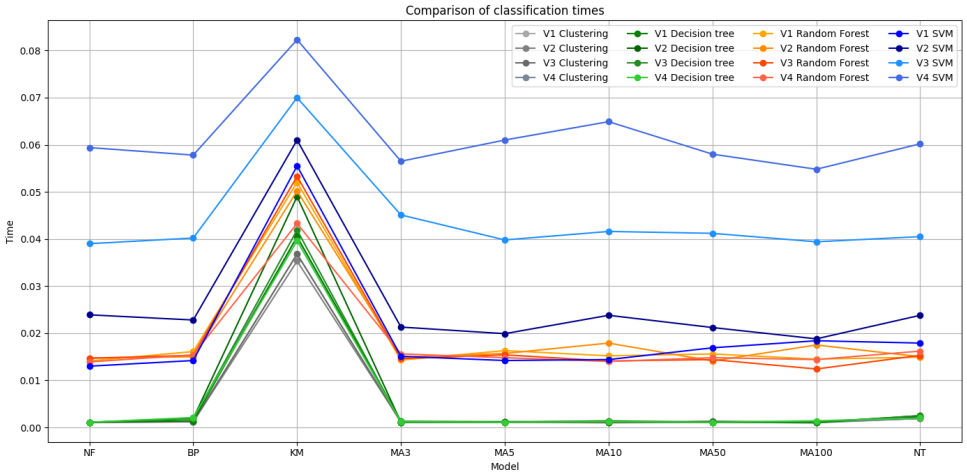


Figure 4.6: Graph of classification time.

method, despite having lower accuracy than the SVM and Random Forest models, stands out for its shortest processing time. This can be particularly advantageous in applications where response speed is more critical than the model's precision. The SVM model, although demonstrating good precision, is one of the slowest, and considering its lengthy training time, it may not be the ideal choice, especially when compared to Random Forest, which exhibited higher accuracies with shorter processing times.

In the process of comparing the percentages of errors and correct classifications among different machine learning models, the *Random Forest* with a moving average filter and

a window of 100 was selected. This choice was based on the observation that this model demonstrated the best accuracies across all its versions, as previously discussed.

More specific details of these comparisons are available in Appendix C.2, where tables detail the hits and misses for each movement in each version of the model. The data for Version 1 (V1) is in table C.5, for Version 2 (V2) in table C.6, for Version 3 (V3) in table C.7, and for Version 4 (V4) in table C.8. An interesting observation is that movement 0 consistently shows the lowest accuracy rate in almost all versions, suggesting a difficulty of the model in accurately classifying this category. This recurring pattern highlights the need to refine data processing or adjust the model specifically to improve the classification of movement 0, aiming to increase the overall precision of the system.

Furthermore, it is notable that movements 1 and 2 exhibit a high rate of confusion with each other, indicating an overlap in the characteristics used by the model to differentiate them. In contrast, movement 3 is characterized by a high rate of correct identification, indicating that it is more distinct and easily recognizable by the model compared to the other movements.

Table 4.5: Comparative of Calculated Accuracies for Each Version

Version	**Accuracy**
V1	92.64%
V2	84.78%
V3	57.63%
V4	65.59%

In Table 4.5, its present accuracies calculated with real-time data collated for each version of the model. This approach allows us to observe the total number of classifications, which increases in proportion to the number of classes. An interesting pattern emerges from this analysis: there is a tendency for precision to decrease as the number of classes increases. However, a notable exception is observed between Versions 3 (V3) and 4 (V4), where V4 shows a higher percentage of correct classifications than V3. This suggests that the introduction of a new class in V4 may have helped to reduce ambiguity among the existing classes. Such an addition appears to enable the model to classify more

accurately, improving the distinction between the categories.

4.2 Identification of the Bracelet Movement

Here, the results related to identifying the bracelet's movements in space are presented. For clearer understanding, the gestures used to control the cursor are illustrated in Figure 4.7. In this figure, the upper part shows the bracelet's movements corresponding to the cursor's horizontal displacement, while the lower part displays movements associated with the cursor's vertical displacement.

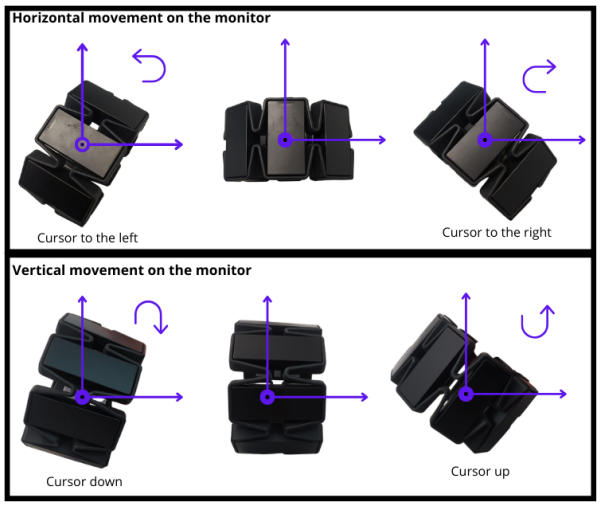
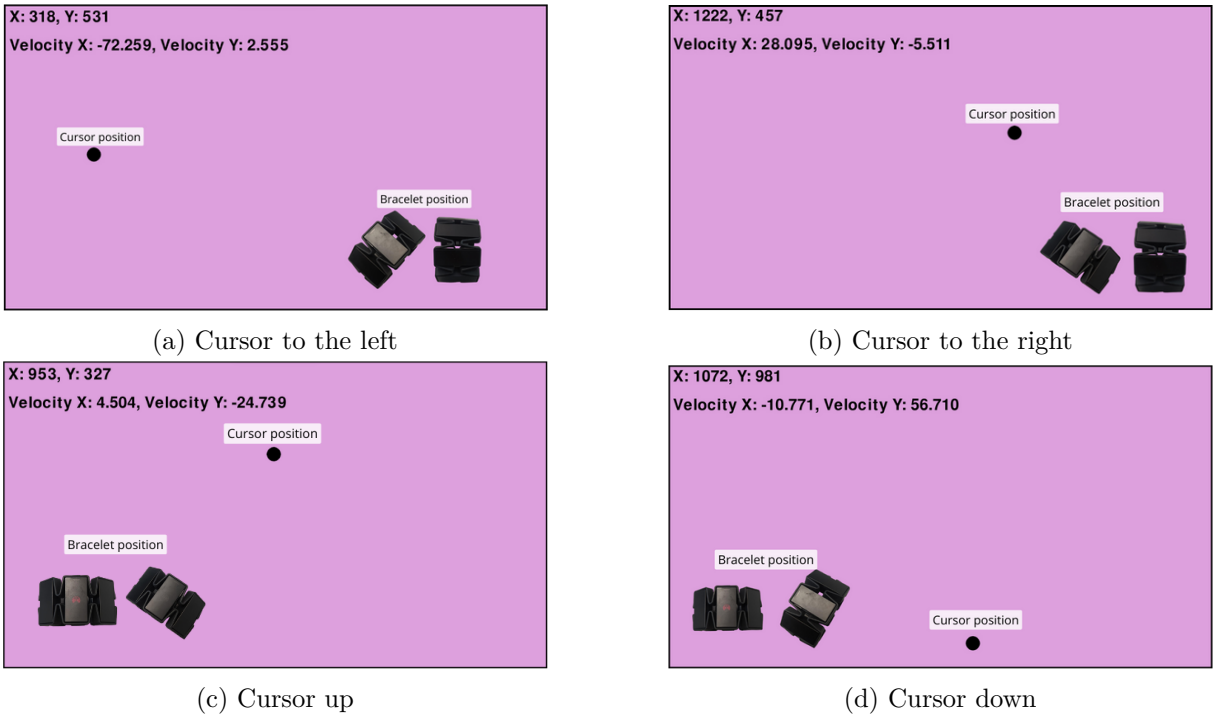


Figure 4.7: Bracelet position related to its effect on the monitor

The functionality of the system is more easily understood through Figure 4.8, which contains four illustrative images. Each image demonstrates the system's response when the bracelet is rotated in one of the four predefined directions. The images include a depiction of the bracelet's positioning, with top and side views, to aid the reader's understanding. In the upper corner of each Figure, the current cursor coordinates and the 'relative velocity' on the X and Y axes are displayed. Additionally, a black circle indicates the current position of the cursor to represent its movement intuitively. In all scenarios, it is assumed that the cursor starts moving approximately from the center of the screen, with initial relative speeds of zero on both axes. It is important to note that the images

presented are screenshots taken during the execution of the program.



(a) Cursor to the left

(b) Cursor to the right

(c) Cursor up

(d) Cursor down

Figure 4.8: Representation of the movement of the cursor on the screen according to the movement of the bracelet

In Figure 4.8a, it is observed that when the bracelet is moved slightly to the left of the screen, the velocity on the X-axis is approximately -72 at the captured moment. This indicates that the cursor will move 72 pixels in the opposite direction of the axis. It's important to note that the origin of the X and Y axes is located in the upper left corner of the screen. Additionally, the relative velocity on the Y-axis is close to 3. This occurs because the sensor is not fixed in place, so it's not possible to ensure that the movement is exclusively along one axis; however, as the value is small, the impact on the screen is almost imperceptible. It should be emphasized that this information is specific to the moment of the screenshot. The velocities are calculated in real-time, meaning that as the bracelet moves, these values will constantly vary depending on the speed and duration of the movement.

Figure 4.8b illustrates the cursor movement to the right. In this scenario, the relative velocity values on the X-axis are positive, leading to the cursor moving in the positive

direction of the axis. As for Figures 4.8c and 4.8d, they depict the vertical movements of the cursor. Here, it is the relative velocity on the Y-axis that changes, showing positive values for upward movements and negative values for downward movements.

The outcomes of the spatial movement identification using the Myo Armband were encouraging. The Python script effectively interfaced with the Myo Armband, translating its movements into corresponding cursor movements on the computer screen. Nevertheless, during the testing phase, a lack of fluidity in the cursor's response was noted. There was a discernible delay in cursor movement, accompanied occasionally by unintended residual motions.

Further investigation led to a crucial insight. The computational logic in the `MyoMouse` class was predicated on the notion that all rotational movements would be centered around the bracelet's axes. However, this assumption does not hold true when the bracelet is worn on the arm, as the true pivot point of movement is the elbow, not the bracelet. This discrepancy introduces unaccounted angles into the calculations.

The `on_orientation` method, responsible for converting quaternions to Euler angles, was significantly impacted by this miscalculation. The yaw, pitch, and roll angles, essential for determining the cursor's direction and magnitude of movement based on the Myo Armband's orientation and velocity, failed to consider the extra angles introduced by the elbow's movement.

Despite these initial challenges, the results remained encouraging. With refinements to include the elbow's rotational angles, it is expected that future iterations of the system will achieve more fluid and intuitive control.

4.3 Control of the Devices

After successfully completing the phases of movement classification and identification, our research progressed to integrating these components, a crucial milestone in assessing the feasibility of combining motion classification with spatial identification in a unified application. The results of this project are encouraging. A harmonious integration of

movement classification and spatial identification of the bracelet was achieved, as envisioned at the start of the project. The outcomes were consistent, although adjustments are needed to optimize usability.

The implementation of device control, although limited by the restricted number of possible actions, proved to be promising. The current system, despite its limitations, indicates significant potential for future improvements. It is believed that with further refinements, the device's control capabilities can be substantially expanded.

A notable aspect of this research is the potential for customizing commands to suit individual user needs. This is particularly relevant for users with disabilities, considering the variation in physical limitations among individuals.

In initial tests using models V1 and V2, which only identified the motion of closing the hand, the system was limited. However, it served well to test the integration of motion classification and identification, showing positive results despite challenges in spatial movement. The identification of hand motion did not present significant issues.

The addition of new commands in models V3 and V4 significantly enhanced the system's functionality. In V3, two particularly notable commands were introduced. The first, essential for practical usability, was a command to activate and deactivate the cursor. This functionality became indispensable, as keeping the cursor constantly active could lead to accidental commands or involuntary cursor movements, especially since the device is continuously worn on the forearm. This could result in a frustrating user experience, with the cursor reacting to every slight arm movement, even when unintended. The second command introduced was a simple click, the most basic and frequently used gesture in electronic device interaction. Both commands were effective, but the overall precision of the system slightly decreased, necessitating the implementation of more refined criteria to ensure correct movement identification.

In Figure 4.9, a visual representation of the application's functionality is presented, particularly in terms of classifying hand movements and their corresponding actions on the device. To illustrate, specific commands like clicking and mouse activation are linked to graphic images representing the corresponding movements. For example, when performing

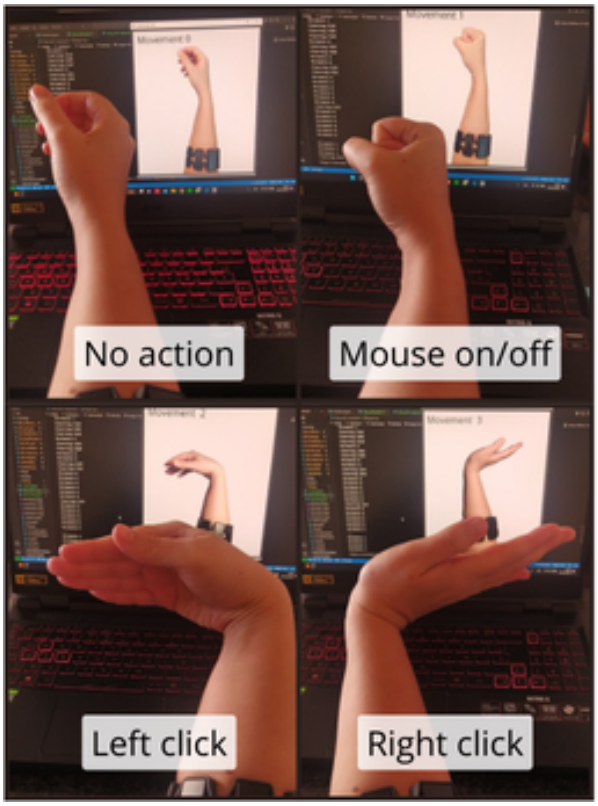


Figure 4.9: Movement associated with action on the computer

the gesture of closing the hand (Movement 1), the program displays an image illustrating this movement, visible in the upper right corner of the figure. This image represents the movement used to activate and deactivate the mouse. In the upper left corner, the representation of the absence of movement, defined as Movement 0, is observed. In the lower left corner, Movement 2 is found, responsible for the left mouse button click. Finally, in the lower right corner, Movement 3 is illustrated, corresponding to the right mouse button click. It is important to note that the images shown are photographs taken at the exact moment of executing each gesture, capturing both the user's hand and the computer interface.

In the V4 models, despite the benefit of an additional action, the device's precision decreased, leading to occasional misclassifications and incorrect actions. However, it is posited that with ongoing improvements, these issues can be eliminated or minimized to the point of being imperceptible.

Another important observation is the presence of a notable delay between the execution of the movement and the corresponding action on the device. This delay highlights the need for further optimizations in the system's processes to ensure that the device can deliver effective results in practical, everyday situations. Minimizing this response time is crucial to improving user experience and ensuring the device's applicability in real-world scenarios.

Chapter 5

Conclusion and Future Work

The aim of this chapter is to distill the main takeaways from the results showcased and reflect on the implications and value added by this study. Here, the discussion circles back to the original goals, evaluates how well they were achieved, and pinpoints the key learnings gathered during the research process. Finally, this chapter will outline possible directions for future research. Suggestions, ideas, and viewpoints for upcoming studies will be presented, delving into areas that could benefit from further exploration or fresh insights.

5.1 Conclusion

This study addressed a significant challenge faced by amputees: the difficulty of accessing suitable assistive and rehabilitation technologies. Many of these technologies are financially inaccessible or do not meet the specific needs of users. An illustrative example is the use of electronic devices. While solutions like voice commands exist, allowing users to control devices through verbal instructions, they are not always feasible. In situations like a classroom or environments requiring silence, using these technologies can be impractical.

On the other hand, more sophisticated solutions, such as myoelectric prostheses, while allowing users to perform a variety of actions, face the challenge of inherent delay in their system. This delay is not only due to the identification and classification of movement

but also to the mechanics of the prosthesis, including motor control. Our study aims to address this issue, seeking to deeply understand these challenges and offer a solution that can effectively assist in the daily lives of people with amputations, improving their interaction with assistive technologies.

Although the data collected in our research primarily come from individuals without amputations, the in-depth analysis of these provided a comprehensive view of the system's functioning as a whole. This overarching understanding is crucial, as it allows for the system's adaptation for use by amputees in a more effective manner. By understanding how myoelectric signals are processed and interpreted in individuals without amputations, we can develop algorithms that compensate for the peculiarities found in amputee users. However, it is important to note that this is just one step in the process. The direct application of these insights into assistive technologies for amputees requires additional and specific studies. These studies should focus on understanding the unique variations and challenges presented by this demographic, thereby ensuring that the developed solutions are not only technically feasible but also practical and comfortable for everyday use.

From analyzing the results of training 144 distinct machine learning models, which included various combinations of filters and versions with an increase in the number of movements to be classified, a remarkable consistency in the results was observed. Some models stood out for their potential, particularly the Random Forest model, which exhibited the best accuracies in all tested scenarios. Its effectiveness was especially notable when combined with the Moving Average filter with a window of 100. This combination not only achieved the highest precision but also demonstrated resistance to variations, reinforcing its viability for implementation in practical applications.

The SVM method also showed positive results, surpassing most scenarios in accuracy, except when compared to Random Forest. However, considering the processing time, SVM proved to be less efficient. This method recorded the longest processing time among all tested models and required a more extensive training time compared to other methods. These factors combined make SVM less ideal, especially when compared to the efficiency and speed of Random Forest.

The Decision Tree model, despite showing slightly lower accuracy compared to SVM and Random Forest, stood out for its shortest processing time. This characteristic suggests that, with the necessary adjustments, the Decision Tree could emerge as the most suitable option for the solution proposed in this work. The importance of a fast system response time is fundamental to ensure a satisfactory user experience, making the Decision Tree a promising candidate for practical applications where agility is essential.

While the clustering method recorded good processing times, its performance in motion identification was significantly inferior to other methods. This limitation in precision makes it impractical for real-world applications, where reliable motion identification is crucial.

Regarding filters, it can be confidently stated that the Moving Average with a window of 100 showed the best results. This was followed by the Moving Average with a window of 50, and then the Bandpass and Notch filters, which had the least impact on the models. However, the limited effectiveness of these latter suggests that their application might not be very beneficial. Training without a filter showed comparable results and has the advantage of not requiring signal preprocessing, saving time. On the other hand, the Moving Average filters with windows of 3, 5, and 10, along with the Kalman filter, negatively impacted the models. The Kalman filter, in particular, not only adversely affected the model's accuracy but also added a considerable processing time, making it the least viable option among the filters applied in this study.

The analyses conducted in this research are corroborated by the demonstrations presented in the article titled 'Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal', included in Appendix B of this document. This article was previously presented at the International Conference on Optimization, Learning Algorithms and Applications 2023 (OL2A 2023).

Regarding the identification of the bracelet's movement in space, some limitations were observed, particularly in precision, which is still not sufficient for real-world applications. However, this aspect shows promise, especially after identifying the possible cause of the imprecision in translating the bracelet's movements into cursor movements. The central

issue identified is that cursor control is based on the premise that the bracelet moves around its own axis. In practice, however, the real axis of movement is located at the user's elbow, not the bracelet. Correcting this factor may allow for more precise and appropriate cursor control, paving the way for significant improvements in the practical application of this system.

Finally, the device control phase, although still requiring improvements, clearly demonstrated the viability of integrating motion classification with spatial identification. This advancement shows that the proposed technology has significant practical potential. Furthermore, the ability to customize commands according to individual user needs was a key aspect demonstrated by the study. This flexibility is crucial in assistive technologies, given the diversity of challenges faced by people with disabilities. The possibility of adjusting the system to meet the specific requirements of each user not only increases accessibility but also promotes more intuitive and effective interaction with the device. This personalized approach underlines the potential of this system to positively transform the daily experience of users, offering them greater autonomy and independence.

In conclusion, although the results of this study did not achieve the absolute ideal, they revealed significant potential. With the necessary adjustments and improvements, the explored solution has the potential to become truly viable and effective.

Moreover, the results obtained can be extremely useful for future research in the field of assistive technologies, especially those that use electromyographic signals. This study contributes to a growing field, providing valuable insights and opening new paths for the development of innovative solutions that can improve the quality of life for people with disabilities.

5.2 Future Work

From the advancements achieved in this study, several promising directions for future research have been identified. These paths aim not only to overcome the challenges identified but also to expand the scope and effectiveness of the technology.

One of the main areas of focus will be the refinement of interpreting movements made with the bracelet. The current model operates under the assumption that the bracelet rotates around its axis. However, as previously observed, the actual movement of the forearm originates from the elbow, introducing additional angles not initially considered. Future iterations of the project will aim to integrate these considerations, with the goal of achieving a more accurate and fluid translation of cursor movement, more precisely aligning with the user's intentions. Adjustments in this direction are essential to improve user experience and the accuracy of interaction with the technology.

In parallel, there is considerable potential in exploring a wider range of filters and machine learning models. Diversifying these models and filters can provide a richer set of comparative data, potentially leading to improvements in performance and precision in motion detection. Such exploration will not only refine the current functionalities but could also uncover new insights into the relationship between filtered electromyographic (EMG) signals and the efficacy of the models. This research effort could significantly contribute to the evolution of the technology, expanding our understanding and ability to create even more efficient and precise solutions.

Another crucial aspect is the recognition of a broader range of muscular movements. Training the models to identify a wider variety of movements could pave the way for the inclusion of additional commands in the device, enriching its functionality. However, it is also essential to conduct training with data collected from individuals who have undergone amputations, considering they are the target audience for the proposed device. As previously mentioned, muscle atrophy may occur in the stump of amputated individuals due to lack of use, which may require a physiotherapy program or specific adjustments to the device to ensure its efficacy. This aspect highlights the importance of a personalized approach in the development of assistive technologies, ensuring that the device is not only functional but also adaptable to the individual needs of the users.

Although the initial focus of this study was computer interaction, there is a significant opportunity to expand this technology to other platforms, including smartphones and tablets. Such an extension would not only increase the versatility of the technology but

would also make it more accessible, opening up possibilities to benefit a broader user base. This expansion could facilitate daily use in a variety of contexts, thus enhancing the autonomy and quality of life of the users.

Finally, it becomes imperative to develop more ergonomic hardware for the bracelet. The current device, although functional, is not ideal for continuous and prolonged use due to comfort limitations. Future efforts should focus on ergonomic improvements, ensuring that users can comfortably use the device for extended periods. In summary, the path ahead is filled with opportunities to refine, expand, and implement the technology in ways that significantly improve user interaction and accessibility. Each of these areas presents its own set of challenges and potentials, and the journey to overcome them promises to be as rewarding as it is challenging.

Bibliography

[1] World Health Organization, *Disability and health*, https://www.who.int/news-room/fact-sheets/detail/disability-and-health, Accessed on: [December 22nd, 2023], 2023.

[2] W. H. Organization *et al.*, *World report on disability 2011*. World Health Organization, 2011.

[3] I. B. de Geografia e Estatística (IBGE), *Pesquisa Nacional de Saúde 2019 - Ciclos da Vida*. IBGE, 2021.

[4] Y. P. d. Souza, A. C. O. d. Santos, and L. C. d. Albuquerque, "Caracterização das pessoas amputadas de um hospital de grande porte em recife (pe, brasil)," *Jornal Vascular Brasileiro*, vol. 18, 2019. DOI: 10.1590/1677-5449.190064.

[5] M. LeBlanc, *"Give Hope - Give a Hand" - The LN-4 Prosthetic Hand*. 2011.

[6] L. L. Paiva and S. V. Goellner, "Reinventing life: A qualitative study on the cultural meanings attributed by amputees to body reconstruction through implantation of prosthetics.," *Interface - Comunic., Saude, Educ*, vol. 12, no. 26, pp. 485–497, 2008.

[7] European Parliament, *A estratégia ambiciosa da ue para a deficiência para 2021-2030*, https://www.europarl.europa.eu/news/pt/headlines/society/20200604STO80506/a-estrategia-ambiciosa-da-ue-para-a-deficiencia-para-2021-2030, Accessed on:[December 22nd, 2023], 2023.

[8] G. J. Tortora and B. H. Derrickson, *Principles of anatomy and physiology*. John Wiley & Sons, 2018.

[9] J. E. Hall, *Tratado de Fisiologia Médica*. Elsevier Health Sciences, 2021.

[10] G. J. Tortora and B. Derrickson, *Anatomy & physiology*. Wiley India Pvt Limited, 2014.

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

[12] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of applied biomechanics*, vol. 13, no. 2, pp. 135–163, 1997.

[13] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, no. 9, pp. 12 431–12 466, 2013, ISSN: 1424-8220. DOI: 10.3390/s130912431. [Online]. Available: https://www.mdpi.com/1424-8220/13/9/12431.

[14] A. V. Oppenheim, R. W. Schafer, M. A. Yoder, and W. T. Padgett, *Discrete-time signal processing*, en, 3rd ed. Upper Saddle River, NJ: Pearson, Aug. 2009.

[15] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, en. Prentice Hall, 1996.

[16] S. K. Mitra, *Digital signal processing: A Computer Based Approach*, en. McGraw-Hill Companies, Jan. 2006.

[17] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.

[18] C. Liu, J. Jiang, J. Jiang, and Z. Zhou, "Enhanced grid-connected phase-locked loop based on a moving average filter," *IEEE Access*, vol. 8, pp. 5308–5315, 2020. DOI: 10.1109/ACCESS.2019.2963362.

[19] H. Xue, M. Ruan, and Y. Cheng, "A fixed length adaptive moving average filter-based synchrophasor measurement algorithm for p class pmus," *Energies*, vol. 12, no. 21, 2019, ISSN: 1996-1073. DOI: 10.3390/en12214168.

[20] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2019.

[22] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[23] A. Alharin, T.-N. Doan, and M. Sartipi, "Reinforcement learning interpretation methods: A survey," *IEEE Access*, vol. 8, pp. 171 058–171 077, 2020.

[24] B. De Ville, "Decision trees," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 6, pp. 448–455, 2013.

[25] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 19, no. 5, p. 1088, 2019.

[26] Y. Ao, H. Li, L. Zhu, S. Ali, and Z. Yang, "The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling," *Journal of Petroleum Science and Engineering*, vol. 174, pp. 776–789, 2019.

[27] A. C. Lorena and A. C. De Carvalho, "Uma introdução às support vector machines," *Revista de Informática Teórica e Aplicada*, vol. 14, no. 2, pp. 43–67, 2007.

[28] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma, "Support vector machines classification with a very large-scale taxonomy," *Acm Sigkdd Explorations Newsletter*, vol. 7, no. 1, pp. 36–43, 2005.

[29] L. C. Coradine, R. V. V. Lopes, and A. F. Maciel, "Mineração de dados: Uma introdução," *Journal of the Brazilian Neural Networks Society*, vol. 9, pp. 168–184, 2011. DOI: 10.21528/LNLM-vol9-no3-art3.

[30] C. C. Aggarwal and C. K. Reddy, Eds., *Data Clustering* (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series). Boca Raton, FL: CRC Press, Aug. 2013.

[31] E. U. Oti, M. O. Olusola, F. C. Eze, and S. U. Enogwe, "Comprehensive review of k-means clustering algorithms," *criterion*, vol. 12, pp. 22–23, 2021.

[32] M. Heydarian, T. E. Doyle, and R. Samavi, "Mlcm: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19 083–19 095, 2022.

[33] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label classifier performance evaluation with confusion matrix," *Computer Science & Information Technology*, vol. 1, 2020.

[34] I. A. Faisal, T. W. Purboyo, and A. S. R. Ansori, "A review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture," *J. Eng. Appl. Sci*, vol. 15, no. 3, pp. 826–829, 2019.

[35] M. Dadafshar, "Accelerometer and gyroscopes sensors: Operation, sensing, and applications," *Maxim Integrated [online]*, 2014.

[36] Vida de Silício, *Sensor de movimento na cabeça*, `https://portal.vidadesilicio.com.br/sensor-movimento-cabeca/`, Accessed: [October 30th, 2023], 2023.

[37] W. Research, *Euler angles – from wolfram mathworld*, Accessed on: [September 2023]. [Online]. Available: `https://mathworld.wolfram.com/EulerAngles.html`.

[38] J. Zeitlhöfler, "Nominal and observation-based attitude realization for precise orbit determination of the jason satellites," Ph.D. dissertation, Jun. 2019.

[39] H. Liu, X. Wang, and Y. Zhong, "Quaternion-based robust attitude control for uncertain robotic quadrotors," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 406–415, 2015.

[40] D. S. Brezov, C. D. Mladenova, and I. M. Mladenov, "New perspective on the gimbal lock problem," *AIP Conference Proceedings*, vol. 1570, no. 1, pp. 367–374, 2013. DOI: `10.1063/1.4854778`.

[41] H. Jeon, J. Min, H. Bang, and W. Youn, "Quaternion-based iterative extended kalman filter for sensor fusion of vision sensor and imu in 6-dof displacement monitoring," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 23 188–23 199, 2022.

[42] H. A. Hashim, *Special orthogonal group so(3), euler angles, angle-axis, rodriguez vector and unit-quaternion: Overview, mapping and challenges*, 2021. arXiv: 1909.06669 [math.OC].

[43] D. M. Henderson, "Euler angles, quaternions, and transformation matrices for space shuttle analysis," Tech. Rep., 1977.

[44] E. Bernardes and S. Viollet, "Quaternion to euler angles conversion: A direct, general and computationally efficient method," *Plos one*, vol. 17, no. 11, e0276302, 2022.

[45] A. Kerr, M. A. Grealy, A. Kuschmann, R. Rutherford, and P. Rowe, "A co-creation centre for accessible rehabilitation technology," *Frontiers in Rehabilitation Sciences*, vol. 2, p. 820 929, 2022.

[46] J. Karki, S. Rushton, S. Bhattarai, and L. De Witte, "Access to assistive technology for persons with disabilities: A critical review from nepal, india and bangladesh," *Disability and Rehabilitation: Assistive Technology*, vol. 18, no. 1, pp. 8–16, 2023.

[47] L. de Witte, E. Steel, S. Gupta, V. D. Ramos, and U. Roentgen, "Assistive technology provision: Towards an international framework for assuring availability and accessibility of affordable high-quality assistive technology," *Disability and Rehabilitation: Assistive Technology*, vol. 13, no. 5, pp. 467–472, 2018.

[48] E. M. Smith, S. Huff, H. Wescott, *et al.*, "Assistive technologies are central to the realization of the convention on the rights of persons with disabilities," *Disability and Rehabilitation: Assistive Technology*, pp. 1–6, 2022.

[49] S. Feng, M. Tang, G. Huang, *et al.*, "Emg biofeedback combined with rehabilitation training may be the best physical therapy for improving upper limb motor function and relieving pain in patients with the post-stroke shoulder-hand syndrome: A bayesian network meta-analysis," *Frontiers in Neurology*, vol. 13, p. 1 056 156, 2023.

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

[51] O. Lennon, M. Tonellato, A. Del Felice, *et al.*, "A systematic review establishing the current state-of-the-art, the limitations, and the desired checklist in studies of direct neural interfacing with robotic gait devices in stroke rehabilitation," *Frontiers in Neuroscience*, vol. 14, p. 578, 2020.

[52] D. Stashuk, "Emg signal decomposition: How can it be accomplished and used?" *Journal of Electromyography and Kinesiology*, vol. 11, no. 3, pp. 151–173, 2001.

[53] S. Conforto, T. D'Alessio, and S. Pignatelli, "Optimal rejection of movement artefacts from myoelectric signals by means of a wavelet filtering procedure," *Journal of Electromyography and Kinesiology*, vol. 9, no. 1, pp. 47–57, 1999, ISSN: 1050-6411. DOI: https://doi.org/10.1016/S1050-6411(98)00023-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1050641198000236.

[54] B. Núñez-Montoya, M. Valarezo Añazco, A. Saravia-Avila, F. R. Loayza, E. Valarezo Añazco, and E. Teran, "Supervised machine learning applied to non-invasive emg signal classification for an anthropomorphic robotic hand," in *2022 IEEE ANDESCON*, 2022, pp. 1–6. DOI: 10.1109/ANDESCON56260.2022.9989874.

[55] N. F. Abdel-Maboud, S. S. Parusheva, M. Alfonse, and A.-B. M. Salem, "Comparative study of machine learning techniques based on tqwt for emg signal classification," in *2022 5th International Conference on Computing and Informatics (ICCI)*, 2022, pp. 374–377. DOI: 10.1109/ICCI54321.2022.9756080.

[56] N. Jarque-Bou, "Toward early and objective hand osteoarthritis detection by using emg during grasps.," *Sensors*, vol. 23 5, 2023. DOI: 10.3390/s23052413.

[57] M. Kristoffersen, "User training for machine learning controlled upper limb prostheses: A serious game approach," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, 2021. DOI: 10.1186/s12984-021-00831-5.

[58] S. Briouza, "Emg signal classification for human hand rehabilitation via two machine learning techniques: Knn and svm," *2022 5th International Conference on Advanced*

Systems and Emergent Technologies (IC_ASET), pp. 412–417, 2022. DOI: 10.1109/IC_ASET53395.2022.9765856.

[59] H. A. Jaber, M. T. Rashid, H. Mahmood, and L. Fortuna, "Incremental adaptive gesture classifier for upper limb prostheses," *IEEE Sensors Journal*, vol. 22, no. 14, pp. 14 273–14 283, 2022.

[60] X. Xiao, Y. Fang, X. Xiao, J. Xu, and J. Chen, "Machine-learning-aided self-powered assistive physical therapy devices," *ACS nano*, vol. 15, no. 12, pp. 18 633–18 646, 2021.

[61] D. Mulfari, G. Meoni, and L. Fanucci, "Machine learning in assistive technology: A solution for people with dysarthria," in *Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good*, 2018, pp. 308–309.

[62] M. Cury, E. Whitworth, S. Barfort, *et al.*, "Hybrid methodology: Combining ethnography, cognitive science, and machine learning to inform the development of context-aware personal computing and assistive technology," in *Ethnographic Praxis in Industry Conference Proceedings*, Wiley Online Library, vol. 2019, 2019, pp. 254–281.

[63] J. White, T. Kameneva, and C. McCarthy, "Vision processing for assistive vision: A deep reinforcement learning approach," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 1, pp. 123–133, 2021.

Appendix A

Original Project Proposal

INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Curso de Mestrado em Engenharia Industrial - Eletrotécnica

Ano lectivo de 2022/2023

Device control system base on classified EMG signals: a machine learning approach

Orientador: José Luís Sousa de Magalhães Lima

Co-orientador: Laercio Simas Mattos

1 Objectivo

A ideia consiste em desenvolver um sistema capaz de controlar dispositivos eletrónicos diversos através dos sinais EMG de pessoas que sofreram de alguma amputação sem a necessidade do intermédio de uma prótese, com intuito de reduzir o tempo de resposta destes equipamentos uma vez que próteses em geral possuem um delay intrínseco.

2 Detalhes

A ausência de meios e ferramentas adaptadas faz com que pessoas que tiveram algum dos seus membros superiores amputados passem por inúmeras dificuldades na execução de tarefas quotidianas, afastando-as ainda mais do que é dito ser uma vida "normal". É neste cenário onde o projeto a ser desenvolvido será aplicado. Uma vez que, por mais avançados que estejam os estudos relacionados com próteses, as mesmas não são capazes de reproduzir todos os movimentos do membro biológico, além de possuírem um alto valor de mercado. Esses fatores limitam a sua utilização pois possuem um delay intrínseco entre o comando e a execução devido a questões mecânicas e de processamento de sinais. Então como pessoas amputadas ainda são capazes de produzir sinais eletromiográficos (EMG) correspondentes aos movimentos mesmo após a perda do membro, a ideia consiste em desenvolver um sistema capaz de controlar dispositivos eletrónicos diversos através desses sinais EMG sem a necessidade do intermédio de uma prótese, reduzindo o tempo de resposta destes equipamentos e permitindo algumas atividades anteriormente limitadas pelas questões citadas. Por exemplo, em videojogos que exigem uma resposta rápida do utilizador, o sistema substituiria a consola, outras aplicações são a possível substituição dos mouses convencionais, e atuação no controle do touch screen de dispositivos móveis. A ideia consiste no desenvolvimento de um produto portátil e compacto constituído por um bracelete com sensores que encaminharão os dados obtidos para um dispositivo controlador que será ligado aos aparelhos eletrónicos a serem comandados.

3 Metodologia de trabalho

O desenvolvimento do projeto necessita da utilização de ferramentas de processamento e armazenamento dos dados dos sensores EMG, acelerômetro e giroscópio para o treinamento da IA. Também será necessário do desenvolvimento de um dispositivo eletrônico onde ocorrerão a interpretação dos dados fornecidos pelos sensores, o mesmo será responsável por transmitir os comandos do usuário para os eletrônicos a serem controlados.

Dimensão da equipa: 3 pessoas

Recursos necessários: Recursos de hardware disponibilizados nos laboratórios do Centro de Investigação em Digitalização e Robótica Inteligente (CeDRI) e recursos de software.

Appendix B

Papers

B.1 Paper 1

This paper, entitled " Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal.", was accepted and presented on International Conference on Optimization, Learning Algorithms and Applications 2023 (OL2A 2023), which took place in Ponta Delgada – Portugal during September 27th to 29th, 2023.

Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal

Ana Barbosa[1,2,3][0009−0002−0121−5401], Edilson Ferreira[1,2,3][0009−0009−7786−3166], Vinicius Grilo[1,2,3][0009−0004−4099−2566], Laercio Mattos[2], and José Lima[1,3][0000−0001−7902−1207]

[1] Research Centre in DIgitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
{viniciusgrilo, jllima}@ipb.pt, edilsonsfc@live.com, cb_ana@outlook.com
[2] Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), Minas Gerais, Brasil
laercio.mattos@gmail.com
[3] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões Montanha (SusTEC), Instituto politécnico de Bragança, 5300-253 Bragança, Portugal

Abstract. Our current society faces challenges in integrating individuals with disabilities, making this process difficult and painful. People with disabilities (PwD) are often mistakenly considered incapable due to the difficulties they face in daily tasks due to the lack of adapted means and tools. In this context, assistive technologies play a crucial role in improving the quality of life for these individuals. However, assistive technologies still have various limitations, making research in this area essential to enhance existing solutions and develop new approaches that meet individual needs, aiming to promote inclusion and equal opportunities. This paper presents a research project that focuses on the study of electromyography (EMG) signal processing generated by individuals who have undergone amputations. These signals are essential in assistive technologies, such as myoelectric prostheses. The study focuses on the impact of different filters and machine learning training methods on this processing. The results of this study have the potential to provide relevant findings for the development of more efficient assistive technologies. By understanding the processing of EMG signals and applying machine learning techniques, it is possible to improve the accuracy and response speed of prosthetics, increasing the functionality and naturalness of movements performed by users, as well as paving the way for the emergence of new technologies.

Keywords: Assistive technologies · Electromyography (EMG) signal processing · Machine learning.

1 Introduction

In our society, certain body features are associated with beauty, health, and perfection, while others are seen as flaws. Different bodies are often viewed as disadvantages, complicating the integration of individuals carrying these features, leading to a challenging and painful process. In this context, people with

disabilities (PWD) are frequently considered imperfect and incapable. This perception is further exacerbated by the lack of adapted means and tools, resulting in numerous difficulties in performing everyday tasks. Consequently, these individuals are further alienated from what society deems a "normal" life.

Amputees, for example, face significant physical and psychological challenges, and the possibility of using assistive technologies such as prosthetics, which represent the most comprehensive solution to this issue, plays a crucial role in rehabilitation and improving the quality of life. However, even with advancements in assistive technology, these technologies still have limitations in replicating all movements of the biological limb and are generally expensive.

In light of this reality, this article explores the use of EMG signals, which represent the electrical activity associated with muscle contraction, along with machine learning algorithms aiming to enhance existing technologies and develop new solutions, focusing on the impacts of different filters on the training methods. Despite the evident applicability of EMG signals and machine learning techniques for the development of advanced assistive technologies, there is still a gap in recent studies evaluating the impact of different types of filters on the performance of machine learning algorithms in classifying multi-channel EMG signals. Previous research has mainly focused on signal classification applications and specific machine learning methods, without conducting a systematic comparison of filters in conjunction with these different classification algorithms.

Therefore, this article employs an approach that applies four filters (bandpass, Kalman, moving average, and Nocth) to EMG signals. These filtered signals are used to train three supervised machine learning methods: Decision Tree, Random Forest, and SVM, as well as an unsupervised clustering method. The goal is to identify the most efficient combinations for the classification and interpretation of EMG signals

2 Theoretical Background and Related Work

In this section, an overview of the relevant theoretical foundations related to the study topic will be presented. Fundamental concepts and theories will be addressed, establishing a solid foundation for understanding and developing the research at hand.

2.1 EMG Signal

The main focus of this study is centered on the analysis and treatment of a specific type of signal that originates from muscle movement.

Muscles play a crucial role in generating force and movement in the human body. Muscle force is the result of muscles' ability to convert chemical energy into mechanical energy. There are three types of muscle tissue in the human body: smooth, cardiac, and skeletal. In this context, our focus is on skeletal muscles, which are directly involved in body movement.

To understand muscle contraction and stretching, it is necessary to explore the macroscopic and microscopic anatomy of muscles. Muscles are composed of muscle fibers, which in turn contain myofibrils. These myofibrils consist of myosin and actin filaments, responsible for muscle contraction. The sarcoplasm,

the fluid filling the spaces between myofibrils, contains various substances and protein enzymes. The sarcoplasmic reticulum stores calcium ions (Ca++) and releases them during muscle contraction.

Voluntary control of skeletal muscles is achieved through somatic motor neurons. Action potentials, which are electrical signals generated in response to stimuli, originate at the neuromuscular junction where the somatic motor neuron connects with the muscle fiber. Communication between these cells occurs through the release of a neurotransmitter called acetylcholine (ACh). When ACh binds to receptors on the muscle fiber, ion channels open, allowing the flow of sodium ions (Na++) and triggering a muscle action potential. This action potential propagates along the muscle fiber, causing the sarcoplasmic reticulum to release stored calcium ions, resulting in muscle contraction. [10]

This brings us to electromyography (EMG), a technique used to record and monitor the electrical activity of muscles. It captures the summation of action potentials generated in muscle fibers during contraction. EMG signals provide valuable information about muscle function and are widely used in studies of muscular diseases, assessment of muscle performance during physical exercises, and the development of assistive technologies. Understanding myoelectric signals and the use of EMG are essential for advancing in the field of rehabilitation, myoelectric prosthetics, and other applications related to the interface between muscles and technology [18].

2.2 Signal filters: definition and types

In the present context, the effects of filters on signal processing will be explored, as they play a role in modifying the spectral content of these signals. The various types of filters employed in this research will be discussed, with the purpose of describing their characteristics and specific applications. The main objective is to provide an in-depth understanding of these filters, thus establishing a solid foundation for comprehending the present study.

- **Bandpass Filter** A bandpass filter allows a specific range of frequencies to pass while attenuating or rejecting frequencies outside that range. It can be created by combining a highpass and a lowpass filter [22, 23].
  The gain of a bandpass filter is ideally maximum within the passband and minimum outside of it, and the attenuation rate is determined by the design requirements [20, 22, 23].
  Designing a bandpass filter involves determining the center frequency, the width of the passband, and the attenuation rate for frequencies outside the passband. There are different configurations of bandpass filters, such as Butterworth and Chebyshev, each with its own frequency response and implementation characteristics [20, 22, 23].
- **Kalman Filter** The Kalman Filter is a mathematical estimator used to solve the linear-quadratic problem by estimating the state of a linear dynamic system affected by random noise. It provides optimal statistical estimation of the state based on measurements that are linearly related to the state but corrupted by white noise [9].
  From a practical perspective, the Kalman Filter has found wide applications in controlling complex dynamic systems such as manufacturing processes,

aircraft, ships, and spacecraft. It allows for inferring missing information from indirect and noisy measurements, enabling a better understanding of system behavior. Furthermore, the Kalman Filter is useful for predicting future behavior in uncontrollable dynamic systems like river flows, celestial body trajectories, or commodity prices [9].

- **Moving Average Filter** Moving Average Filters (MAFs) are cost-effective and easy to implement linear-phase FIR filters. They operate by continuously calculating the average value of input signals within a sliding time window T_W. This helps reduce undesired harmonics and distortion. MAFs exhibit a low-pass filtering behavior, providing unity gain at zero frequency and attenuating frequencies at regular intervals determined by the window size. They are commonly used as ideal low-pass filters under specific conditions [15, 28]. Due its characteristics, the MAF suffers a response delay on T_W. Typically, a smaller T_W means a shorter MAF response delay; however, the filtering performance is also dependent on T_W [28].
- **Notch Filter** A notch filter is a linear filter that selectively eliminates a specific frequency component, known as the notch frequency, while maintaining unit gain at other frequencies. These filters are widely used in various applications where targeted frequency components need to be removed [23].
  However, FIR notch filters often have a wide bandwidth, resulting in significant attenuation of nearby frequencies. To address this, poles can be introduced into the filter to create resonance and reduce the notch bandwidth. However, this may introduce a slight ripple in the passband, which can be minimized by adding more poles and/or zeros. Nevertheless, this approach is considered ad hoc and relies on trial-and-error, presenting a notable challenge [23].

2.3 Machine learning: concepts and applications

Machine Learning is a field of artificial intelligence research in prominence and in constant evolution today that focuses on the development of algorithms and models capable of making predictions or decisions from data collection without the need for explicit programming, as in classical programming. These algorithms cover a wide range of concepts and techniques that allow the extraction of patterns and decision-making from data sets, with applications in several areas, such as signal processing, computer vision, data analysis, among others [26].

For Machine Learning concepts to be applied, a few core elements are required. First of all, is needed to have a relevant amount of data. This data is used to train the model and needs to be representative and contain meaningful information for the problem. The preparation and processing of data are also extremely important steps in building a good machine learning model.

The learning techniques of a machine learning model can be divided into three types:

- **Supervised:** Supervised learning involves the presence of an external instructor who imparts knowledge about the environment through sets of examples in the format of input-output pairs. The machine learning algorithm

leverages these examples to extract a representation of the acquired knowledge. The primary goal is to generate a representation that can accurately produce desired outputs for novel inputs that have not been encountered before [11].

- **Unsupervised:** Unsupervised learning operates without the presence of a teacher, meaning there are no labeled examples provided. This algorithm, employed in such scenarios, strives to acquire a representation or grouping of the given inputs based on a quality measure. These techniques are predominantly utilized when the goal is to uncover patterns or trends that facilitate comprehension of the data [24].
- **Reinforcement:** Reinforcement Learning (RL) is a dynamic method to learn optimal actions through trial and error in an interactive environment. Unlike other approaches that rely on pre-existing labeled data, RL operates in an environment where the agent learns by taking actions and observing the resulting feedback or rewards. This feedback serves as a signal for the algorithm to adjust its decision-making process and maximize its long-term objective, known as the reward function [3].

2.4 Training methods

Some examples of training methods are:

- **Decision tree** emerged as versatile tools for prediction and classification, and were one of the pioneering statistical algorithms to be implemented in electronic form during the widespread adoption of digital circuits for computational purposes in the last decades of the 20th century. Over time, they have evolved into highly adaptive and computationally intensive methods that find applications in a variety of disciplines. Decision trees now serve as general-purpose mechanisms for prediction and classification, as well as indispensable components in artificial intelligence, machine learning, knowledge discovery, and inductive rule construction. Their use extends to a diverse range of tasks in data mining, knowledge discovery, machine learning, and artificial intelligence [8].
  A key feature of decision trees lies in their recursive subset approach, employed to analyze a target data field based on the values of associated input fields or predictors. This methodology facilitates the creation of partitions within the data set, resulting in descending subsets of data called leaves or nodes. Notably, these leaves or nodes contain progressively similar intra-leaf (or intra-node) target values, while exhibiting progressively different inter-leaf (or inter-node) values at each level of the tree. This hierarchical organization allows decision trees to effectively discern patterns and relationships within the data, enabling robust predictive and classification capabilities. This unique feature, combined with the computational intensity and interdisciplinary applicability of decision trees, contributes to their broad utility in a variety of research and practice domains [8].
- **Random forest** The method called random forest is an extension of decision tree models and consists of creating multiple individual decision trees through a process called "ensemble learning". Each tree in a random forest

is built and trained using variables or a set of random information in the dataset, allowing each tree to learn different characteristics of the data set decision trees [27].
When a prediction is required, after the model is trained, each tree performs its individual prediction and then a final decision is made by the model. This collective process reduces overfitting and consequently improves the generalizability of the model.
One of the main advantages of Random Forests is their ability to handle large data sets while maintaining robustness, since in a sense the problem is divided into several decision trees. Furthermore, they are less sensitive to outliers and noise when compared to individual decision trees [4].

- **Support Vector Machine or SVM** is a widely used machine learning algorithm for classification and regression. Its main goal is to find the optimal hyperplane that separates the input data into distinct classes, seeking the largest margin between the closest training samples. Such samples are called support vectors and play a crucial role in the definition of the hyperplane [17].
  For non-linearly separable data sets we use mapping tricks to a higher dimensional space, where linear separation through a hyperplane is possible [17].
  This machine learning technique has good generalization capabilities, since the optimization process is based on margin maximization, which tends to avoid unwanted overfittings. The possibility of application to non-linearly separable problems makes SVMs extremely flexible and able to handle a wide variety of classification and regression problems [17].
  However, the training process of an SVM can become computationally demanding for large data sets, so it is necessary to evaluate the size of the data set when using the SVM as a learning algorithm [16].
- **Clustering** is a technique used to group similar objects or data points based on their intrinsic characteristics, aiming to discover patterns, structures, or relationships within a dataset without prior knowledge of the specific groups or classes [7].
  The challenge lies in dividing the data into groups in a way that maximizes their similarity. Clustering has been extensively studied due to its diverse applications in data mining and machine learning, including summarization, segmentation, and targeted marketing [2].
  There is a wide variety of problems that can be solved with clustering algorithms, such as Collaborative Filtering, Data Summarization, Biological Data Analysis, and Social Network Analysis [2].

2.5 Related Work

The study of electromyographic (EMG) signals has advanced significantly, with applications in medicine, biomedical engineering, biomechanics, and neuroscience [19]. Improvements in the acquisition, processing, and interpretation of EMG signals have been achieved, allowing for a greater understanding of muscle function and the diagnosis of neuromuscular disorders, in addition to the control of assistive devices [14]. Techniques such as decomposition into individual motor units, machine learning, and brain-machine interfaces have been explored [25].

However, EMG signals can be compromised due to noise and artifacts. Studies have sought ways to remove these noises and recover the actual muscle signal. [6] analyzed four filtering procedures, finding that the wavelet method stood out for preserving all information and having greater accuracy.

Recent studies have also explored the application of machine learning in EMG signal analysis. One study developed a control system for a robotic hand using an Artificial Neural Network (ANN), achieving an accuracy of 84.78% in the identification of hand movements [21]. Another study used the Q-Tuned Wavelet Transform (TQWT) for feature extraction and applied various machine learning classifiers, achieving the highest success rate with Random Forest (RF), 98.64% in the classification of neuromuscular disorders [1].

There are also works exploring the use of EMG features in early detection of medical conditions, such as hand osteoarthritis [12]. In assistive technology, EMG signal processing and machine learning techniques are used to enhance prosthesis functionality, increasing accuracy and response speed [13]. Studies like [5] also demonstrate significant advancements in motion estimation and EMG signal classification to enhance prosthesis functionality.

In sum, the field of EMG signal processing and the use of machine learning is very promising. Advancements in this field have the potential to improve the quality of life for many individuals.

3 Methodology

This section describes the methods and procedures used in this study to collect and analyze the data. It provides a clear overview of how the research was conducted.

3.1 Data Acquisition

For data acquisition, the Myo Gesture Armband, developed by THALMCLABS, was used, which is capable of capturing electromyographic (EMG) signals from the forearm muscles. In addition to EMG signals, the armband also provides additional information such as accelerometers and gyroscopes. Although this device was not originally designed for this specific purpose, it can be used as a non-invasive tool to capture the electrical signals generated by the muscles during contraction. With its 8 channels, it is possible to obtain signals from distinct muscles during movement, allowing comprehensive data acquisition.

The main objective of this step was to create a suitable database for training the machine learning methods presented earlier. For this purpose, data were collected from 12 volunteers, including 5 women and 7 men, all between the ages of 20 and 30, in good health and without amputations. The armband was placed on the right forearm of each volunteer, and 60 files were recorded for each of them. Of these, 30 files correspond to the execution of the hand closing and opening movement, varying in duration and the force applied in each capture. The other 30 files correspond to the absence of movement, meaning the volunteer did not perform any specific movement. Each capture had a duration of 3 seconds. The stored files vary in the number of samples, ranging from 600 to 800 samples, depending on the file.

Subsequently, the collected data was saved in CSV files, where each column represents an EMG channel, facilitating data processing. In Figure 1, It is possible to observe at the top in black the 8 channels during the execution of the movement, while at the bottom of the figure we have in blue the same 8 channels during the absence of movement.. These collected and organized data files provide the foundation for the subsequent steps of analysis and training of the machine learning algorithms.

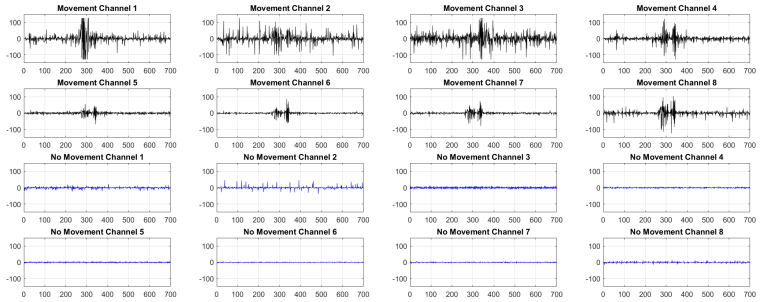


Fig. 1. Comparison of 8-channel EMG signals during movement execution (Black) and absence of movement (Blue).

Upon observing Figure 1, it is evident that the movement is characterized by signals with higher amplitudes. Specifically, when the volunteer performs the movement, the EMG signals exhibit larger amplitudes, whereas in the absence of movement, only small amplitudes are observed. Furthermore, it can be noted that, for the hand-closing movement, the signal is not uniformly reproduced across all channels, indicating varied muscular activation intensities based on their respective contributions to the execution of the movement.

3.2 Data Filtering

After properly storing the data, we proceeded to process it. In order to gain a better understanding of the filters that operate in the frequency domain, we performed the Fourier transform to visualize the frequencies present in the signal, as well as their magnitude. Figure 2 displays the applied transform for each channel, represented in black for the movement data and in blue for the non-movement data.

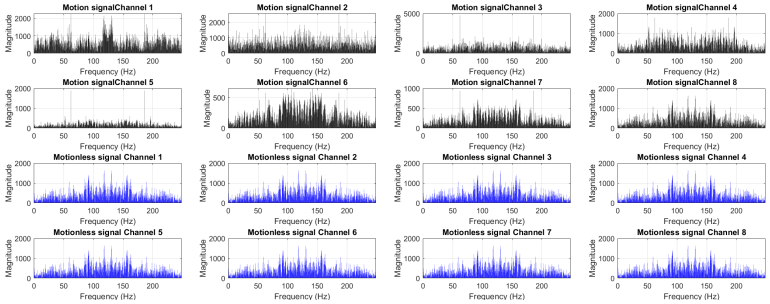


Fig. 2. Fourier Transform of Raw Signal during movement execution (Black) and absence of movement (Blue).

The first filter applied was the bandpass filter, with a passband ranging from 5 Hz to 500 Hz. This range was chosen based on the frequency range typically present in healthy skeletal muscle action potentials.

Next, with the cutoff frequencies defined, the bandpass filter was applied to all files in the database. The filter was applied to each column of the files, using the Fourier transform to perform the filtering. The result of this filtering can be observed in Figure 3, where the original signal is represented in black and the filtered signal is represented in red. For better visualization, only channel one is being presented.

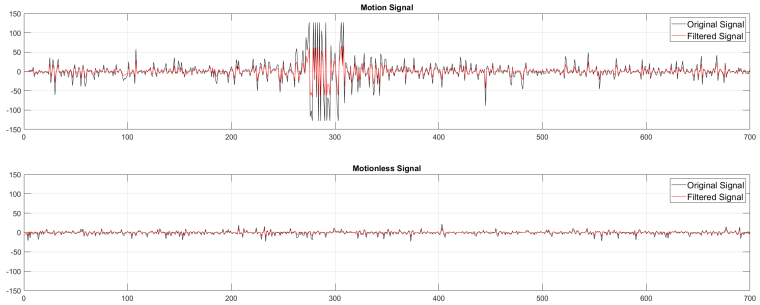


Fig. 3. Bandpass Filtering

It is evident that the filter performed well in attenuating the signal, even with the removal of a relatively narrow frequency range. It is important to note that although the filter was designed to eliminate all frequencies below 5 Hz and above 500 Hz, only frequencies below 5 Hz were effectively eliminated. This is because in the original signal, the maximum frequency present is around 250 Hz, so there were no frequencies above 500 Hz to be removed. The application of the Fourier transform to the filtered signal allows visualizing the filter's effect. As shown in Figure 4, frequencies below 5 Hz were efficiently suppressed after the filter application.

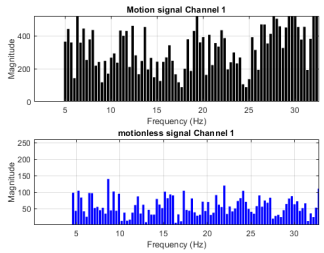


Fig. 4. Fourier Transform of Bandpass Filtering

Another filter used for frequency elimination is the notch filter, which aims to remove a specific frequency. In this case, the filtering process was performed to eliminate the 50 Hz frequency, which represents the interference caused by the power grid in Portugal, where the data was captured. Similar to the passband filter, the notch filter is also applied to each column of the files. The effectiveness of the filtering can be observed in Figure 5, where the original signal is represented in black and the filtered signal is represented in red. It is important to note that only channel one is presented for better visualization of the results.

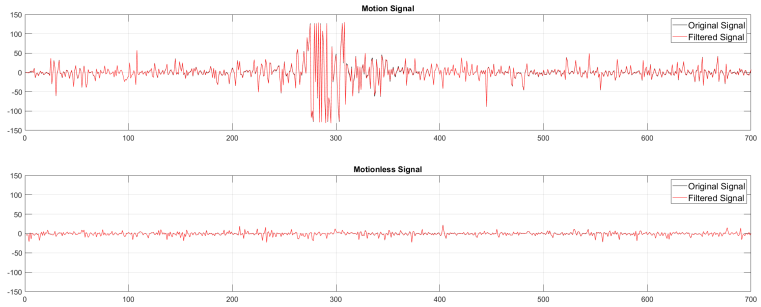


Fig. 5. Notch Filtering

By performing the Fourier transform of the filtered signal, as illustrated in Figure 6, we can observe that the filter was effective in attenuating the frequencies near 50 Hz. However, it is notable that the elimination of this single frequency did not have a significant impact on the signal, as the filtered signal overlaps almost completely with the original signal. This indicates that the 50 Hz interference caused by the power grid was not a dominant source of distortion in the captured signal.

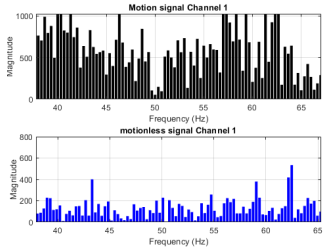


Fig. 6. Fourier Transform of Notch Filtering

The next step involved applying the moving average filter, testing different window sizes, including 3, 5, 10, 50, and 100 samples. The result of this filtering can be observed in Figure 7, where the left side shows the filtering of the signal with movement, and the right side shows the filtering of the signal without movement. It is noticeable that as the window size increases, the signal is more attenuated, meaning there is a greater smoothing of the signal.

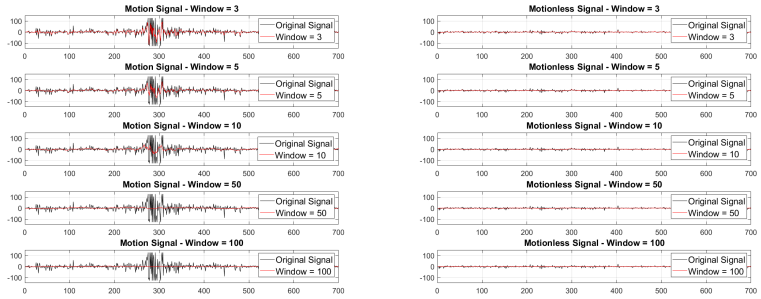


Fig. 7. Moving Average Filtering

Finally, the last filter applied was the Kalman filter, which is implemented in the prediction and update steps. In the prediction step, the filter estimates the next state of the system based on the previous state, while in the update step, the filter corrects the estimated state based on the actual measurement. The result of this filtering can be observed in Figure 8, where a significant attenuation of the signal is evident.

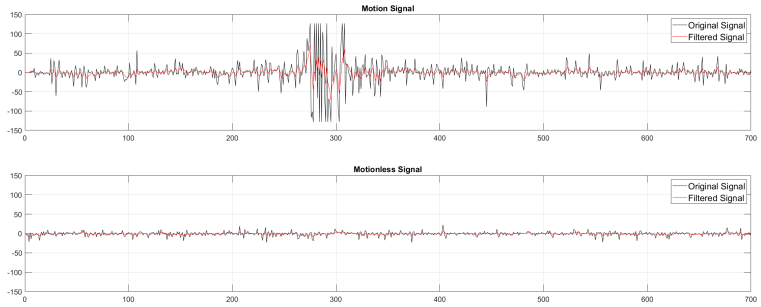


Fig. 8. Kalman Filtering

3.3 Training of Machine Learning Methods

After the proper data prepossessing, it's time to start training the models. We have selected three supervised models and one unsupervised model, which will be explained below. Training and evaluating these models will allow valuable insights into the relationship between filtered EMG signals and the accuracy of the tested models.

- **Supervised models** In this supervised training process, the general operating structure remains consistent, differing only in the applied training model, which includes Decision Trees, Random Forests, and Support Vector Machines (SVM). Each of these models has its peculiarities, but the overall workflow to apply them remains the same.
  The first step of this process involves reading data from CSV files. These files contain data classified as 'movement' and 'non-movement,' which form the two classes that the models will try to predict. The data is read and combined into a single DataFrame, one for each class, to facilitate subsequent processing. This step is crucial as it allows all information to be gathered into a single coherent data structure.
  After data consolidation, the next step is to split the data into training, validation, and test sets. These sets represent 70%, 20%, and 10% of the data, respectively. Splitting the data this way is a common practice in machine learning. The training set is used to train the model, the validation set is used to adjust hyperparameters and assess the model's performance during training, and the test set is used to evaluate the model's performance on unseen data. One point to consider is that many machine learning models cannot handle null values. To address this, in this process, any missing values in the data are filled with the mean of existing values.

After data preparation, the selected model is trained. The model learns patterns from the training data that enable it to predict whether a given example belongs to the 'movement' or 'non-movement' class. Then the model is evaluated by running the validation and test sets and comparing the model's predictions with the actual labels. This provides a measure of the model's performance, usually expressed as a percentage of correct predictions, known as accuracy. Finally, the trained model is saved to a file, allowing it to be reused later without the need for retraining.

– **Unsupervised model**

In this context, the process of unsupervised training maintains a similar operational structure to supervised cases, albeit with some specific nuances. The learning model used here is k-means, a clustering algorithm. The first step involves reading data from CSV files and combining this information into a single DataFrame. Similar to the supervised case, this data is then partitioned into training, validation, and test sets, following the proportions of 80%, 10%, and 10%, respectively. Like many models, k-means also does not handle missing values well, so these values are filled with the mean of existing ones.

The data preparation phase continues with the selection of specific columns from the DataFrame that represent Electromyography (EMG) signals. These signals are then normalized to ensure that all signals are on the same scale, facilitating the clustering process. Next, the k-means algorithm is applied to the training data, learning the centroids of the clusters. Once trained, the model is used to make predictions on the validation and test sets, and the predicted labels are stored for further evaluation.

The model's performance is evaluated by comparing the predicted labels with the true ones, generating an accuracy measure. Finally, just as in supervised cases, the trained model is saved in a file to be reused later, eliminating the need for retraining.

The four machine learning models, including K-means, Decision Tree, Random Forest, and SVM, consistently utilize "train_test_split", "test_size", "random_state", and "SimpleImputer". Specific parameters include "n_clusters" for K-means, and the default values of "gini" and "min_samples_split" for Decision Tree and Random Forest, as well as C and gamma for SVM. Although adjustable, these parameters remained unchanged to prevent the introduction of an additional variable to the analysis, which could lead to overfitting of the data. The focus of the study is the impact of the filter on the data, maintaining the model parameters at their default values to establish a consistent reference.

4 Results

The results of the accuracy calculations for each of the training methods are presented in Table 1, showing the accuracies of the supervised training methods for each of the filters applied to the dataset.

Table 1. Accuracy of Trained Models

		Moving Average						
	BandPass	**3**	**5**	**10**	**50**	**100**	**Kalman**	**Notch**
Decision Tree								
Accuracy Test	0,811	0,775	0,762	0,769	0,825	0,866	0,771	0,805
Accuracy Validation	0,809	0,775	0,759	0,772	0,823	0,865	0,770	0,802
Random Forest								
Accuracy Test	0,876	0,850	0,838	0,845	0,882	0,901	0,843	0,871
Accuracy Validation	0,874	0,850	0,836	0,844	0,881	0,900	0,843	0,869
SVM								
Accuracy Test	0,819	0,812	0,805	0,807	0,799	0,788	0,806	0,812
Accuracy Validation	0,820	0,810	0,803	0,806	0,798	0,788	0,806	0,811

Based on the obtained results, it can be observed that the Random Forest model consistently demonstrated the highest accuracies for all applied filters. In particular, the Moving Average filter with a window size of 100 exhibited exceptional performance, achieving an accuracy of 0.90. The SVM and Decision Tree models showed significant variations in performance depending on the applied filter. Except for the Moving Average filter with a window size of 100, which yielded satisfactory results of 0.866 in the test set and 0.865 in the validation set for the Decision Tree method, no other combination surpassed the performance of the Random Forest. These results suggest that the Random Forest was the most effective method for the analyzed dataset.

When evaluating the different filters, it was consistently observed that the Moving Average filter with a window size of 100 displayed the best performance in the Decision Tree and Random Forest models. Additionally, the Moving Average filter with a window size of 50 yielded satisfactory results, while the BandPass filter also achieved good results. The Notch filter and Moving Average filters with window sizes of 3 and 10 produced intermediate results. The Kalman filter and Moving Average filter with a window size of 5 exhibited inferior performance compared to the other filters in both machine learning models considered.

Regarding the SVM method, it was found that the BandPass filter obtained the best performance, followed by the Notch filter. The Moving Average filters with window sizes of 3, 5, and 10, as well as the Kalman filter, showed intermediate results, while the Moving Average filters with window sizes of 50 and 100 had the lowest accuracy compared to the other filters applied to this method.

Analyzing the validation results, it was observed that the BandPass filter performed well in all methods, being particularly effective in SVM. The Notch filter also stood out by consistently yielding accuracies above 0.800 across all methods. On the other hand, the Moving Average filter with a window size of 5 displayed lower results in all methods, suggesting a limited impact on improving accuracy compared to the other filters.

The results indicate that the SVM method exhibited less variation in performance with respect to the different filters, suggesting that this method may be less sensitive to the considered filter types and may not benefit as much from the applied variations. The analysis of moving average filters reveals significant

differences in the performance of different machine learning models depending on the window size used. The choice of window size for the moving average filter should consider the nature of the machine learning model to be used, as well as the trade-off between noise reduction and data reactivity preservation.

For instance, window sizes of 100 and 50 demonstrated the best performance in Decision Tree and Random Forest models. This can be attributed to the larger window's ability to more effectively smooth out data noise, allowing the model to focus on general trends rather than short-term fluctuations. However, larger window sizes may introduce more significant delays, which could be a disadvantage in real-time applications.

On the other hand, the same window sizes of 100 and 50 showed inferior performance in the SVM method, which seeks to find the optimal hyperplane that maximizes the margin between classes. Excessive data smoothing, which can occur with a large moving average filter window, may make the features of support vectors less distinct, and these features are crucial for constructing the SVM's decision boundary. In contrast, smaller window filters may not smooth out data noise sufficiently for Decision Tree and Random Forest models, resulting in a model that could be overly influenced by short-term variations. However, these smaller window filters may be useful for SVM, preserving some important details for constructing the decision boundary.

These assumptions are based on the observations of the results and may vary depending on the nature and quality of the data under consideration. Therefore, the choice of window size should be made carefully, considering the machine learning model to be used and the specific characteristics of the data.

The analysis of the unsupervised model showed considerably lower training accuracies (0.535 for test data and 0.534 for validation data) compared to the supervised methods discussed earlier. These results suggest that the clustering method may be less effective in classifying the dataset used in this article compared to supervised methods. However, it is worth noting that further tests, including the use of filtered data or a different dataset size, can provide a more reliable evaluation of the clustering method's performance in comparison to the supervised methods

The goal of testing this method was to explore the possibility of classifying motion without the need for filtering, which could be of interest in the future for classifying multiple motions. The idea behind the used clustering was to group the data based on similar characteristics. Eliminating the need for filtering would reduce the processing time required for real-time motion classification, contributing to reducing the response time of related technologies. However, the results obtained were not satisfactory for the dataset used in this project.

5 Conclusion

This study focuses on analyzing electromyographic (EMG) signals from individuals with amputations and utilizing machine learning techniques for training. It explores the effects of different filters (bandpass, Kalman, moving average, and

notch) and training methods (decision tree, random forest, and support vector machines) to optimize the classification and interpretation of EMG signals.

This research contributes to the field of assistive technology by demonstrating the potential of using EMG signals and machine learning to enhance the functionality and efficiency of assistive technologies. By investigating various filter and training method combinations, it aims to advance assistive technologies, promote inclusivity, social integration, and equal opportunities for individuals with disabilities.

In summary, this study investigated the effectiveness of specific filters in signal analysis, creating possibilities for future research. To further advance this field, future studies will include different types of filters and explore various combinations. Comprehensive analyses of the results, including testing multiple movements, are crucial. Additionally, upcoming tests will analyze EMG signals from individuals with amputations. These future efforts will improve our understanding of filter applications and their impact on outcomes, leading to the development of more accurate and efficient techniques in signal analysis..

Acknowledgment

The authors are grateful to CeDRI (UIDB/05757/2020, UIDP/05757/2020), SusTEC (LA/P/0007/2021) and SmartHealth (NORTE-01-0145-FEDER-000045).

References

1. Abdel-Maboud, N.F., Parusheva, S.S., Alfonse, M., Salem, A.B.M.: Comparative study of machine learning techniques based on tqwt for emg signal classification. In: 2022 5th International Conference on Computing and Informatics (ICCI). pp. 374–377 (2022). https://doi.org/10.1109/ICCI54321.2022.9756080
2. Aggarwal, C.C., Reddy, C.K. (eds.): Data Clustering. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, CRC Press, Boca Raton, FL (aug 2013)
3. Alharin, A., Doan, T.N., Sartipi, M.: Reinforcement learning interpretation methods: A survey. IEEE Access **8**, 171058–171077 (2020)
4. Ao, Y., Li, H., Zhu, L., Ali, S., Yang, Z.: The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. Journal of Petroleum Science and Engineering **174**, 776–789 (2019)
5. Briouza, S.: Emg signal classification for human hand rehabilitation via two machine learning techniques: knn and svm. 2022 5th International Conference on Advanced Systems and Emergent Technologies ($\mathrm{IC}_A SET$)$pp.412--417(2022).https : //doi.org/10.1109/IC_ASET53395.2022.9765856$
6. Conforto, S., D'Alessio, T., Pignatelli, S.: Optimal rejection of movement artefacts from myoelectric signals by means of a wavelet filtering procedure. Journal of Electromyography and Kinesiology **9**(1), 47–57 (1999). https://doi.org/https://doi.org/10.1016/S1050-6411(98)00023-6, https://www.sciencedirect.com/science/article/pii/S1050641198000236
7. Coradine, L.C., Lopes, R.V.V., Maciel, A.F.: Mineração de dados: Uma introdução. Journal of the Brazilian Neural Networks Society **9**, 168–184 (2011). https://doi.org/10.21528/LNLM-vol9-no3-art3
8. De Ville, B.: Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics **5**(6), 448–455 (2013)

9. Grewal, M.S., Andrews, A.P.: Kalman filtering: Theory and Practice with MATLAB. John Wiley & Sons (2014)
10. Hall, J.E.: Tratado de Fisiologia Médica. Elsevier Health Sciences (2021)
11. Haykin, S.S.: Neural networks: A comprehensive foundation. Prentice Hall (1999)
12. Jarque-Bou, N.: Toward early and objective hand osteoarthritis detection by using emg during grasps. Sensors **23 5** (2023). https://doi.org/10.3390/s23052413
13. Kristoffersen, M.: User training for machine learning controlled upper limb prostheses: a serious game approach. Journal of NeuroEngineering and Rehabilitation **18** (2021). https://doi.org/10.1186/s12984-021-00831-5
14. Lennon, O., Tonellato, M., Del Felice, A., Di Marco, R., Fingleton, C., Korik, A., Guanziroli, E., Molteni, F., Guger, C., Otner, R., et al.: A systematic review establishing the current state-of-the-art, the limitations, and the desired checklist in studies of direct neural interfacing with robotic gait devices in stroke rehabilitation. Frontiers in Neuroscience **14**, 578 (2020)
15. Liu, C., Jiang, J., Jiang, J., Zhou, Z.: Enhanced grid-connected phase-locked loop based on a moving average filter. IEEE Access **8**, 5308–5315 (2020). https://doi.org/10.1109/ACCESS.2019.2963362
16. Liu, T.Y., Yang, Y., Wan, H., Zeng, H.J., Chen, Z., Ma, W.Y.: Support vector machines classification with a very large-scale taxonomy. Acm Sigkdd Explorations Newsletter **7**(1), 36–43 (2005)
17. Lorena, A.C., De Carvalho, A.C.: Uma introdução às support vector machines. Revista de Informática Teórica e Aplicada **14**(2), 43–67 (2007)
18. Merletti, R., Parker, P.J.: Electromyography: physiology, engineering, and non-invasive applications, vol. 11. John Wiley & Sons (2004)
19. Merletti, R., Parker, P.J.: Electromyography: physiology, engineering, and non-invasive applications, vol. 11. John Wiley & Sons (2004)
20. Mitra, S.K.: Digital signal processing: A Computer Based Approach. McGraw-Hill Companies (jan 2006)
21. Núñez-Montoya, B., Valarezo Añazco, M., Saravia-Avila, A., Loayza, F.R., Valarezo Añazco, E., Teran, E.: Supervised machine learning applied to non-invasive emg signal classification for an anthropomorphic robotic hand. In: 2022 IEEE ANDESCON. pp. 1–6 (2022). https://doi.org/10.1109/ANDESCON56260.2022.9989874
22. Oppenheim, A.V., Schafer, R.W., Yoder, M.A., Padgett, W.T.: Discrete-time signal processing. Pearson, Upper Saddle River, NJ, 3 edn. (aug 2009)
23. Proakis, J.G., Manolakis, D.G.: Digital Signal Processing: Principles, Algorithms, and Applications. Prentice Hall (1996)
24. Souto, M.C.P., Lorena, A.C., Delbem, A.C.B., Carvalho, A.C.P.d.L.F.: Técnicas de aprendizado de máquina para problemas de biologia molecular (2003)
25. Stashuk, D.: Emg signal decomposition: how can it be accomplished and used? Journal of Electromyography and Kinesiology **11**(3), 151–173 (2001)
26. Suvrit, S., Sebastian, N., Stephen J., W.: Optimization for Machine Learning. Neural Information Processing Series, The MIT Press (2012), https://mitpress.mit.edu/9780262537766/optimization-for-machine-learning
27. Xu, G., Liu, M., Jiang, Z., Söffker, D., Shen, W.: Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. Sensors **19**(5), 1088 (2019)
28. Xue, H., Ruan, M., Cheng, Y.: A fixed length adaptive moving average filter-based synchrophasor measurement algorithm for p class pmus. Energies **12**(21) (2019). https://doi.org/10.3390/en12214168

Appendix C

Tables

C.1 Tables of processing times for machine learning models

Table C.1: V1 - Filtering and total time for different methods in seconds

	Clustering		Decision tree		Random Forest		SVM	
	FT	TT	FT	TT	FT	TT	FT	TT
NF	0,0000	0,0011	0,0000	0,0011	0,0000	0,0141	0,0000	0,0130
BP	0,0012	0,0012	0,0012	0,0014	0,0011	0,0161	0,0012	0,0142
KM	0,0364	0,0368	0,0400	0,0405	0,0373	0,0520	0,0415	0,0555
MA3	0,0010	0,0011	0,0010	0,0013	0,0010	0,0143	0,0010	0,0151
MA5	0,0010	0,0011	0,0011	0,0012	0,0010	0,0163	0,0011	0,0142
MA10	0,0010	0,0011	0,0008	0,0012	0,0010	0,0152	0,0010	0,0144
MA50	0,0010	0,0012	0,0009	0,0011	0,0011	0,0156	0,0010	0,0169
MA100	0,0010	0,0011	0,0009	0,0010	0,0010	0,0145	0,0010	0,0184
NT	0,0017	0,0020	0,0018	0,0025	0,0015	0,0149	0,0021	0,0179

legend:
FT - Filtering Time (seconds)
TT - Total Time (seconds)

Table C.2: V2 - Filtering and total time for different methods in seconds

	Clustering		Decision tree		Random Forest		SVM	
	FT	TT	FT	TT	FT	TT	FT	TT
NF	0,0000	0,0011	0,0000	0,0011	0,0000	0,0147	0,0000	0,0239
BP	0,0010	0,0012	0,0016	0,0019	0,0011	0,0150	0,0013	0,0228
KM	0,0348	0,0353	0,0484	0,0490	0,0357	0,0502	0,0410	0,0610
MA3	0,0011	0,0012	0,0010	0,0011	0,0009	0,0149	0,0013	0,0213
MA5	0,0010	0,0011	0,0010	0,0012	0,0010	0,0157	0,0009	0,0199
MA10	0,0010	0,0010	0,0010	0,0011	0,0010	0,0179	0,0011	0,0238
MA50	0,0009	0,0011	0,0011	0,0012	0,0010	0,0141	0,0010	0,0212
MA100	0,0009	0,0012	0,0009	0,0012	0,0010	0,0175	0,0011	0,0188
NT	0,0017	0,0019	0,0019	0,0023	0,0015	0,0150	0,0020	0,0238

Legend:
FT - Filtering Time (seconds)
TT - Total Time (seconds)

Table C.3: V3 - Filtering and total time for different methods in seconds

	Clustering		Decision tree		Random Forest		SVM	
	FT	TT	FT	TT	FT	TT	FT	TT
NF	0,0000	0,0011	0,0000	0,0011	0,0000	0,0147	0,0000	0,0390
BP	0,0010	0,0012	0,0013	0,0018	0,0010	0,0153	0,0011	0,0402
KM	0,0362	0,0368	0,0414	0,0419	0,0379	0,0533	0,0324	0,0700
MA3	0,0010	0,0011	0,0010	0,0011	0,0010	0,0146	0,0010	0,0451
MA5	0,0010	0,0011	0,0010	0,0011	0,0013	0,0154	0,0011	0,0398
MA10	0,0010	0,0011	0,0009	0,0014	0,0011	0,0141	0,0011	0,0416
MA50	0,0011	0,0013	0,0009	0,0011	0,0010	0,0144	0,0009	0,0412
MA100	0,0009	0,0011	0,0008	0,0011	0,0011	0,0124	0,0011	0,0394
NT	0,0015	0,0019	0,0022	0,0025	0,0013	0,0153	0,0017	0,0405

Legend:
FT - Filtering Time (seconds)
TT - Total Time (seconds)

Table C.4: V4 - Filtering and total time for different methods in seconds

	Clustering		Decision tree		Random Forest		SVM	
	FT	TT	FT	TT	FT	TT	FT	TT
NF	0,0000	0,0010	0,0000	0,0011	0,0000	0,0139	0,0000	0,0594
BP	0,0017	0,0020	0,0015	0,0021	0,0011	0,0154	0,0010	0,0578
KM	0,0402	0,0404	0,0388	0,0396	0,0306	0,0433	0,0308	0,0823
MA3	0,0010	0,0011	0,0009	0,0012	0,0010	0,0156	0,0010	0,0565
MA5	0,0009	0,0012	0,0010	0,0011	0,0011	0,0148	0,0011	0,0610
MA10	0,0011	0,0013	0,0010	0,0012	0,0010	0,0140	0,0010	0,0649
MA50	0,0011	0,0011	0,0010	0,0011	0,0010	0,0148	0,0011	0,0580
MA100	0,0010	0,0010	0,0011	0,0014	0,0011	0,0144	0,0010	0,0548
NT	0,0017	0,0019	0,0018	0,0020	0,0014	0,0162	0,0018	0,0602

Legend:
FT - Filtering Time (seconds)
TT - Total Time (seconds)

C.2 Confusion matrixes

Table C.5: Confusion Matrix for V1

	Predicted M0	**Predicted M1**
Actual M0	240	40
Actual M1	4	314

Table C.6: Confusion Matrix for V2

	Predicted M0	**Predicted M1**
Actual M0	209	91
Actual M1	4	320

Table C.7: Confusion Matrix for V3

	Predicted M0	**Predicted M1**	**Predicted M2**
Actual M0	185	46	99
Actual M1	4	160	122
Actual M2	0	129	199

Table C.8: Confusion Matrix for V4

	Predicted M0	**Predicted M1**	**Predicted M2**	**Predicted M3**
Actual M0	125	77	63	73
Actual M1	9	181	62	72
Actual M2	16	60	235	3
Actual M3	0	12	1	313