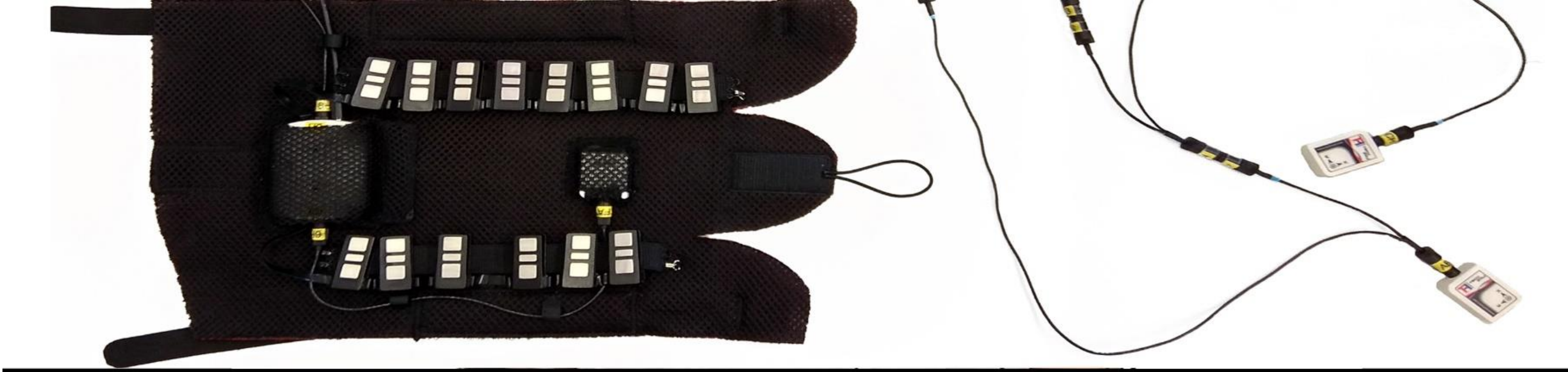
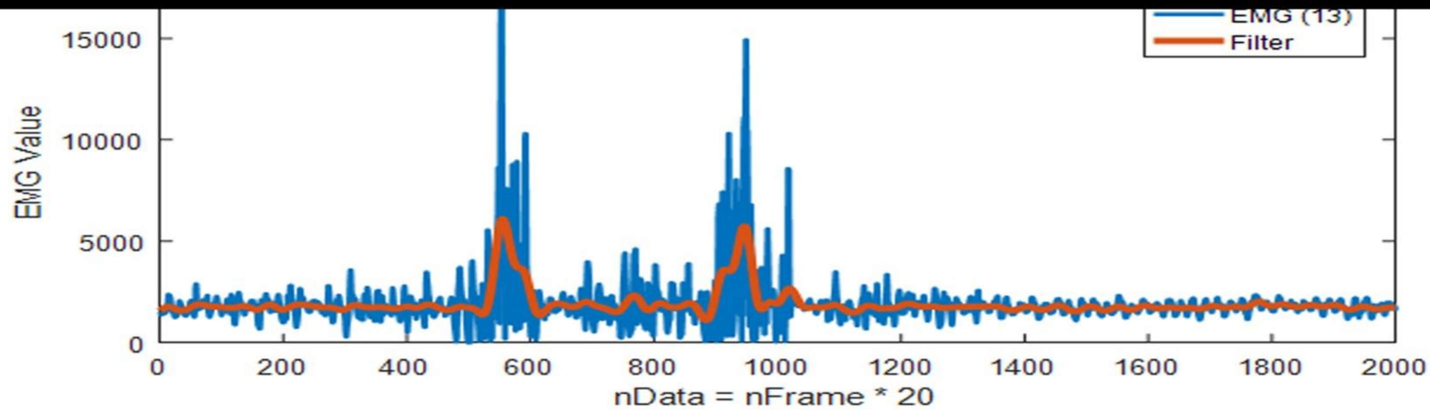




João Ricardo Ferreira dos Santos Silva

Classificação de gestos a partir de dados EMG

Dissertação de Mestrado em Engenharia Mecânica
na Especialidade de Produção e Projeto

Julho/2018

Universidade de Coimbra

FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE ENGENHARIA MECÂNICA

Classification of Gestures from EMG Data

Submitted in Partial Fulfilment of the Requirements for the Degree of Master in Mechanical Engineering in the speciality of Production and Project

Classificação de gestos a partir de dados EMG

Author

João Ricardo Ferreira dos Santos Silva

Advisor

Pedro Mariano Simões Neto

Jury

President	**Professor Doutor Cristovão Silva** **Professor da Universidade de Coimbra**
Vowels	**Professor Doutor Nuno Alberto Marques Mendes** **Professor Auxiliar da Universidade de Coimbra**
Advisor	**Professor Doutor Pedro Mariano Simões Neto** **Professor da Universidade de Coimbra**

Coimbra, July, 2018

Live slow, die whenever.

Unknown.

ACKNOWLEDGEMENTS

The work here presented is only possible with the help and support of my friends and family to whom I must start by paying my tribute.

First and foremost, my supervisor, professor Pedro Neto who gave me the opportunity to work alongside him and explore an area so interesting as robotics, making me feel comfortable with his support and friendship throughout this whole process.

To the whole robotics laboratory with people such as Nuno Mendes, André Brás, Miguel Simão, Pedro Moreira, Modi Safeea, Mihail Babcinshi, between others, that have been present from the beginning. They received me with open arms and with their friendship inspired me every day with their work ethics, showing me that we can have fun and work at the same time.

To all the friends I've made on this university journey, for making these last few years the best of my life and for all the good times we had together.

To all my hometown friends that have been present and that will never be forgotten.

To Portugal 2020 project DM4Manufacturing POCI-01-0145-FEDER-016418 by UE/FEDER through the program COMPETE2020.

And last but not least, to my parents for putting me on this earth and making me the man I am today, without them nothing would be truly possible.

Abstract

This work has as its main objective the segmentation and classification of gestures through the data collected from surface electromyography (sEMG) sensors. The performance of the sensors is evaluated through the collection of data to create an offline dataset. The implementation of different features and their threshold values is studied both for the segmentation and classification of gestures. The best offline results were obtained using a balanced dataset and for a library of 7 gesture patterns. For segmentation, the result showed an error of 3.7% using a Gentle boost ensemble classifier. For classification, the results showed an error of around 0.9% and were obtained for a K - Nearest Neighbors (K-NN) Spearman classifier with one nearest neighbor and seven implemented features. The online implementation is briefly presented with the error for segmentation of 15% and an error for classification of 35%.

Keywords EMG, Features, Segmentation, Classification.

Resumo

Este trabalho tem como objetivo principal a deteção, reconhecimento e classificação de gestos através de dados recolhidos a partir de sensores de eletromiografia superficiais (sEMG). A performance dos sensores é avaliada através a recolha de dados com intenção da criação de uma base de dados offline. A implementação de features e o valor do threshold das mesmas é estudado tanto para a segmentação como para a classificação de gestos. Os melhores resultados obtidos na avaliação offline foram conseguidos com uma base de dados equilibrado entre gestos e não gestos. Para a segmentação os resultados apresentaram um erro de (3.7%) com recurso ao classificador *ensemble Gentleboost*. Para a classificação os resultados mostram um erro na ordem de (0.9%) e foi obtido com recurso a um classificador *K-nn Spearman* com 1 vizinho mais próximo e 7 features implementadas. A implementação online começa a ser explorada e é rapidamente apresentada, com erro de 15% na segmentação e 35% na classificação.

Palavras-chave: EMG, Features, Segmentação, Classificação.

Contents

LIST OF FIGURES

LIST OF TABLES

SIMBOLOGY AND ACRONYMS

Symbology

G1-G9 – Gestures 1 to 9

RMS – Root Mean Square

SSC – Slope Sign Change

MAV – Mean Absolute Value

WL – Waveform Length

ZC – Zero Crossing

Var – Variance

Wamp – Willison amplitude

SD – Standard Deviation

AR – AutoRegressive model

WT - Wavelet Transform

Fc – Frequency Cut-off value

M.N.S. - Maximum Number of Splits

N.L. - Number of Learners

L.R. – Learning Rate

K-nn – K nearest neighbours

Thr – Threshold Value

Acronyms

EMG – Electromyography

IMU – Inertial Motor Unit

sEMG – surface EMG

SVM – Support Vector Machine

HOS - High Order Statistics

STFT - Sort-Time Fourier Transform

1. INTRODUCTION

Interactive work between robots and humans in a smart and flexible manufacturing station is a subject that has been studied continuously in recent years.

The recognition and classification of gestures with data from electromyography (EMG) signals, is one of the most intuitive methods that allow interaction between human and machine. It allows us to have semiautonomous robots that realize defined tasks based on the gestures defined by users. Usually, robots follow a stipulated sequence to accomplish some complex task. However, it will be the responsibility of the programmer to define the tasks associated with each movement.

The segmentation of gesture data has been studied in the last few years, for example (Simão et al. 2017) addressed this problem with the use of a Cyber Data Glove to detect the hand and arm gestures of the user. However, this method has the drawbacks of being expensive and not practical. Also, in (Lopes, João 2016) this problem is addressed using a MYO Armband that uses EMG sensors combined with inertial movement unit (IMU). This bracelet had the advantage of being affordable, but it was not always the most reliable in terms of data, presenting some error using both the sensors together or separately.

In order to solve the problems mentioned above, a research was conducted that culminated in a project called ColRobot. This is a project that counts with the collaboration of several research centers and companies that perform different tasks, but all focused on the main objective of creating a viable collaborative robot to perform and carry out various tasks with humans at an industrial level.

The success of the industrial implementation of this system is vital if we want to have a robot that acts as a "third hand" for an operator who can interact through gestures while ensuring his safety.

A prototype from TECHNAID consists of a system similar to MYO, but with 14 sEMG sensors instead of the 8 possessed by MYO. This prototype also has IMU sensors, but this thesis will only focus on the recognition, classification, and creation of a dataset of EMG sensors, leaving out the IMU system.

This work will start with a state of art review, followed by a presentation of the studied device and the methods used to obtain results. In chapter 5 results are shown and discussed. Online implementation it’s also mentioned in chapter 5.

2. STATE OF ART

2.1. EMG

According to (Kim, Mastnik, and André 2008) electromyography (EMG) is a method that measures electrical currents that are generated in a muscle during its contraction and represent neuromuscular activities. In (Dario Farina et al. 2014) the authors claim that when the neuromuscular system is probed for information extraction, the interfacing can be realized at the level of the brain and other areas such as, above the spine, using peripheral nerves, or muscle and that among the option presented the muscle interfacing is currently the only viable way to control external in commercial and clinical systems.

In (Norali and Mat Som 2009) it is stated that the electrochemical transmission between nerves that has its origin on the brain produces action potential which propagates through nerve fibers until finally stimulate the skeletal muscle which will contract the muscle resulting on the movement of the human limb.

Every movement corresponds to a different activation of a set of nerves and muscles, identifying this information is beneficial for the control of various things as stated in (Rojas-Martínez et al. 2013). According to (Clancy, Morin, and Merletti 2002) the EMG, can be measured from the muscle or from the skin surface overlying the muscle. Surface EMG (sEMG) is most commonly used because of its non-invasiveness. This thesis and the prototype presented is made up of surface EMG. It is the only type of EMG here presented, leaving the invasive type of EMG out of this presentation.

2.1.1. Applications of EMG

EMG has been used for quite some time in medical applications, such as gaining an insight into muscle activity, identifying muscle fatigue and causes of pain, as demonstrated in (Kleine et al. 2000). Recent advances in technology allowed for a deeper analysis of the muscle signals, such as the natural biological signal to infer motion intention of the user regarding the direction of movement, in addition to its strength. It evolved to identify patterns related to muscle activity as stated in (Rojas-Martínez et al. 2013). This development allowed to apply EMGs in such areas as the control of human-machine interactions, interfaces like prostheses or orthoses, as well as other rehabilitation devices, games and computer-based training programs (Rojas-Martínez et al. 2013).

2.1.2. Problems influencing the surface EMG

According to (Dario Farina et al. 2014) muscle recordings are peripheral measurements and are not obtained directly from neural cells, as they contain neural information on the executed motor tasks. It is also mentioned that most myoelectric control schemes use macro-features of the sEMG, that depends on the neural and peripheral information contained in the signal, and that their performance is influenced by factors that can affect the motor unit action potential shapes or the neural drive to the muscle.

In (Simão et al. 2017) is stated that in sEMG, signals have inherently low signal to noise ratio, which means that they are very susceptible to environmental noise. It can be a result of inherent noise in electronics equipment, ambient noise, motion artefact and inherent instability of the signal, described in (Lopes, João 2016). Noise are electrical signals that are not part of the desired EMG signal. Therefore, signal processing methods are required for noise reduction through signal filtering.

According to (Dario Farina et al. 2014) there are multiple factors, both physiological and non-physiological that influence the surface EMG, but among all, the only one that deserves special mention is, crosstalk among muscles.

Crosstalk occurs when electrodes are placed on the skin overlying a target muscle but there is also signal generated by other muscles in its vicinity being recorded. It can be seen as an addition of extra spike to the surface EMG generated by muscles, other than the target muscle, which will impact the measure of EMG signal and its features, such as power. In some cases, the detected EMG signal is naturally the mixture of the activities of several muscles that are anatomically and functionally close, such as the forearm muscles or wrist tendons, in which crosstalk is at least as large as the target muscle activity. This is of interest to this project since the studied prototype is placed on the forearm and wrist of the user and therefore is subject to crosstalk muscle activity. However, this is not detrimental to the study, since for a specific gesture to be classified may be characterized by a set of EMG recordings from different muscles, all influenced by a certain amount of crosstalk. If the activity containing crosstalk is consistent across trials of the same task, task classification can be performed correctly, and crosstalk may even become a source of information to better differentiate the tasks.

2.1.3. sEMG

The use of sEMG presents some issues, as stated in (Simão et al. 2017) "a targeted electrode is defined as a surface electrode that is carefully positioned above the target muscle, an untargeted electrode is a surface electrode that is placed above the muscle of interest but positioned without regard for its accuracy. An example of an untargeted electrode is an element of a linearly spaced array of electrodes is likely to capture signals from several muscles at the same time".

In (Scheme and Englehart 2011) it is also mentioned issues such as electrode shift, variation in force and position of limb and transient changes in EMG, alongside muscle fatigue, electrode lift-off, and electrode impedance changes.

Electrode shift: Occurs when electrodes are used and then settle in a slightly different position relative to the underlying musculature. This may also occur because of loading and positioning of the limb. This change in position can result in an increment of classification errors.

Variation in force: When a gesture is performed, the force applied from the user it influences the reading of the sensors data, so it's important to calibrate the device for a medium strength from the overall testing subjects to perform a better reading result.

Variation in position of limb: Mentioned in (Lopes, João 2016), it requires that the tests be realized with the device in different positions because when two different users use the device is likely that the position of the contact between the sensors and the muscles won't be the same, which will cause a degradation of the pattern recognition, in both static and dynamic movements, but with a more severe impact on pattern recognition on dynamic changes of arm position.

Electrode lift-off: It is the loss of contact between electrode and skin and is also mentioned as an issue to be considered and in this case is resolved by strapping the EMG to the forearm of the user using Elastic Bands and a textile Sleeve.

3. DEVICE PRESENTATION -TECHNAID PROTOTYPE

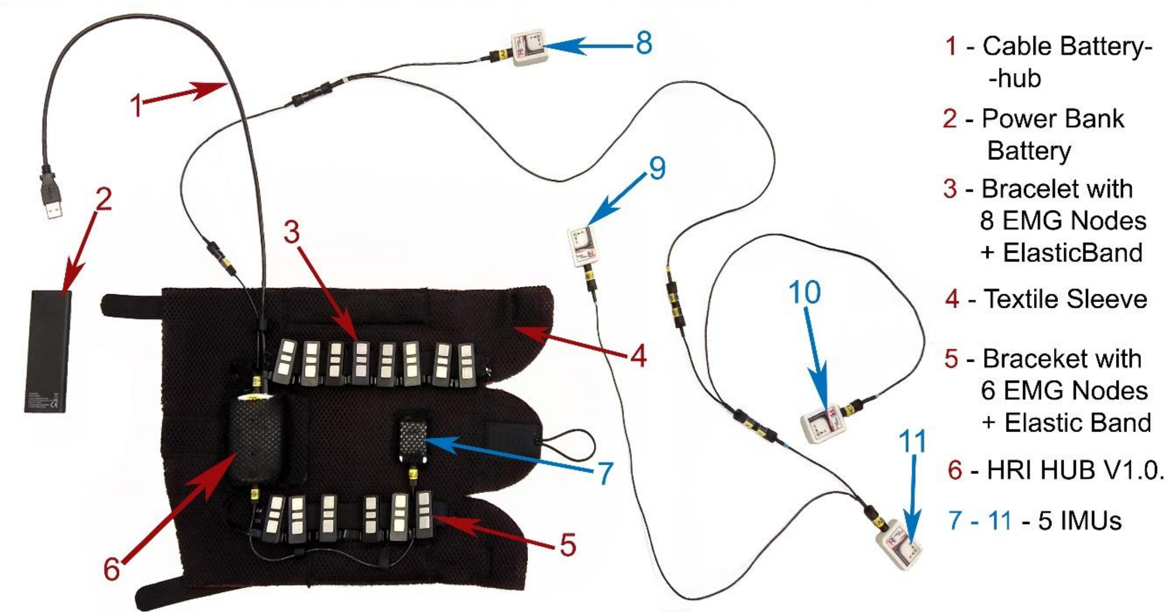


Figure 3.1 – Technaid Tech-HRI prototype

The Technaid Tech-HRI prototype is a device developed by Technaid, meant to be worn on the forearm. Has the purpose of detecting forearm and wrist muscles activity, by using 14 stainless steel sEMG muscle sensors, combined with 5 nine-axis IMU containing a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. This armband contains a HUB that processes the information received from all the sensors and communicates the data to the computer through wireless communication.

Although this device has EMG and IMU components, the thesis here presented is aimed to gesture spotting with sEMG. Thus, IMU is not extensively presented.

IMU, as said by (lopes, João 2016) are "devices used to measure acceleration and angular rate through the use of different types of inertial sensors, accelerometers, gyroscopes and magnetometers.". This will allow determining the location, orientation, and velocity of a moving object in a three-dimensional space in relation to the chosen axis.

The sleeve with sEMG sensors is to be applied on the right forearm of the user, it contains the 14 sEMG sensors, a hub that receives and communicates data and an IMU to measure the orientation and position of the forearm in a three-dimensional space.

The other IMUs are applied in other parts of the body. In accordance with the figure 3.1, IMU 8 is put on an arm strap on the right arm alongside the battery that is connected by a USB cable to the HUB. IMU 9 is applied on the chest of the user, in a chest harness. IMU 10 is applied on the left arm of the user, once again in an arm strap. Finally, IMU 11 is to be applied on the left forearm of the user inside a forearm strap for IMU. When applied, the system should look like the figure 3.2.

Figure 3.2 - Mounted System

3.1. First contact and initial problems

Being a prototype, there was not much information available to work within the presented device. So, the contact made was a result of a process of trial an error. The cable connections and the device, in general, are fragile, that ensued some initial problems with cable connections to the HUB at the start. Sometimes some of the EMG Data was not available and we figured the problem of the cables connecting with the hub.

It was provided a software for data acquisition that allowed to acquire data online but did not allow to save the collected data. It was also provided a program using the software Visual Studio that also presented some problems in the connection with the device and in the collection of data, reason why it was necessary to repair the Visual Studio program and create a new MATLAB program that allows data reception using sockets.

Later, in the testing phase, after running the program and collecting data, if there was a stop in the acquisition, re-acquire data without restarting the device was found not to be feasible. Instead of collecting new data, the device reproduced the previously acquired data and once it was finished, it entered an error state giving the maximum values of all EMG continuously. While collecting data for the dataset, not rebooting the device would also result in an error in which the device only sent half the supposed data in every frame.

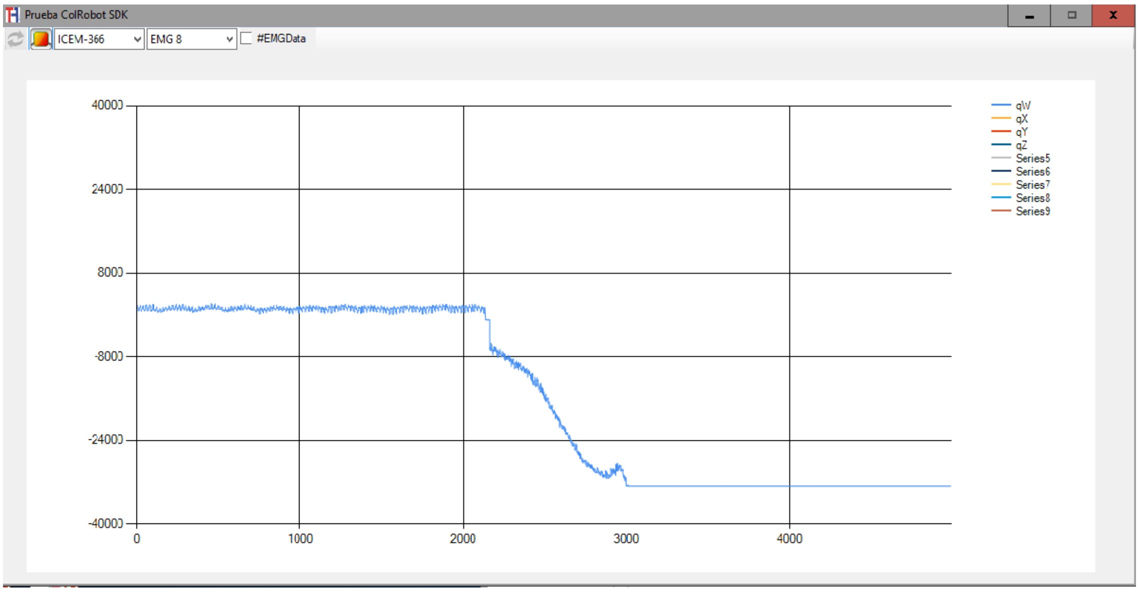


Figure 3.3 - Example of acquisition error time/EMG value

While testing the device with the intention of creating the dataset, it was noted that once the battery of the power bank provided with the device ended, connecting it to another power source wasn't really an option. This happens because different power sources would result in a fatal resonance error making the data collected unusable since it did not correspond to the pretended signal that is verified while making gestures.

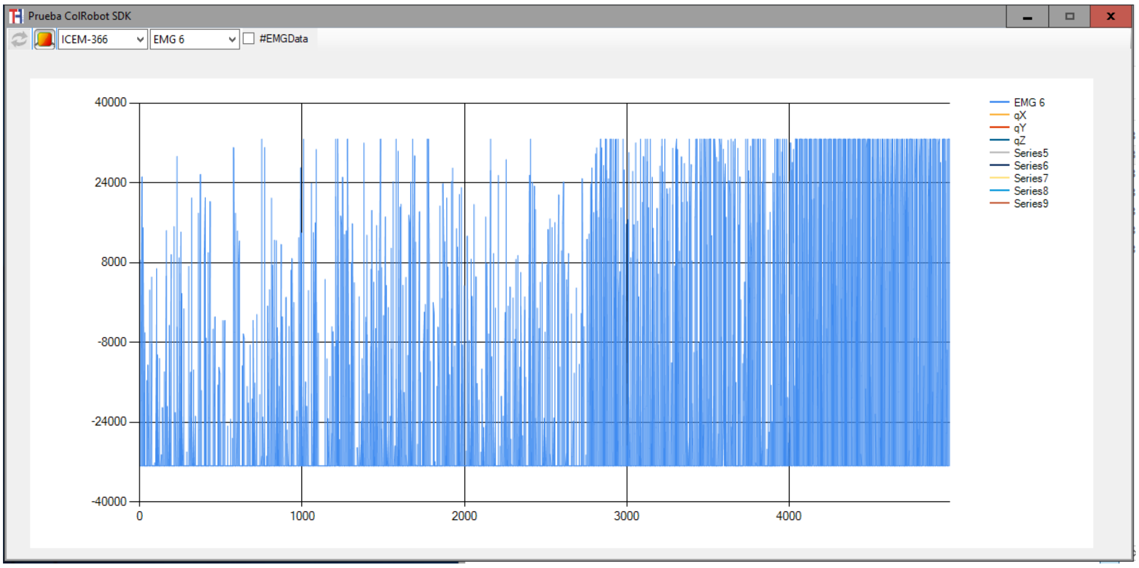


Figure 3.4 - Example of acquisition error in time/EMG value

In addition, while collecting the data for the dataset it was noted that the system consumed a lot of resources of the computer in which it was connected and once connected it needs some time to stabilize the signal, or else the data would present spikes that were not normal, reaching the maximum value for the data, as it can be seen in figure 3.5.

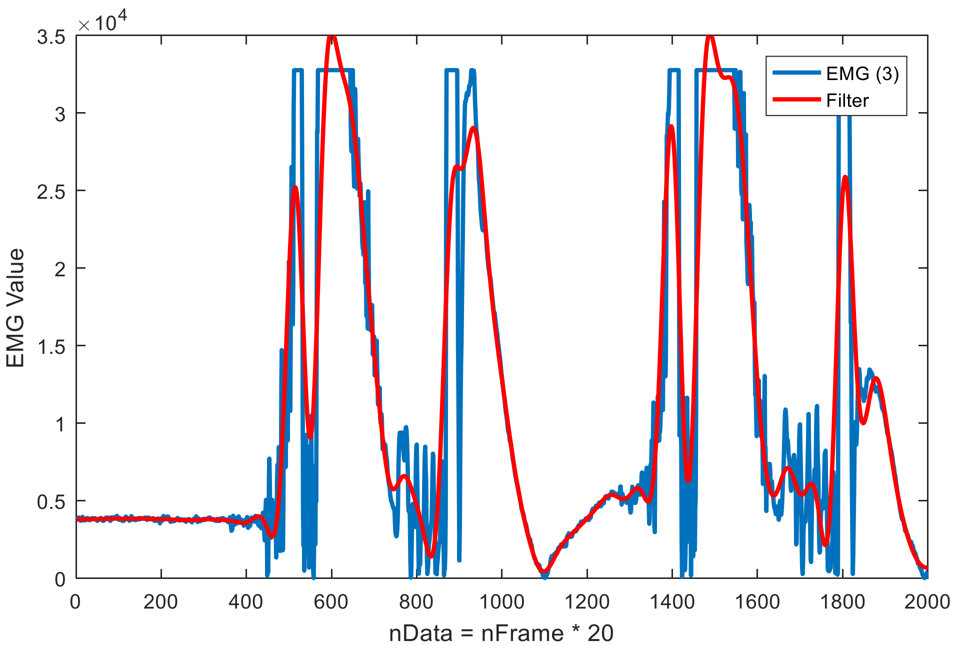


Figure 3.5 - Example of acquisition with unstable device

To end this subchapter, it is worth mentioning that there was the need to change the positioning of the IMU placed in the sleeve. This turned out to be a problem since the sensor would be in different positions in different users with different forearm diameter. To solve this, we used a forearm strap placed in front of the sleeve, over the wrist, so the IMU position would be the same in all the users, with the objective of collecting IMU data that can be stored and potentially used in the future.

3.2. EMG Positioning

The studied device is composed of two strips of EMG's. The first one with 8 sEMG positioned closer to the elbow, and a second with 6 sEMG positioned closer to the wrist. We can see the illustration of the positioning of the device in the following figures 3.6 and 3.7.

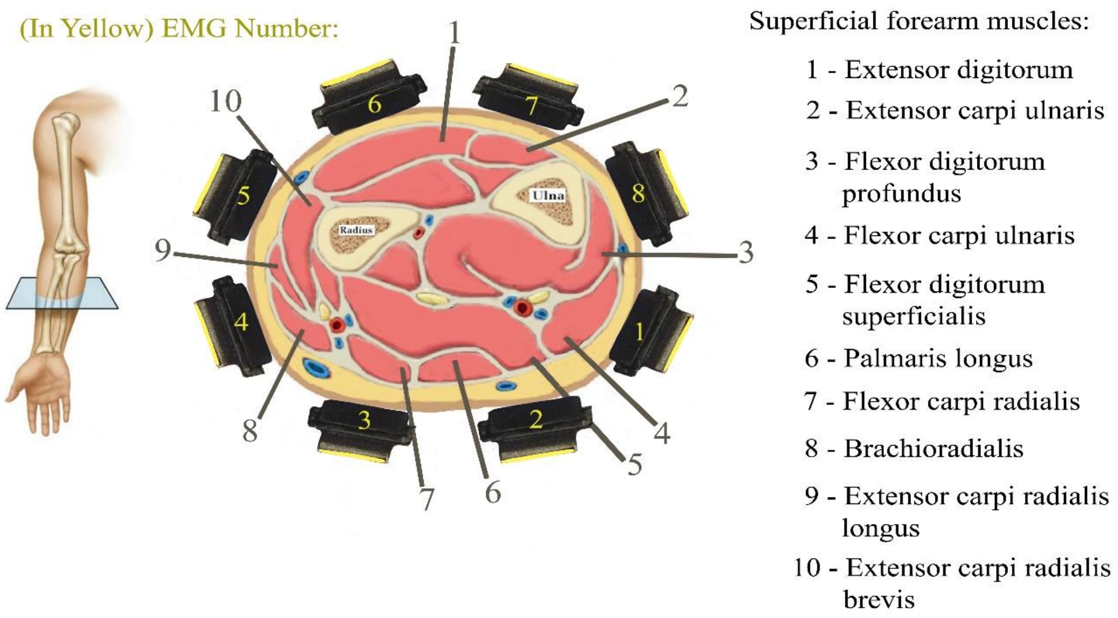


Figure 3.6 – Cross Section of the right forearm musculature indicating the approximate locations of the superficial EMG

The objective is to cover most of the forearm musculature as possible. But, as mentioned in the chapter 2.3.1 sEMG, since we are using superficial sensors, individual muscles can’t be specified to be analysed. A way of contradicting this fact is trying to put the sensors in a similar position in everyone who wears it. The major catch of this type of device is that there is no fixed position of the sensors in comparison to the user, and different users have different sizes of forearm, so the muscles that are read for one user won’t be necessarily the same for the next one even if the device is put in the same way. For this reason, training of the device requires the use of different size users. Retracting as many samples as possible, from a great variety of users is an important factor in the segmentation and classification of the process as it will be seen in a later chapter of this Thesis.

There are changes to the forearm anatomy throughout its length. The major differences are that the bones (ulna and radius) cover a bigger area of the cross-section and the muscles assume a different function, acting as connecting tendons between the muscle and the hand. Although tendons and muscles are structurally different, as mention in (D. Farina and Merletti 2001), the action potential generated at the neuromuscular junction propagate toward the tendons, and so, the recognition of signal works the same way for both muscle and tendon.

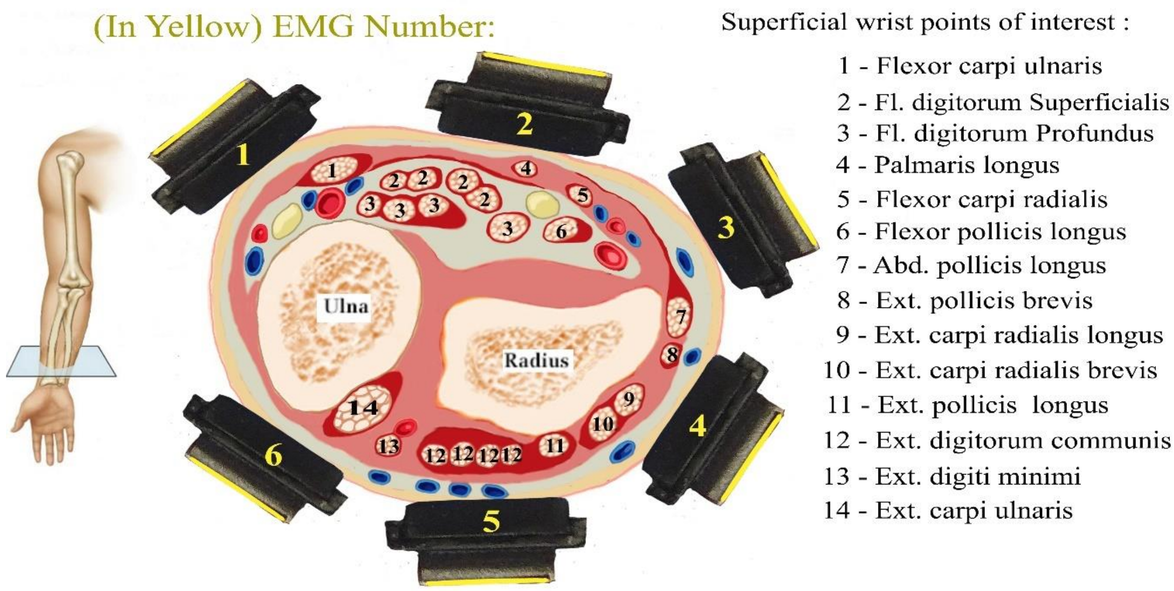


Figure 3.7 – Cross-section of the right wrist musculature indicating the approximate locations of the superficial EMG

In figures 3.6 and 3.7, can be observed major points of interest for the cross-section of different parts of the forearm. In figure 3.6, it is clear all the musculature of the upper region of the forearm. The position of the sensors is intended to be distributed equally along the region to be able to occupy the largest number of possible muscles.

It is also perceptible from the cross-section of the wrist that we have a relatively large portion without any point of interest surrounding the ulna, from the tendon 14 - Exterior Carpi Ulnaris, to tendon 1 - Flexor Carpi Ulnaris.

The fact that some muscles (e.g. Flexor digitorum superficialis) branch out in several different tendons in addition to tendons occupying a small area, are factors that do not help to the cross-talking between different tendons nor to specify a portion to analysis.

In the next figure, 3.8 it is clear the point of separation between the two human structures along the forearm, muscles, and tendons.

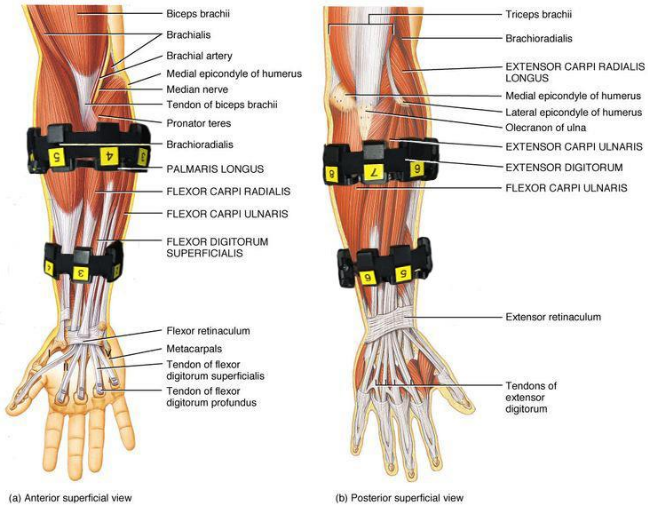


Figure 3.8 – Forearm muscular anatomy and positioning of EMG sensors in the forearm from an anterior and posterior superficial view

3.3. Gestures Used

It was established the use of a closed group of gestures within the scope of the project. Seven gestures were required to do the tasks in an industrial environment and 2 non-gestures were also implemented to facilitate the training of the machine. The seven defined Gestures are dynamic, which means that there is movement associated with the gesture. The objective is to have a group of easy to do/easy to remember gestures that are not natural to the human movement to avoid accidental activation of a certain task. As previously said, different gestures activate a different set of muscles and thus a different set of EMG, so it's also convenient that the gestures are varied to make for a different and clear set of activated EMG. In the end, the following group of gestures was defined:

Figure 3.9 – G1 – 2x Close hand; G2 – 2x open hand; G3 – 2x Wave in; G4 – 2x Wave out; G5 – 2x double tap in neutral position; G6 - 2x double tap in down position; G7 – 2x double tap in upward position

The gestures chosen for this project, consist of the repetition of one simple gesture. These gestures were chosen with help from feedback received in the preparation of this project. Some gestures, both static and dynamic, were presented but in the end, the dynamics were found to be more attractive for most people involved, and so, the result is what is shown.

The gestures were all performed in the same way, starting in a rest position, followed by the performance of the designated gesture and finishing back at the initial position.

Having a repetition of a simple gesture is also helpful when visualizing the graphics formed with EMG data since once activated they will also produce a double spike related to the movement.

The Gesture 8 (G8) and Gesture 9 (G9) that were also performed but are not represented in the images are non-gestures performed to help in the classification. G8 consists in having the user perform the rest position in several different positions throughout the duration of the session, as seen in figure 3.2. G9 consists in performing any gesture that is not the ones listed above and trying to avoid the rest positions. For instance, having a static close hand gesture, or making numbers with the hand, are some performed non-gestures.

4. METHODS

4.1. EMG Gesture Spotting procedure

At the beginning of the project, the following procedure was defined and followed throughout the duration of the study. It was mostly used the MATLAB (MathWorks Inc., Natick, MA) software and the help of literature whenever it was necessary, as well as the device available for retrieving data. The present subchapter is aimed to present the procedure followed, as seen in figure 4.1.

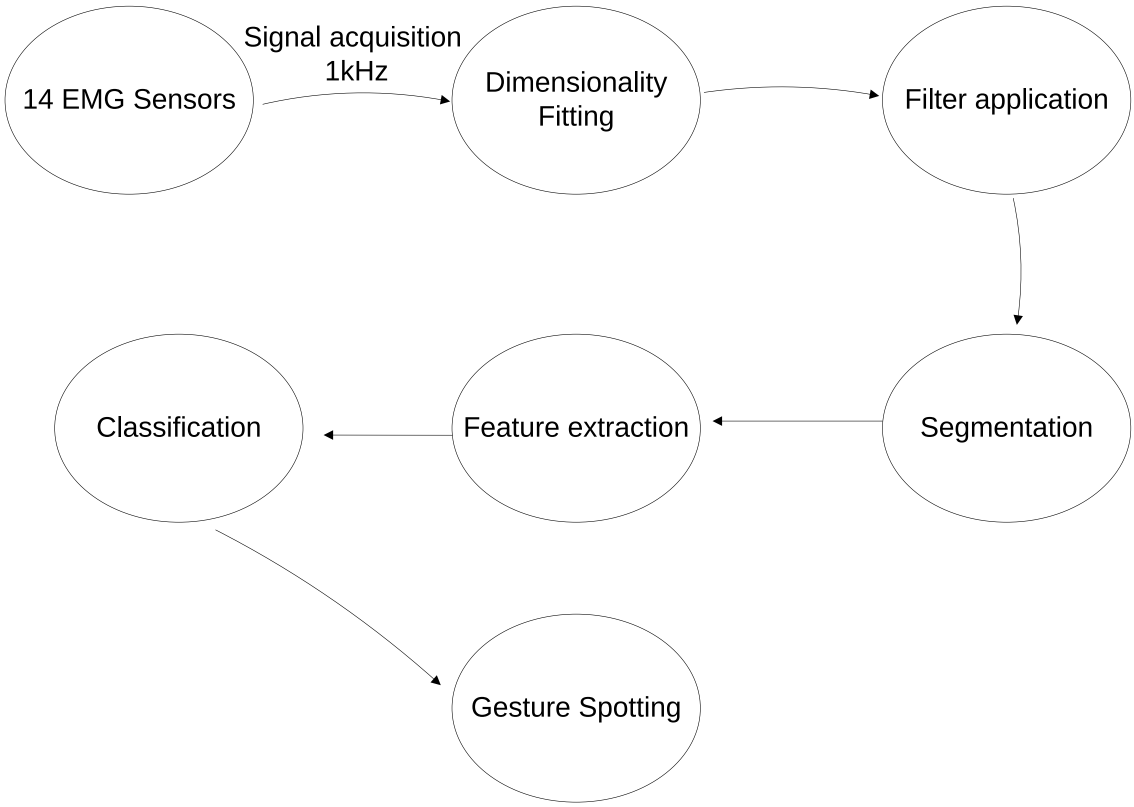


Figure 4.1 - EMG Gesture Spotting procedure

4.1.1. Data Acquisition

To the data acquisition, 6 subjects with ages from 18-26 years volunteer for this study. A set of 9 gestures were performed 50 times each, with a window of 2 seconds for each gesture, which corresponded to the longest activity in our experiment, verified for

G1 (2x close hand). The power bank provided by TECHNAID had to be used in every session and between gestures there was a need to reboot the system in order to avoid the errors presented in chapter 3.1.

The definition of the dataset for this project turned out to not be an easy task taking into consideration all the problems faced by the device already presented in previous chapters. For each session, taking the window of 2 seconds per sample and collecting 50 samples per gesture, the effective time performing gestures per session is of 9*50*2=900 sec or 15 minutes. It was taken at most 4 seconds in between collecting samples, which means, adding all up, that at most each session should take about 45 minutes to complete.

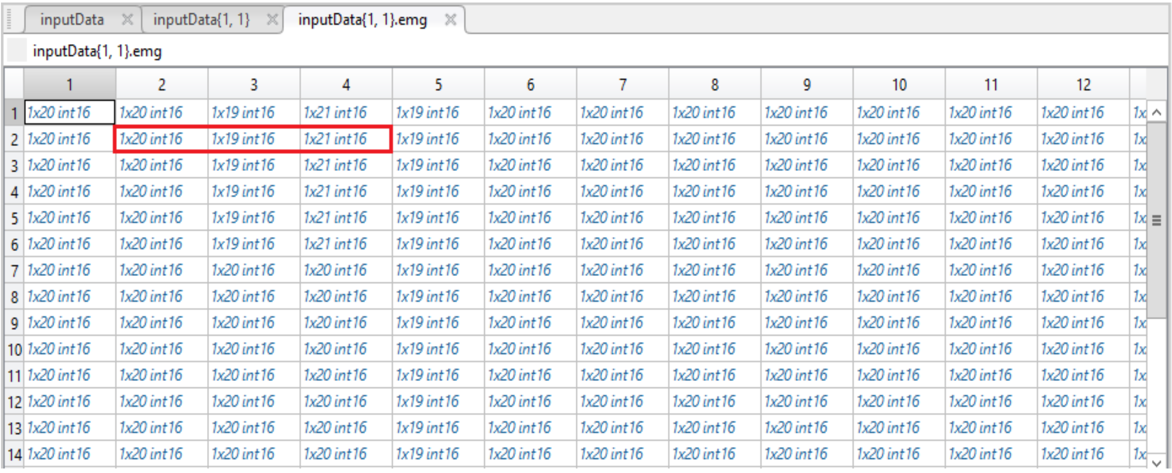
Each session took over 1 hour to complete, meaning that the remaining time was spent trying to solve the connection issues faced every time the device was connected to the computer. Having to use a battery that lasts about 3 hours of continuous use between charges, not allowing to connect with other power sources, meant that at most 2 complete sessions of collecting data could be realized in between charges.

At the end of each session the data collected had to be analysed to make sure the signal was received without errors making it a good fit for the dataset. All the time taken with these tasks made it difficult to perform more than 2 sessions of data acquisition per day

The device was used with a frequency of 1000 Hz to the acquisition. In the placement of the device, care was taken to put it in the same way for all the users, thus trying to cover the same set of muscles for all. The gestures were performed in a standing position in a way that the subjects felt more comfortable through this process, it was also asked to do a natural force to the movements, not imposing an abnormal amount of force to the sEMG.

4.1.2. Dimensionality Fitting

When the data is received from the device, not all the frames come with the same dimensions, as seen in figure 4.2.



	1	2	3	4	5	6	7	8	9	10	11	12	
1	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
2	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
3	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
4	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
5	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
6	1x20 int16	1x20 int16	1x19 int16	1x21 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
7	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
8	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
9	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
10	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
11	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
12	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
13	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x
14	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x19 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x

Figure 4.2 - Data before dimensionality fitting

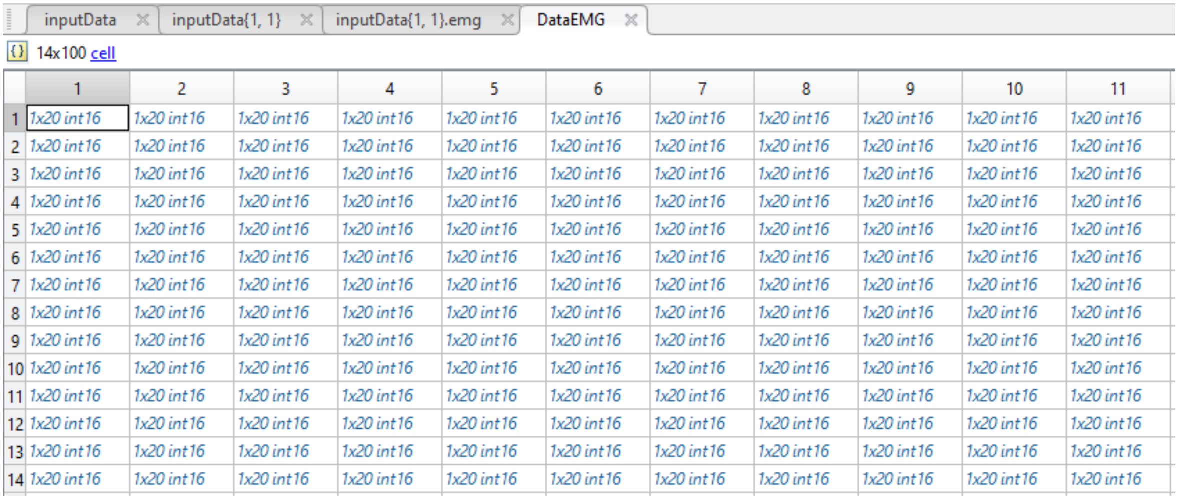
The different dimensions are an issue that must be addressed before processing the data since data with different dimensions will result in errors during the analysis besides also making the graphic representation harder.

At 1kHz of frequency acquisition, it was supposed to be received 20 data per frame per EMG, so the input data should be the number of frames with 1x20 data values since they were disposed of in a row. As observed in figure 4.2 this was not the case and the need arose for dimensionality fitting.

When adjusting the sizes of the cells, three situations could be verified:

1. If the cell came with the correct size (1x20) there would be no need to change nothing, therefore, no adjustments would be made.
2. If the cell presents more values than expected (1x>20), we only consider the first 20, discarding the remaining ones.
3. If the cell has a shortage of values (1x<20), the cell will be filled with the last recorded value until there were 20 values of data. The reason we use the last value instead of filling it with zeros is so to maintain a sequence of logic values, instead of changing it to 0 that would not make much sense in the sequence.

The result will be a normalized set of data that will not have cells with different data sizes, as seen in figure 4.3


inputData | inputData{1, 1} | inputData{1, 1}.emg | DataEMG

14x100 cell

	1	2	3	4	5	6	7	8	9	10	11
1	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
2	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
3	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
4	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
5	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
6	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
7	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
8	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
9	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
10	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
11	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
12	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
13	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16
14	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16	1x20 int16

Figure 4.3 - Data after dimensionality fitting

To end the dimensionality fitting, a matrix of values is created that has all the values from the frames displayed in order, making the graphic representation possible. Since there are 20 values per frame and 100 frames per reading, we end up with a 14x2000 matrix corresponding to 2000 data for 14 EMG.

4.1.2.1. Signal Filtering

A filter was applied to the raw data that is very similar to the one used in (Lopes, João 2016) work using MYO Armband. In essence, the filter is used to treat the raw data, reducing the inherent noise of the system with the objective of creating a pattern that allows us to identify the muscle activity for each gesture. The data treatment started with the appliance of the *abs* function to the signal of the 14 EMG. Then it was applied a low-pass Butterworth function, to remove some noise and unwanted signals to obtain the filtering coefficients, which are then applied to a zero-phase digital filter using the function *filtfilt*. In the *butter* function, the following parameters were applied:

- Sampling frequency of 1000 Hz that goes in accordance with the EMG data frequency obtained by the device.

- A cut-off frequency of 50 Hz and a filter order of 4. These values were obtained by trial and error, comparing with other cut-off values and selecting the one that gave us the best filter line, avoiding significant signal peaks due to noise while also avoiding excessive smoothing of the data.

The comparison between raw and filtered data can be seen if we plot them together. We can do this using the MATLAB software. This is also used to see if there is any lag between the 2 types of data. Figure 4.4 represents the problem of using a lower sampling frequency, Figure 4.5 shows the problem of not adjusting the cut-off frequency and subsequent excessive smoothing of the data and the final figure 4.6 shows us the result of the parameters described above.

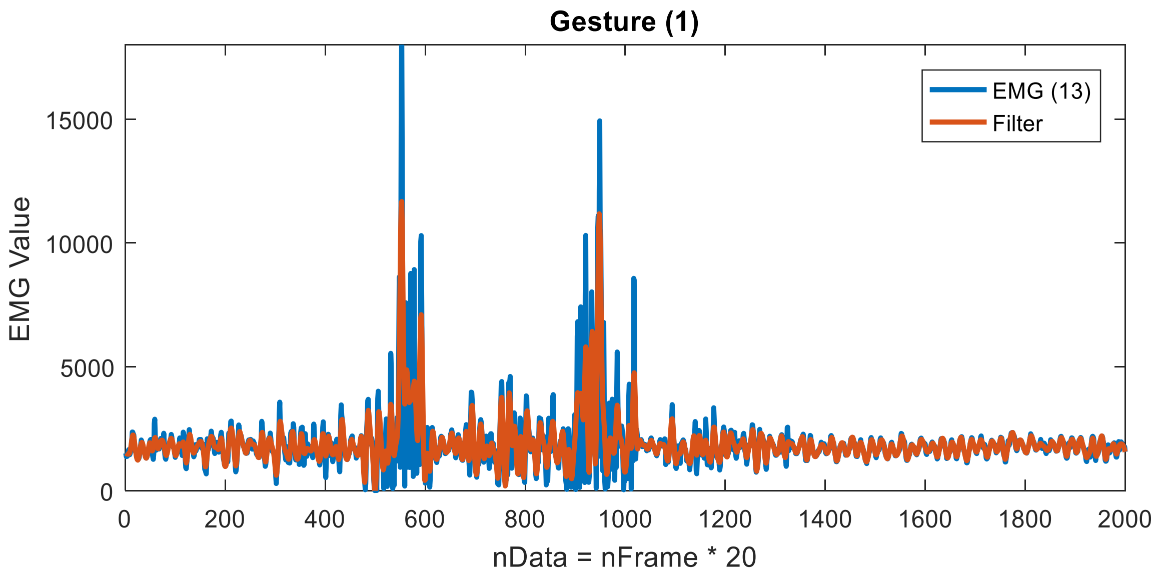


Figure 4.4 - A high frequency cut-off results in too many peaks following the raw data

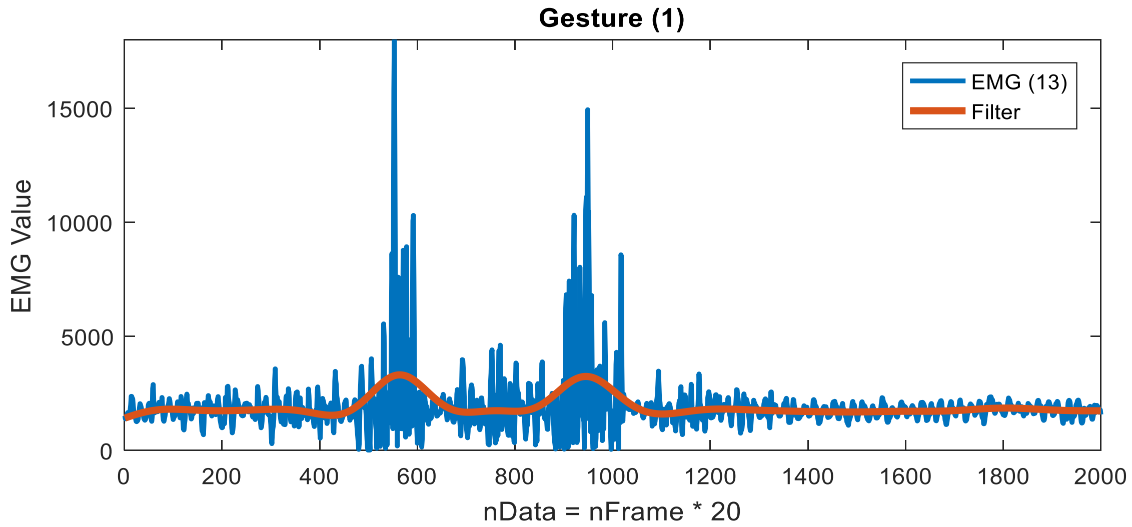


Figure 4.5 - Low cut-off frequency results in excessive smooth filter line

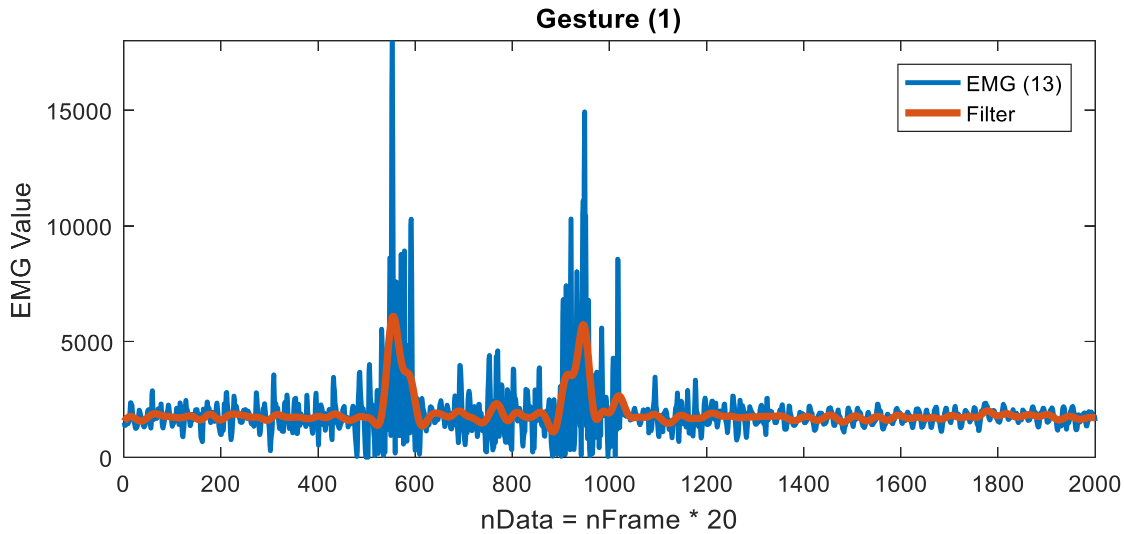


Figure 4.6 - Filter line following the described parameters

4.1.3. Segmentation

As stated in (Lopes, João 2016) and (Attal et al. 2015) segmentation allows to obtain the part of the Data processing that is used to extract features. To do this the data signal is divided into smaller time segments and then classification algorithms are applied to each window.

The segmentation is used to recognise if there is a gesture being made in a given sample. This is done by calculating the variance and analysing its values. If the variance of data values is higher than a certain threshold than it is classified as being a gesture. If the threshold is not exceeded than its classified as a non-gesture. The segmentation is also used to prepare data for classification since only if a sample is classified as being a gesture there will be the need to advance for the classification.

A window of frames of the analysed data is used in order to extract features. In our case, the window is 100 frames (2000 EMG data values), which means that one of each feature is extracted from each sample.

4.1.4. Feature extraction

As described by (Kumar and Bhatia 2014), "Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure." This means, using features is a way of trying to find what defines a certain pattern, in our case, the principal characteristics that define each EMG sensor, whether it be the signal amplitude, slopes, number of turns, or others, that allows us to easily identify the performed gesture in the classification.

4.1.4.1. Selected features

With the help of the literature and testing, a certain group of features was chosen to be used in the project. According to (Veer and Sharma 2016) there are three types of features that can be used in the EMG signal, they are time domain, frequency domain, time-frequency domain features. Both (Chen and Wang 2013) and (Phinyomark et al. 2014) present the most popular time-domain features that we will use in this study.

The first feature is used in the segmentation and it is the Variance (Var) of the EMG, that consist of using the power of the sEMG signal as a feature and is generally characterised by being the "mean value of the square of the deviation of that variable", it is represented by the equation (4.1).

$$Var = \frac{1}{N-1}\sum_{t=1}^{N} {x_t}^2 \tag{4.1}$$

.

N denotates the length of the signal and x(t) represents the sEMG signal amplitude in a segment.

Throughout the literature the most used feature is the Mean Absolute Value (MAV) that can be calculated using the moving average of the full wave rectified EMG (Phinyomark et al. 2014) and it is calculated by taking the average of the absolute value of the sEMG signal as shown in the equation (4.2).

$$MAV = \frac{1}{N}\sum_{t=1}^{N}|x_t| \tag{4.2}$$

Root Mean Square Value (RMS) is another feature that is used along with the mean absolute value to estimate EMG amplitude (Phinyomark et al. 2014). (Phinyomark, Limsakul, and Phukpattaranont 2009) "Is modelled as amplitude modulated Gaussian random process whose RMS is related to the constant force and non-fatiguing contraction". It is expressed by the equation (4.3).

$$RMS = \sqrt{\frac{1}{N}\sum_{t=1}^{N}{x_t}^2} \tag{4.3}$$

The Waveform Length (WL) is "the cumulative length of the waveform over the time segment, and it is related to the waveform amplitude, frequency and time" (Phinyomark, Limsakul, and Phukpattaranont 2009). It is given by the equation (4.4).

$$WL = \sum_{t=1}^{N-1}|x_{t+1} - x_t| \tag{4.4}$$

Zero Crossing (ZC) is defined by the number of times that the amplitude value of signal passes through zero (Xi et al. 2017) It can be formulated by the equation (4.5).

$$ZC = \sum_{t=1}^{N-1} u(-x_t \times x_{t+1}) \qquad threshold = \begin{matrix} 1, if\ x > threshold \\ 0, otherwise \end{matrix} \tag{4.5}$$

Slope Sign change (SSC) similarly to zero crossing represents the frequency information of sEMG signals represents the number of turns between positive and negative slope among consecutive segments (Phinyomark, Limsakul, and Phukpattaranont 2009). In other words, it is the number of signal peaks (Xi et al. 2017) and is represented by the equation (4.5).

$$SSC = \sum_{t=1}^{N-2} u[(x_{t+1} - x_t)(x_{t+1} - x_{t+2})] \quad threshold = \begin{matrix} 1, if\ x > threshold \\ 0, otherwise \end{matrix} \tag{4.6}$$

.

The Willison Amplitude (WAMP) is mentioned throughout the literature and it is described in (Phinyomark, Limsakul, and Phukpattaranont 2009) as being “the number of times that the difference between sEMG signal amplitude among two adjacent segments that exceeds a predefined threshold to reduce noise effects as ZC or SSC”. It is defined by equation (4.7). from our case study, the results showed an improvement in a similar result with a slight improvement in the mean of all the different classifications and so it was kept throughout the experiment.

$$WAMP = \sum_{i=1}^{N-1} f(|x_t - x_{t+1}|) \qquad threshold = \begin{matrix} 1, if\ x > threshold \\ 0, otherwise \end{matrix} \tag{4.7}$$

The Standard deviation was also added to the feature list, it is described in (Gerla et al. 2006) as a simple feature to detect movement artefacts and it is defined by the equation (4.8).

$$SD = \sqrt{\frac{\sum_{t=1}^{N}(x_t - \bar{x})^2}{N-1}} \tag{4.8}$$

4.1.4.2. Excluded Features

The features presented in subchapter 4.1.4.1 are those who were used in the project. However, there are other features that have been studied but have not been used for one reason or another, but are worth mentioning, for example:

- In (Chen and Wang 2013) it is mentioned a feature called High Order Statistics (HOS) that is said to perform well for stationary or weak sense stationary signals. There was an interest in this type of feature because of his weak sense sensibility but it was ultimately abandoned due to the fact that this project is composed of dynamic gestures.
- Also, in (Chen and Wang 2013) features like the Wavelet Transform (WT) and the Sort-Time Fourier Transform (STFT) both require the original data segments to have the same length and since that is not verified in our study these features are not used.
- Autoregression model (AR) coefficient, the signal samples are estimated by the linear combination of their earlier samples (Xi et al. 2017). It is represented as the equation (5.1).

$$x_n = -\sum_{i=1}^{p} a_i + x_{t-1} + w_t \tag{5.2}$$

where x_t is a sample of the model signal, a_i is AR coefficients, w_t is white noise or error sequence, and p is the order of AR model (Phinyomark et al

2009). Various experimental and theoretical studies have shown that the model order P = 4 or P=6 is suitable for EMG signals (Xi et. al 2017). But the results of this study concluded that the classification actually produces a worse result when using the autoregression model and so it was excluded.

Finally, in (Chen and Wang 2013) it is said that the SSC feature tends to contribute in either negative way or with no significant effect, but in our case, the non-use of this feature resulted in lower values of valid classification, and so it was kept as a valid feature.

4.1.5. Classification

In the book Data mining: Concepts and techniques (Han, Kamber, and Pei 2012) is stated that “Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The model is derived based on the analysis of a set of training data…” “…The model is used to predict the class label of objects for which the class label is unknown.” In the case of this study, the features are retrieved from the segmentation and then used to classify a pattern to identify a gesture.

In (Duda, Hart, and Stork 1999) it’s said that the perfect classification of this type of data is almost impossible and so, it is instead determinate the probability of a certain gesture to occur for each of the possible categories.

The classification is the culmination of everything that was previously developed. It is the evaluation of the features that will identify the gesture being made by the user and it is only possible with the help of everything previously talked about.

As presented, Gestures 5, 6 and 7 are the same but in different positions. Since we are not evaluating the IMU part of the system, the EMG data will be similar for these three gestures, and so, we will consider them as being the same gesture in the classification. This concept is also applied to the non-gestures since gestures 8 and 9 are to be classified as non-gestures they can also be considered as one non-gesture with the fusion of the data from both. This will result that in the classification that is a lack of balance in data, meaning that we will have three times more data for gesture 5 and two times more data for the non-gestures in relation with the other gestures. This topic will be further explored in the results chapter.

4.1.5.1. Studied Classifiers

The chosen classifiers were chosen based on the best-obtained values in the classifier testing. Results will be chosen in the next chapter, being this chapter dedicated for a brief explanation of what are these classifiers. One of the best explanations found was the ones available in the course of Machine Learning with MATLAB available online in the MATLAB academy.

- K-nn or K nearest neighbours is defined by (MATLAB n.d.) as a classification model that categorize new samples in accordance to samples found in the K nearest neighbours, being K defined by the user. When using this method, we do not have to make any assumptions about the underlying distribution of data. If the nearest neighbour is an outlier of the classification, then the new data will be falsely classified. This error is resolved by adding neighbours for a better classification.

- SVM or Support Vector Machines is defined by (MATLAB n.d.) by being a classifier that uses data by finding the “best” hyperplane that separates all data points. It can be a linear separation, but if that does not show the best results, other separation might also be useful as for example a quadratic separation.

- AdaBoost is defined by (MATLAB n.d.) as being an ensemble algorithm for multiclass classification that is “adaptive because it uses multiple iterations to generate a single composite strong learner. AdaBoost creates the strong learner (a classifier that is well-correlated to the true classifier) by iteratively adding weak learners (a classifier that is only slightly correlated to the true classifier)”. This process of adding consist of adding a weak learner to the ensemble and a weighting vector that focuses on the misclassified data of the previous round. In the end, this process results in a classifier that has higher accuracy than the weak learners. In our study there were 3 parameters that could be changed (being the maximum number of splits, number of learners and the learning rate) they were obtained by trial and error, changing each value until a maximum classification rate was found. It uses Decision trees as the learner type.

5. RESULTS AND DISCUSSION

5.1. Features

5.1.1. Threshold Calculus

Some of the presented features require the use of a threshold to retrieve the best values associated with that certain feature. This is required so that the values of the feature change whenever there is movement both from a gesture and the associated signal.

In a feature like Zero crossing, Slope sign change or Willison Amplitude, when the user is not making any gesture, being with the hand in a rest position, the represented signal for the EMG follows a line with no significant variance, figure 5.1:

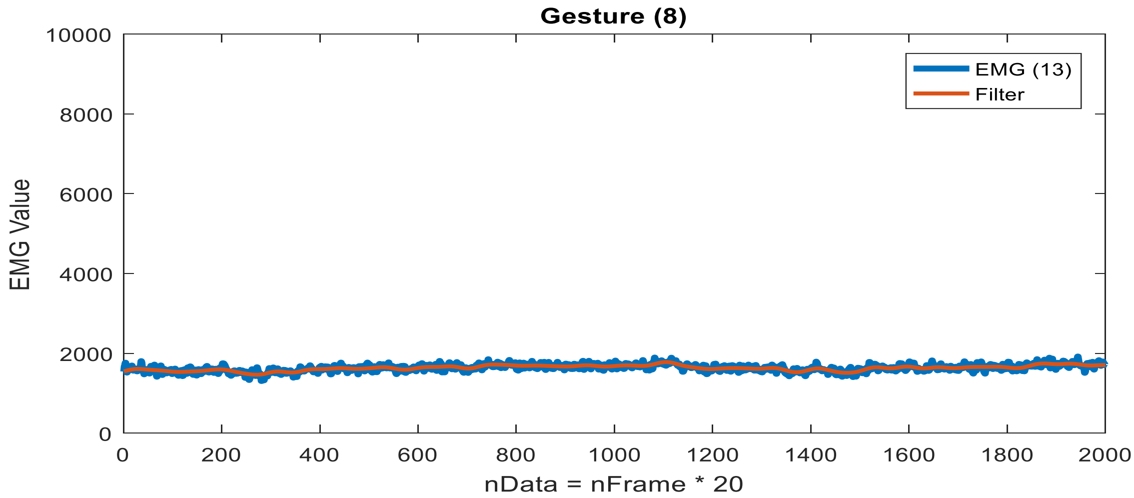


Figure 5.1 - Example of a signal found in a rest state

When we impose a threshold, it will count variances in the signal data superior to the defined threshold, and the feature value will be the sum of the number of times the signal went above it for each EMG.

To identify the threshold value for each feature that requires it, we analysed the data from a non-gesture G8 and adjusted the threshold to a value that produces a feature value of 0 in every EMG. Following this method, it was defined a threshold of 10 for Zero-crossing and 50 for both Slope sign change and Willison amplitude.

5.2. Segmentation Results

The segmentation results consist in separating the samples in gesture or non-gesture. The global variance for each EMG is calculated because a static gesture (a non-gesture) should not have a significant variance. A threshold of 0.0001 is defined, and if the variance is observed to be above that value then the sample is classified as a gesture.

We can use the classification learner app as in the classification, but because this is a separation of gestures and non-gestures, some of the classification methods can be applied that it is not possible for the "real" classification. There are classification methods such as Ensemble Gentle boost that can be used for binary results but can't be used for multiple class classification, and so, it is good to analyse the properties applied in the segmentation, but it would not be used in the classification results.

The properties analysed in this section started with the cut-off frequency and then the variance threshold. The results presented in table 1 are presented as percentage of accuracy of the total values, a confusion matrix was used to analyse the number and percentage of false positives as showed in previous chapters.

Table 1 – Results for different frequency cut off values

	Fc=10Hz	Fc=15Hz	Fc=25Hz	Fc=50Hz	Fc=200Hz
Knn cosine 2	92.3 %	92.9 %	93.1 %	93.5 %	94.0 %
Knn Euclidean 4	92.6 %	92.6 %	93.0 %	93.3 %	93.7 %
Knn Correlation 2	92.4 %	93.3 %	93.4 %	94.0 %	94.5 %
Knn CityBlock 1	94.0 %	94.5 %	94.7 %	94.8 %	95.2 %
Knn CityBlock 2	93.6 %	94.1 %	95.0 %	94.5 %	95.2 %
Knn CityBlock 3	93.7 %	93.9 %	94.1 %	94.5 %	94.6 %
Ensemble RusBoost	93.5 %	93.9 %	93.4 %	93.9 %	93.7 %
Ensemble Bagged	94.2 %	94.7 %	94.5 %	94.3 %	94.8 %
Ensemble GentleBoost	95.2 %	95.0 %	95.3 %	96.0 %	96.3 %

The data used for the segmentation and later used for classification is divided into training and testing data, but the presented results are global, being the junction of both. This process chooses randomly the samples used for testing and training, so when searching for the best parameters it was made the classification 3 times for each parameter change and then the calculated mean of the 3 results. This result was the taken value for the different classification method. This method can be observed in table 3

As it is observable in the table above, the classification learner, after reaching a cut-off value of 50Hz the difference in the results for the classification do not differ meaningfully, and as such the cut-off value is kept as 50Hz for the rest of the testing.

Having this value, it was time to calculate the optimal threshold value for the variance. This is calculated in the same way as the cut-off value, starting with a threshold of 0.0001 the following table can be constructed.

Exploiting the best cut-off value, the following table for the threshold is formed:

Table 2 - results of classification with different variance threshold value

	Thr=0.0001	Thr=0.001	Thr=0.01	Thr=0.1
Knn cosine 2	93.5 %	93.8 %	93.9 %	93.3 %
Knn Euclidean 4	93.3 %	93.4 %	93.3 %	93.5 %
Knn Correlation 2	94.0 %	94.1 %	94.1 %	94.1 %
Knn CityBlock 1	94.8 %	95.1 %	95.1 %	94.9 %
Knn CityBlock 2	94.5 %	94.8 %	94.8 %	94.7 %
Knn CityBlock 3	94.5 %	94.5 %	94.6 %	94.4 %
Ensemble RusBoost	93.9 %	94.1 %	94.0 %	93.9 %
Ensemble Bagged	94.3 %	94.3 %	94.3 %	94.2 %
Ensemble GentleBoost	96.0 %	96.1 %	96.2 %	96.1 %

As its observable in table 2, there is no significant difference in the chosen threshold for the individual EMG variance, so the threshold was kept with its original value of 0.0001 throughout the rest of the testing.

The data was also observed with recourse to a confusion matrix that let us identify where are the issues on the classified data. An example of a confusion matrix for the segmentation follows:

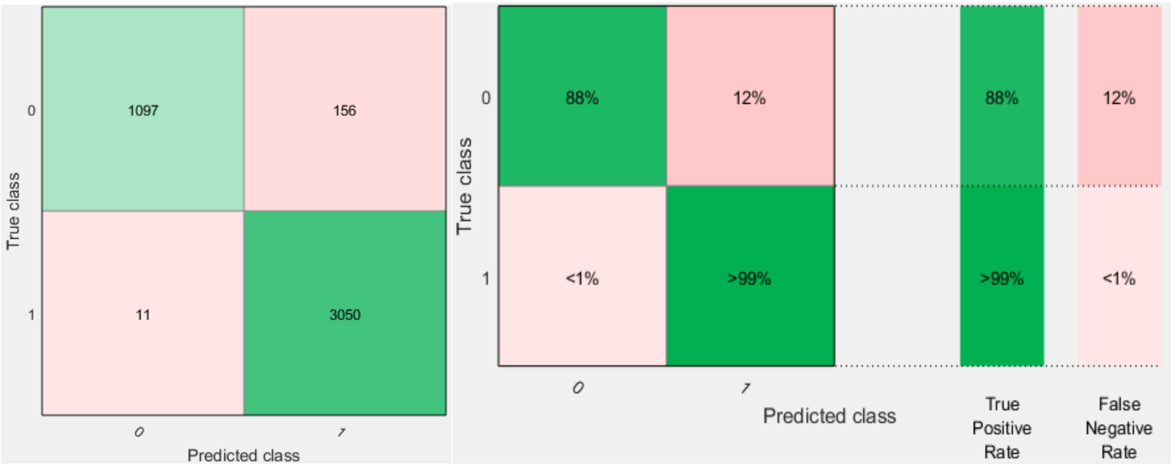


Figure 5.2 - Confusion Matrix for segmentation in numbers and percentage

This confusion matrix is related to a classification method using decision trees with a *GentleBoost* Ensemble. In green are the samples correctly identified in the segmentation, in red the ones that were mistaken during the segmentation. The samples belonging to the gestures and non-gestures will always be decompensated since we have 7 gestures and 2 non-gestures. But observing the confusion matrix, we can see that the biggest problem is identifying non-gestures as gestures. This result can be explained by the similarity of some of the performed non-gestures in comparison to the gestures studied. There is more than a 99% chance of identifying a gesture correctly and a 12% chance of mistaking a non-gesture for a gesture.

5.2.1. Influence of a balanced Dataset in Segmentation

As it is explained in more detail later, in the subchapter about the influence of the dataset for the classification, the segmentation also started to be analysed with an unbalanced dataset as it can be seen in figure 5.2.

Analysing the influence of balancing the data set in the segmentation and using the same parameters and classifier as the previous figures we obtain the following.

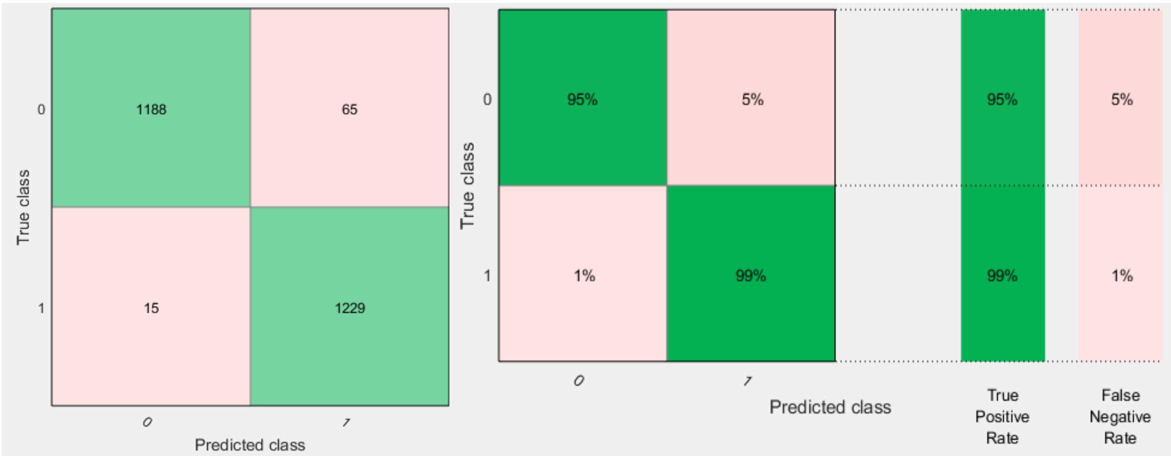


Figure 5.3 - Confusion matrix for Segmentation with a balanced dataset

As it is perceptible there is a significant reduction in the false negative rates for the non-gestures, meaning there are fewer non-gestures recognized as gestures when we have a balanced dataset. The overall result of the segmentation increased by 0,8% being around 96,8% for ensemble *GentleBoost* using a cut off value of 50 and a variance threshold of 0.0001.

5.3. Classification Results

The classifiers were chosen from the MATLAB library through testing, using the classification learner APP available on the software and evaluating the results. The available classifiers were tested with different parameters. There were made more than 100 tests, and, in the end, 14 classifiers with classification results above 90% were chosen:

1. K-nn Cosine, with the 3 nearest neighbours
2. K-nn Euclidean, with the 3 nearest neighbours
3. K-nn Correlation, with the 4 nearest neighbours
4. 4 K-nn City Block, with 1,2,3 and 4 nearest neighbours
5. 3 K-nn Spearman, with 1,2 and 3 nearest neighbours
6. SVM Linear with a box constraint level of 6
7. SVM Quadratic with a box constraint level of 6
8. Decision trees with Ensemble AdaBoost with 80 maximum number of splits, 200 number of learners and learning rate of 1

9. Decision trees with Ensemble Bag with 80 max number of splits and 200 number of learners

Table 3 - Different classifiers used for classification results enquiry

	Cutoff = 15; AR = 4; all features			mean
K-nn cos3	93,3	93,7	93,7	93,57
K-nn EU3	93	93,4	93,2	93,2
K-nn COR4	93,4	93,1	93,1	93,2
K-nn CB1	95,6	95,4	95,7	95,57
K-nn CB2	94,5	94,2	94,3	94,33
K-nn CB3	94,7	95	94,8	94,83
K-nn CB4	94	94,2	94,5	94,23
K-nn SP1	96	96,2	96,2	96,13
K-nn SP2	95,5	95,6	95,6	95,57
K-nn SP3	95,7	95,7	95,8	95,73
EADABOOST	97,8	97,5	97,6	97,63
EBAG	97,2	97	97,1	97,1
SVML6	94,3	94,4	94,6	94,43
SVMQ6	96,7	96,8	96,5	96,67
				95,16
				96,81

Table 3 shows a typical evaluation of a parameter. The study follows the same logic presented for the segmentation, being made 3 measurements for each change and then being calculated the mean value of each.

To evaluate the parameters for using the decision trees classifiers with Ensemble parameters, a study of the individual parameters had to be made. Table 4 demonstrates the difference of the results obtained by trial and error throughout this study, using all presented features, a cut-off of 100 Hz and a sixth order autoregression coefficient.

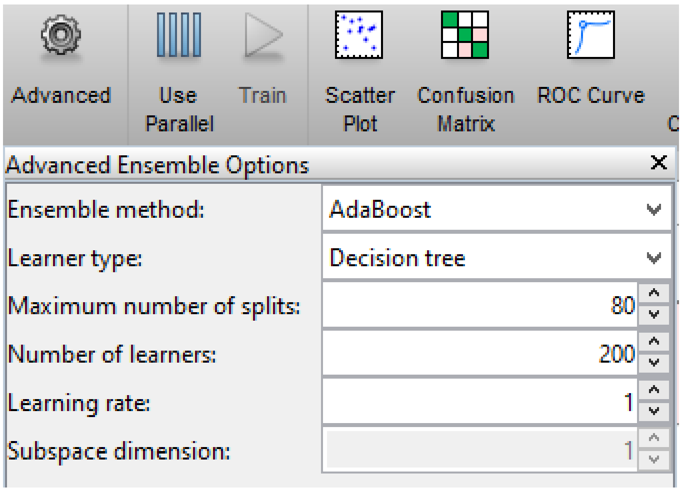


Figure 5.4 - Different parameters available for ensemble

Table 4 - Variance of classification results with alternation of property values of ensemble classifier

M.N.S.	20	50	80	90	80	80	80	80	80	80	80	80
N. L.	30	30	30	30	80	150	200	300	200	200	200	200
L. R.	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.6	0.9	1.0
Result %	91.6	95.9	96.2	95.8	97.8	98.2	98.4	98.4	98.6	98.7	98.8	99

M.N.S. represents the maximum number of splits, N.L. represent the number of learners and L.R. represents the learning rate, all parameters found on the Ensemble AdaBoost method. This trial and error method to find the best parameters was used for all classifiers used in the study, being always used the values that appear to produce the best results.

5.3.1. Influence of a balanced Dataset in classification

At the beginning of this study it was used an unbalanced dataset, due to the merging of gestures 5, 6 and 7, that are the same gesture performed in different positions, and the merging of the non-gestures 8 and 9, meaning that the gestures (1 to 4) end up with a third of the data in comparison to gesture 5 and a half in comparison to the non-gesture 8.

An uncompensated dataset will have an impact on gesture recognition since gestures with more samples will naturally produce more errors and the errors present in the gestures with fewer samples will have less impact on the final percentage of the classification.

To correct this issue, some adjustments have been made, collecting some samples for the gestures where there was a lack of samples and despising some of the acquired data for the gestures with too many samples. This was both used for the segmentation and classification part of the process.

The major difference in terms of results was that the ensemble AdaBoost classifier fell from being the one showing the best results and K-nn classifiers got an improvement to their results.

The following images show the confusion matrixes using the same variables but comparing a balanced to an unbalanced data set.

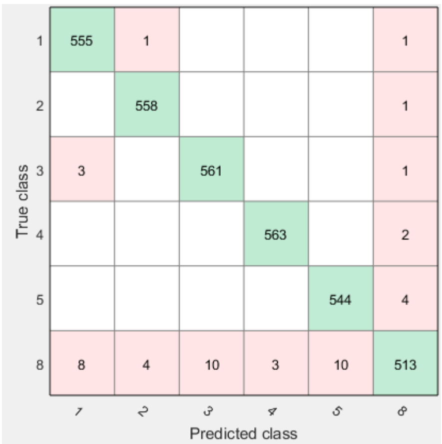


Figure 5.5 - Confusion matrix balanced dataset

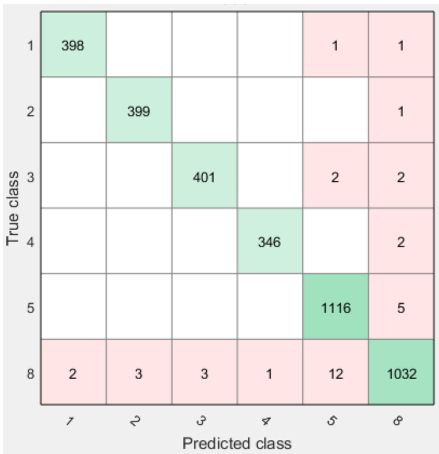


Figure 5.6 - Confusion matrix unbalanced dataset

Both these figures, it was used Ensemble AdaBoost for the classification and the same features were applied. We can see that the issue in both is the false positives of the non-gestures and that in the balanced dataset this issue gets worse. But considering that there are more samples for the gestures in the balanced dataset, this also means that the percentages of true gesture recognition are higher. Because of this, the overall result stays practically the same for both situations.

The confusion matrix for the classification works in the same way it does for the segmentation, being in green the true positives. These are the times that the classifier correctly identified the gesture being made and in red the false positives is the times that the classifier incorrectly identified the gesture being performed for another.

5.3.2. Influence of Features

To evaluate the influence of the features in the classification results, two methods can be applied, the first one consists in classifying the gestures with the parameters already defined but with only one feature at the time. The other method consists in making an opposite strategy, using all the features except one for the classification and observing the results. Although the test was always made for the 14 different classifiers mention in table 3, when presenting the results, it will only be presented the best results for each type of classifier. As such is only presented the classifier for K-nn Spearman with 1 nearest neighbour, Quadratic Support Vector Machine SVM with a box constrained level of 6 and Ensemble AdaBoost with the parameters already mentioned.

Table 5 - Influence of the use of a single feature in the classification results of the three best classifiers

	RMS	WL	ZC	MAV	SSC	Wamp	AR6	SD
Knn (%)	93.1	96.1	92.2	91.5	92.0	95.6	78.7	95.2
SVM (%)	95.1	96.7	95.1	92.7	94.6	96.0	83.2	94.8
EAda (%)	93.6	96.5	95.2	94.3	95.6	96.5	83.2	95.8

From these results of table 5, is clear that the feature Auto Regression model (AR) produces results much lower than the other features. This will prove to have a negative impact on the classification using all the features at the same time, being this the reason for the exclusion of this model. The other features produce a very positive result on their own.

This study method can also be used to evaluate the chosen thresholds for each feature. An example is shown in table 6.

Table 6 - Influence of changing the threshold value in Classification with Zero-crossing feature alone

	ZC threshold = 1	ZC threshold = 10	ZC threshold = 50
Knn (%)	81.5	92.2	95.5
SVM (%)	86.8	95.1	96.1
EAda (%)	87.1	95.2	96.5

It is observable that the threshold value plays an important role in the classification results of a certain feature. The results reach a peak when the threshold value is 50 for all the three features that require a defined threshold value. However, for zero-crossing it is opted to use of the threshold equal to 10, due to it producing a better result when combined with other features. This phenomenon is also visible for slope sign change, where it is used a threshold of 20. However, in Willison amplitude it is better to use a higher threshold value, thus being defined as 50.

5.3.2.1. Combining Features

Once the results for the classification with individual features is evaluated, we can make the combination of several different features to study the performance of results. Using a balanced data set and the same method and variables already presented, the following table is created:

Table 7 - Combination of features for classification results

	All Features	WL; SD; Wamp	Without AR	Without AR and SSC
Knn (%)	97.7	98.2	99.1	98.9
SVM (%)	98.5	98.3	98.8	98.6
EAda (%)	98.6	98	98.6	98.6

When analysing table 7, we can clearly see that adding the autoregression model it will only damage the pretended results. It was studied the possibility of using only the three best individual features, but the results show that the performance decreases by roughly 1% and so this idea was abandoned. When testing other alternatives, it was studied the demotion of the number of features, but the results always decreased when removing more than one features. Although the difference is not significant it was chosen to maintain the seven features presented in chapter 4.1.4.

With the method described, the best-obtained result was a 99.3% and the confusion matrix is as follows:

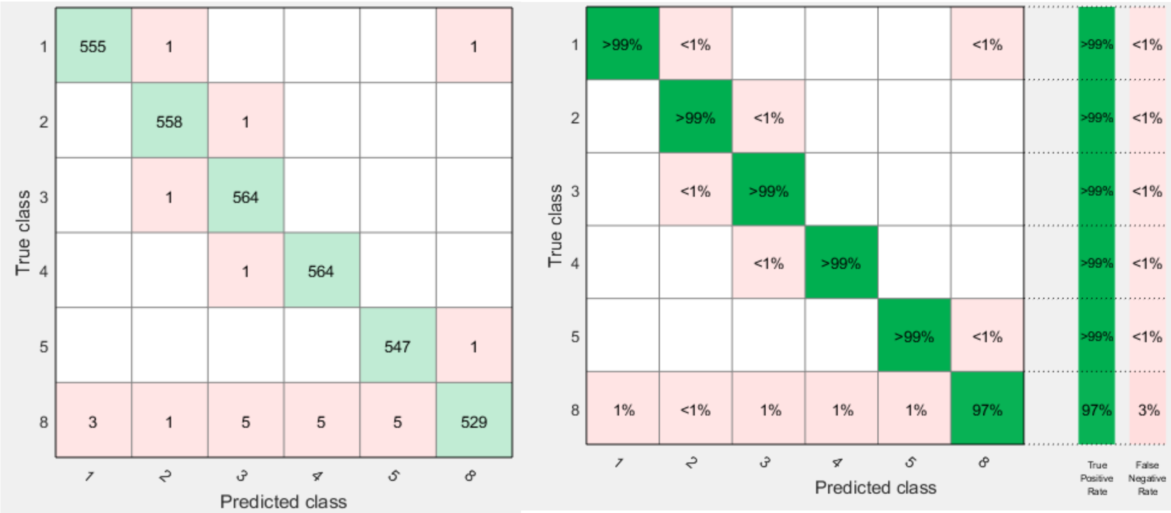


Figure 5.7 - Best obtained value for a confusion matrix, displayed both in numerical and percentage

This confusion matrix shows that the old problem of recognising some non-gestures as gestures is still alive, but it also shows promising results occurring with less frequency when comparing with other classification methods presented for offline classification.

5.4. Online implementation

Although the offline implementation of this program presents good results, the online implementation is in another league of its own. The online implementation consists of the junction of everything that was previously studied but in a live environment.

It should be reminded that the studied device is a prototype and it was not expected to be perfect, to begin with. For example, we can choose the best samples from the dataset to train the classifiers, the outliers or the corrupt data can be eliminated and there aren’t multiple things happening at the same time. Collecting, processing segmenting and classifying data are made separately permitting that the conditions become ideal for good results.

In the online implementation, there are some issues that must be assessed. The issues that have been mention at the beginning of this thesis are still present. Using the visual studio to collect data, we can observe a certain lag while collecting frames. There is a frame drop usually when various gestures are made in a row. This phenomenon can be visualized in a way by figure 5.8.

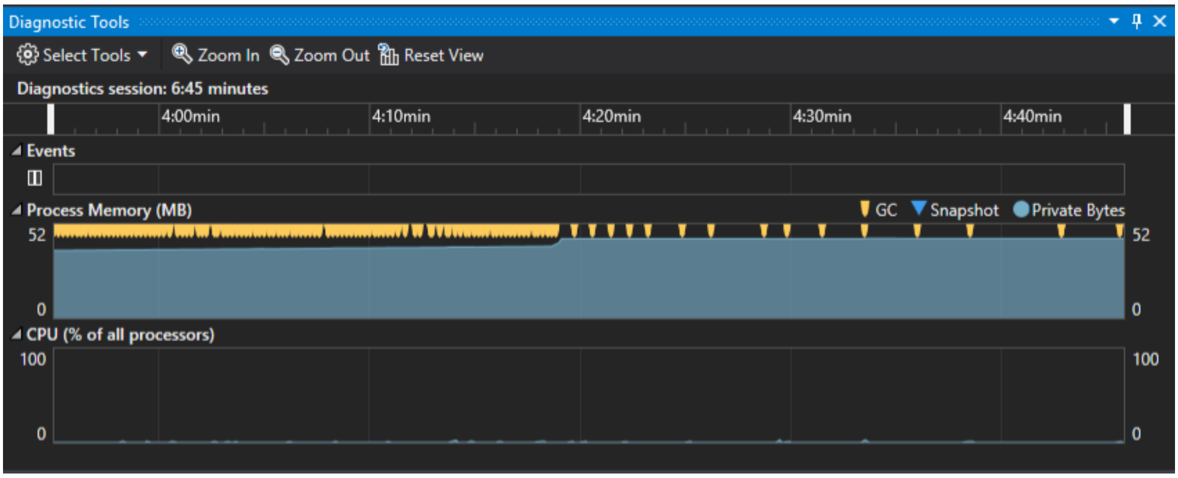


Figure 5.8 - Diagnostic tools on visual studio

The yellow ticks in the process memory correspond to the timestamps that indicate the start of a manage heap garbage collection. The heap garbage collection process consists in a continuous replacement of data. As data is being transmitted it is being saved and after the transition of data, there is a need to relocate space for data. So, when the transition is concluded the older data is moved to the garbage.

As it is perceptible from the figure 5.8 when the frame data flows normally (3.55min-4.10min timestamp) there is a big density of heap garbage collection cycles and when the lag is noted (4:20min-onwards) there is a smaller density of this process occurring. Also, in MATLAB the command window is supposed to update every two seconds with the message of the detected segmentation and classification values. As the lag is occurring this is not what happens, being the time of the refresh higher than its supposed.

At the arrival of the message, only one message appears as usual, what leads me to believe that there are problems from the connection of the hub to the computer which result in a portion of time without any data being analysed. This problem can possibly happen due to the use of the live data requiring more resources from both the computer and the hub that simply aren't there. Changing for a better computer resulted in an easier time connecting the devices, but the change of hub or the connecting cables can't be tested due to lack of replacement parts.

Another noticeable problem is that sometimes in a sample, some frames of some EMG do not transmit any message. This results in empty frames of data, being every frame corresponding to 20 data of an EMG. This is the usual corrupt data that are eliminated in the offline analysis. This happens sometimes in the connection usually following the lag portions and the blank data vectors can range from a few to dozens at the time, with no defined value. An example of the previously described phenomenon is presented in figure 5.9.

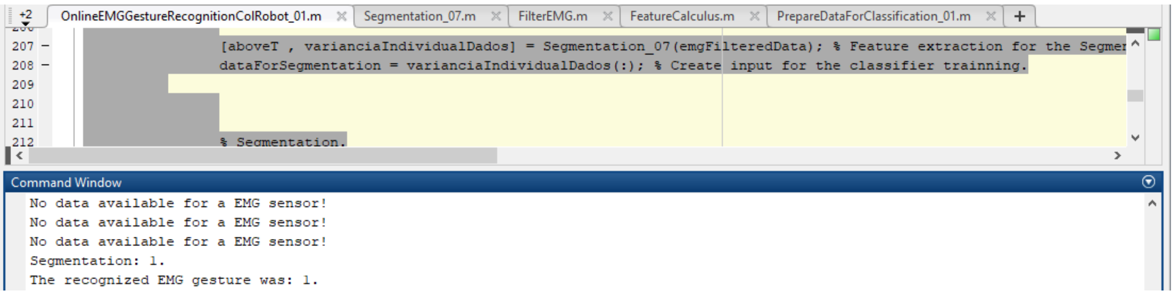


Figure 5.9 - Example of an error occurred in the online processing

With the conditions presented, taking into account that these errors would influence in a negative way the results, we tested the online accuracy of classification. To make this test, we started by randomizing 10 rows of 20 gesture each. Then after connecting the device, the sequence of gestures was made, and the results noted. In total 200 tests were made. In this, is evaluated both the segmentation and the classification with the segmentation showing better results than the classification. when performing a gesture, 30 of the 200 performed gestures were detected as non-gestures, which means a segmentation accuracy of 85%. As for the classification, 72 of the performed gestures weren't detected as being the correct gestures, which means a classification accuracy of 64%. This result takes into account the wrongly classified and segmented gestures, which means that from the 170 correctly segmented gestures, 42 were after wrongly classified, being the classification accuracy of 75.3%.

6. CONCLUSION

When the signal is acquired with success it is possible to identify very accurately the gesture being made which is a good signal that the device being created has potential. The good results don't mean that it can be applied right away to online classification because of all the issues that have been mentioned throughout the work. Without the proper fixes of the aspects already mentioned, it won't be recommended the use in an online environment.

Offline segmentation showed an accuracy of 96.3% and the offline classification results showed an accuracy of around 99.1% with the best result reaching 99.3%.

It was found that the best results came from the conjugation of 7 different features and that some features can have a bad influence on the results. In the same way, the chosen threshold for the features has an impact on the results.

Online implementation showed much worse results with segmentation accuracy of around 85% and classification results of 65%.

For the future work, the design flaws should be assessed, a more practical and comfortable version of the device could be created where there wasn't a need for 5 IMU. With just 2 IMU, (in the arm and forearm of the implemented device) is possible to create a wide range of gestures. Apart from the design flaws, the online implementation should be worked on, a bigger data set for the gestures should be created but also the connection problems must be tested in order to obtain a better performance in an online environment. It should also be tested for the performance of different users. Exploration and implementation of the IMU data in the gesture spotting is also something to consider.

BIBLIOGRAPHY

Attal, Ferhat et al. 2015. “Physical Human Activity Recognition Using Wearable Sensors.” *Sensors* 15(12): 31314–38.

Chen, Xun, and Z. Jane Wang. 2013. “Pattern Recognition of Number Gestures Based on a Wireless Surface EMG System.” *Biomedical Signal Processing and Control* 8(2): 184–92. http://dx.doi.org/10.1016/j.bspc.2012.08.005.

Clancy, E A, E L Morin, and R Merletti. 2002. “Sampling, Noise-Reduction and Amplitude Estimation Issues in Surface Electromyography.” *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology* 12(1): 1–16.

Duda, Richard O., Peter E. (Peter Elliot) Hart, and David G. Stork. 1999. *Pattern Classification*.

Farina, D., and R. Merletti. 2001. “A Novel Approach for Precise Simulation of the EMG Signal Detected by Surface Electrodes.” *IEEE Transactions on Biomedical Engineering* 48(6): 637–46.

Farina, Dario et al. 2014. “The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges.” *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22(4): 797–809.

Gerla, Vaclav, Lenka Lhotska, Vladimir Krajca, and Karel Paul. 2006. “Multichannel Analysis of the Newborn EEG Data.” *Itab 2006* (May 2014): 1–6.

Han, Jiawei, Micheline Kamber, and Jian Pei. 2012. San Francisco, CA, itd: Morgan Kaufmann *Data Mining: Concepts and Techniques*.

Kim, Jonghwa, Stephan Mastnik, and Elisabeth André. 2008. “EMG-Based Hand Gesture Recognition for Realtime Biosignal Interfacing.” *Conference paper*.

Kleine, B U, N P Schumann, D F Stegeman, and H C Scholle. 2000. “Surface EMG Mapping of the Human Trapezius Muscle: The Topography of Monopolar and Bipolar Surface EMG Amplitude and Spectrum Parameters at Varied Forces and in Fatigue.” *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology* 111(4): 686–93.

Kumar, Gaurav, and Pradeep Kumar Bhatia. 2014. “A Detailed Review of Feature Extraction in Image Processing Systems.” In *2014 Fourth International Conference on Advanced Computing & Communication Technologies,* IEEE, 5–12.

Lopes, João. 2016. " Gesture Spotting from IMU and EMG Data for Human-Robot Interaction".

MATLAB. “AdaBoost - MATLAB & Simulink.” https://www.mathworks.com/discovery/adaboost.html.

———. “Machine Learning with MATLAB - Nearest Neighbour Classification.” https://matlabacademy.mathworks.com/R2017b/portal.html?course=mlml#chapter=4&lesson=2&section=1.

———. “Machine Learning with MATLAB - Support Vector Machines.” https://matlabacademy.mathworks.com/R2017b/portal.html?course=mlml#chapter=4&lesson=6&section=1.

Norali, A N, and MH Mat Som. 2009. “Surface Electromyography Signal Processing and Application: A Review.” In *Proceedings of the International Conference on Man-Machine Systems (ICoMMS),* , 11–13.

Phinyomark, Angkoon et al. 2014. “Feature Extraction of the First Difference of EMG Time Series for EMG Pattern Recognition.” *Computer Methods and Programs in Biomedicine* 117(2): 247–56. http://dx.doi.org/10.1016/j.cmpb.2014.06.013.

Phinyomark, Angkoon, Chusak Limsakul, and Pornchai Phukpattaranont. 2009. “A Novel Feature Extraction for Robust EMG Pattern Recognition.” *Journal of Computing* 1(1): 71–80.

Rojas-Martínez, M., M.A. Mañanas, J.F. Alonso, and R. Merletti. 2013. “Identification of Isometric Contractions Based on High Density EMG Maps.” *Journal of Electromyography and Kinesiology* 23(1): 33–42.

Scheme, Erik, and Kevin Englehart. 2011. “Electromyogram Pattern Recognition for Control of Powered Upper-Limb Prostheses: State of the Art and Challenges for Clinical Use.” *The Journal of Rehabilitation Research and Development* 48(6): 643.

Simão, Miguel, Nuno Mendes, Olivier Gibaru, and Pedro Neto. 2017. “University of Coimbra Dear Editor , This Review Paper Is the Revised Version of Document Number BSPC-D-17-00647 . According to the Reviewers Comments the Paper Was Improved and Updated : (1) Novel References ; (2) Comparison and Discussion among Diffe."

Veer, Karan, and Tanu Sharma. 2016. "A Novel Feature Extraction for Robust EMG Pattern Recognition." *Journal of Medical Engineering and Technology* 40(4): 149–54.

Xi, Xugang, Minyan Tang, Seyed M. Miran, and Zhizeng Luo. 2017. "Evaluation of Feature Extraction and Recognition for Activity Monitoring and Fall Detection Based on Wearable SEMG Sensors." *Sensors (Switzerland)* 17(6).