AALBORG UNIVERSITY
STUDENT REPORT

Robotics

P4 Project - Spring 2016

# Remote Surgery

***Sensing the Surroundings***

Group 16GR461

AALBORG UNIVERSITY
STUDENT REPORT

**4th Semester**
**Det Teknisk-Naturvidenskabelige Fakultet**
**School of Information and**
**Communication Technologies**
Fredrik Bajers Vej 7B
9220 Aalborg East
webinfo@es.aau.dk

Title:
Remote Surgery

Theme:
Sensing the Surroundings

Subtheme:
Biomedical Robotics

Project period:
P4, Spring 2016
D. 01/02/2016 - 27/05/2016

Group number:
16GR461

Group members:
David Černý
Biranavan Pulendralingam
Emil Blixt Hansen
Rasmus Eckholdt Andersen
Steffen Madsen
Marike Koch van Amstel

Supervisor:
Ernest Nlandu Kamavuako

Co-supervisor:
John Hansen

Prints:
9

Pages:
98

Appendices:
5

Abstract

This project investigates the use of electromyographic (EMG) pattern recognition of hand motions as an alternative control method for a robot in the context of remote surgeries. Remote surgery is a developing field as it eliminates the distance between patient and surgeon. All remote surgical robots investigated in this project are controlled by a joystick console interface. The goal is to create a supervised machine learning algorithm to correctly classify seven hand motions based on recorded EMG signals. The final product is a system, which can record EMG signals and recognize the seven defined motions. The system is able to classify off-line recordings with a prediction accuracy of min. 95%. This was done using six features (MPR, SSC, AM WAMPL, ZC, LOG) with a PCA dimensional reduction algorithm and a LDA classifier. The system was tested with a Fitts's law test, to check its stability. However, the test was not successful against the required minimum $R^2$ value of 0.9. Five of the seven motions were classified with only minor errors, however the two other motions, which were very similar, were prone to confusion with each other.

# Preface

This report is written by $4^{th}$ semester student group 461 on the bachelor study of Robotics at Aalborg University. The project was carried out during the time from January to May 2016. The complexity of technical and programming language is on a level expected of students on this education program to use and understand. All programming was carried out in MATLAB environment. The topic of this project is remote surgery and usage of EMG signal for surgical robot control. The goal of this project is not to deliver a solution to remote surgery. The main goal is to create a concept and to carry out the signal processing part of the system.

A thanks belongs to Ernest Nlandu Kamavuako, the project supervisor, who helped the authors and encouraged them in the process of writing the report.

**Reading Directions**
This report uses modified Chicago style of referencing. All sources are listed in an alphabetical order in the end of the report. References to those sources in text are written after paragraph in brackets (author's name, year). References to figures are written in similar manner in caption of a figure. If there is no reference in the caption, the figure has been created by the authors. In the report all test and approaches are described in two separate chapters. One for the methodology and one for test result. This report contains appendices which can be found after the content. The report also comes with a CD containing:

- Video of the GUI test.
- MATLAB scripts.

Written by:

Biranavan Pulendralingam

Emil Blixt Hansen

Rasmus Eckholdt Andersen

Steffen Madsen

Marike Koch van Amstel

David Černý

# Contents

# Chapter 1

# Introduction

The majority of the world's population lives in cities. It is estimated that over 70% of the world's population will inhabit urban areas by 2050. This indicates that it is more attractive to live in or around cities, than in rural areas. Based on research it is assumed that the reason for this are the opportunities, lucrative jobs and wealth in the cities. Therefore it becomes more difficult to employ specialised workforce, e.g. brain surgeons and even harder to get recent graduated surgeons in rural areas. (Eugenie L. Wacter 2001) Generally, surgeons in rural areas of USA tend to be male with an age above 50 years. It is supposed that such surgeons are more likely not to consider income as a factor when choosing an area to open a practice. Besides of the urbanising tendencies, another reason for the lack of specialised surgeons in rural areas is that the education to become a surgeon often takes point in sophisticated practices, usually located in urban areas. It can be expected that rather outdated practices in a rural area are less desirable for a young surgeon to work in. Also because they do not provide the same career opportunities as the larger cities. Due to this, surgeons are more likely to choose sophisticated practices in urban areas as their workplace, which has led to a problem with lack of surgeons in the rural areas of USA. (Surg 2007)

A way to counter this problem could be to use robots for remote surgeries. Remote surgeries make it possible for patients to receive medical attention carried out by a professional surgeon, no matter what distance lies in-between. Using a surgery robot provides the opportunity for a patient, in a rural or remote area to have a surgery done with the newest technology, and a specialised surgeon, which is an advantage for both the patient and the surgeon. This also makes it possible to use robots for surgeries in war-zones. In this case, it is possible for the surgeon to perform a surgery in a dangerous area, while staying in a safe place outside the war-zone. (Overall 2013)

The usage of surgery robots may affect the quality of the surgery carried out by the surgery. According to Herlev Hospital the use of a surgery robot will have the following positive effects for the patient: (Bergqvist 2016)

- Less blood loss.
- Smaller scars.
- Lower risk of infection.
- Less pain inflicted.
- Shorter hospitalisation time.

- Less complications.

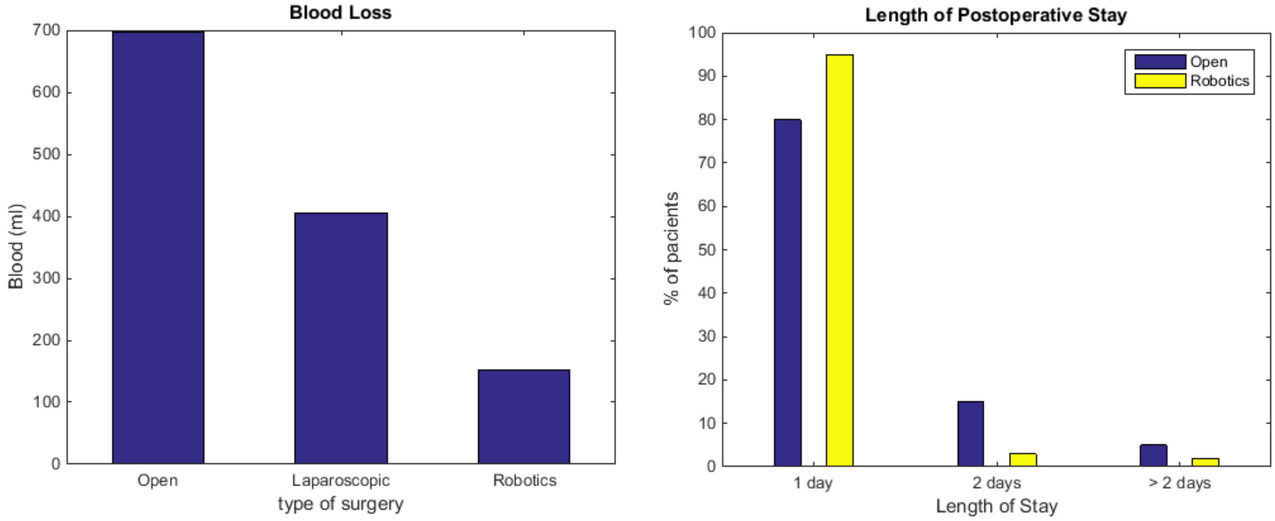


**Figure 1.1:** *Comparison of blood loss and hospital stay among different surgical approaches. (Samadi 2016)*

Figure 1.1 supports the statements from Herlev Hospital with statistics from David Samadi, Chariman of Urology, Chief of Robotic Suergery at Lenox Hill Hospital. The figure compares blood loss and length of hospitalisation of different surgical methods applied for prostate surgery. In case of this survey, the blood loss is indeed reduced and the postoperative stay is shortened. Based on this it can be presumed that using a surgery robot results in less complications and shorter hospitalisation time which benefits the patient and prevents overpopulating of hospitals.

The benefits mentioned are compared to traditional surgery. The same benefits can be achieved when using normal minimal incision procedures (MIP). When comparing the procedures (robot vs. MIP), a conducted study shows that using the robot causes an increase in procedural costs. There is a small but significant increase in the risks of intraoperative complications. (D. Wright et al. 2014) This partly contradicts Herlev Hospital statement about complications, however, the application of the robotic solutions creates new opportunities for surgeries to occur where they otherwise would not. Therefore, these disadvantages might be rather irrelevant. This is e.g. the case when talking about surgery in war-zones or remote or rural areas that are either far away or not easily accessible.

# Chapter 2

# Existing Methods

In 1984 a robotic manipulator named Arthrobot was first used as an assisstive device in a surgery room. The Arthrobot was used doing an orthopaedic surgery where it held the patients limbs in the right position while the surgeon performed the surgery. The Arthrobot was voice-controlled by the doctor, so there was no need for physical control of the manipulator. (Lechky 1985)

In 2001 the first remote surgery was introduced with the Zeus robotic surgical system. This transatlantic surgery took place in Strasbourg, while the surgeon operated 6000 km off-site in New York. The visual-feedback and data were transmitted by a 14 000 km fiberoptics circuit, which enabled the delay of only 155 milliseconds. Such delay was inside the range of the estimated safe lag of 330 milliseconds delay. (Gottlieb 2001)

In this chapter a surgical system designed for remote surgery is examined. The system, the da Vinci surgical robot, introduced in 2000, is examined in order to obtain information about the interfaces of a modern surgical system. The main focus of section 2.1 is to describe how the robot and its controls work. This chapter will furthermore look into a future perspective for robot assisted surgery, and investigate other alternative surgery robots available on the market (section 2.2). This will be done in order to investigate which methods of controlling a surgical robot exist today and to set a motivation for an alternative method for controlling an assistive surgical robot.

## 2.1 Da Vinci Surgical System

The next section is based on the company Intuitive Surgical own information from (Intuitive-Surgical 2015) about the da Vinci system. The da Vinci surgical system is a surgical robot created by the company Intuitive Surgical, and is designed to perform and facilitate complex surgeries using a minimally invasive approach. Furthermore the da Vinci makes it possible to carry out remote surgery in rural areas.

The initial price of the system is over \$1 000 000,- (6 648 800,- DKK) and thus requires a larger initial investment than the surgeon. Furthermore a fourth arm costs \$175 000,- (1 167 460,- DKK) to purchase, but it replaces the need to have an OR assistant nurse at the table (around \$80 000,- (533 980,- DKK) per year).

There are currently three versions of the da Vinci surgical robot, though the method of controlling is the same in all three versions. This section will therefore not go into details with the different versions, but rather describe the da Vinci system in general. This technology consists of the following three key components; Ergonomic control system, Patient-side cart with four interactive robotic manipulators and High definition 3D vision system.

### 2.1.1 The Setup

The da Vinci system consists of two subsystems communicating with each other. At the surgeons site a console is located that allows the surgeon to operate the system. The surgeon seated at the console has a clear 3D image of the surgical field from the patient's site and is able to control the robotic part of the system located at the site of the patient, through the use of joysticks (see figure 2.1). The system is able to register the surgeon's movements of the joysticks, and apply them to the manipulators at the patients site. The movements of the joysticks are scaled, filtered and translated control signals for the manipulators. The system runs safety checks during procedures to enhance the surgical precision and control. The system is able to filter hand-tremor, and scale the surgeon's motion into micro movements.

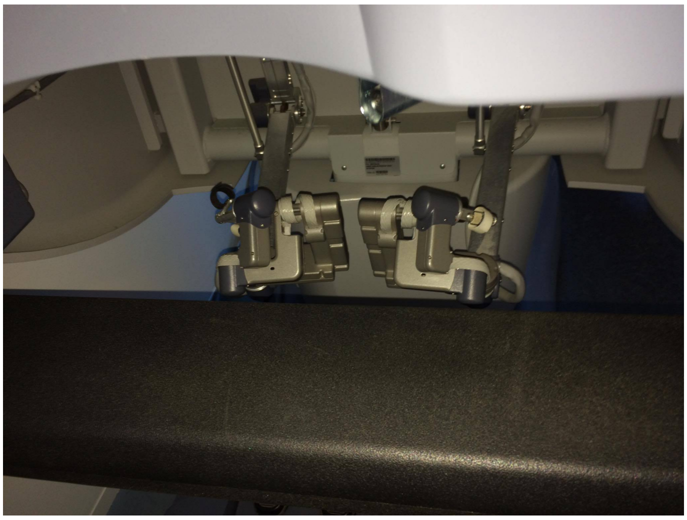
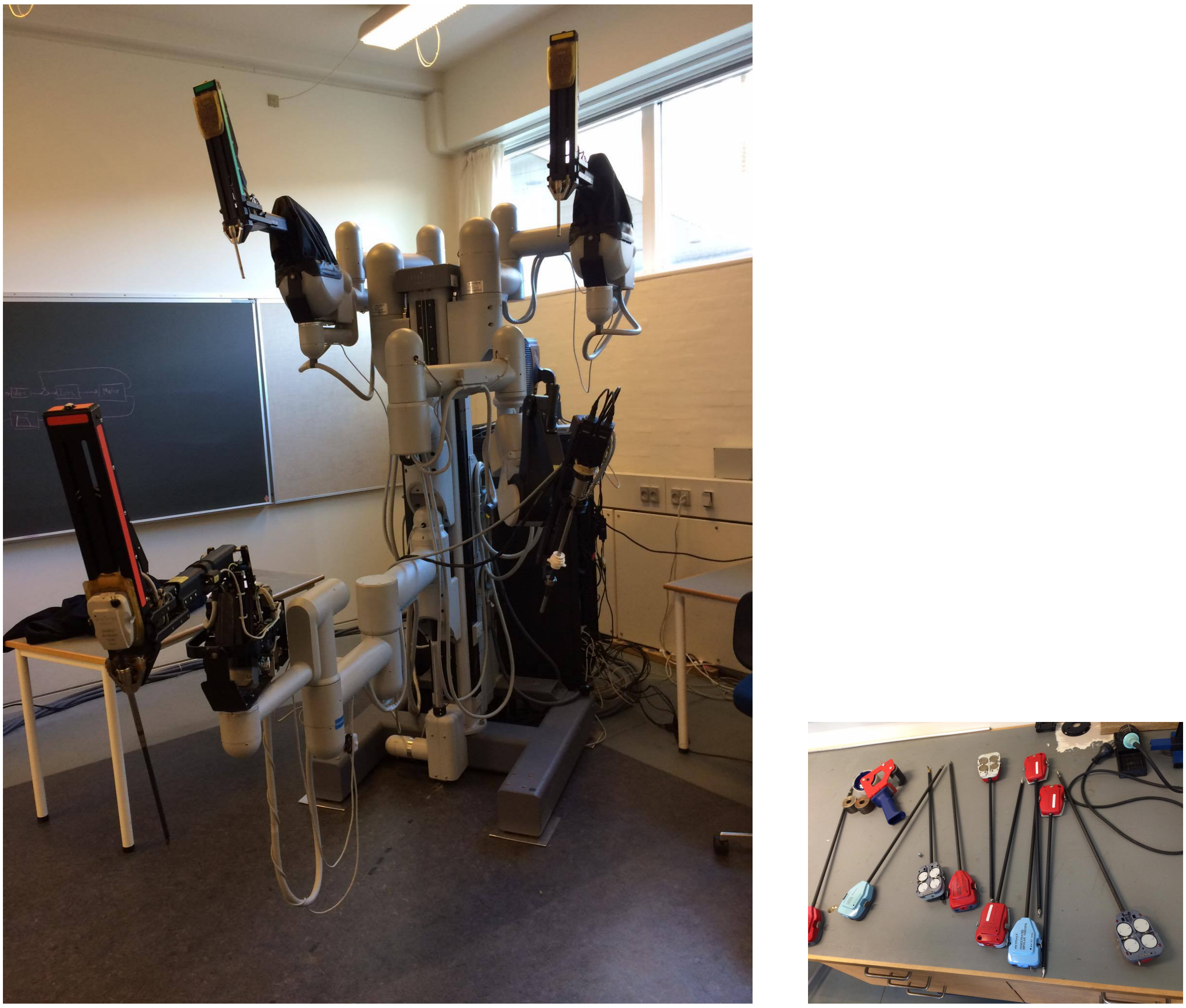


**Figure 2.1:** *The joystick used to control the da Vinci.*

At the site of the patient the surgical robot cart, as can be seen in figure 2.2a, is placed. On the cart the "da Vinci HS" camera is located, which provides a high definition three dimensional magnified view. A touch screen monitor on the cart provides the surgical team with the intern footage or other important chosen visual information. The screen also has a Smart-board function where nurses can guide the surgeon while cutting, by touching the screen and drawing. This is then visualised at the operator's console. The cart as mentioned has four manipulators. The manipulators can be equipped with different surgical instruments called "EndoWrists" which all are easily exchangeable due to the simple click-locks they are mounted with. The EndoWrists come in a wide selection of specialised tools with destinct tip designs (see figure 2.2b). The EndoWrist instruments themselves have seven degrees of mechanical freedom (DoF) and 90 degrees of articulation. The da Vinci EndoWrist instruments are designed to mimic a human-being's hand movements. The EndoWrists are designed with an unique wrist architecture that enables the system to perform surgical manoeuvres, which would not be possible to perform with conventional laparoscopic tools.



**(a)** *The da Vinci with the four arms.*

**(b)** *The da Vinci's EndoWrist.*

**Figure 2.2:** *The parts of the da Vinci system at the patients site*

The system is commonly used in a variety of surgical fields such as; cardiac surgery, colorectal surgery, gynecologic surgery, head and neck surgery, thoracic surgery, and urologic surgery.

## 2.2 Zeus

Another solution for remote surgeries is the Zeus surgical system (see figure 2.3), made by the company Computer Motion. The first Zeus prototype came in 1995. In 2001 it received a clearance to assist during certain types of surgeries and it was marketed in 2003 for $975 000,- (6 490 000,- DKK). (University 2016)

This system consists of three robotic arms. One of them is a six DoF endoscope-holding robotic arm called AESOP (Automated Endoscopic System for Optimal Positioning), which is voice activated, meaning that the surgeon can control its behaviour by using predefined voice commands. AESOP has been used before as a stand alone device for holding and operating an double camera endoscope. It is used for visualisation and transmits a high quality steady image. The two remaining arms are extensions of the surgeon's left and right arm. They can be equipped by specially manufactured end-effectors resembling conventional endoscopic instruments. Each arm posses four DoF, although if equipped by newer MicroWrist end-effectors, the number of DoF can be raised to six. A surgeon operates the whole system from a console using two joysticks. The surgeon wears special 3D glasses, such that the surgeon perceives a 3D image. The controls are placed ergonomically to maximise dexterity of the operator. Due to this the surgeon does not have to be in the same room as the manipulator, and can operate by watching the transmission of AESOP camera. The controls have a safety counter-measure in a form of a clutch pedal at the surgeon's foot, that is required to be pushed down in order to transmit signals to the robotic arms. When the clutch is not pushed, the surgeon can position the two joysticks without initiating movement of the robotic arms. (Masatoshi Eto 2007)

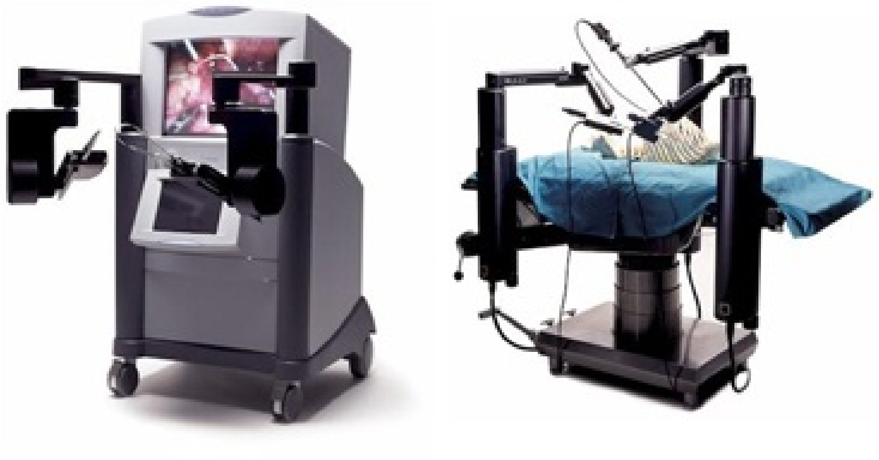


**Figure 2.3:** *Picture of Zeus system: three robotic arms mounted on the operating table controlled by screen console. (prWeb 2010)*

## 2.3 Lack of Force Control

As the two previous sections have indicated, both the da Vinci and the Zeus are advanced systems, fully capable of performing surgeries. Though they both have the common weakness of missing force control. In a traditional surgery a surgeon has tactile control when grasping, e.g. the skin of a patient. The surgeon thus ensures that the force put in the grasping is suitable and thereby minimises possible complications related to grasping.

When, e.g. looking at a surgery performed with the da Vinci system, this tactile force feedback is not present in the controls. Using the da Vinci system the surgeon does not have any kind of force control besides visual feedback. A study concluded that 38% of grasping tasks in laparoscopic surgeries were not successful, due to the lack of force control. The same study showed that this leads to an increased risk of complications and an increased operating time. (Paul J. Johnson 2014)

## 2.4 Summary

The purpose of this chapter was to investigate and to show what kind of solutions are on the market and how they are controlled. As both the da Vinci and the Zeus systems are very similar and thus suffer from similar limitations. The one most significant being the missing tactile feedback. In both systems the surgical manipulators are controlled by joysticks without force feedback and relies heavily on visual feedback.

# Chapter 3

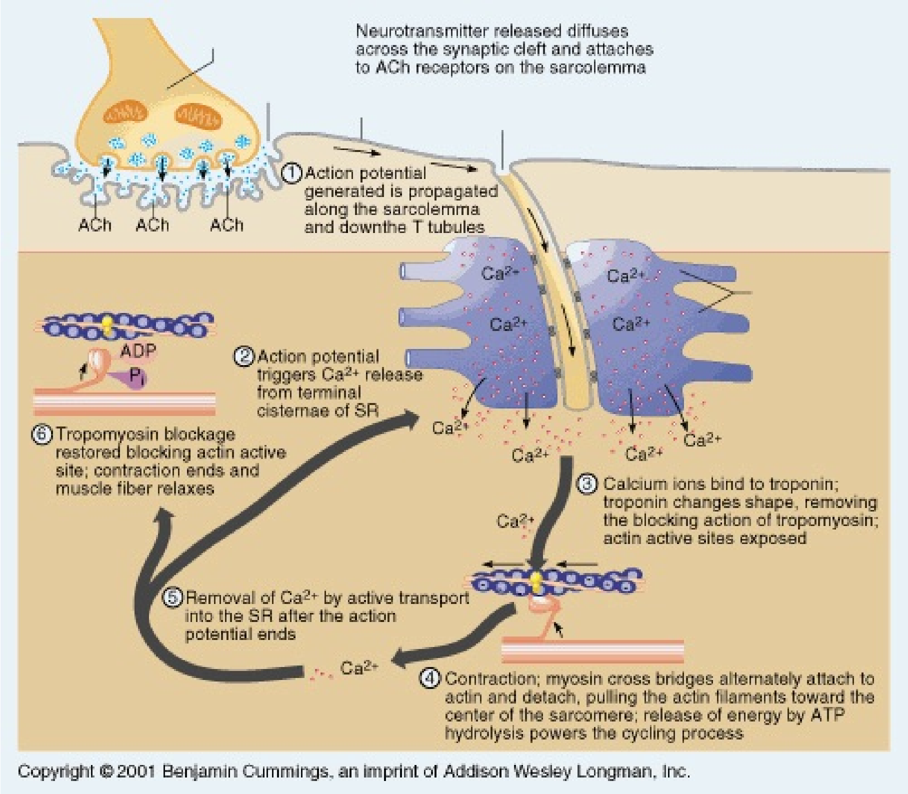
# Technical Challenges

When developing and designing a system for remote surgery, the research in chapter 2 indicates that there do not exist any alternative methods to control a surgical robot, other than with a joystick.

One alternative method for controlling surgical robots could be with the use of EMG signals. This control method arises several technical challenges. This chapter describes the different relevant challenges and provides basic knowledge on the different areas related to pattern recognition with the use of EMG signals.

## 3.1 EMG

When the muscle fibre receives a signal to contract, it activates gates through the membrane of the muscle which allows sodium to diffuse through the membrane. The diffusion of the sodium creates an action potential which travels along the membrane of the muscle fibre. This action potential causes a depolarisation leading to electricity from the action potential going through the muscle fibre. The flow of electricity causes great quantities of calcium ions to be released into the sarcoplasm. The calcium ions create attraction between actin and myosin filaments causing contraction of the filaments and thereby contraction of the muscle fibre. Immediately after the contraction has been made, the calcium ions are removed. (Hall and Guyton 2011)
This whole process is illustrated in figure 3.1


Neurotransmitter released diffuses across the synaptic cleft and attaches to ACh receptors on the sarcolemma
ACh
ACh
ACh
① Action potential generated is propagated along the sarcolemma and downthe T tubules
$Ca^{2+}$
ADP
$P_i$
② Action potential triggers $Ca^{2+}$ release from terminal cisternae of SR
⑥ Tropomyosin blockage restored blocking actin active site; contraction ends and muscle fiber relaxes
③ Calcium ions bind to troponin; troponin changes shape, removing the blocking action of tropomyosin; actin active sites exposed
⑤ Removal of $Ca^{2+}$ by active transport into the SR after the action potential ends
④ Contraction; myosin cross bridges alternately attach to actin and detach, pulling the actin filaments toward the center of the sarcomere; release of energy by ATP hydrolysis powers the cycling process

Copyright © 2001 Benjamin Cummings, an imprint of Addison Wesley Longman, Inc.

**Figure 3.1:** *The process of the muscle activation. (Marieb 2001)*

The EMG signal is in fact a representation of the depolarisation of the muscle fibre's membrane. EMG signals are measured as a voltage drop of the depolarisation and are on a scale around micro- ($\mu V$) to millivolts ($mV$). Usually the EMG signal has an amplitude range of 0 to 10 mV (-5 to +5) and a frequency in the interval of 20 Hz to 500 Hz. When the force applied by the muscle increases it is regulated by activation of more muscle fibres and by increasing the rate by which the action potential is initiated. Both ways of regulation eventually lead to an increase in depolarisations of a certain area, which also eventually leads to a greater voltage drop and a greater amplitude of the EMG signal. EMG signals can be recorded either intramuscularly (with intramuscular electrodes) or on the surface of the skin (with surface electrodes). The more tissue there is between the membrane surface, where the depolarisation happens, and the electrode, the more "external" influences are present in the signal. Even the composition of the in-between lying tissue affects the signal recorded. (Merletti and Parker 2004)

In the system described and developed in this project, the purpose is for a human to connect to the system without it being too intricate, and it is not a responsible solution to use intramuscular electrodes in this case. Therefore the chosen method and the method implemented throughout the project is electrodes on the surface of the skin, i.e. surface EMG.

The signal by itself is fairly rough and can contain a lot of unwanted influences and noise. Section 3.2 will clarify what has to be considered when obtaining the signal and what has to be done to prepare the signal for further use, and also how the signal can be analysed and implemented afterwards.

## 3.2 EMG Acquisition and Processing

To be able to obtain an EMG signal and afterwards process it to prepare it for control purposes, different pre-acquisition and processing operations have to be considered. The pre-acquiring operations include adding different hardware to modify the signal in order to optimise for the desired processing that takes place afterwards. The pre-acquisition operations can be amplification and filtering, i.e. steps related to or preparing the signal for feature extraction. Another step is to consider the sampling rate. Once the signal has been acquired the processing can continue such that desired features can be extracted. When the features are found, a classification method can be applied. For this purpose different methods can be used. See appendix D for more information about feature extraction and classification.

### 3.2.1 Obtaining the Signal

In relation to obtaining the signal certain choices regarding the hardware used should be made. These choices are often highly influenced by the aim of the use of the EMG signals. As mentioned in section 3.1, the electrodes used in this project will be surface electrodes, but in general this is the point where to consider which kind of electrodes to use. Even when limited to surface electrodes one still has to consider things like the quality and the size of the electrodes used, the distance between the electrodes, the amount of channels necessary, and the placing of the electrodes. As an example the requirement of classifying various inputs influences how many EMG channels are necessary to distinguish between or classify the given inputs.

In general it should be mentioned that the technical choices made, should be mainly related to the final requirements to the application at hand. The purpose should be to detect as appropriate a signal as possible in relation to the application of the signal.

### 3.2.2 Pre-acquisition Operations

As mentioned, different pre-acquisition operations can be done after obtaining the signal but before sampling it, one might want to do some signal conditioning in order to acquire a proper signal with the proper amplitude and the relevant signal inputs as isolated from other disturbances in the signal as possible before sampling. In relation to the pre-acquiring, a variety of methods can be applied, some of which are described in this section. The pre-acquisition operation and a proper acquisition has the purpose of preparing the signal for later processing and analysis.

The raw EMG signal has a rather small amplitude. To be able to analyse the signal properly and apply it for the desired purpose, amplifying of the signals can be necessary. When amplifying the signal, one has to be considerate of the limitations given by the analogue to digital (A/D) converter acquiring the signal, such that the signal is not amplified more than the maximum value of the A/D

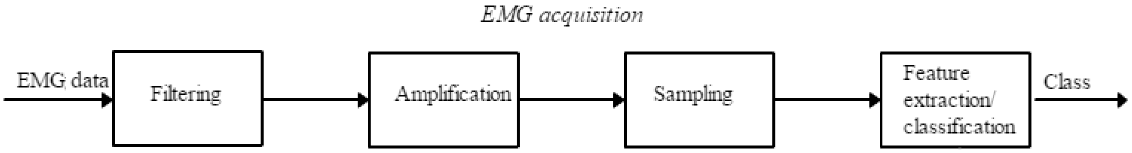


**Figure 3.2:** *Simplified EMG acquisition and processing system*

converter. However, the signal can not be too small in comparison to the least significant bit of the A/D converter otherwise important features of the signal might be lost in the conversion. See detailed information about A/D converting in appendix B.

As mentioned in section 3.1 there are a lot of factors influencing the signal. The raw measured EMG can contain noise and disturbances from different sources. Therefore, applying filters can be both beneficial and necessary. Filters can be applied analogue before sampling or digitally after sampling. It might be beneficial to filter out irrelevant frequencies before amplifying and sampling. With filters applied, frequencies with a high content of input caused by noise and disturbances can be eliminated.

Sometimes the obtained signal has a constant offset that is irrelevant to the further processing and application of the signal. In this case it might be relevant to remove the offset through offset correction before sampling.
For more details about analogue processing and filters, see appendix A.

### 3.2.3 Acquisition of the Signal

When acquiring, i.e. sampling the signal, it is important to consider the sampling rate and also the A/D converter applied. The sampling rate influences how much data is recorded.It is important that the sampling frequency is larger than the highest frequency contained in the signal. If the sampling rate is too low, important information in the signal might be lost. According to the Nyquist sampling theory the sampling rate should be at least two times the bandwidth of the sampled signal.

As mentioned in section 3.2.2, the A/D-conversion defines the limits for the amplitude of the EMG signal and therefore has to match the wanted maximal amplitude. Also the amount of bits and the resulting least significant bit are crucial for the amount of details in the digital replica of the analogue signal. (Merletti and Parker 2004)
See appendix B for further explanation of least significant bit.

### 3.2.4 Processing

After acquisition, further processing can be done. The operations done at this point are all digital. The operations done in the analogue domain can also be done digitally by using mathematical operations and programming. This includes filters, rectifying ect., see more details about the digital domain in appendix C. Furthermore an analysis of the signal can be done. This analysis can be related to control, how the control algorithm is designed and what inputs it takes. For this part, limits or thresholds can be implemented or more complex analyses can be done such as feature extraction and classification. Also some enveloping might be applied for different purposes e.g. smoothing the signal or analysing the signal over time.

In section 3.3 the different possibilities for ways of control through the use of EMG signals will be elaborated.

## 3.3 Myoelectric Control

This section describes the different ways EMG signals can be applied for control.

A simple approach for control using EMG that has been applied a lot is simply coupling the control of one degree of freedom (DOF) to one set of electrodes consisting of two channels placed on a set of muscles often respectively the agonist and the antagonist. In this case the contraction of the two different muscles is coupled to opposite functionalities e.g. flexion and extension of a joint. This approach opens for few possibilities of more complex control based on this approach. The first one being simply adding more sets of electrodes if control of more DoFs is necessary. Another possibility is to add a function to switching between controlling the single DoFs. This can be done in more ways for example by a switch-mode actively accessed by the subject when providing a certain trigger action or it can be done by segmenting the amplitude range of the absolute value of the signal with several thresholds. This is also called level-coding. Another method is rate-coding, where the rate of change in the signal is the feature controlling which functionality is activated. This is done by integrating them to find the area under the curve and compare the value to certain thresholds. This is very similar to level-coding. For all the cases it is possible to define whether the control signal is only an activation signal or is proportional to the EMG signal provided. However, in some cases this choice should be in consideration of what suits the control algorithm the best.

Regarding the focus of this project the surgeon has to have fine control over the robot. In that case it is not beneficial that the surgeon only can control one DoF at a time and has to switch through joints all the time to be able to carry out the wished actions. This will be a intricate and tedious process. On the other hand adding new sets of electrodes for every DoF will result in many electrodes being placed and there might be too little space to have enough room for sets of electrodes for all the DoFs. However, it is still important to be able to identify the exact movements of the surgeons hand in order to know what the robot should do. In this case another approach can be used. The approach of pattern recognition. Since different movements have distinct patterns with distinct composition of contributions from different muscles, recording the signal and comparing it to previous recorded signal pattern references, called training data. This gives the opportunity to get a more detailed picture of the muscle contraction using fewer electrodes. The method of using training data to create a pattern reference is called supervised machine learning. This also enables activation of functionalities by generation of certain patterns. As one can imagine the mapping of the patterns becomes more thorough, which decreases the risk of errors when more channels are used. The benefit of using pattern recognition is also that a greater quantity of functionalities can be implemented with a smaller amount of EMG signals i.e. a smaller amount of electrodes. (Merletti and Parker 2004)

When the use of pattern recognition is applied it can be implemented in several ways. Different functional patterns can be recognised and implemented as a unit. This could for example be different predefined gripping configuration that are combined with certain patterns recognised in the EMG signal. However, it can also be that different patterns across several channels are connected to certain movements such as flexion and extension of a certain joint. (Li and Kuiken 2009)

Since pattern recognition is a central focus point of this project an explanation of how pattern recognition is carried out is given in section 3.4.

## 3.4 Pattern Recognition

This section describes the process of making successful pattern recognition. To be able to conduct pattern recognition of the EMG signal consisting of signals from several channels, certain steps are necessary. First of all, the acquisition of signals is needed. The next two steps which are specific to pattern recognition are feature extraction and classification. In the following sections 3.4.1-3.4.3 the three steps will be elaborated.

### 3.4.1 Pre-processing

As pattern recognition requires multiple channels, there are some technical considerations to do in regard to the amount of channels and the placing the electrodes. The important part is that the channels provide enough signals with enough varying in the signals to differentiate between and classify the movements successfully. The goal is to classify with a high success rate but at the same time the amount of electrodes are optimally kept at a minimum to decrease the complexity and the cost.

Conditioning and simple preparatory processing operations should be done to prepare the signals for the actual analysis. This is all done according to the principles described in section 3.2.

### 3.4.2 Feature Extraction

To be able to distinguish between different motions, one needs different parameters and features that characterise the signals, and thus associates the motion with features.

For EMG signals, features can be both in the time domain or in the frequency domain. Features in the time domain can be e.g. the mean absolute value, slope sign change, zero crossings, etc. Features of the frequency domain can be median frequency and mean frequency. See appendix E for detailed information about different features the EMG signal possesses. (Li 2011)

In order to do feature extraction it is needed to observe a fragment of the EMG signal in the time domain, therefore, the signal is segmented or windowed. This means that each entry in the set of analysed data is coupled to the window. There will be specified time intervals in between the windows which also affects the level of detail in the recordings represented by the features. (Li 2011)

### 3.4.3 Classification

Once the features are extracted it is possible to classify the recorded signal. This can be done using one of possible methods. No matter the method chosen, the system has to be trained by being given at least one data set that defines the different classes, i.e. motions. It is by comparing new signal features to the trained data set in one way or another that the signal can be classified. One method is using linear discriminant analysis and another method is using the K-nearest neighbour approach. K-nearest neighbour approach compares the Euclidean distances between all classes and takes a selected amount of the nearest points. Afterwards it takes a majority-vote of the nearest points. If e.g. the five nearest classes are of two different types, the system will classify the input as the class which had three nearest neighbours.

Another approach is using the Bayes classifier. This approach is based on probability which means that the system can take into account if one class is occurring statistically more often than another. The model compares the likelihood of an input belonging to certain classes, and selects the class which is most likely. See appendix D for further explanation.

## 3.5 Delays

When EMG is used for control purposes, where an output or actuators are activated by the analysed signals, the delay of the system response is an important aspect to consider. Naturally different operations and analysing processes in the line of control will cause a delay. In particular when the current output value of a certain operation depends on several previous inputs, which is the case in the different methods where segmentation is included. Dependent on the application of the system it leads to certain requirements for the system and also to limits of how big delays are acceptable. Keeping that in mind it is relevant to investigate what limits are acceptable for the application in this project.

The area of using EMG to control a robot for surgery has certain similarities to the field of myoelectric prosthetics. In the field of myoelectric prosthetics there are certain relations with pros and cons to consider. A bigger delay leaves more time for analysis and data gathering which also increases the accuracy of classification, however, a big delay also causes a decrease in the responsiveness of the system. Naturally there is a limit to how little time can be invested in the classification in order to get a reliable result, and also to how big a delay is acceptable before the responsiveness of the system is too slow. In between the two outer limits lies an interval with acceptable abilities of the system but where one can adjust between the two opposed benefits. Research in the field of myoelectric prosthetics has shown that the optimal delay lies in interval $100 - 175$ $ms$. Furthermore, literature through time has stated that the delay should not be bigger than $300 - 400$ $ms$ and that a delay smaller than $300$ $ms$ is not perceivable for the user. Yet, other source states that a delay of more than $200$ $ms$ is unacceptable. (R. Farrell and F. Weir 2007)

# Chapter 4

# Problem Formulation

As mentioned in chapter 1, the still developing systems for remote surgery provides possibilities within the field of surgeries. The new technology provides the possibility of performing surgeries, no matter what distance lies between patient and surgeon. This is relevant because of the urbanisation, leaving the rural areas with few chances of receiving specialised medical attention.

Chapter 2 showed that assistive surgical systems are already in use for remote surgeries. Furthermore they enable high precision during minimal incision surgeries. All of the investigated surgical systems are controlled by a joystick interface. This project investigates an alternative method for controlling surgical robots. As described in chapter 3, EMG can be processed and used for controlling purposes.

EMG is generally used in the medical world to locate muscle damage, in research projects concerning prosthetics and robotic solutions for paralysis. However, there is no solution utilising EMG signals in order to control surgery robots, even though making controls based on EMG signals is possible. Assuming that controlling robotic arms with the use of EMG signal would be more intuitive than using a joystick, the possibility of EMG control is investigated. Signal processing and classification are essential aspects in making EMG based control. These aspects are the central focus points of this project.

The analysis above lead to this question:

*How can EMG signals be processed using pattern recognition to output control signal for a manipulator?*

## 4.1 Solution Strategy

The goal of this project is to deliver a supervised machine learning algorithm which uses up to eight EMG channels perceived from the subjects forearm, and uses the signal to recognize hand poses. This procedure can then be used to control a provided BeBionic robotic hand and make it replicate the subject's poses. The EMG signals are to be captured by a set of electrodes that should be placed on certain surface muscles of the arm. The signals are to be streamed to a computer on which the EMG processing is carried out. In order to do that, the knowledge and techniques of standard signal processing, like pattern recognition, can be applied. The idea is to interpret the EMG signals in a way that would make an applicable control input for a robotic hand.

# Chapter 5

# Requirement Specifications

In this chapter the requirement specifications are stated. The project will be delimited so that the focus is solely on creating a subsystem capable of making pattern recognition on EMG signals. Different requirements will be elaborated. This is done in order to create an overview of the system created in this project and listing the requirements of the system. The requirement specifications are furthermore used while developing the solution proposed in section 4.1, and the solution will be tested against these requirements.

## 5.1 Delimitation

The general remote surgery system using EMG signals as control input can be seen in figure 5.1. However the focus of this project is not to create a finished alternative system, but to investigate the subsystem where EMG signals are converted into robot commands, as illustrated with the red box.

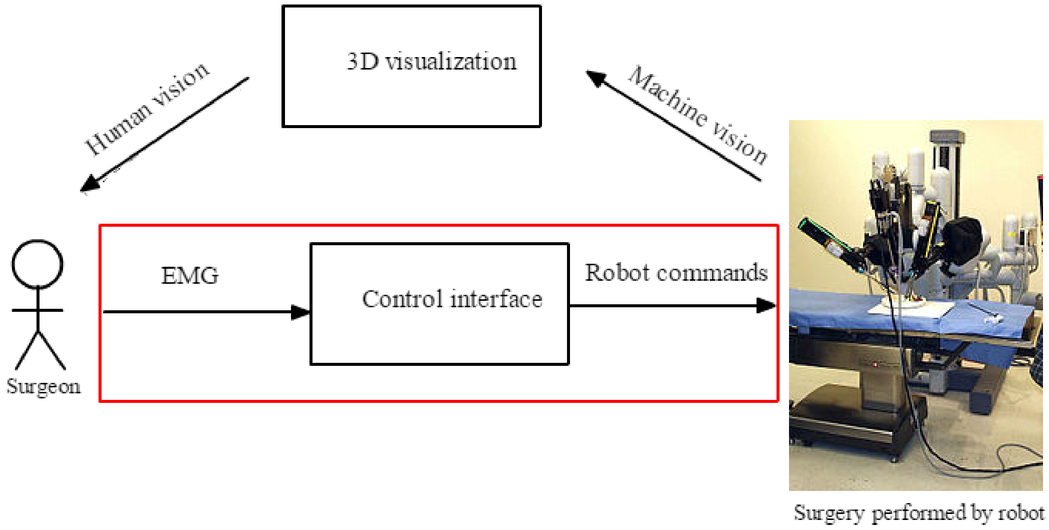


**Figure 5.1:** *The overall system, where the focus of this in project is marked with a red box.*

The goal of the system designed is to use a BeBionic robotic hand, which is further described in section 6.2, and make it move to predefined poses using EMG signals. This is done in order to have the BeBionic hand replicate the poses of the surgeon. The poses will be described in section 5.2.1. It is assumed that the product is not able to carry out a surgery, therefore the use of robot is unimportant and the BeBionic hand is sufficient for a prototype. The same applies for the subject as no medical knowledge is necessary. The point of the system is to verify that it is possible to copy movement from a subject's hand to a robot, using EMG signals, and not if it is possible to carry out a surgery using the system. This is illustrated in figure 5.2, the system is expanded and further explained in chapter 6.

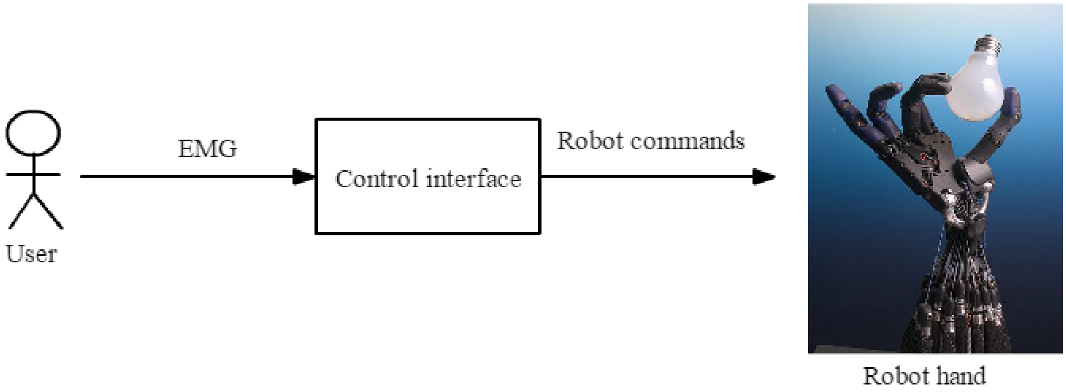


**Figure 5.2:** *The system of this project with outputs and inputs.*

## 5.2 Specification

In this section the requirements specifications for the system, illustrated in figure 5.2, are determined and elaborated. This is done in order to create a background for developing the system.

#### 5.2.0.1 Given Hardware and Software

In this project following given hardware and software was used during the work with EMG signals. The amplifier used is the Prima - Biomedical & Sport EMG 16. The amplifier has a fixed pass band between 10 $Hz$ and 500 $Hz$. The customised settings chosen for this device are mentioned in the introduction to chapter 7. The digital to analogue converter used is the National Instruments USB 6212. Between the EMG electrodes and the amplifier active adapters from OT Bioelettronica are used. The electrodes given and applied for recording are the Ambu WhiteSensor 0415M electrodes. Furthermore a stationary computer is used to record the digital signals. Due to safety reasons all hardware used with a power supply is isolated from the power grid through the use of an optocoupler. The used optocoupler is the Noratel IMED 2000 power supply. The software used was BioPatRec, which is an open source toolbox for MATLAB. The customised settings set in this software are mentioned in the introduction to chapter 7. The BeBionic robotic hand was provided for the purpose of this project. All parts of the digital signal processing algorithm will be made in the MATLAB environment using different libraries.

### 5.2.1 Performance Requirements

Performance requirements sets out to give the system the most stability. They are all related to the performance of the system and could e.g. be the number of classifications per minute.

#### 5.2.1.1 Delay

In order for the system to act intuitive for the user, it is important to investigate the delay from when the EMG signal is measured to when the signal is classified as a movement. To make the system intuitive for the user, this delay must be approximating the existing natural delay, from when the brain initiates a contraction to when the muscle contracts. As pointed out in section 3.5, a delay larger than 300 $ms$ is unacceptable. Therefore the requirement for the delay is set to be 300 $ms$.

#### 5.2.1.2 Poses

In order to verify that the system is able to receive EMG signals as input, and a specific robot command as output, six hand movements are chosen (plus a relaxed state), which the system is required to classify using a classification method. The BeBionic hand has several preprogrammed hand poses. The poses the system should be able to classify are chosen from these preprogrammed poses while maintaining the relevance for a surgeon. This way a direct link between the poses done by the surgeon, and the poses the BeBionic hand is capable of doing, can be drawn. The following list contains the movements that the system is required to classify and the background for choosing them:

1. **No motion (relaxed)**

   The relaxed state is important to recognise in order to differentiate between when a pose is being performed and when it is not.

2. **Flex wrist**

3. **Extend wrist**

   Both the flex and extend wrist, and the movements to get in those positions, are basic elements of any hand motion of any person. With these poses and movements the surgeon is able to move any of the poses done by the fingers in order to use them to palpate and/or grab the right area or make an incision.

4. **Open hand (extend fingers)**

   This pose is useful to push and hold tissue to the side.

5. **Close hand (flex fingers)**

   This pose is useful to grab bigger objects, and the motion to get to this pose can also be used for pushing or holding tissue to the side while working with the other hand.

6. **Pinch close**

   This pose is a grip where the index finger and the thumb work together. This is used to pick or palpate small things.

7. **Close tripod**

   This pose is a grip consisting of three fingers, the thumb, the index finger and the middle finger, all applying force towards one centre point between them. This pose could be used by surgeons for holding tools. Hence the importance for the surgeon's work.

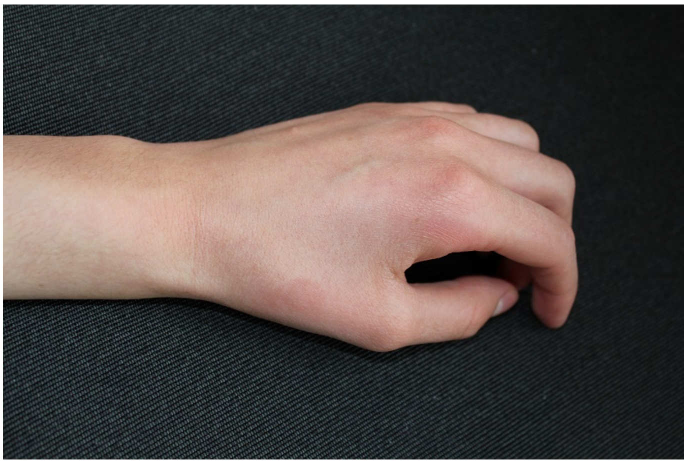
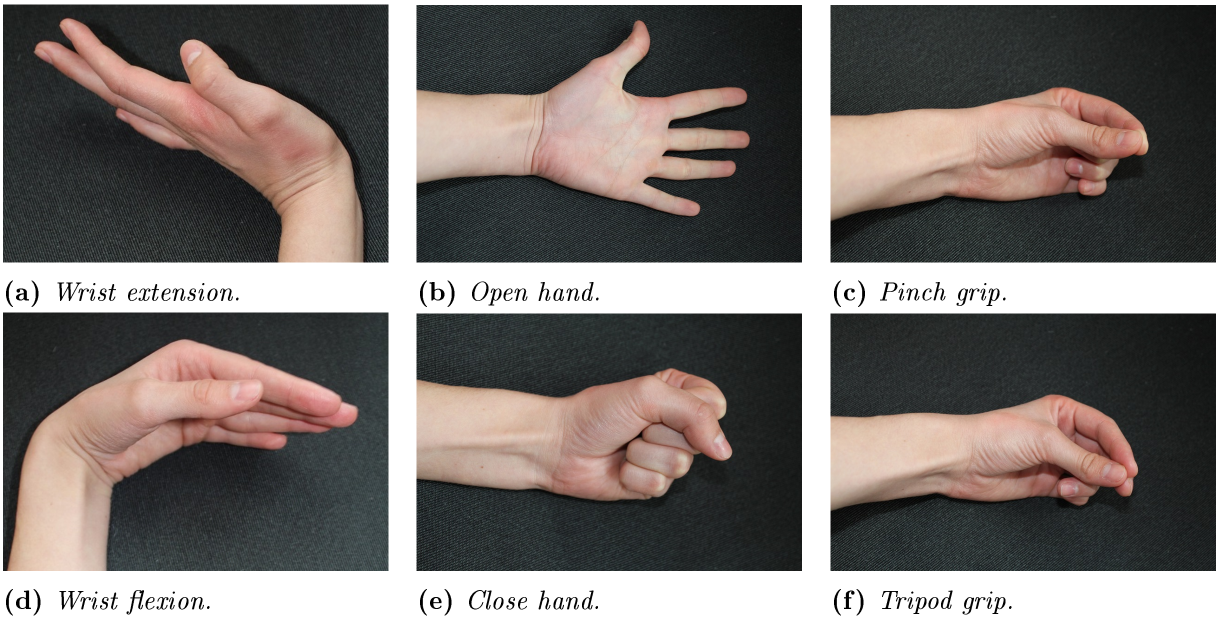


**Figure 5.3:** *Relaxed hand.*



**(a)** *Wrist extension.* **(b)** *Open hand.* **(c)** *Pinch grip.*

**(d)** *Wrist flexion.* **(e)** *Close hand.* **(f)** *Tripod grip.*

**Figure 5.4:** *The six selected hand poses.*

All the poses above have in common that they are carried out by contraction of muscles located in the lower forearm.

#### 5.2.1.3 Success Rate

The requirements for success rate of the supervised machine algorithm is that the system has to be able to classify with a prediction accuracy of min. of 95%, for all the movements. Meaning that the system should be able to record and classify off-line movements with min. 95% accuracy, using test data (20% of recorded data). Furthermore this success rate should remain after 2 hours and by testing according to Fitts's law the system should obtain a coefficient of determination ($R^2$) of atleast 0.9. The system should also be able to classify on-line, from the same person whom trained it off-line.

## 5.3 Final Requirement Specification

The following list contains a collection of all the requirements stated in this chapter:

1. The responsive delay of the system should not be bigger than 300 $ms$.
2. The prediction accuracy for the machine learning algorithm has to be min. 95% across all poses for individual subjects.
3. A subject should be able to use the system in at least two hours (the system should keep the min. 95% prediction accuracy).
4. The system should be able to classify at least six specified hand poses, and when the hand is at rest.
5. A classification model has to be created from training data, and training the system before using the system should not exceed a time limit of four minutes while still fulfilling requirement two.
6. The classification model should be able to predict and classify movements in on-line recordings made by the same person as the off-line recordings.
7. Features have to be extracted from the training data and used in classification of EMG signals in off-line and on-line signal processing.
8. An interface has to provide a feedback, that illustrates the movement that the system is classifying.
9. By testing according to Fitts's law the system should obtain a coefficient of determination ($R^2$) of atleast 0.9.

# Chapter 6

# System Description

This chapter gives an introductiond to the system seen in figure 6.1. The first step of the system is the analogue processing. Analogue processing is where the unwanted part of the signal is filtered out. It has a bandpass filter from 10 $Hz$ to 500 $Hz$. It also applies a gain of 2000 to the signal in order make feature extraction and classification easier. There are numerous ways to design a bandpass filter, and the one used in this project is the Bessel filter. For more examples and explanation of analogue processing methods, see appendix A and for more information on the analogue to digital conversion see appendix B.

The next step is the analogue to digital converter where the EMG signal is converted and imported to a computer.

The third step is signal classification where the feature extraction and classification is done. The signal classification is further described in section 7.1

The fourth step takes the output of the classifier and converts it to a control signal for the fifth step, which is the BeBionic hand.

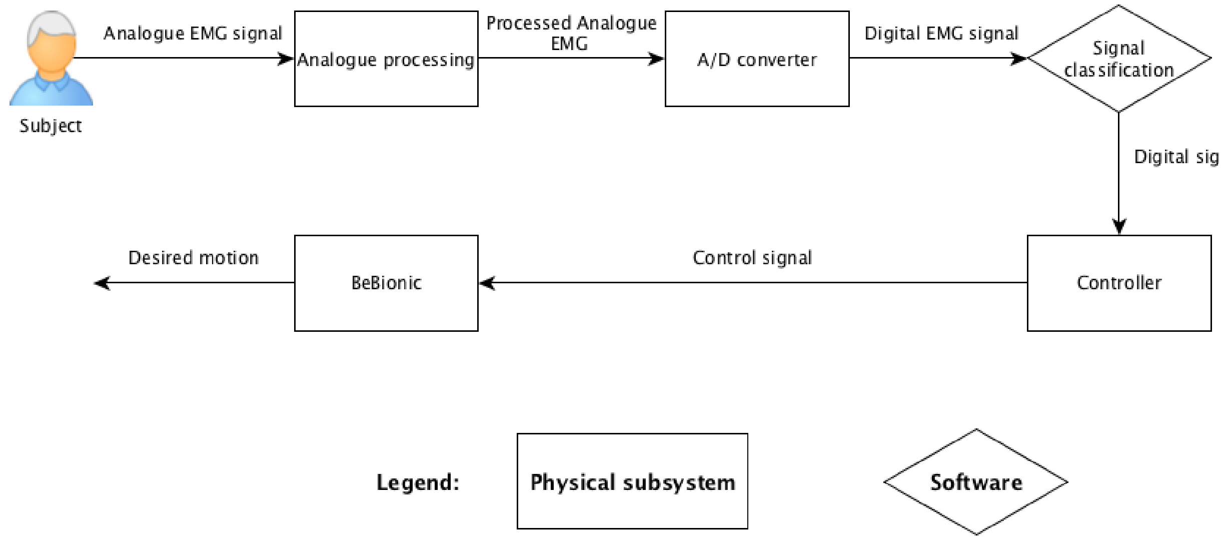


**Figure 6.1:** *Relations between all subsystems in the overall system.*

## 6.1 Placing of Electrodes

Before any operation or processing can be done on the analogue signal from the subject, the analogue signal has to be obtained through surface electrodes. In this section the placement of the electrodes is presented. Mainly the muscles, that enable hand movement and movement of the wrist joint will be mentioned.

In the hand, the wrist, and the forearm more than 30 muscles are working together to ensure the high degree of mobility. The composition of these muscles can be seen in figure 6.2. The muscles responsible for the movements in the hand and the wrist are mainly located in the lower arm.

When placing the electrodes it is important to consider which muscles should be targeted. This also depends on the amount of electrodes used. It is important to note, that readings with surface electrodes have a high chance of being influenced, by neighbour muscles. It can been seen on figure 6.2 that the proximity between muscles is small. Sometimes this is inappropriate but in some cases like this it might be beneficial because a bigger area of muscle can be covered with few electrodes. If classification across several channels is done, it will still be able to distinguish between different poses because the composition of all the signals will be different between the poses.

In this project four channels are used. They are placed according to the placement in the article (Xing et al. 2014), such that the four following areas are targeted as according to figure 6.2.

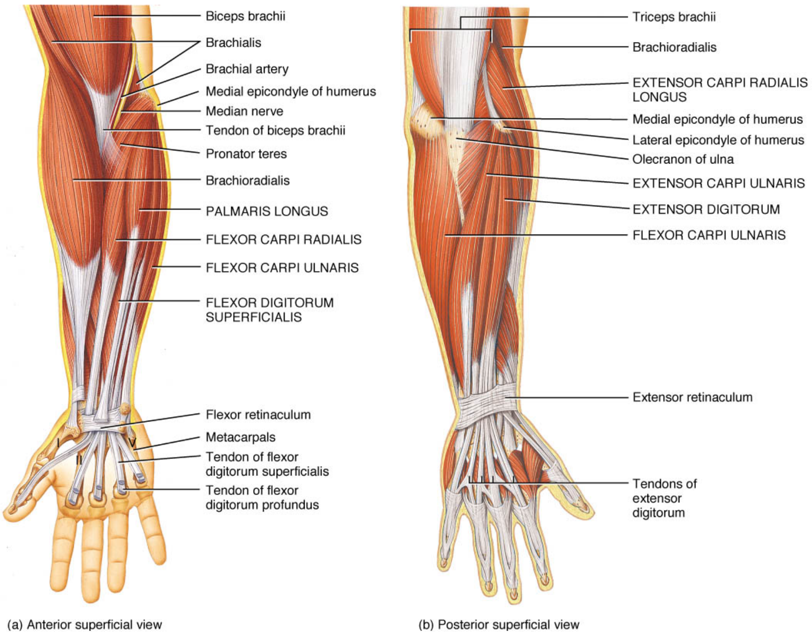
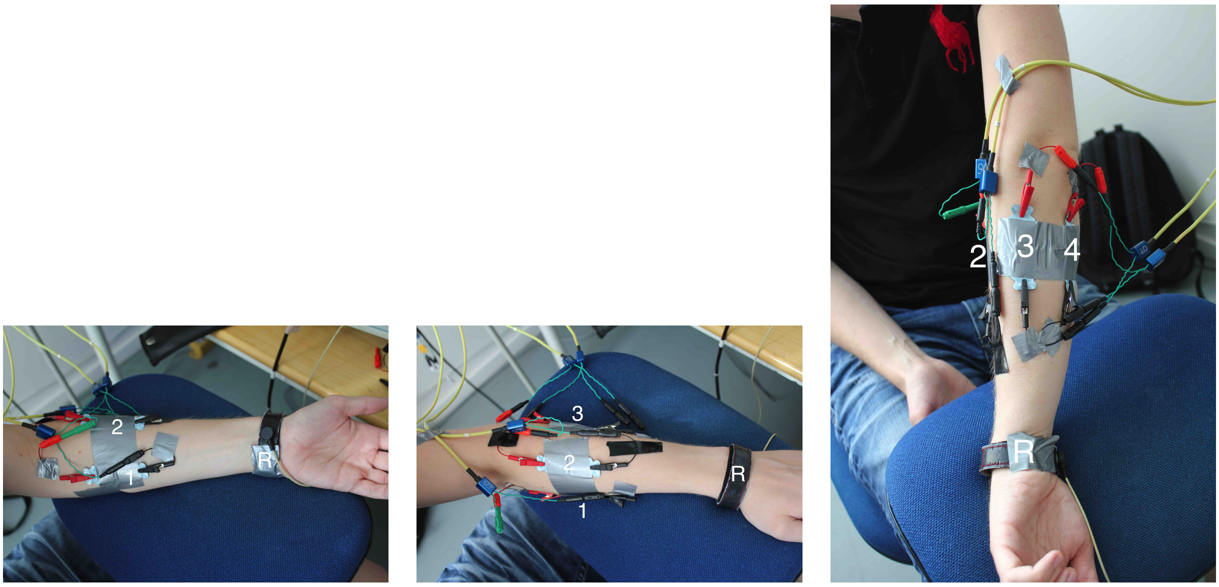


**Figure 6.2:** *The muscular anatomy of the lower arm. (Medicine 2016)*

- The first channel is placed on the area on top of Palmaris Longus, and Flexor Carpi Radialis (See figure 6.3a and figure 6.3b).
- Channel two is covering the area on top of Brachioradialis (See figure 6.3a, figure 6.3b, and figure 6.3b).
- The third channel covers the area on top of Extensor Digitorium, and Extensor Carpi Ulnaris (See figure 6.3b and figure 6.3b).
- Channel four is placed on the Flexor Carpi Ulnaris (See figure 6.3b).



**(a)** *Channel one and two.* **(b)** *Channel one, two, and three.* **(c)** *Channel two, three, and four.*

**Figure 6.3:** *Pictures of the placement of the different channels and the reference.*

With patch placements illustrated in figure 6.3, not only signals from the muscles, mentioned in the listing above, are measured but also signals from the underlying and neighbouring muscles are influencing the measured signal. Once the placing of the electrodes is decided and applied, signals can be obtained and different choices in regard to processing can be made.

## 6.2 BeBionic Robotic Hand

The BeBionic hand is a product of Steeper company, which focus's on prosthetic products and assistive technology. The first version of this product was presented in year 2010 and has been developed until today. Now, BeBionic hand is considered the most advanced prosthetic hand in the world.

This next-gen multi-articulating prosthetic hand is an accurate anatomic representation of human hand, which has five DoFs and are able to replicate many poses of the real human hand. Each finger then, except the thumb, is ran by its individual motor, which offers high variation of different gestures and grips the hand can form. The motors have also tracking devices allowing accurate feedback of the current position which improves repeatability of the various gestures. (Steeper 2016)

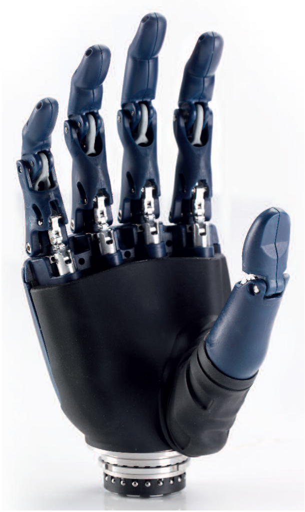


**Figure 6.4:** *Picture of BeBionic hand. (BeBionic 2016)*

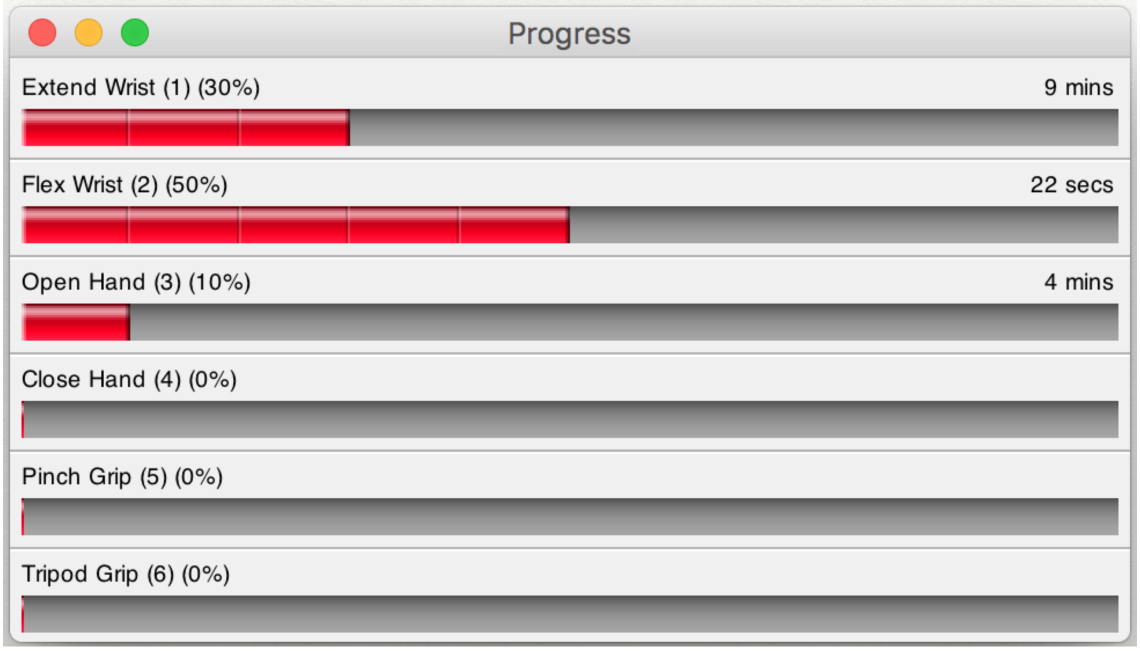
When the signal classification is trained for the on-line control, the MATLAB can start either outputting a signal to BeBionic's controller or a control signal directly to actuators of the robotic hand. The actuators can then output the desired movement.

Controlling the hand is only a secondary objective of this project, however this goal is still pursued, and even if the hand is not part of the final product, the program will be made in a compatible way, so the BeBionic hand can be part of the future work (e.g. gestures in requirements specifications 5 chosen from gestures compatible with BeBionic hand).

## 6.3 Visualisation of Classification

One of the goals of this project is to get a visual representation of the classification. Optimally this will be the BeBionic hand responding with the right classified hand poses in relation to what the subject is performing. However, during the work with implementing the on-line processing and classification it is necessary to have visual representation of the system response and the classification. For that purpose, a small graphical user interface (GUI), has been prepared.

The GUI is simple and consists of six loading bars each split into ten sections. Each section represents 10% of the full loading bar. The six loading bars each represent one of the six active hand poses described in section 5.2.1. "No Motion", however, is not represented in a bar because no motion in this case is equivalent with none of the other poses happening, which means that it can be indicated by the six bars for the other hand poses being empty (See figure 6.5).



Progress

Extend Wrist (1) (30%) 9 mins

Flex Wrist (2) (50%) 22 secs

Open Hand (3) (10%) 4 mins

Close Hand (4) (0%)

Pinch Grip (5) (0%)

Tripod Grip (6) (0%)

**Figure 6.5:** *The Graphical user interface for the purpose of portraying the classification.*

The GUI is made as a function in MATLAB and takes a class (a number between 0-6) as argument. The number-class correlation can be seen in table 6.1.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| No Motion | Extend Wrist | Flex Wrist | Open Hand | Close Hand | Pinch Grip | Tripod Grip |

**Table 6.1:** *The correlation between the number given as argument to the MATLAB function and the class it represents.*

The function works in a simple manner. It increases the bar of the class given as argument by ten, if it is not already a 100, and decreases the bars of the other classes with ten if they are not already zero. If the argument class is no motion, all six bars of the other six poses are decreased by ten if they are not already zero. The fact that the system is working in this manner also means that it takes at least ten calls of the function with the same class as argument before the bar of the class is full.

With this GUI the requirement mentioned in section 5.3 of having an interface to give feedback and illustrate the classified movements is fulfilled.

## 6.4 Digital Filtering Suggestions

The purpose of digital filtering is to reshape the spectrum of waveform to ones advantage. This can be achieved by reducing noise, making the point-of-interest features more obvious and increasing signal-to-noise ratio (SNR). There are many different filters that are able to carry out the spectral reshaping (like Butterworth, or Chebyshev). What makes them different, is their approach to achieve the reshaped state. All the filters are generally sorted into two major groups: Finite Impulse Response (FIR), and Infinite Impulse Response (IIR). See appendix C for more details about FIR and IIR filters, as well as about digital transfer functions. (Dekker 2004)

The way to apply the chosen filter is also different. Depending on what is necessary to bring up in the filtered signal, the filter can be either:

**Low-pass**
Filter out the signal under the specified cut-off frequency.

**High-pass**
Filter out the signal above the specified cut-off frequency.

**Bandstop**
Filter out the signal between two specified cut-off frequencies.

**Bandpass**
Filter out all of the signal except the part contained between two specified cut-off frequencies.

In this section the collected EMG signals are investigated. Based upon that, specific digital filtering measures are suggested. The purpose is to create preliminary filtering applications, which can be later applied in final solution (if it benefits the cause). The investigation is mostly concerned about detecting noise in the signal and possibly removing it, if the noise is considered problematic enough. However, different application is concerned later in this section.

Before investigating the signal itself, it is necessary to choose a filter type best suitable for this project. It has been decided that Butterworth filter between second and fourth order should be ideal for such application.

The IIR was chosen over FIR because this application benefits from its ability to come closer to the shape of optimal filter, than FIR filter (see figure 6.6). IIR filters also require less computing time than FIR.

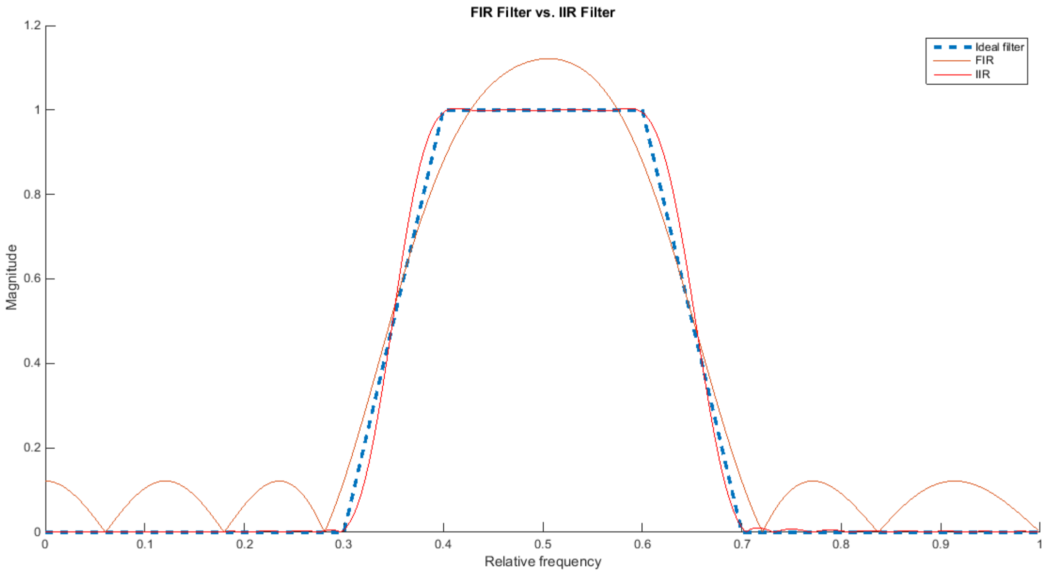


**Figure 6.6:** *Comparisson of IIR and FIR filter.*

The Butterworth filter was chosen for its flat passband, which does not distort the original signal that much (unlike Chebyshev). Such trait is considered an advantage in applications like noise removal (see appendix A for more information about Butterworth and Chebyshev filters). The order of the filter directly affects steepness of the filter's roll-off, i.e. the higher the order, the more the filter looks like the optimal filter. However, one needs to exercise caution when deciding the order, since too high order can make the filter unstable. Also, the higher the order, the more processing power the application requires. Lower order should however be safe from becoming unstable, that is why it has been decided to stick to second up to fourth order.

A noise that should be considered removed is the 50 Hz noise. However, in the EMG recordings (see one of them in figure 6.7) the 50 Hz noise is basically not visible. It is then not necessary to threat it in any way, especially because filtering out 50 Hz would erase the actual signal around 50 Hz. That in this case is not very good trade, because, as mentioned before, the noise is not interfering and is almost invisible. It is also important to consider that EMG signals are mostly concentrated in a band between 50 Hz and 200 Hz. This can be observed in a frequency plot in figure 6.7.

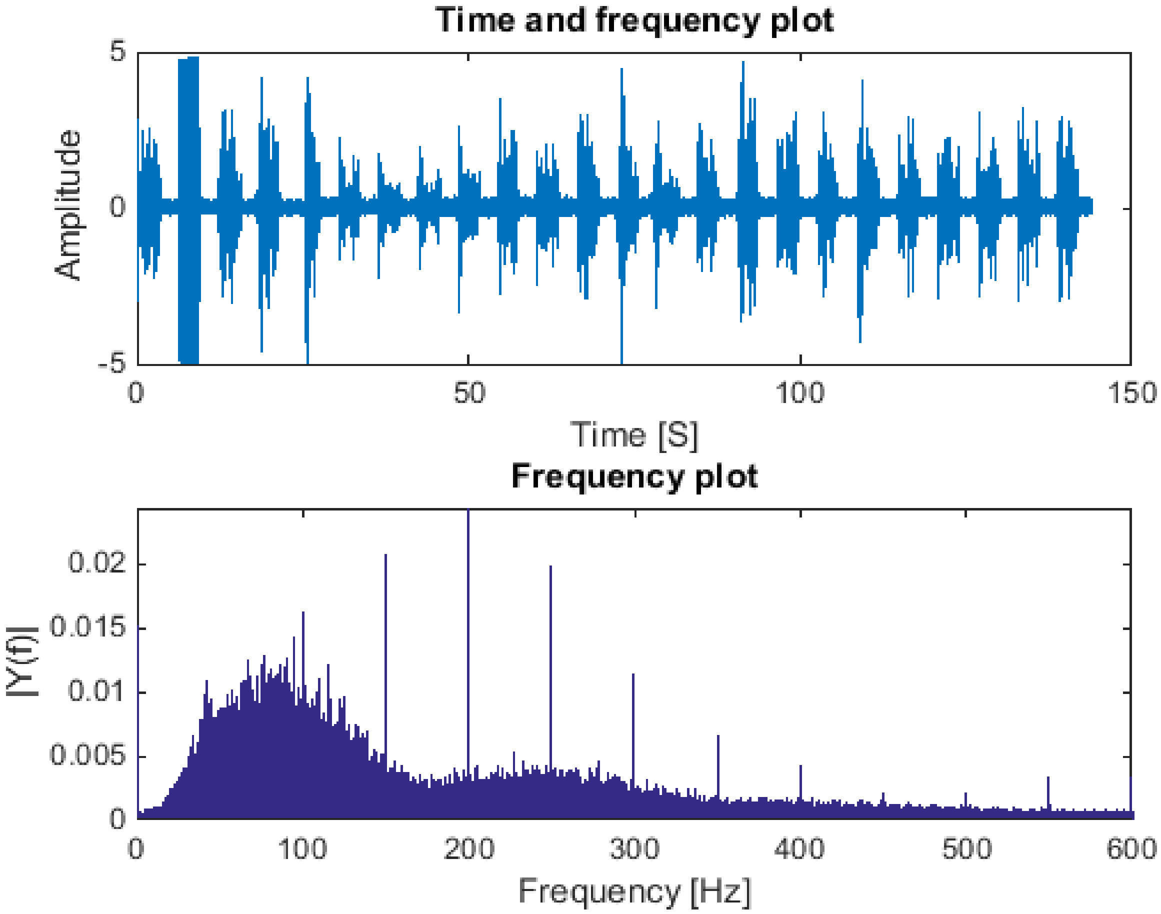


**Figure 6.7:** *Frequency plot from MATLAB of one gesture recordings.*

Looking at the figure 6.7, it can also be seen resonance at frequencies, which are iterations of 50. This phenomenon occurred in some of the recordings and the one in figure 6.7 is one of them. The solution here to get rid of the resonance would be to apply notch (special bandstop) filter to every iterations of 50 Hz (figure 6.8). However, the resonance does not distort the actual signal in any significant way, so applying the array of notch filters would be a waste of processing power and signal data.

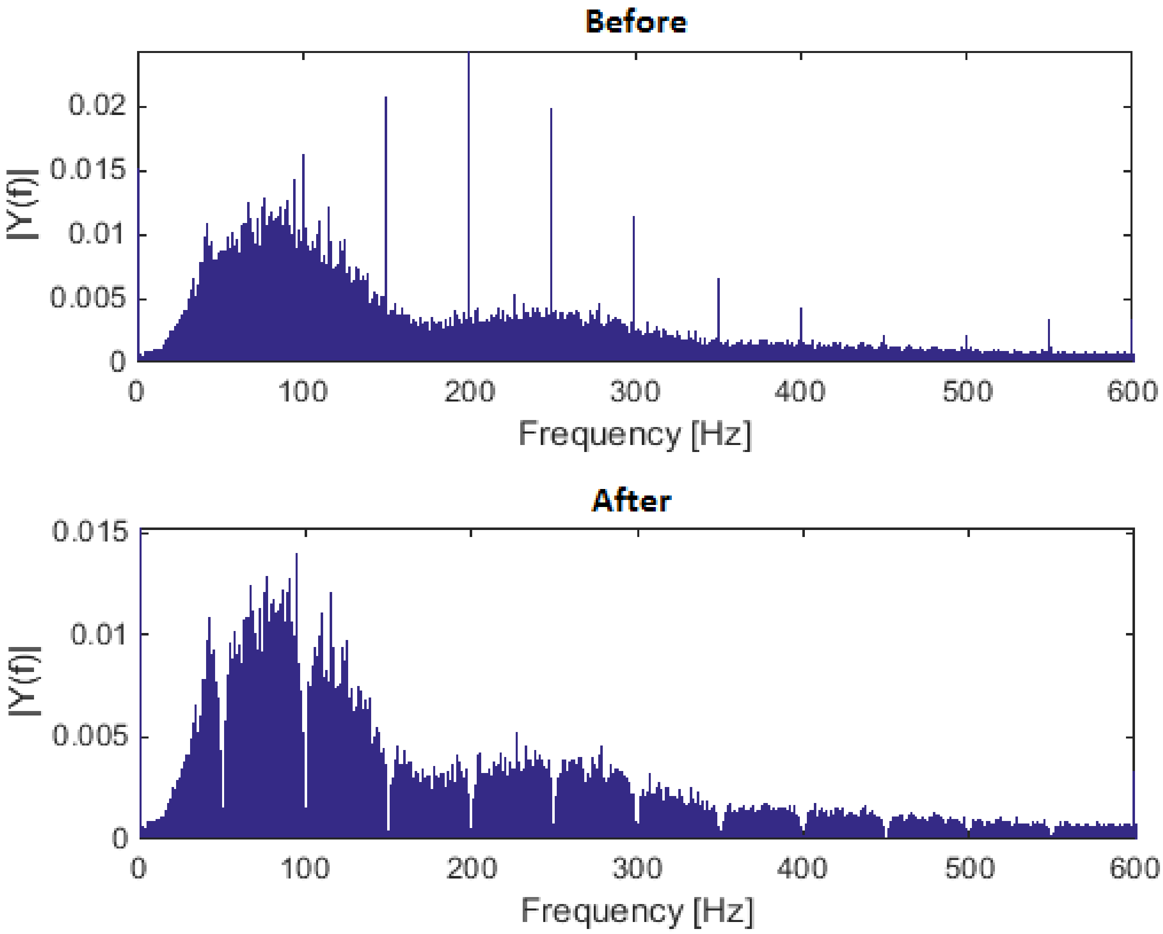


**Figure 6.8:** *Frequency plots before and after filtering out the resonance.*

Apart from filtering out the noise, the digital filtering might be used to one's advantage by re-shaping the signal.

As a practical example of that, see figure 6.9, which shows a part of recorded EMG signals in the lab. Top part shows the original signal from one channel, and the bottom part is the signal with applied low-pass second order Butterworth filter with cut-off frequency at 100 Hz. The filter cuts out big part of the signal, because the amplitude drops down to approximately half of its original value. However, the static in-between the two activities present in the original recording has been filtered out almost completely and the characteristic features of the signal have been preserved. So in this case, although the filter has cut out a part of the signal, it is possible that such modification made the signal easier usable in feature extraction (e.g. for applying thresholds).

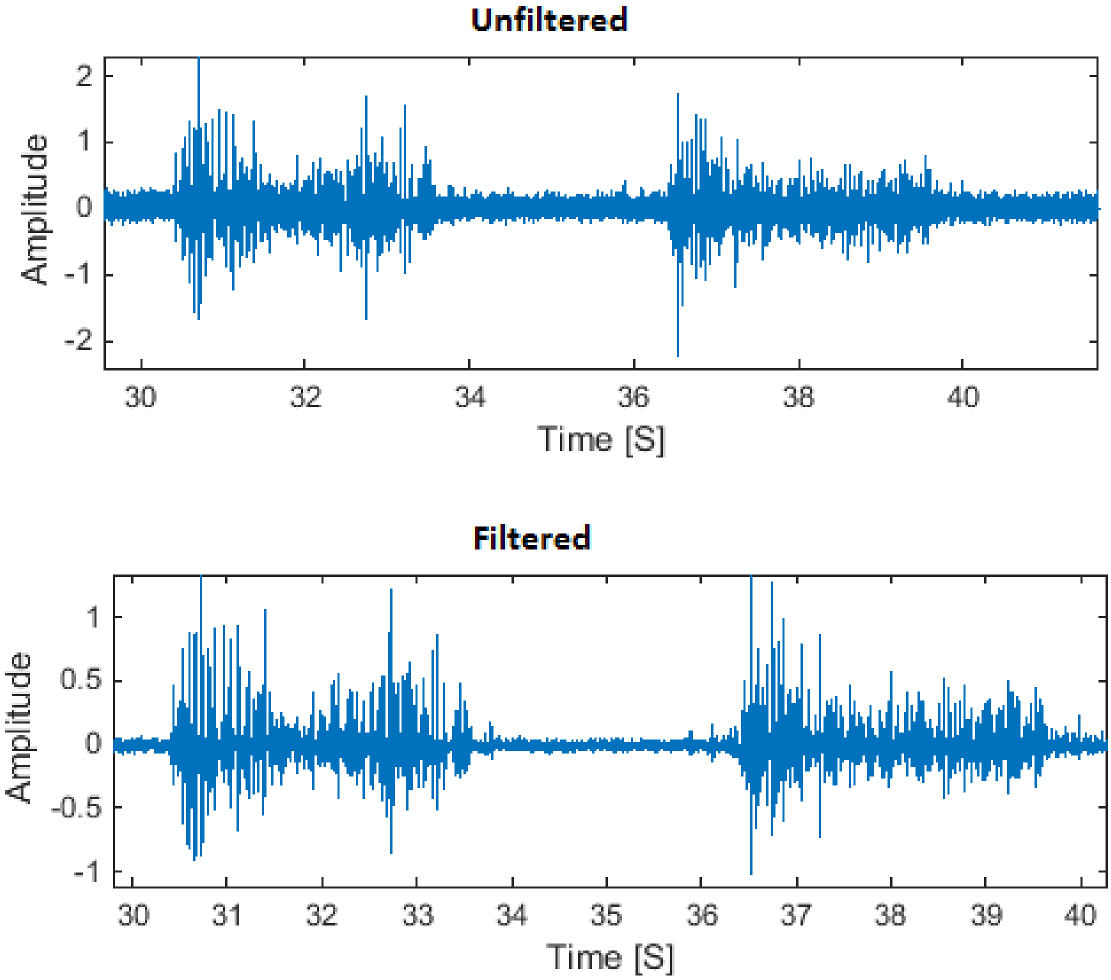


**Figure 6.9:** *A signal segment recorded in lab shown in both filtered and unfiltered form.*

This sections described certain filters which can be useful for application in this project. However, some of the filters were immediately neglected, since it was already obvious from signal investigation that such measures are not helpful at all in case of the investigated readings (50 Hz noise, resonance).

The claim of benefiting of having a low-pass 100 Hz filter was retracted after making a test, where a second order 100 Hz low-pass Butterworth filter was applied to the signal before doing feature extraction. It was discovered that this caused the error to rise. After changing the cut-off frequency to 200 Hz, the error dropped in order of tenths of one percent.

This result is logically expected as a EMG signal has a bandwidth between 10 to 500 Hz and is concentrated around 200 Hz. Thus the filter would cut out a large portion of the dominant frequencies.

# Chapter 7

# Methods

This chapter contains a methodical description of the exact steps undertaken to fulfil the requirements stated in chapter 5, and to address the problem stated in chapter 4. The focus is on the off-line supervised machine learning algorithm, and how the performance of the algorithm will be tested on the on-line recorded data. The results of the tests described in this chapter can be found in chapter 8.

All the EMG data recordings used in this project are obtained by the use of an open source MATLAB toolbox called BioPatRec. The specifications of the recordings, can be seen in table 7.1, and the placement of electrodes, follows the explanation and illustration in section 6.1.

| | |
|---|---|
| Number of movements | 7 |
| Sample frequency | 2000 Hz |
| Number of EMG channels | 4 |
| Gain | 2000 |
| Number of repetitions per movement | 5 |
| Time per sequence | 4 sec. |
| Rest after sequence | 2 sec. |
| Validation size | 20% of input data. |
| Training size | 60 % of input data. |
| Test size | 20% of input data. |

**Table 7.1:** *The specification for recording session.*

Furthermore the amplifier uses a differential setup meaning that all channels is the voltage difference between two electrode pads and the reference. The gain of 2000 is selected to amplify the signal while still avoiding the risk of going above the 5V, that is the maximum value the A/D converter can read. As mentioned in section 3.2.3 the sampling rate according to Nyquist has to be atleast two times the maximum bandwidth. In this case the bandwidth ranges from 10 to 500 Hz. Therefore a sampling rate of 2000 Hz is chosen.

Since the combined time it takes for all seven motions to be trained adds up to 3.5 minutes, the requirement of maximum four minutes training time is achieved.

## 7.1 Off-line

In this section, the method undertaken to create a training model for the supervised machine learning algorithm is explained. The chapter consists of a description of the specification chosen for the algorithm, and the methodology used to address the requirement for algorithm (a prediction accuracy on min. 95%, see chapter 5).

A supervised machine learning algorithm is a method where a training model is inferred, using a set of known supervised training data, which furthermore can be used to predict movements from a new set of unknown EMG data doing an on-line recording. The algorithm is illustrated by the red box in figure 7.1. The following description shows step by step what is required from the algorithm shown in figure 7.1:

1. **Feature extraction.**
   In this part several features are selected and extracted from the input data, and a feature matrix is obtained. In this step it is required to be able to characterise difference in the EMG signal for the different poses.

2. **Feature reduction.**
   In this part the dimensions of the feature matrix are reduced. The requirement for the feature reduction, is to simplify the feature matrix (obtained in part 1), and secure minimal redundancy of the features, while preserving variation of the features.

3. **Validation of training data.**
   In this part different sets of training data are tested using a classifier, and a finve fold cross test. From this step it is required for the algorithm, that it has to be able to evaluate different sets of training data, and to give the most satisfying training data as output.

4. **Identifying the training model.**
   In this part the training model is identified, and tested on a set of test data. The requirement for the training model is to have a minimum prediction accuracy on 95%, when classifying a set of test data.

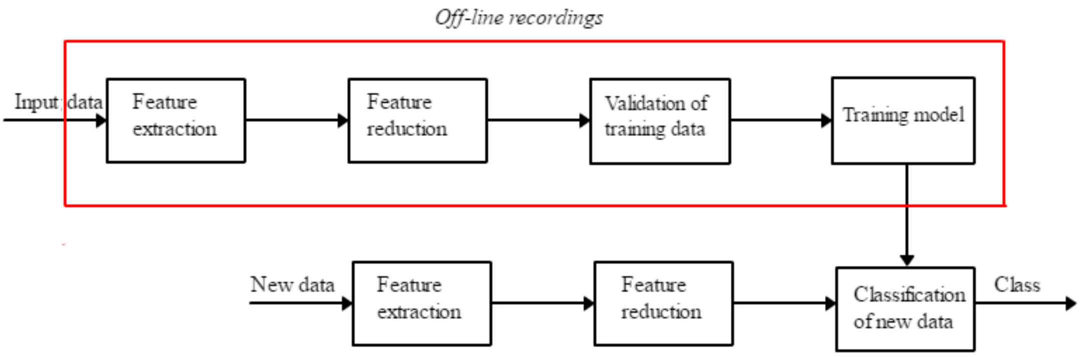


**Figure 7.1:** *A general illustration of the supervised machine learning algorithm.*

### 7.1.1 Feature Extraction

This section consists of a specification of the segmentation of the EMG data, the methodology of the feature selection and a description of the feature matrix.

#### 7.1.1.1 Segmentation of Data

Before extracting features from a raw EMG signals, a method is chosen to carry out segmentation of the data. The applied method is an overlapping windowing technique. A general illustration of the technique can be seen in figure 7.2. The overlapping windowing technique is chosen while it generates a contentious stream of data, because the ideal data classification time is equal to the step size (50 ms). This makes the stream feel more intuitive for the user, rather than having a buffer time equal to the window size (250 ms). By using this method, it is possible to divide the EMG signals into windows, from where features can be extracted window by window. (Li 2011)

The chosen specifications for the method are:

- $Windowsize = 250\ ms$
- $Stepsize = 50\ ms$
- $Samplingfrequency = 2000\ Hz$

This corresponds to a window size = 500 and a step size = 100 measured in EMG data points (for the calculation see equation 7.1). For each segmented window a 500 × 4 matrix is obtained (500 = amount of EMG points per window, and four numbers of channels).

$$500\ points = \frac{2000\ Hz \cdot 250\ ms}{1000} \tag{7.1}$$

Note, the same overlapping windowing technique (with the same specification), is used doing on-line recording.

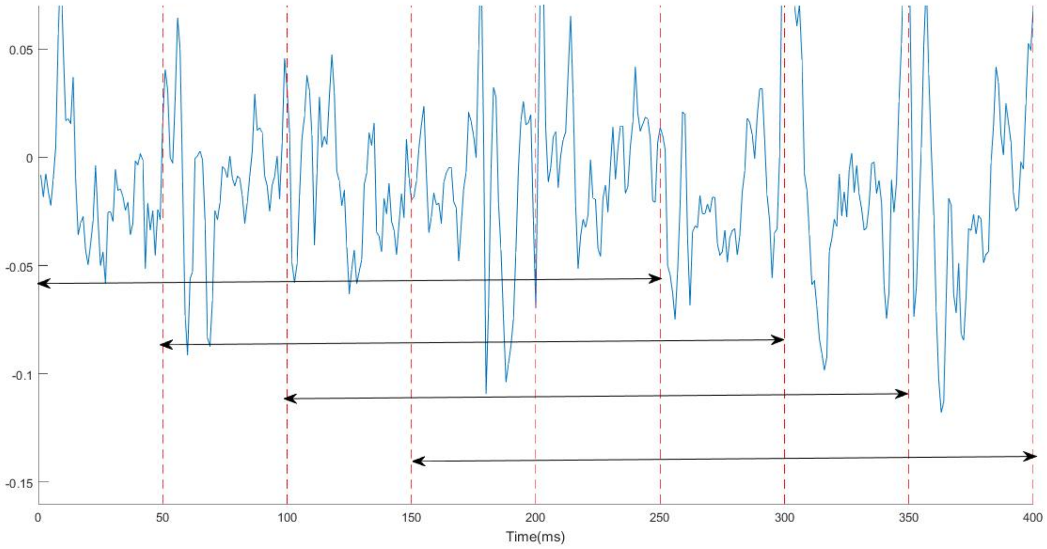


**Figure 7.2:** *A general illustration of an overlapping windowing technique (window size = 250 and step size = 50).*

#### 7.1.1.2 Feature Selection

The features selected, for the machine learning algorithm, are all features in the time domain based on statistical calculations. The argument for the feature selection is that, time domain features are simple, easy to access, and do not require much calculation time. The time available for calculations is only 50 ms. This leaves out frequency domain features, as converting a signal to the frequency domain might cause heavy load calculations.

For the feature extraction 12 features are chosen, and from these, six features are identified, and used in the machine learning algorithm. The amount of features are delimited to six, to reduce the complexity of the system. To find the combination of features with the best performance, it would be optimal to test all the combinations of features. However, since this would be too time consuming, a search for a satisfying feature set (a set which generates a prediction accuracy of min. 95%) is made, using an iterative process. The process is explained in the following list:

- A combination of six features is chosen.
- A set of EMG data is obtained following the specification in table 7.1, from where a set of test and training data is identified, by a five fold cross test.
- The test set is classified using a LDA classifier.
- The test is repeated, with same EMG data set, but a new combination of features. This is done until a satisfying prediction accuracy is obtained.

The 12 following features were used for the search (for a mathematical description of the features see appendix E):

1. Wilson amplitude (WAMPL).
2. Slope sign change modified (SSC2).
3. Myopulse percentage rate (MPR).
4. Root mean square (RMS).
5. Standard derivation (STD).
6. Log detector (LOG).
7. Absolute mean (AM).
8. Power (PWR).
9. Variance (VAR).
10. Zero crossing (ZC).
11. Wave form length (WFL).
12. Median (MD).

Of the 12 features three of them have a defined threshold. The three features with a threshold are WAMPL, MPR, and ZC. In general, the applied thresholds were found through a trial and error process. The process was executed with a starting point in 50 $\mu V$ in correlation with the generally suitable threshold interval between 50 $\mu V$ and 100 $mV$ suggested in the article (Angkoon Phinyomark 2012). Several iterations were made until the optimal thresholds were found. The identified thresholds are for WAMPL 40 $mV$, for MPR 30 $mV$, and the threshold for ZC is set to 0 $mV$.

The chosen set of features and discussion of the choice logic is described in section 7.1.1.

#### 7.1.1.3 Verification of Selected Features

In order to verify that the performance of the 6 selected features satisfies the requirement of a prediction accuracy of min. 95%, a test is undertaken. In the test a LDA classifier and 14 sets of different training and test data are used. Before the test recordings are obtained following the specification in table 7.1, and the recorded data are tested using the five fold cross-test (see section 7.1.3). For each of the 14 data sets a prediction accuracy is obtained by classifying a given test data set using the corresponding training data set. If the mean of the 14 prediction accuracies is min. 95%, the selected features are verified and accepted as a satisfying set of features. The result of this test can be found in section 8.1.1.1.

#### 7.1.1.4 Feature Matrix

When the features are selected, a feature matrix can be obtained. The feature matrix contains all the extracted features for each EMG channel. This section investigates the feature matrix, and how it is obtained from the segmented EMG training, validation, and test data. In the example illustrated in figure 7.2 it can be seen how the feature matrix for the LOG feature is obtained; each $j^{th}$ and $i^{th}$ entry of the LOG feature matrix, is extracted from the data in a $j^{th}$ column of the $i^{th}$ segmentation window matrix. This applies for all the features and it means that there is a $m \times 4$ feature matrix related to each feature (m = number of segmentation window).

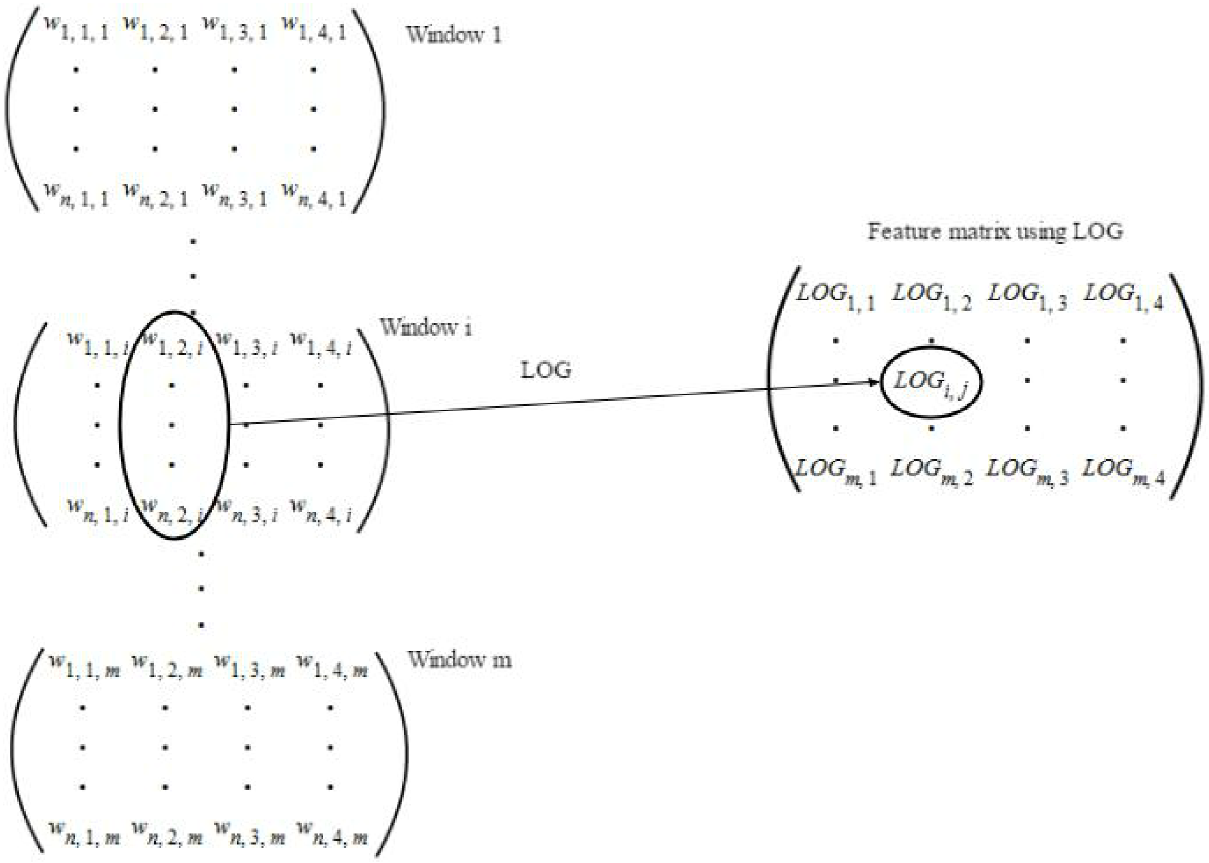

Window 1
Window i
Window m
LOG
Feature matrix using LOG
$w_{1,1,1}$ $w_{1,2,1}$ $w_{1,3,1}$ $w_{1,4,1}$
$w_{n,1,1}$ $w_{n,2,1}$ $w_{n,3,1}$ $w_{n,4,1}$
$w_{1,1,i}$ $w_{1,2,i}$ $w_{1,3,i}$ $w_{1,4,i}$
$w_{n,1,i}$ $w_{n,2,i}$ $w_{n,3,i}$ $w_{n,4,i}$
$w_{1,1,m}$ $w_{1,2,m}$ $w_{1,3,m}$ $w_{1,4,m}$
$w_{n,1,m}$ $w_{n,2,m}$ $w_{n,3,m}$ $w_{n,4,m}$
$LOG_{1,1}$ $LOG_{1,2}$ $LOG_{1,3}$ $LOG_{1,4}$
$LOG_{i,j}$
$LOG_{m,1}$ $LOG_{m,2}$ $LOG_{m,3}$ $LOG_{m,4}$

**Figure 7.3:** *Illustrated how features are extracted from segmented data windows, using a specific feature function (LOG, see appendix E), and how parts of the feature matrix are obtained from the given feature*

Once all the $m \times 4$ feature matrices are retrieved for each function of the features, a general $m \times 24$ feature matrix can be obtained as in matrix 7.2.

$$FM = \begin{pmatrix} WAMPL_{1,1x4} & SSC2_{1,1x4} & MPR_{1,1x4} & RMS_{1,1x4} & STD_{1,1x4} & LOG_{1,1x4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ WAMPL_{m,1x4} & SSC2_{m,1x4} & MPR_{m,1x4} & RMS_{m,1x4} & STD_{m,1x4} & LOG_{m,1x4} \end{pmatrix} \tag{7.2}$$

Note that when obtaining features from on-line recorded data, the same procedure is applied. However only one window matrix is used (m=1).

#### 7.1.1.5 Feature Reduction

Principal component analysis (PCA), is a feature reduction algorithm which by analysing the variance of an original feature matrix, makes it possible to transform the dimensions of the feature space. The following list shows the PCA algorithm:

1. Eigenvalues and eigenvectors are identified from a covariance matrix of the original feature matrix.

2. The eigenvalues are sorted from the highest value to the lowest.
3. A threshold (e.g. 0.95) is set and the $k$ largest significant eigenvalues, constituting the threshold (in %), of the sum of eigenvalues, are found. This threshold is refereed to as ETH.
4. The eigenvectors of the corresponding $k$ largest eigenvalues are gathered, in a matrix and projected to the training data and thereby the dimensions can be reduced. These new eigenvectors represent the new dimensions and are called principal components.
5. The gathered matrix (from step 4) with the eigenvectors are kept, and projected on test/new data.

The PCA algorithm is worth considering using in the machine learning algorithm to transform and reduce the dimensions of the feature space defined by the matrix obtained in subsection 7.1.1.4. The motivation for doing so, is that a lower dimensional space provides a simpler input to the classifier. By only choosing the significant features, PCA might secures that redundant features are left out, and information about the variance of the classes kept.

In order to verify, that PCA is a satisfying feature reduction method (usable for the machine learning algorithm), a test is undertaken. The test investigates the relation between, the size of ETH, and the prediction accuracy. The test is the following:

1. For the test a LDA classifier and 14 sets of training, are used (the same 14 recordings as in section 7.1.1.3).
2. For each of the 14 data sets, four prediction accuracies are obtained, each representing a ETH(the values for ETH are 0.95, 0.97, 0.98 and 0.99).
3. Four box-plots are made representing the prediction accuracies of the Four different ETHs.
4. If the prediction accuracy, corresponding to the lower quartile, of all the four box-plots are min. 95%, the test is accepted as a satisfying result and the PCA is verified as an satisfying feature reduction method.

The test results can be found in section 8.1.1.2.

### 7.1.2 Two Hour Test and Classification Selection

A system used for surgery, like the one in this project, needs to be stable. The system should not only perform with a certain accuracy shortly after the training of the system. It should also be able to maintain the accuracy over time. This is important because the system has to be continuously reliable throughout a surgery even if it takes several hours.

To test the long term reliability of the system, a test will be conducted where the system is tested using training data to classify test data after zero hours, one hour and two hours. If the success rate is above 95% in all of the three cases the system fulfils the requirement set for the system.

Furthermore, to make the error lowest in all three cases, three different classifiers (Linear Discriminant, K Nearest Neighbour and Naive Bayes, see appendix D) will be tested. The classifier with the lowest error in all three cases will be selected and used for this project.

The error can be calculated by having known motions and then classify the test data and see how it differs. Figure 7.4 visualises the error and shows where the classifier may have trouble.

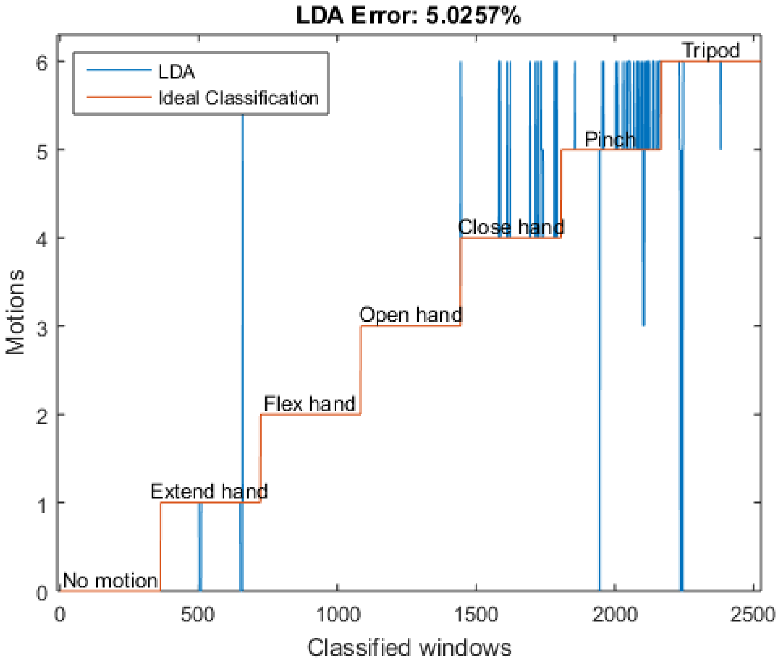


**Figure 7.4:** *Visualisation of error using Linear Discriminant Analysis to classify test data with two hour old training data.*

The error obtained this way can be compared with the error of the two other classifiers. All the calculated errors can be seen in section 8.1.2.

### 7.1.3 Identifying Training Model

In order to obtain a training model, which is the last step of the off- line algorithm illustrated in figure 7.1, a full set of recorded data is tested. The most satisfying part of the data is chosen as training data. This test is done using a five fold cross-test algorithm and is carried out using the features, feature reduction and classification method determined in chapter 8. The algorithm is explained in the following list, and illustrated in figure 7.5:

1. A set of recorded EMG data, is divided sequentially into five groups consisting of four validation sets and one testing set, each contributing 20%.
2. four tests are carried out, one for each validation set, where the representative set is classified using the three surplus validation data as training data. For each test the error is noted.
3. The combination of validation set, generating the lowest error, is chosen as training data and tested by classifying the test set. If this test fulfils the requirements of having a precision accuracy on min. 95%, the representative three validation sets are chosen as training data.

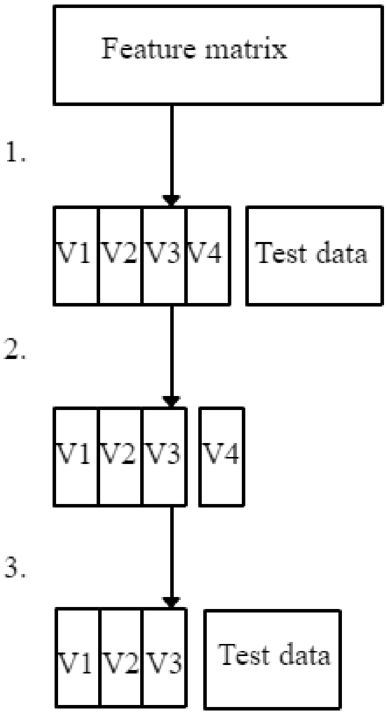


**Figure 7.5:** *An illustration of how a set of recorded data is validated using cross validation method.*

### 7.1.4 Validation of Machine Learning Algorithm

In order to verify the machine learning algorithm, identified in this chapter, fulfils the requirement of a prediction accuracy on 95% for all movements (explained in chapter 5), a test is made. If the test shows that all the movements are classified with a prediction accuracy on min. 95% the algorithm is verified as being a satisfying algorithm for further use. In the test all the specification and the parameters identified in this chapter is used. The test description is the following:

1. eight data sets are obtained using the specifications from table 7.1.

2. The data sets are split into training and test data, using the k fold cross test (one test set and a corresponding training set, for each data set).

3. The test sets are classified a corresponding training set.

4. The prediction accuracy is noted for all the eight recordings, from where a confusion matrix is obtained.

The obtained confusion matrix is described in section 8.1.2.1 along with validation results.

## 7.2 On-line

This section describes the method, and the background for testing the performance of the system. It is done in order to reproduce the method steps undertaken to fulfil the requirements (see chapter 5), for the performance of the system, and to address the problem stated in chapter 4.

As mentioned throughout the report the final goal for the on-line processing was to use the developed supervised machine learning algorithm to create control input for the BeBionic hand. Due to technical issues with the BeBionic hand it was not possible to integrate it in a prototype. Therefore the final prototype of the system was provided with a renewed GUI, which also is used during testing.

The system has to be able to classify the classes successfully repeatedly. In order to check how well the system performs, two tests are undertaken. One to determine if the system is capable of classifying movements within set time limits, and one to determine the stability of the system.

### 7.2.1 Successful Classification

In order to make the test the GUI was changed and a command window randomly telling the subject which motion to do was added. Also an extra bar representing the "No Motion"- class was added. This was done in order to evaluate how well the system detects pause in motion, since this it is essential for a surgeon to be able to rely on the system to do motion when that is what it is supposed to do. The modified GUI can be seen in figure 7.6. The subject in the test would have ten seconds to successfully reach the goal on the GUI. The goal in this test was reach 100% of the bar. The bar increment of the bar was 5% which means that the class has to be classified successfully 20 times before reaching the goal. The GUI "punished" the subject by removing a step of 5% from the wanted movement and adding it to the classified movement if the system classified wrongly. This means that each wrongly classified entry has to be corrected by an extra successful classification. The "punishment" was implemented because it is a way to demand more of user. Furthermore, the "punishment" is a way to simulate the fact that the subject has to correct for the wrong movements.

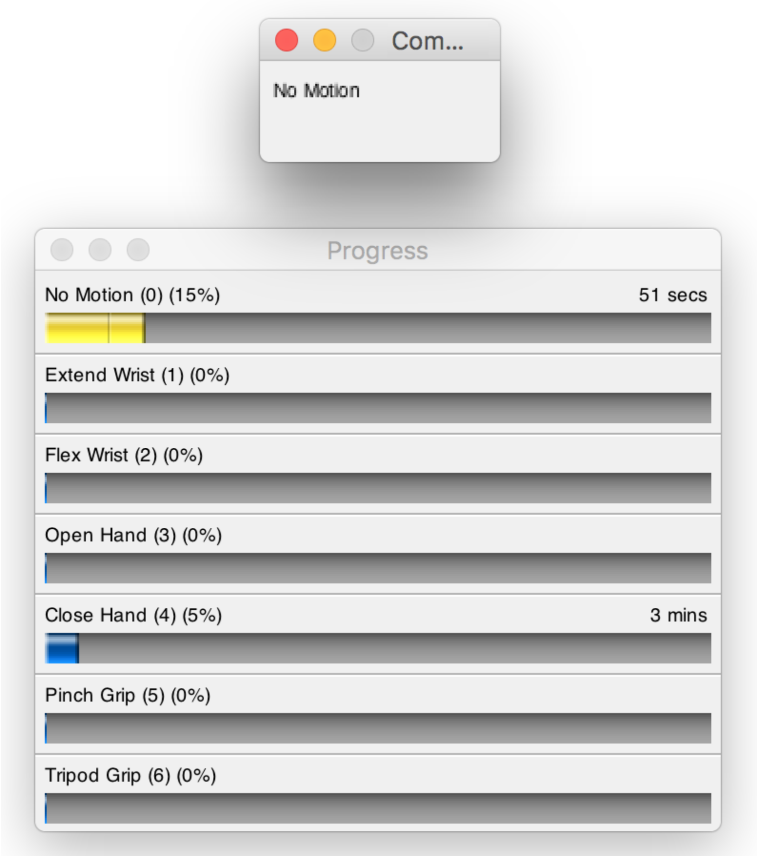


**Figure 7.6:** *The modified graphical user interface with the command window. The bar of the demanded motion is yellow and the others are blue. If the subject is successful it turns green and by failure is turns red.*

In the conducted test, the subject had visual feedback from the GUI as seen on figure 7.7). However, the visual feedback also lets the subject correct the movement according to the feedback in order to be more successful. This will most likely affect the performance. The subject might not do the things as intuitively as it was done during the recording of training data. In some cases the subject's possibility of correcting the motion according to the feedback might lead to better results than if it was done intuitively, however the subject might over-correct in some cases and thereby worsen the performance. In order to test the influence of the visual feedback a "blind test" was conducted where the subject only saw the commanded motion and afterwards whether they were successful or failed. In both tests each motion was tested 25 times giving a total of 175 measurements in each of the two tests. For each measurement it was noted if the bar indicator of the motion was successfully filled before ten seconds, but also the time it took before the bar indicator was filled.

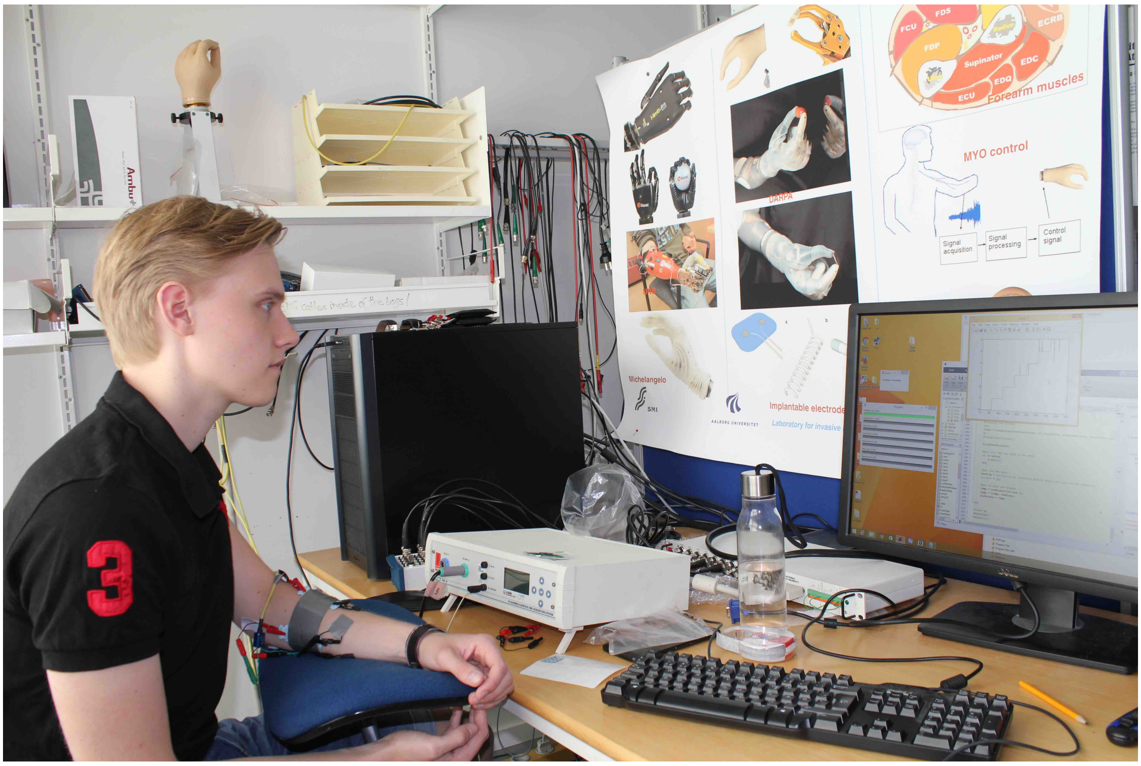


**Figure 7.7:** *The subject doing the test. The subject looks at the screen to see the commanded motions and gets visual feedback from the bars.*

To be able to determine the delay of the system and estimate, whether the time limit set in the test of ten seconds is reasonable, a small test was made to investigate how fast the system could reach success without classification errors. The time of ten successful measurements of 20 classifications without error was measured. The fastest time measured for classifying the same motion 20 times without any classification error was 2.9180 $s$. Whether this means that the requirements to the system about delay have been met will be evaluated in section 8.2.1. However this measurement tells that the time limit first set of ten seconds is a rather long time. From the measured time the lowest number of available classifications within 10 seconds can be calculated.

$$Number\ of\ classifications\ within\ 20\ s = \frac{10\ s}{\frac{2.9180\ s}{20\ classifications}} = 68.5\ classifications \tag{7.3}$$

This gives the subject a rather big margin to reach the target. Suppose the time is halved, this is already a more sensible and interesting time limit to investigate the data with. The time limit of 5 *seconds* corresponds to the shortest time for classifying 20 times plus a few seconds for the subject to react to the command and maybe correct a small error. For this reason the data will also be analysed in regard to success if the time limit is 5 *seconds* as opposed to 10 *seconds*.

The analysis of the data and the findings will be presented in section 8.2.1.

### 7.2.2 Fitts's Law

A way to figure out, how easy and fast the subject can move from one point to another, is by Fitts's law. It is commonly used in GUI design on computers, tablets etc. to determine where to place different objects and buttons, along with what size they should have. Fitts's law is derived by formula 7.4. (Zhao 2002)

$$MT = a + b \cdot log_2(\frac{2A}{W} + 1) \tag{7.4}$$

Equation components:

***MT*** Average time.

***a*** constant from specific application.

***b*** constant from specific application.

***A*** Amplitude from start to target center.

***W*** Width from target center.

Equation 7.5 is called "Index of difficulty" ($ID$) and its units are bits. The higher the number, the more difficult it is to perform. The equation and is derived as follows:

$$ID = log_2(\frac{2A}{W} + 1) \tag{7.5a}$$

$$ID = log_2(\frac{A}{W} + 1) \tag{7.5b}$$

This will express the relation between the amplitude and width in bits, without the application constants.

#### 7.2.2.1 Fitts's Law Test

For this application, $A$ and $W$ are from the GUI (described in section 6.3), where the amplitude is the distance from 0% to the target border, while the target is 97%. The GUI can be seen in figure 6.5. To find $a$, $b$, $A$ and $W$ a test with the GUI is performed. This will include the change of how many steps it takes to reach the target, as well as the size of the target. The test is made to investigate the stability of the system, to see if the increase of the $ID$ will have a linear relation with the time it takes to perform the motions.

The test, were done with three chosen width and three chosen amplitudes, these can be seen in table 7.2.

These values give the $IDs$ shown in table 7.3. As shown, the smaller amplitude and higher target, the easier the movement is to perform.

| **A** | **W** |
|---|---|
| 10 | 5 |
| 20 | 10 |
| 50 | 15 |

**Table 7.2:** *The three different chosen values for A and W*

| Amplitude | Width | ID | Amplitude | Width | ID | Amplitude | Width | ID |
|---|---|---|---|---|---|---|---|---|
| 10 | 5 | 1.5850 | 20 | 5 | 2.3219 | 50 | 5 | 3.4594 |
| 10 | 10 | 1 | 20 | 10 | 1.5850 | 50 | 10 | 2.5850 |
| 10 | 15 | 0.7370 | 20 | 15 | 1.2224 | 50 | 15 | 2.1155 |

**Table 7.3:** *Table showing the IDs on the different values.*

| | | Width | | |
|---|---|---|---|---|
| | | **5** | **10** | **15** |
| **Amplitude** | **10** | 10 | 9 | 9 |
| | **20** | 18 | 17 | 16 |
| | **50** | 45 | 43 | 40 |

**Table 7.4:** *Table of how many steps needed to reach the target.*

The amplitude was normalized so the distance is the number of steps needed to reach the target. While the target is set to 97 and the width is around 97. In table 7.4 the actual number of steps needed to reach the border of the targets is shown.
The Fitts's Law testing was done with one test-person, who was accommodated to control the system. The test subject carried out randomly selected motions. The first two data of each motion were chosen as test data.

In section 8.2.2 the results of the test are explained and discussed.

# Chapter 8

# Results

This chapter contains a collection of all test results, and validation outcomes performed throughout the project and mentioned in methodology chapter 7. This was done to separate all specific results and conclusions from general methodology that was used for purposes of this project.

## 8.1 Off-line

Results of all tests and validations mentioned by methodology in section 7.1 can be found in this section, i.e. all tests and validations of feature extraction and classification, and classifier tests.

### 8.1.1 Feature Selection Results

The features for the algorithm are selected by an iterative process, where the prediction accuracy of 12 features is tested in different combinations of six features, using a linear discriminant analysis (LDA) classifier. In order not to test all the combinations of features, a search for a satisfying feature set (a set which generates prediction accuracy on min. 95%) was undertaken.

The search showed that the following six features generated a satisfying prediction accuracy (the verification of the six features is displayed in section 8.1.1.1):

1. Wilson amplitude (WAMPL).
2. Slope sign changes modified (SSC2).
3. Myopulse percentage rate (MPR).
4. Zero crossing (ZC).
5. Absolute mean (AM)
6. Log detector (LOG).

The threshold values for MPR, ZC and WAMPL, are equally determined by an iterative process, where $50\mu V$ where set as a starting threshold, and adjusted to a satisfying value.

#### 8.1.1.1 Feature Verification Test

In this section the feature set selected in section 8.1.1, is verified(the verification is based on the method described in section 7.1.1.3).

In figure 8.1, the prediction accuracies for 14 recording can be seen. Only one (recording nr. 13) out of the 14 recordings does not fulfil the requirement on a prediction accuracy of 95%. However, while the mean of the 14 prediction accuracies is equal to 99, the six selected features are accepted as a satisfying combination of features, and will be used for the machine learning algorithm.

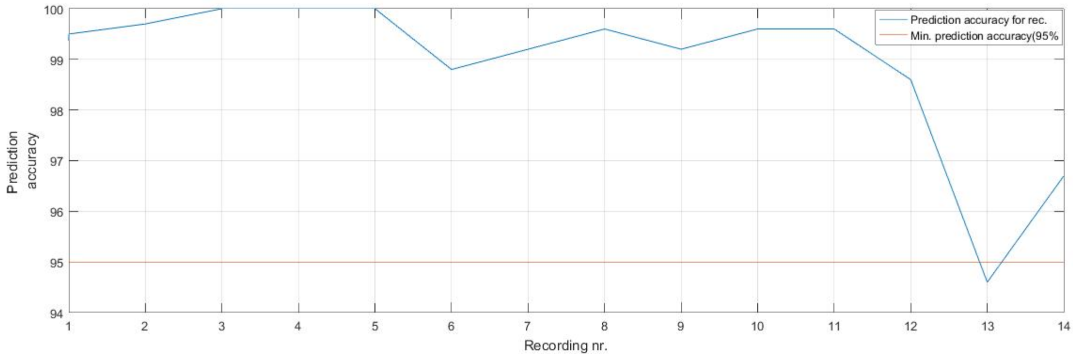


**Figure 8.1:** *Illustrates the prediction accuracy for 14 different recordings (using a LDA classifier and the six selected features). The limit for a min. prediction accuracy is illustrated with the red line.*

#### 8.1.1.2 PCA Verification Results

In the following test the PCA algorithm is tested, using the method stated in section 7.1.1.5. The test takes point in the box-plots of figure 8.2. If the lower quartiles of all the plots are min. 95%, PCA is verified as a satisfying feature reduction method (usable for the supervised machine learning algorithm).

The box plot (see figure 8.2) shows that the ETH, with the highest lower quartile, is 0.99 (dimensions 8-9), and the second highest is 0.98 (dimensions 6-7). This indicates, that for this test a bigger ETH gave a lower error. This means that the bigger the ETH, and dimensions are, the more variance is preserved in the feature space, which in some cases contributes to a smaller error. It can be considered an advantage when choosing an ETH on 0.99. The disadvantage of having a ETH on 0.99, is that the dimensions increase with the threshold, which means that the classifier gets more inputs, and makes the classification more complex.

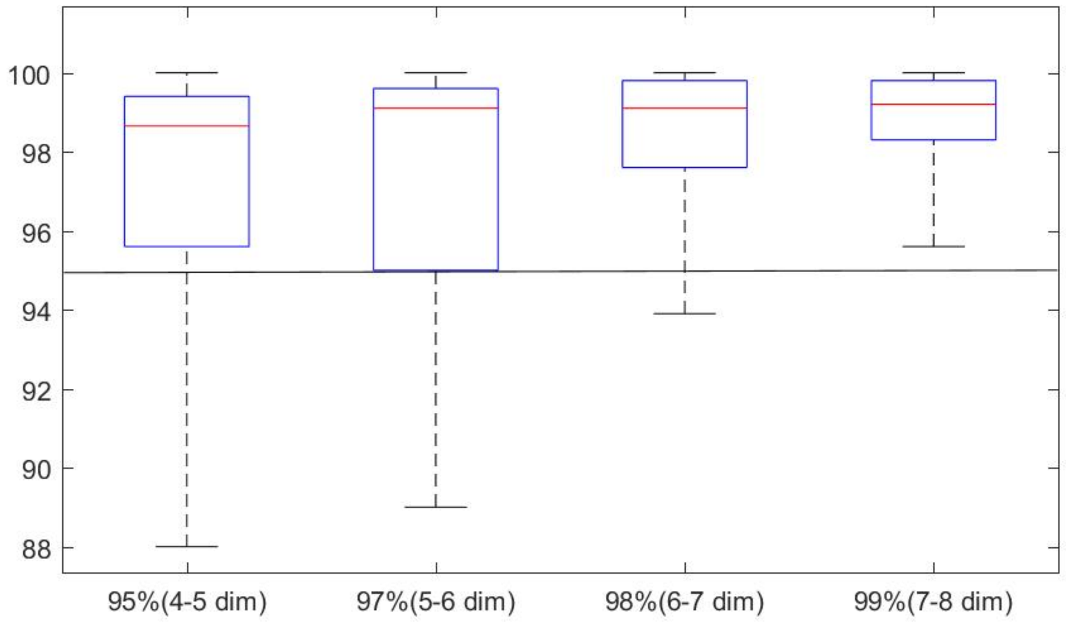


**Figure 8.2:** *A box plot showing the prediction accuracy when using different threshold for the eigenvalue, and the corresponding dimension.*

It does also appear from figure 8.2, that the highest extreme (in all four box-plots) is 100%. Which indicates, that for some training data a threshold on 0.95 is enough to have 0% error. As well, the relation between ETH and the prediction accuracy varies from data to data. Therefore, it might be an advantage to adjust the ETH every time having a new set of training data. The aim of adjusting the ETH, is to have as few dimensions and as high prediction accuracy as possible.

Based fact, that the lower quartile for all the box-plots, are higher than 95%, the PCA algorithm is verified and will be used in the machine learning algorithm explained in chapter 7.1.

### 8.1.2 Two Hour Test and Classification Selection

Results from the two hour test can be seen in table 8.1. The first row shows the three classifiers (see appendix D). The mean and standard deviation is used to make a conclusion on the test success.

| Classifier | | **LDA** | | | | **KNN** | | | | **Bayes** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Test data age | | 2 | 1 | 0 | | 2 | 1 | 0 | | 2 | 1 | 0 |
| Subject 1 (error pct) | | 7.04 | 22.20 | 0 | | 17.61 | 19.51 | 0 | | 15.28 | 21.96 | 1.43 |
| Subject 2 (error pct) | | 7.08 | 14.60 | 6.67 | | 7.99 | 15.51 | 15.95 | | 9.93 | 17.06 | 17.14 |
| Subject 3 (error pct) | | 10.84 | 15.31 | 0 | | 10.92 | 13.96 | 0.48 | | 19.82 | 16.19 | 0 |
| | | | | | | | | | | | | |
| Mean | | 8.32 | 17.37 | 2.22 | | 12.17 | 16.32 | 5.48 | | 15.01 | 18.40 | 6.19 |
| Standard Deviation | | 2.18 | 4.20 | 3.85 | | 4.93 | 2.86 | 9.07 | | 4.95 | 3.11 | 9.51 |

**Table 8.1:** *Comparison of the classifiers using test data two, one and zero hours old.*

It can be seen that the Linear Discriminant has a lower mean and standard deviation in two of the three cases. This indicates that the classifier is able to classify motions with an average lower error than the two other classifiers and furthermore does so in a way where the standard deviation is low compared with KNN and Bayes. This is desirable since the error the classifier produces does not vary in large margins from subject to subject. This makes the system more capable to adapt to different training data from different subjects.

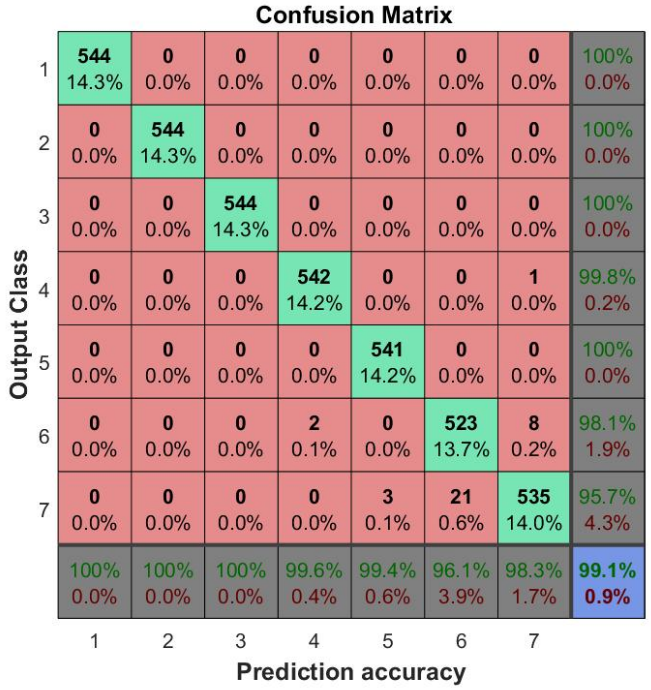
The table also shows how the error rises when the test data is one hour old only to decrease when the test data is two hours old. Furthermore, none of the one or two hour old errors are below the 5%, which is the requirement. This could be because of the missing feedback. The subject does not have a way of getting any feedback during the training, and therefore may change the motions slightly between recordings and thus changing the signal. This means that if the subject received feedback from the classifier, such as the GUI or the BeBionic hand, the numbers may have been different.

Further tests to explain the increase in the error when using one hour old test data could be to have more tests within the two hours. This way it would increase the resolution of the data and it may be possible to see a relation between the increase and decrease in errors.

Due to this, the requirement of a successful classification ratio of 95% will not be confirmed in this test. But the classification method can be selected as it is still advantageous to have a versatile classifier capable of classifying motions, even when the training data is not fresh or the motions changed slightly but maintain the overall motion. The classifier selected is Linear Discriminant Analysis.

#### 8.1.2.1 Algorithm Validation Test

The confusion matrix in figure 8.3 shows that (for the given test) no motion, extend hand and flex hand, have a prediction accuracy on 100%. Open hand has a prediction accuracy on 99.6%, where two out of 544 samples are wrongly classified as class pinch close. Close hand has a prediction accuracy on 99.4%, where 3 out of 544 samples are wrongly classified as tripod. Pinch close has a prediction accuracy on 96.1% where 21 out of 544 samples are classified as tripod, and tripod has a prediction accuracy on 98.3% where one out of 544 samples are wrongly classified as open hand, and where eight are wrongly classified as pinch close. This means that pinch close is the class with the lowest prediction accuracy, followed by tripod. Looking at the scatter plot in figure 8.4, it can be seen that the variance between pinch close and tripod, is not significant, which indicates that the EMG signals generated by the motions are closely related. This might be why the prediction accuracy is lower for pinch close and tripod than the rest of the classes, and why both classes are frequently classified wrongly as each other.



Confusion Matrix

| Output Class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| 1 | **544**<br>14.3% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | 100%<br>0.0% |
| 2 | **0**<br>0.0% | **544**<br>14.3% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | 100%<br>0.0% |
| 3 | **0**<br>0.0% | **0**<br>0.0% | **544**<br>14.3% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | 100%<br>0.0% |
| 4 | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **542**<br>14.2% | **0**<br>0.0% | **0**<br>0.0% | **1**<br>0.0% | 99.8%<br>0.2% |
| 5 | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **541**<br>14.2% | **0**<br>0.0% | **0**<br>0.0% | 100%<br>0.0% |
| 6 | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **2**<br>0.1% | **0**<br>0.0% | **523**<br>13.7% | **8**<br>0.2% | 98.1%<br>1.9% |
| 7 | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **3**<br>0.1% | **21**<br>0.6% | **535**<br>14.0% | 95.7%<br>4.3% |
| | 100%<br>0.0% | 100%<br>0.0% | 100%<br>0.0% | 99.6%<br>0.4% | 99.4%<br>0.6% | 96.1%<br>3.9% | 98.3%<br>1.7% | **99.1%**<br>**0.9%** |

Prediction accuracy

**Figure 8.3:** *A confusion matrix showing the test results(classes; 1 = no motion, 2 = extend hand, 3 = flex hand, 4 = open hand, 5 = close hand, 6 = pinch close, 7 = tripod).*

However, despite the condition that pinch close and tripod have closely related EMG signals, the machine learning algorithm is verified (based on the test), since the test showed that the algorithm was able to satisfy the requirement of a prediction accuracy on min. 95%.

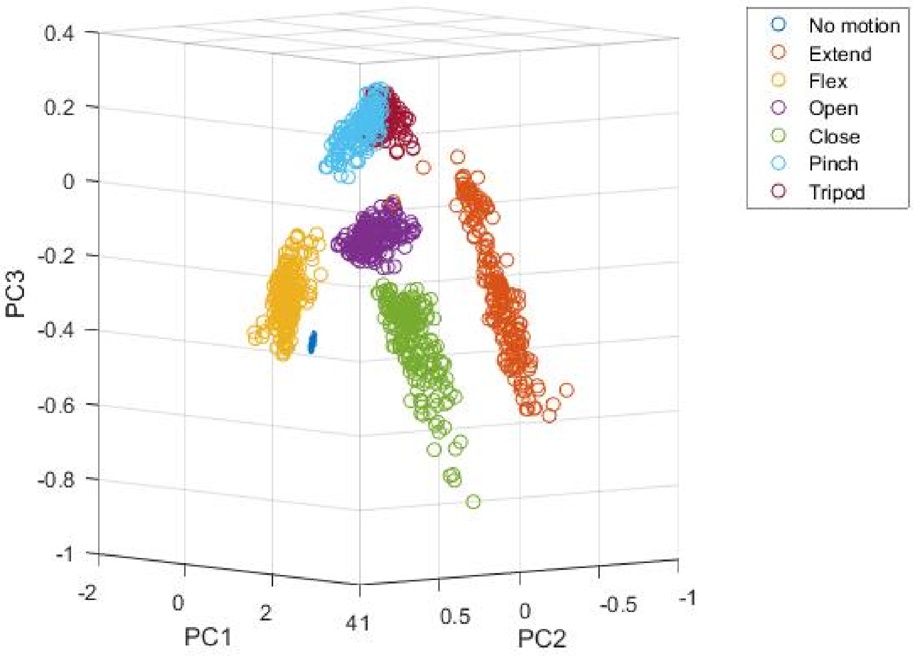


**Figure 8.4:** *Illustrating a scatter plot for one of the eight recordings used for the test.*

## 8.2 On-line

This section consists of tests investigating the performance of the supervised machine learning algorithm, identified in section 8.1 (the algorithm is illustrated in figure 8.5).

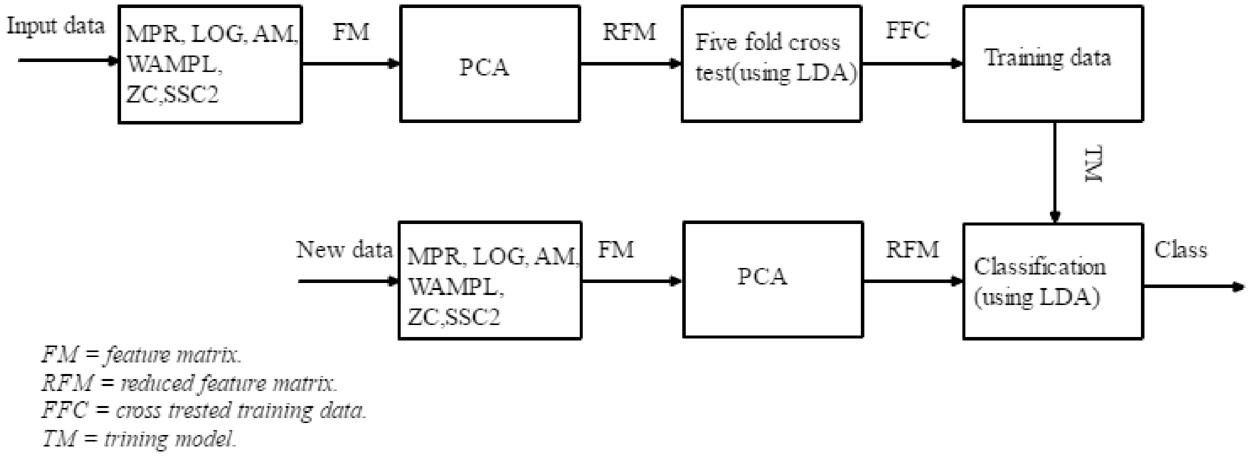

Input data
MPR, LOG, AM, WAMPL, ZC,SSC2
FM
PCA
RFM
Five fold cross test(using LDA)
FFC
Training data
TM
New data
MPR, LOG, AM, WAMPL, ZC,SSC2
FM
PCA
RFM
Classification (using LDA)
Class
FM = feature matrix.
RFM = reduced feature matrix.
FFC = cross trested training data.
TM = trining model.

**Figure 8.5:** *A general illustration of a supervised machine learning algorithm.*

### 8.2.1 GUI Tests

This section investigates and elaborates on the findings of the classification success test.

#### 8.2.1.1 Evaluation of System Delay

The first result gained during the test was a the delay time of the system, i.e. the time it takes for the system between classifying and displaying two inputs. As mentioned in section 7.2.1 the fastest time measured for classifying the same motions 20 times without any classification, the error was 2.9180 seconds. This means that the fastest classifying time for the system can be calculated as approximately $\frac{2.9180\ s}{20\ classifications} = 0.147\ \frac{s}{classification}$. Although, the fastest time might not be representative for the general picture. Therefore the times of ten successful motions for each 20 successful classifications without any errors. The result can be seen in table 8.2.

| Test no. | Time |
|---|---|
| 1 | 2,918 |
| 2 | 2,9909 |
| 3 | 2,9213 |
| 4 | 2,951 |
| 5 | 2,9495 |
| 6 | 2,9563 |
| 7 | 2,9601 |
| 8 | 2,9668 |
| 9 | 2,952 |
| 10 | 2,9618 |

**Table 8.2:** *The ten measured times for classifying the same class 20 times in a row without errors.*

The mean of the ten measurements is $\mu = 2.9528$ and the standard deviation is $SD = 0.0200$ which is rather small. This means that the measured times do not vary much from the mean. If the delay is calculated from the mean, the delay will be $\frac{2.9528\ s}{20\ classifications} = 0.148\ \frac{s}{classification}$.

According to the information above, the system has a delay of 148 $ms$. This means that the delay of the system lies within the interval of optimal delay, which is between $100 - 175$ $ms$. The optimal delay is described in section 3.5. Furthermore, the system meets the requirement described in section 5.3 to the delay. The system has a delay smaller than 300 $ms$ and thereby fulfils the requirement.

#### 8.2.1.2 Success Test

As mentioned in section 7.2.1 where two tests, a normal and a blind test, each giving a total of 175 data-points, were carried out. Afterwards, the two datasets were analysed in regard to success rate, mean time and standard deviation. In the graphs in this section, classes are represented by numbers. The class number correlation is as follows:

**Class 0**
No Motion.

**Class 1**
Extend Wrist.

**Class 2**
Flex Wrist.

**Class 3**
Open Hand.

**Class 4**
Closed Hand.

**Class 5**
Pinch Grip.

**Class 6**
Tripod Grip.

The first investigation was about whether success was achieved within the time limit of 10 *seconds*. Figure 8.6 shows the success rate for the classes in the two tests. As each class was repeated 25 times, each repetition corresponds to 4% of the bars. On the graph it can be seen that in the normal test only one failure of reaching the target within 10 *seconds* was noted. This was in the class 5, "Pinch Grip". During the blind test, the success rate of the first five classes remained unchanged, while the success rate of class 5, "Pinch Grip", decreased even further and an error occurred in class 6, "Tripod Grip", as well. Figure 8.7a shows the general success rate. It can be seen that the failure rate increased when blind testing. However, the success rate was above 95% in both tests.

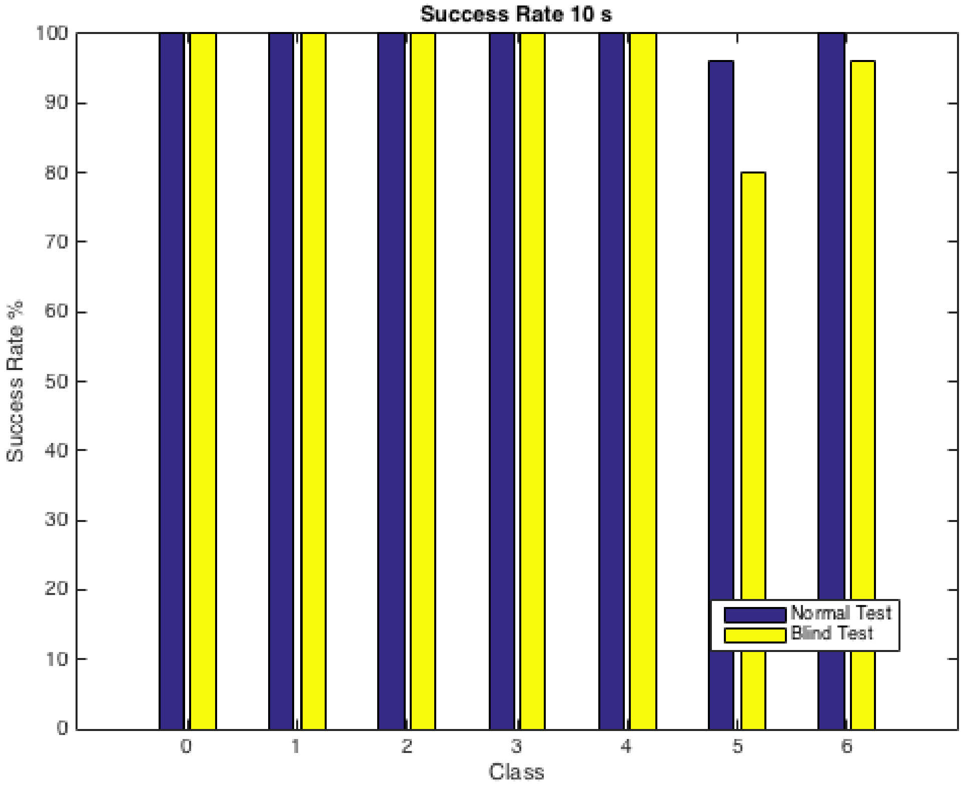
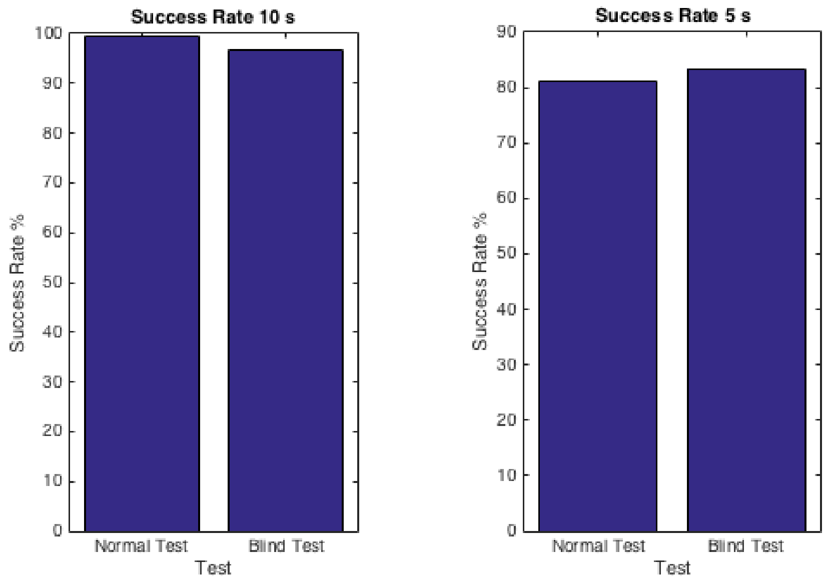


**Figure 8.6:** *The success rate of the classes separately in the two tests with a time limit of* 10 *seconds.*

Figure 8.8 shows the success rates for reaching the target within 5 *seconds*. As it can be seen there is a decrease in the success rate in all classes except by "No Motion" in the blind test. Furthermore, there are several cases where the success rate is higher in the blind test than the normal test. This is also the case when the observing the general picture as portrayed in figure 8.7b. There can be several reasons for this tendency. One reason could be the fact that the motions are done more intuitively in the blind test and therefore better match the training data, which is based on motions done intuitively. This could be the case because the subject does not have visual feedback and is not trying to "satisfy" the GUI and over-correcting. Another reason could be the fact that when the subject has the bars to look at the subject tends to be less focused on the command window and instead seeks information from the bars and their colours. Therefore the "reaction time" is longer by the normal test than by the blind test.



**(a)** *The general success rate across all classes in the two tests with a time limit of* 10 *seconds.*

**(b)** *The general success rate across all classes in the two tests with a time limit of* 5 *seconds.*

**Figure 8.7:** *General success rates for the two tests.*

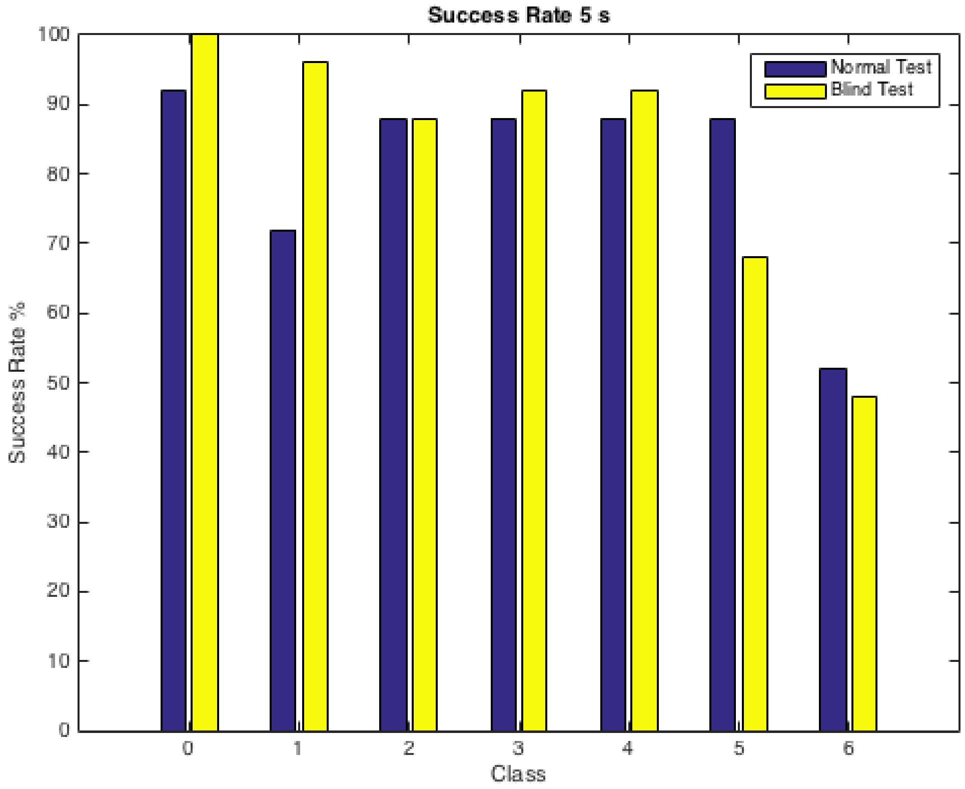


**Figure 8.8:** *The success rate of all the classes separately in the two tests with a time limit of* 5 *seconds.*

Another thing to notice figure 8.8 is that the success rate of the classes 5 and 6 drop even further and where the other classes have a higher success rate in the blind test those two classes both have a lower success rate in the blind test. This observation suggests that those two classes in average take a longer time to achieve success than the first five classes. Furthermore, it suggests that it takes in average longer time to reach success without visual feedback than with. In order to investigate whether this suggestion is true, the average time for success and the standard deviation is investigated. The bar graph in figure 8.9 shows the mean of the measured times and the standard deviation. The data underlying the graph can be seen in table 8.3.

| | Normal Test | | Blind Test | |
|---|---|---|---|---|
| Class | Mean | Standard Deviation | Mean | Standard Deviation |
| No Motion (0) | 4.1228 | 0.9518 | 3.5201 | 0.5955 |
| Extend Wrist (1) | 4.7198 | 0.9538 | 4.2565 | 0.4993 |
| Flex Wrist (2) | 4.6172 | 1.0206 | 4.5256 | 0.6953 |
| Open Hand (3) | 4.2643 | 0.4327 | 4.3459 | 0.3526 |
| Close Hand (4) | 4.4595 | 0.5470 | 4.5409 | 0.8320 |
| Pinch Grip (5) | 4.3444 | 1.6147 | 5.4534 | 2.5993 |
| Tripod Grip (6) | 4.9884 | 0.9279 | 5.1631 | 1.6143 |

**Table 8.3:** *The mean of the completion time and the corresponding standard deviation of the different demanded classes in the two tests. If an attempt was unsuccessful the time for that attempt is included as ten seconds.*

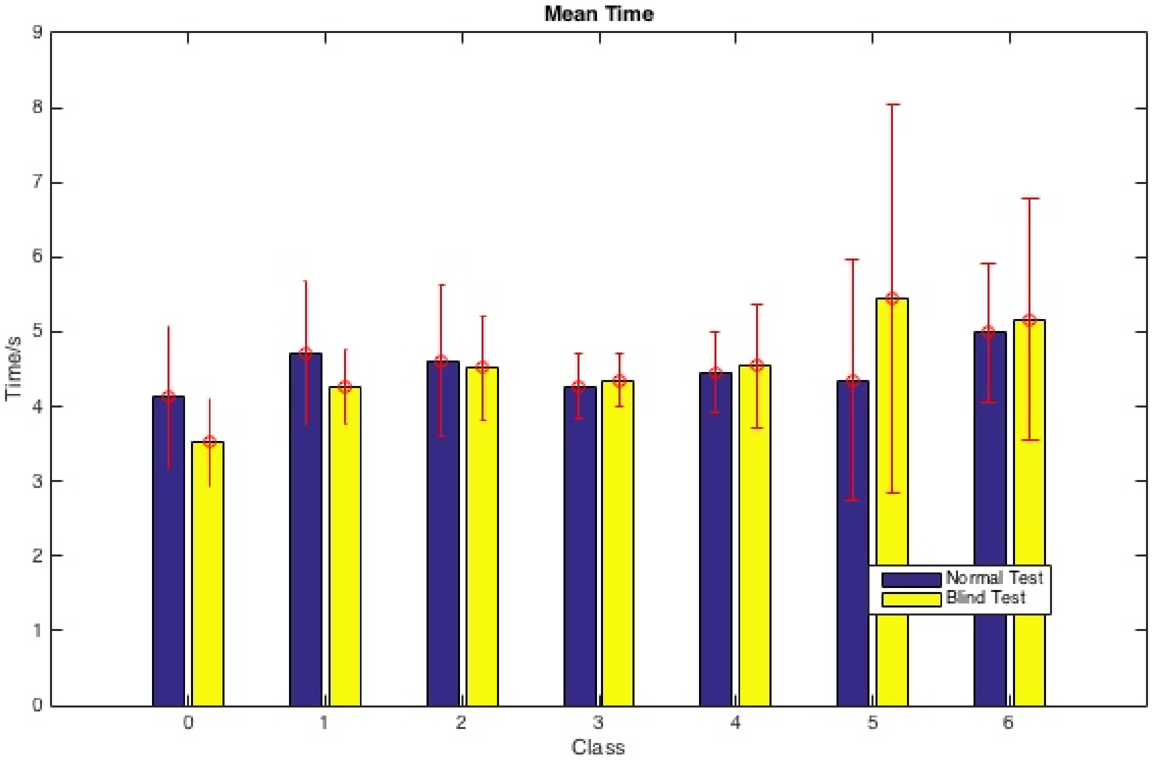


**Figure 8.9:** *The mean of the time to succeed of the different classes separately in the two tests. The red lines indicate the standard deviation.*

Looking at figure 8.9 it can be seen that during the blind test the average time for success is higher for class 5 and 6 compared to the other classes. It can also be seen that the standard deviation is rather large for the two last classes in both the normal and the blind test compared to the other classes. This was also very distinct in the blind test. The observation suggests that the system is less stable when classifying class 5 and 6.

A logical reason for these problems could be that they are hard to classify. With simple reasoning one can easily conclude that the motions behind the two classes, the "Pinch Grip" and the "Tripod Grip" are very similar but is this also the case from a classification point of view. To get an answer, the training data is examined. Figure 8.10 shows that there is a small classification error during the five fold cross-test, where class 6 is confused with class 5 training data.

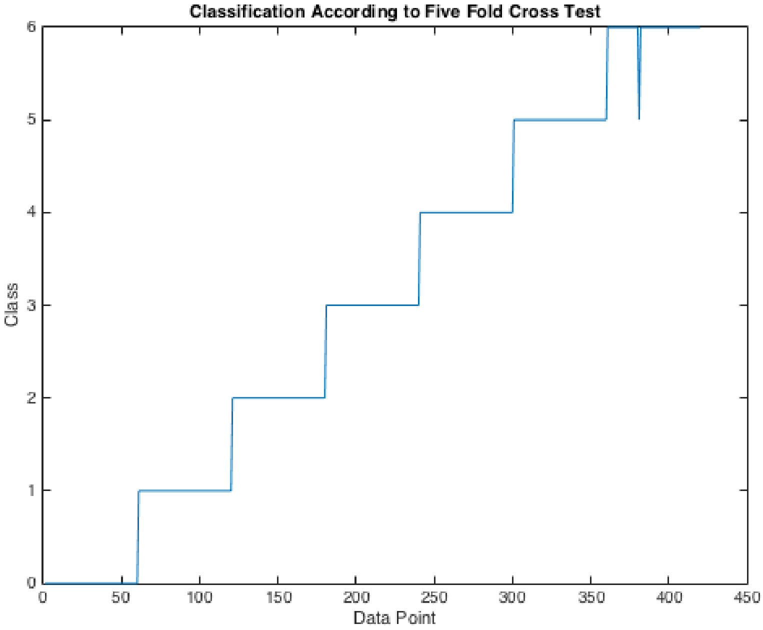


**Figure 8.10:** *The classification of the training data according to the five fold cross-test. A confusion between "Pinch Grip" and "Tripod Grip" can be seen.*

If all data points from the training data are plotted in respect to different principal components identified by the PCA an overlap between classes can be seen. As can be seen in figure 8.11 the classes "Pinch Grip" and "Tripod Grip" overlap in the space of the second, third, and fourth principal component. Furthermore, as shown in figure 8.12 "Tripod Grip" also overlaps with "Open Hand" in the space of the first, second, and fourth principal component. The overlapping of those classes might just be the reason that the system is less stable in the fifth and sixth class because the system might be making classification errors in when the subject attempts to do class 5 and 6. This also correlates with the confusion matrix about off-line data in section 8.1.2.1, which shows that the greatest confusion between classes is between class 5 and 6.

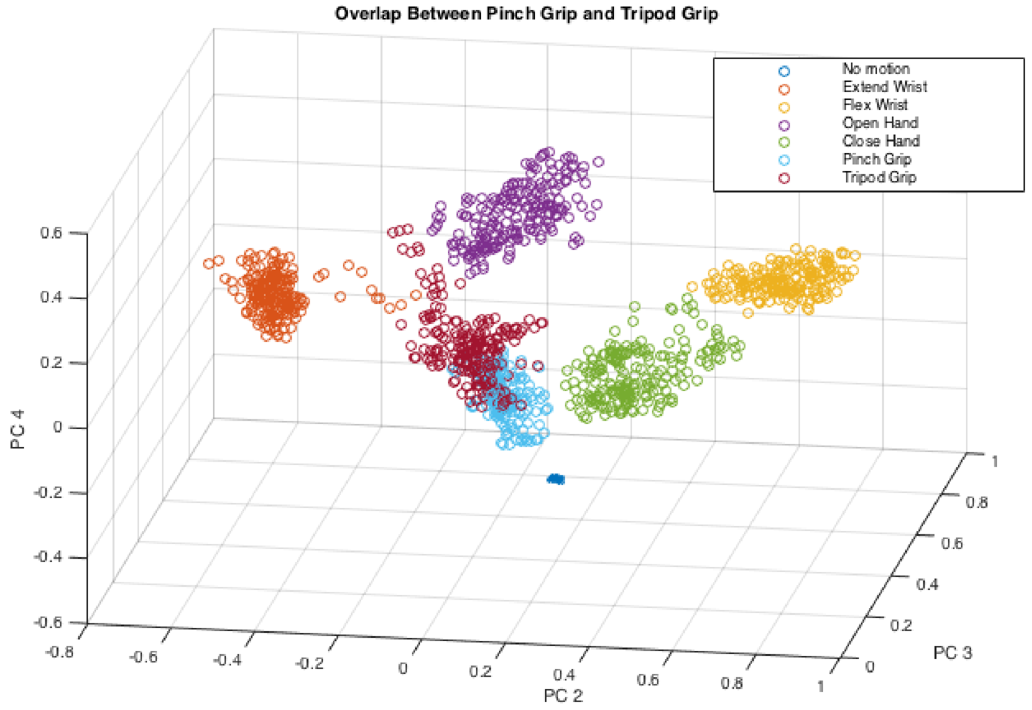


**Figure 8.11:** *The distribution of the data points of the seven classes according to the second, third, and fourth principal component. The five of the classes can easily be separated by a plane as opposed to "Pinch Grip" and "Tripod Grip" which overlap.*

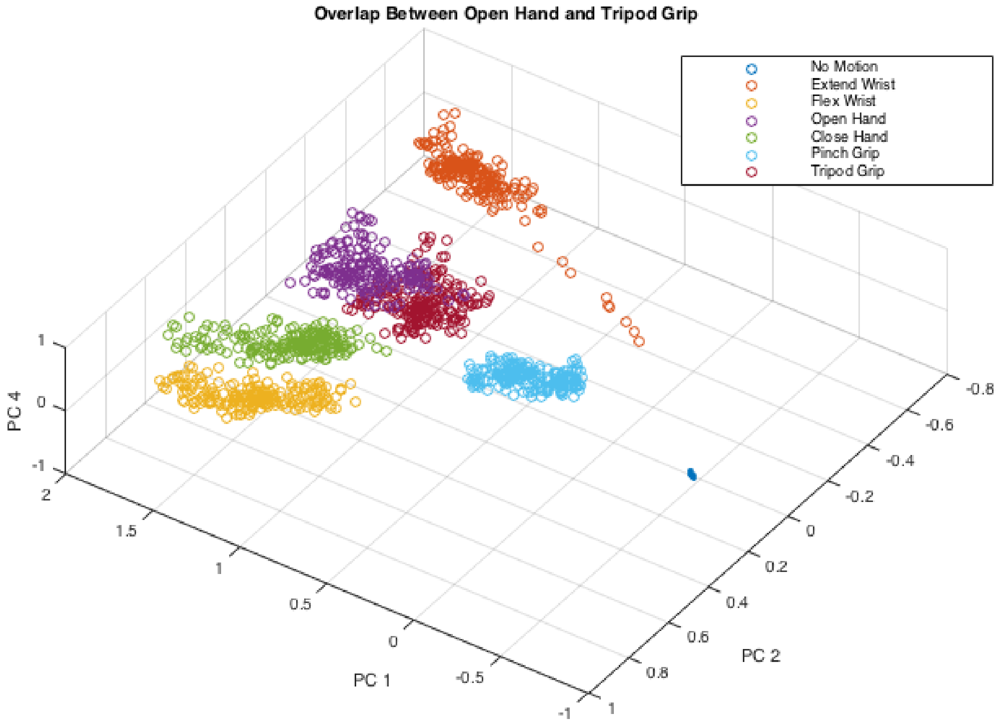


**Figure 8.12:** *The distribution of the data points of the seven classes according to the first second and fourth principal component. The five of the classes can easily be separated by a plane as opposed to "Open Hand" and "Tripod Grip" which overlap.*

The general time for classification and the standard deviation for the system in the two tests are shown in table 8.4 and figure 8.13. From the figure it can be seen that the average success time is below the second limit set of 5 *seconds*, which definitely is preferable. Besides that it can be seen that the average time is a bit longer for the blind test and that the standard deviation also is larger compared to the normal test. This is expectable considering figure 8.9 and the data about classes 5 and 6 obtained from the blind test.

| | Normal Test | | Blind Test | |
|---|---|---|---|---|
| | Mean | Standard Deviation | Mean | Standard Deviation |
| Across all data | 4.4914 | 1.0199 | 4.6008 | 1.4137 |

**Table 8.4:** *The mean time of completion and the standard deviation in the two tests across all classes. If an attempt was unsuccessful the time for that attempt is included as 10 seconds.*

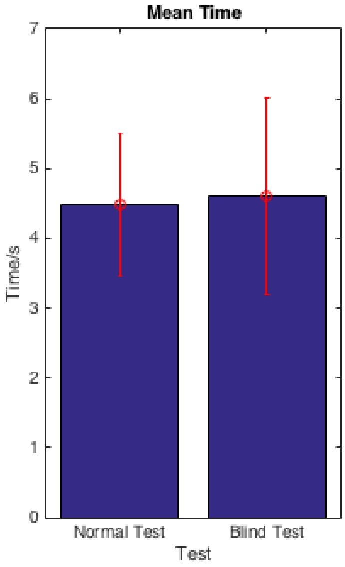


**Figure 8.13:** *The mean of the time to succeed across all classes in the two tests. The red lines indicate the standard deviation.*

In relation to the tests conducted there are some insecurities that affect the results. One of such insecurities is the fact the subject in the tests has to react to the commands given by the test program. This means that within the time measured is also the reaction time. Another insecurity, that might affect the time measured, or ultimately whether it was a successful attempt according to the test, is human errors. During the testing the subject accidentally sometimes made errors, and started out making the wrong motions. This delays the attempt to do the right motion and eventually causes a longer measure time. This cannot be seen in the data, and since the program has no way to know what the subjects intentions were it is hard to correct. The data recorded in such cases will indicate that the system responds ineffectively even though it may have responded perfectly to what the subject was doing. Last but not least the fatigue of the subject building up through out the test might affect the results. The test requires certain exercises to be done in order to obtain sufficient data, which puts a heavy workload on the subject's arm. During the test the subject expressed increased difficulty doing the demanded motions throughout the testing. However, this is not something that can be prevented since the two tests have to be conducted with the same training data in order to obtain valid data. Also, the longer the tests take, the longer time it will be since the last system training, which might influence the data.

To summarise, the success rate of both tests was rather high. Moving the time limit from 10 *seconds* to 5 *seconds* caused a decrease in success rate especially in the last two classes. The success rate across all classes was still above 80%. In average the system is able to reach success within 5 seconds both in the normal test and in the blind test. The system is able to classify quite well however the classification of class 5 and 6, "Pinch Grip" and "Tripod Grip", that are very similar poses a challenge. This influences the stability of the classifications and the time it takes to reach success.

### 8.2.2 Fitts's Test Results

The result of this can be shown in figure 8.14, where the relation between the time and the $ID$ is shown. The relation is linear, and with linear regression it is possible to get the $a$ and $b$ for Fitts's formula. In the figure the STD is also added to the points, where it is visible that most of the data is close to the mean.

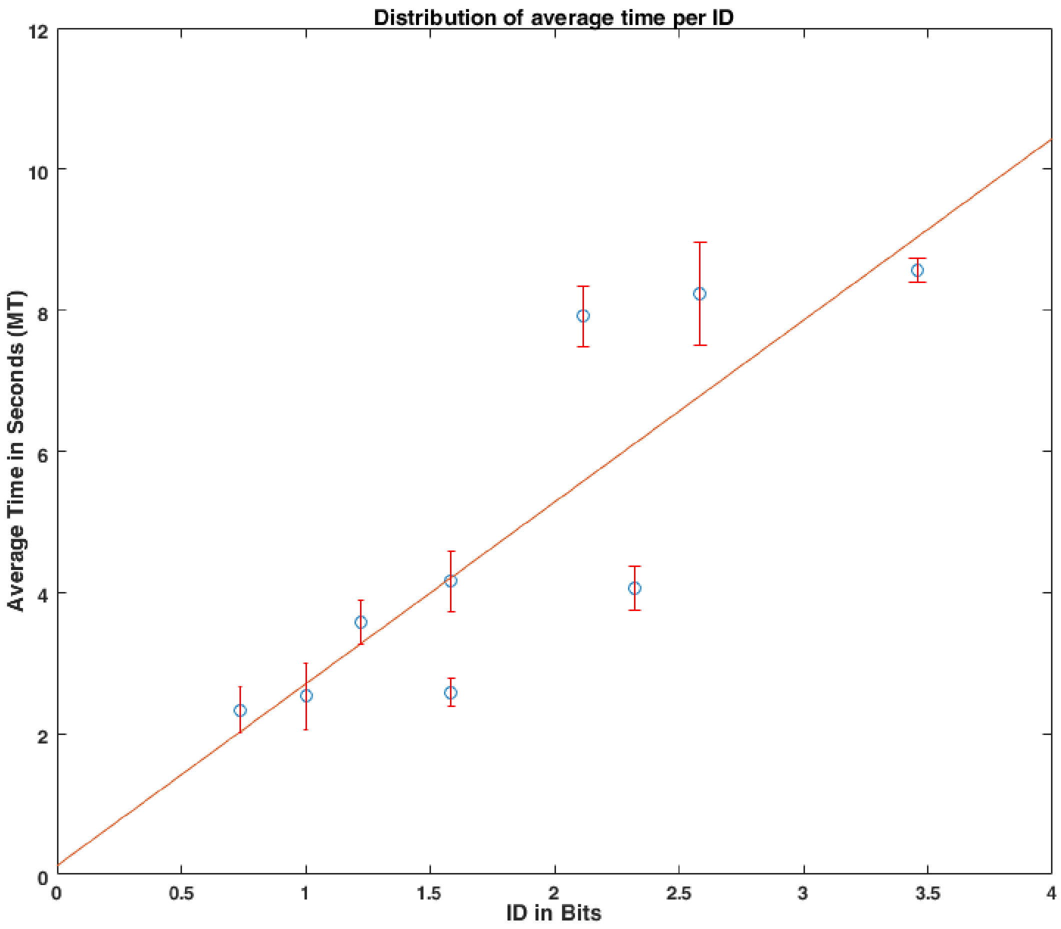


**Figure 8.14:** *An overview of the Fitts's test with the linear regression.*

The $R^2$ value is 0.726. Although if it is not the optimal value for $R^2$, it still gives an image of a linear relation between the $ID$ and $MT$. With the linear regression from figure 8.14, we know that $a$ and $b$ equal to 0.1269 and 2.5767 respectively. With this, equation 8.1 is the final Fitts's Law equation for this application.

$$MT = 0.1269 + 2.5767 \cdot log_2(\frac{A}{W} + 1) \tag{8.1}$$

This gives the average time for the chosen length and target width.

A reason to why the data not gives a higher $R^2$ value, can be because of the time. The test took around 1.5 hour to perform and the test subjects muscles may have started to fatigue. Also, the motions where randomly selected, thus reaction time may have differed throughout the test. Nonetheless, the relation is still clearly visible that the $ID$ and $MT$ have a linear relation.

In this project, it is used to test to see if there were any linear consistency between $ID$ and $MT$

of the application. As the requirement states, it is desired to reach a $R^2$ value of 0.9, however, the requirement was not reached, thus it can be concluded that the system is not as stable as desired.

# Chapter 9

# Conclusion

This project is concerned by implementing EMG signals, perceived from forearms, to carry out controls of a surgery robot. The analysis in the start of the project showed that the most used method of controlling surgery robots is by a joystick controller. Among the variety of surgical robot systems, the da Vinci is most frequently used. The analysis also concluded that the precision of robot surgery is higher and it is quicker to do surgeries with a robot. However, it was also concluded that there is a large training time in learning to operate a robotic surgery, because of the joystick controller. This led to a problem formulation alongside the requirements for the system.

An analysis of the arm muscles was done to see what muscles humans use to make different hand and wrist movements. Alongside the analysis the BeBionic robotic hand was given for a real set-up environment. The robotic hand inspired the six different movements needed to be performed by the surgeon.

The project was delimited to only focus on the classification of motions because of technical issues, thus not using a fully functional robot. The classification was done first with a machine learning algorithm, which extracts the features of the EMG signal thus creating a training model. Afterwards the redundant features are reduced. Lastly the recorded movement were classified and compared against new data. For visual feedback a GUI was created, which was also used in the different tests.

In this application a delay was a concern. However, the system has a delay smaller than 300 ms which complies with the requirements. The machine learning algorithm was able to classify all the six poses with more than 95% success rate. The system was also tested with a two hour test, to investigate if the system was suitable for making a successful classification after the given interval. Since the system was not able to maintain the success rate of 95% the requirement of two hour was not met. The findings indicated that this may not be the correct way to test it, since there was noway to ensure that the poses are carried out in the same manner. The final solution was required to carry out on-line recordings, provided four minutes of training data. The requirement is considered accomplished thanks successful classification and feature extraction. The system was also tested on how easy it is to comprehend by different users. This was done with a GUI, where the user had a time limit to reach the target. This was done to see if there were any patterns in how the different movements were performed. The test concluded that the system works well, although there are certain difficulties with "pinch grip" and "tripod", since they are similar. The system was also tested by Fitts's law, to see if the system was stable. This was done with the test of different lengths of the GUI bars, to see if there was a linear relation between the elapsed time and the length of the bar. The system was concluded to be unstable according to specified requirement.

It can also be concluded that majority of the requirements were successfully reached and most of the tests were carried out with success.

Further studies could be implementing a robotic hand and evaluate the system including the hand, investigate how the recording of signals could be optimised which might affect the stability, or lastly examine whether the system is actually an intuitive way to control a manipulator not resembling the human hand.

# Bibliography

Angkoon Phinyomark Pornchai Phukpattaranont, Chusak Limsakul (2012). “Feature reduction and selection for EMG signal classification”. In:

BeBionic (2016). *Product Brochure.* URL: http://mindtrans.narod.ru/pdfs/bebionic-Product-Brochure-Final.pdf.

Bergqvist, Jesper (2016). “Ofte stillede spoergsmaal”. In: *Herlev hospital.*

Burrus, C. Sidney (2016). *Chebyshev Filter Properties.* OpenStax CNX. URL: http://cnx.org/contents/DqoXKCgk@2/Chebyshev-Filter-Properties.

Christian U. Grosse, Hans W. Reinhardt (2002). *Signal conditioning in acoustic emission analysis using wavelets.* URL: http://www.ndt.net/article/v07n09/08/08.htm.

D. Wright, Jason et al. (2014). “Comparative Effectiveness of Robotically Assisted Compared With Laparoscopic Adnexal Surgery for Benign Gynecologic Disease”. In: *Obstet Gynecol.*

Dekker, Marcel (2004). *Biosignal and Biomedical Image Processing.*

EDCH (2015). *CHAPTER 8 ANALOG FILTERS.* URL: http://www.analog.com/library/analogDialogue/archives/43-09/EDCh%208%20filter.pdf?doc=ADA4666-2.pdf.

Eugenie L. Wacter, Susan M (2001). *City in the Twenty-First Centery: Global Urbanization.*

Gottlieb, Scott (2001). *Surgeons perform transatlantic operation using fibreoptics.* URL: http://search.proquest.com.zorac.aub.aau.dk/docview/204004647/83615E2C0D784CF6PQ/3?accountid=8144.

Hall, John E. and Arthur C. Guyton (2011). *Guyton and Hall - Textbook of medical Physiology.* 12th. Elsevier.

Intuitive-Surgical (2015). *da Vinci Single-Site Instruments and Accessories.* URL: http://www.intuitivesurgic com/products/davinci_surgical_system/da-vinci-single-site/.

jojo (2011a). *Higher Order Filters.* Circuits Today. URL: http://www.circuitstoday.com/higher-order-filters.

— (2011b). *Type of Active Filters.* Circuits Today. URL: http://www.circuitstoday.com/active-filter-types.

Lechky, Olga (1985). “World’s first surgical robot in B.C.” In: *The Medical Post* 21.23. URL: http://www.brianday.ca/imagez/1051_28738.pdf.

Li, Guanglin (2011). *Electromyography Pattern-Recognition-Based Control of Powered Multifunctional Upper- Limb Prostheses.* InTech.

Li, Guanglin and Todd A. Kuiken (2009). “EMG Pattern Recognition Control of Multifunctional Prostheses by Transradial Amputees”. In: *31st Annual International Conference of the IEEE EMBS.*

Marieb, ElaineN. (2001). *Human Anatomy & Physiology.* Fifth edition. Pearson Education, Benjamin Cummings.

Masatoshi Eto, Seiji Naito (2007). *Robotic Surgery Assisted by the ZEUS System.* Tech. rep. Department of Urology, Graduate School of Medical Sciences, Kyushu University. URL: http://eknygos.lsmuni.lt/springer/137/39-48.pdf.

Medicine, Anatomy (2016). *The muscles of the arm and hand*. Anatomy Medicine. URL: http://anatomy-medicine.com/uploads/posts/2015-11/1447074703_arm-and-hand-anatomy.jpg.

Merletti, Roberto and Philip A. Parker (2004). *Electromyography - Physiology, Engineering, and Non-invasive Applications*. John Wiley & Sons.

miniDSP (2016). *FIR vs IIR filtering*. URL: https://www.minidsp.com/applications/dsp-basics/fir-vs-iir-filtering.

OpendTect (2015). OpendTect. URL: http://www.opendtect.org/relman/4.6.0/unpacked/4.6.0/doc/User/base/appendix_frequency-filter.htm.

Overall, Michael (2013). "Hillcrest Medical Center robot can perform war-zone surgery". In:

Paul J. Johnson David E. Schmidt, Umamaheswar Duvvuri (2014). "Output control of da Vinci surgical system's surgical graspers". In: *Journal of Surgical Research* 186. URL: http://www.sciencedirect.com/science/article/pii/S0022480413007105.

prWeb (2010). *Picture*. URL: http://ww1.prweb.com/prfiles/2007/05/10/525578/iBotAGoogleB.jpg.

R. Farrell, Todd and Richard F. Weir (2007). "The Optimal ConControl Delay for Myoelectric Prostheses". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.1.

Rose, William (2014). *Electromyogram Analysis*. URL: https://www.udel.edu/biology/rosewc/kaap686/notes/EMG%20analysis.pdf.

Samadi, David (2016). *Prostatectomy - Prostate Removal Surgery*. URL: http://www.roboticoncology.com/oncology-articles/prostate-removal-surgery/.

Stanford (2016). *Stability Revisited*. URL: https://ccrma.stanford.edu/~jos/fp/Stability_Revisited.html.

Steeper (2016). *BeBionic hand*. URL: http://bebionic.com/.

Surg, J. (2007). "Rural and remote surgery". In: *Royal Australasian College of Surgeons*.

Thede, Les (2004). *Practical Analog and Digital Filter Design*. Artech House Microwave Library. Artech House.

University, Brown (2016). *da Vinci Surgical System*. URL: http://biomed.brown.edu/Courses/BI108/BI108_2005_Groups/04/davinci.html.

Wikipedia (2005). *Ideal Band Pass Filter Transfer Function*. URL: https://commons.wikimedia.org/wiki/File:Ideal_Band_Pass_Filter_Transfer_Function.PNG.

Xing, Kexin et al. (2014). "A real-time EMG pattern recognition method for virtual myoelectric hand control". In: *Neurocomputing* 136.

Zhao, Haixia (2002). *Fitt's Law: Modeling Movement Time in HCI*. Computer Science University of Maryland. URL: https://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/fitts.html.

# Appendix

# Appendix A

# Analogue Processing

Analogue processing is a way to reshape signal in the analogue domain, which includes filtering and amplification. This can be done with different methods, which will be discussed in this appendix. EMG signals have a voltage of 0 to 10 mV peak to peak and have a frequency range of 20 Hz to 500 Hz. The signal needs to be filtered and amplified so the signal is better suited for the system and analogue to digital converter, which includes removal of noise.

## A.1 Filtering and Amplification

An analogue amplifier device receives the raw EMG signal from the muscles. This includes the EMG signal, DC noise and 50 Hz noise. All the noise should be removed before the signal is used. First the signal is amplified with a gain of 2000, which however also includes noise. Afterwards the signal goes trough a low- and high-pass filter. On an amplifier device, the high-pass filter can be set to 20 Hz and the low-pass filter to 500 Hz. This removes all noise outside of the EMG spectrum, including baseline drift (i.e. very low frequency) and DC offset (i.e. constant voltage offset). It also enables the possibility of avoiding aliasing in the sample data. (Rose 2014)

## A.2 Bandpass Filter and Method

Low- and high-pass filters let every frequency through above or below defined values, bandpass filter will only let some certain frequency through. In the ideal world, as seen in figure A.1, the frequency will only be let through when it is in the defined range. This will give the bandpass a rectangular shape.

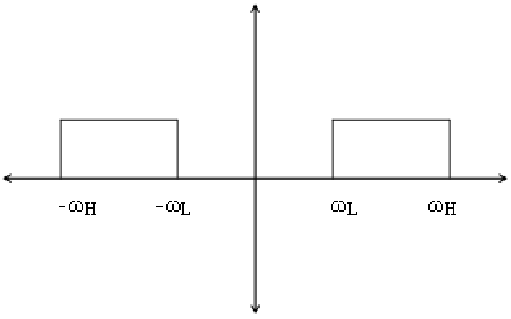


**Figure A.1:** *An ideal sketch of two bandpass filters. (Wikipedia 2005)*

In a real scenario the ideal bandpass filter is not possible. Therefore the bandpass needs to be designed to have a rectangular shape, which creates ripples. This is illustrated in figure A.2, where the rippling is visual in the stopband as well in the passband. The sharp edges are also gone, instead there are slopes which are defined within a certain area.

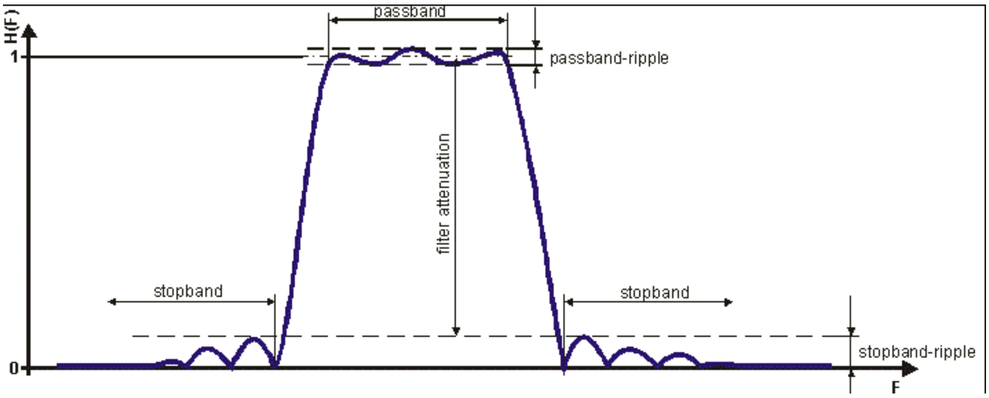


**Figure A.2:** *A real sketch of one bandpass filter. (Christian U. Grosse 2002)*

From figure A.2 it can be seen that the slopes and the rippling are not as rectangular as the ideal scenario. There are though numerous methods to accomplish a more rectangular formed signal.

### A.2.1 Butterworth Filter

Butterworth filter is a signal processing filter, which is specially designed to have a flat passband. In equation A.1 the Butterworth equation is shown, which finds the magnitude frequency response, where $B$ shows that the transfer function used by the Butterworth filter. Here $\omega_0$ is the cut-off frequency of the filter and $n$ is the order of the frequency. In equation A.2 $\varepsilon$ is described as the passband gain adjustment factor. (Thede 2004)

$$\left| H_{B,n}\left[ j\left(\frac{\omega}{\omega_0}\right)\right]\right| = \frac{1}{\sqrt{1+\varepsilon^2\cdot\left(\frac{\omega}{\omega_0}^{2\cdot n}\right)}} \tag{A.1}$$

Where

$$\varepsilon = \sqrt{10^{-0.1\cdot a_{pass}} - 1} \tag{A.2}$$

When designing a Butterworth filter, the $n$-order is needed to be chosen. In figure A.3 five different orders are shown. It can be seen, that the higher the order the more rectangular shaped the signal will be.

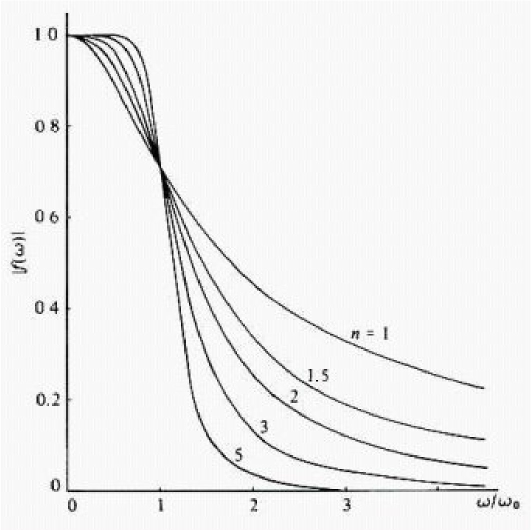


**Figure A.3:** *A sketch of the some Butterworth orders. (OpendTect 2015)*

Although that the signal will be more like the ideal signal, higher order is not always the best. The reason behind this is that the higher the order, the more complex and expensive it also becomes. Therefore, when the most important is a flat passband and not a more precise stopband or resources do not allow for a higher order, a lower order Butterworth is commonly used. (jojo 2011a)

### A.2.2 Chebyshev Filter

Unlike the Butterworth filter, the Chebyshev is not designed to have a flat passband or stopband, it is designed to have a steeper curve than the Butterworth. This steeper roll-off makes it come closer to the ideal scenario, but with the trade-off of the ripple in the passband and stopband.
In equation A.3, the Chebyshev equation can be seen. (Thede 2004)

$$\left|H_{C,n}\left[j\left(\frac{\omega}{\omega_0}\right)\right]\right| = \frac{1}{\sqrt{1+\varepsilon^2\cdot C_n^2\left(\frac{\omega}{\omega_0}\right)}} \tag{A.3}$$

Where $\varepsilon$ is the same as in equation A.2.
$C_n(\omega)$ is the Chebyshev polynomial, with the first kind of degree $n$. We define the normalized Chebyshev polynomial $\omega_0 = 1$ as seen in equation A.4.

$$C_n(\omega) = cos\left[n\cdot cos^{-1}(\omega)\right],\ \omega \leq 0 \tag{A.4a}$$

$$C_n(\omega) = cosh\left[n\cdot cosh^{-1}(\omega)\right],\ \omega > 0 \tag{A.4b}$$

In figure A.4 three examples of Chebyshev filter are shown. The first shows the lowpass, where the ripple is seen in the passband. The second shows the highpass where the ripple is also present. However the slopes are more steep and look more like the ones in the ideal scenario.

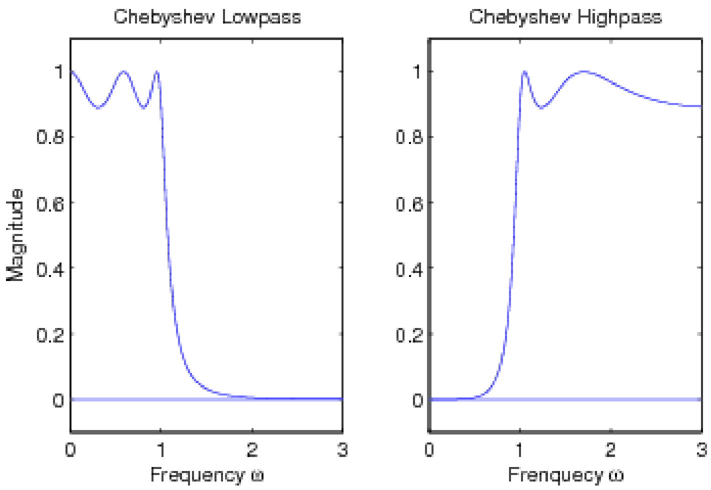


**Figure A.4:** *A sketch of Chebyshev filters. (Burrus 2016)*

### A.2.3 Bessel Filter

The Bessel filter is much like the Butterworth filter, designed to have a flat passband and stopband. In equation A.5 the Bessel filter is shown.

$$H(s) = \frac{\theta_n(0)}{\theta_n\left(\frac{s}{\omega_0}\right)} \tag{A.5}$$

Here $\theta_n(s)$ is the reverse Bessel polynomial. The other variables is the same as the previously filters. The Bessel filter is designed to have a minimal overshoot and ripple. In figure A.5 the filter is illustrated. (EDCH 2015)

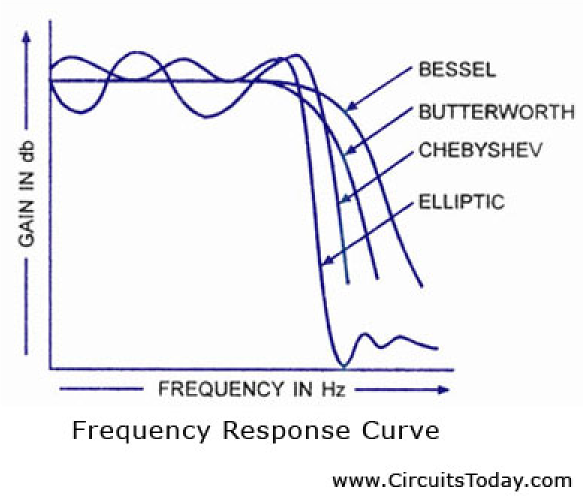


**Figure A.5:** *A sketch of Bessel filter compared to the other filters. (jojo 2011b)*

# Appendix B

# Analogue to Digital Converter

Analogue to digital converters (ADC) are used when an analogue signal is given and has to be converted to digital signal, e.g. in order to be processed by a computer. The important factors when designing or deciding on an ADC are based on the incoming signal, what resolution is desired when the analogue signal is represented as digital numbers and what amplitudes, measured in voltage, the signal has.

## B.1 ADC Limits

AD converters are designed to associate specific intervals of an incoming signal to specific output values. Since digital numbers are represented as bits and it is not possible to have infinite number of bits, there is a limit to how big a number a given amount of bits can represent.
While EMG signals lie in the microvolts to mini volts, they are often amplified. This amplification makes the amplitudes of the signal bigger. But since the ADC has a max voltage, it is important to design the amplification so that the maximum expected signal amplitude does not exceed that of the ADC. If it does, the ADC will not be able to associate the measured amplitude with a correct value, and will instead read it as the maximum voltage even though it might be higher. This scenario can be seen in figure B.1.

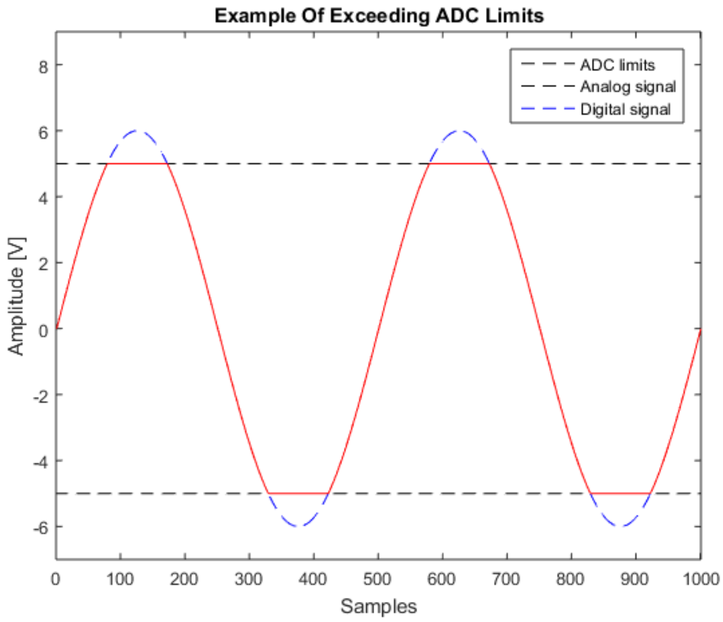


**Figure B.1:** *Example scenario where the amplitude of the incoming signal (±6$V$) is larger than the limits of the ADC (±5$V$).*

Another problem to consider is when the signal has DC noise. This means that e.g. if the signal is lying between $+5V$ and $-5V$, the same goes for the ADC, and there is a DC noise, parts of the signal will be lost as well. This is due to the mean of the signal not lying at zero, but instead at whatever value the DC noise provides, thus the signal fluctuates about the DC noise (see figure B.2).

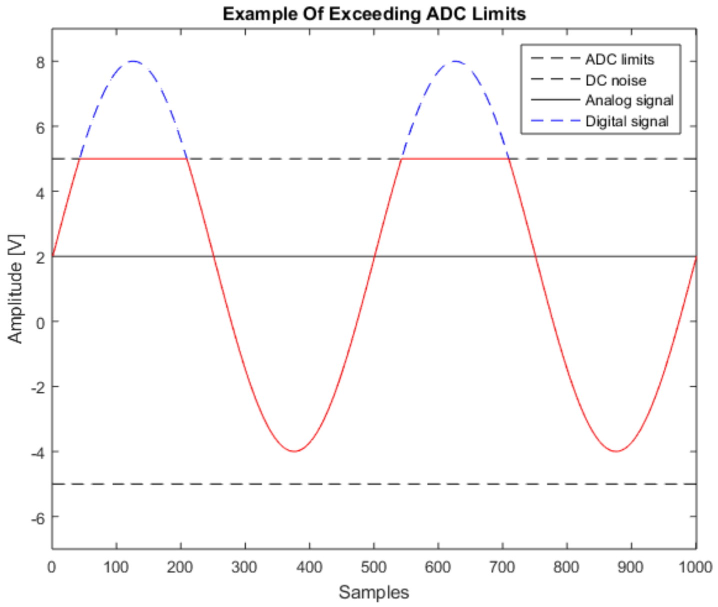


**Figure B.2:** *Example scenario where the amplitude of the incoming signal (±6$V$) is larger than the limits of the ADC (±5$V$) with DC noise present.*

Therefore it is important to not amplify the signal so much, that it will exceed the limits of the ADC, or it is important to select a ADC with limits higher than the maximum amplitude that can be expected with from the incoming signal.

## B.2 Least Significant Bit

The least significant bit also called the resolution is a description that tells how precise the ADC is capable of representing the analogue signal. The resolution is a measure of how many bits the analogue signal is converted into. The more bits available the less changes in amplitude can be described. If an ADC with too few bits is selected, the converted signal will not describe the analogue signal very well (see figure B.3).

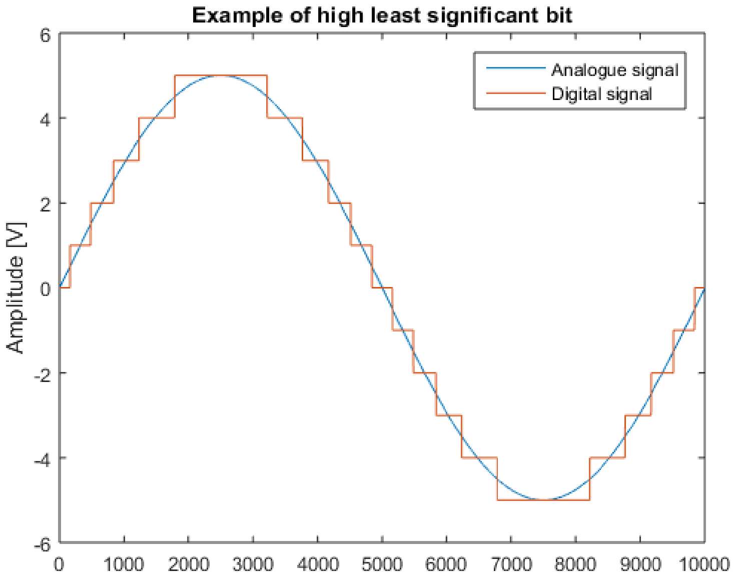


**Figure B.3:** *Here the least significant bit will be equal to a change in voltage of 1.*

In the extreme case of having an analogue signal with amplitudes of $-5V$ and $+5V$ converted to a 1 bit value, the output of the ADC will always be either 0 or 1 corresponding to $-5V$ and $+5V$.

## B.3 Sampling Rate

The sampling rate is telling the ADC how many times per second it should measure the voltage and convert it to a number. If e.g. the sampling rate is 100 Hz, the ADC will read the voltage and convert it every 10 ms. In theory it is desirable to have a very high sampling rate so the smallest changes in the signal can be detected. In practice, however, this is not always the case. The higher the sampling rate the more data will be created, and this is sets higher requirements for the computer that is to process it all. Therefore it is often desirable to select the lowest sampling rate which is still capable of representing the analogue signal to a satisfying rate.

If a too small sampling rate is selected, the discrete signal may not represent the analogue signal but an entirely different signal. This scenario is called aliasing and can bee seen in figure B.4.

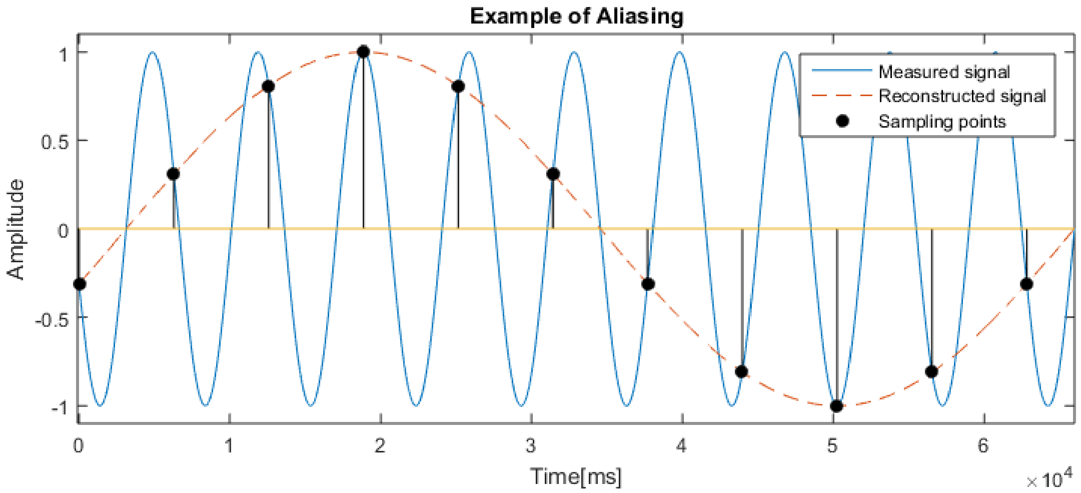


**Figure B.4:** *Low sampling rate resulting in aliasing.*

The lowest possible sampling frequency is two times the frequency desired to be measured. If the sampling is any lower than that, the aliasing is guaranteed. A rule of thumb says to use 10 times the maximum frequency of the analogue signal.

The sampling rate for EMG is often set to 2000, even though the maximum frequency of EMG is 500 Hz and by the rule of thumb therefore have to sample at 5000 Hz. This lower sampling rate is due to the fact that the EMG signal is concentrated around the middle frequencies, i.e. 50-200 Hz. This can be seen when plotting an EMG signal in the frequency domain as seen in figure B.5.

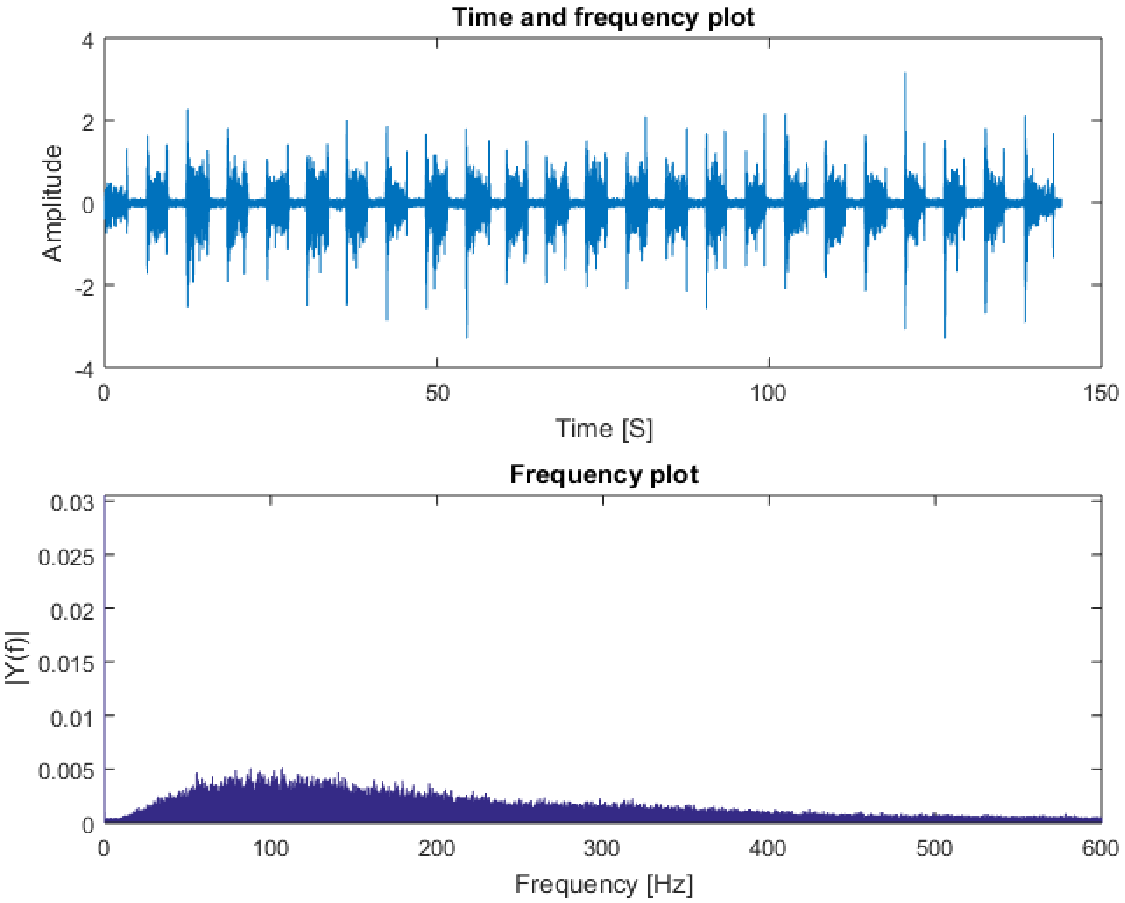


**Figure B.5:** *EMG signal in the frequency domain.*

Thus the sampled data can be kept at the minimum by selecting the right sampling rate, without losing important data.

# Appendix C

# Z-transform and Digital Filters

As the Laplace transform operation is used in analogue filtering, so is Z-transform (operation analogous to Laplace transform) used for digital filters. Using Z-transform, the digital equivalent of transfer function can be defined, which makes this operation convenient. Figure C.1 illustrates the transfer function. In real applications though, the transfer function is more complicated, having polynomials in both numerator and denominator, thus forming zeros and poles, like transfer functions of Laplace transform. The digital transfer function is defined by equation C.5, where X(z) stands for the input and Y(z) is the output.

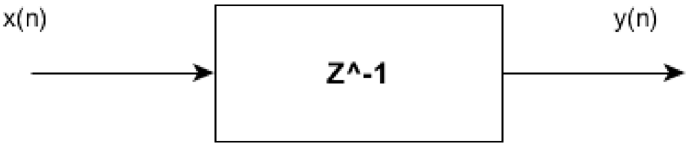


**Figure C.1:** *Illustration of digital transfer function.*

$$H(z) = \frac{Y(z)}{X(z)} \tag{C.1}$$

From the transfer function formula, the output can be expressed as follows in equation C.2.

$$Y(z) = H(z)X(z) = X(z)\frac{\sum\limits_{k=0}^{N-1} b(k)z^{-n}}{\sum\limits_{l=0}^{D-1} a(l)z^{-n}} \tag{C.2}$$

By applying the inverse transformation (thus expressing the equation in sample domain) we arrive at equation C.3.

$$y(n) = \sum_{k=0}^{K} b(k)x(n-k) - \sum_{l=0}^{L} a(l)y(n-l) \tag{C.3}$$

## C.1 FIR against IIR Filters

The main difference between FIR and IIR simply lies in their digital transfer function. FIR's transfer function does not have any denominator (see equation C.4), which causes the impulse response to be finite. Notice also that FIR for that reason does not have "a" coefficients, thus, when it comes to expressing the output, the second summation containing "a" in equation C.3 will be omitted.

It generally applies that FIR is more complicated to implement, and it requires more computation time, and memory. However there are certain advantages over IIR that FIR possesses:

1. FIR is always stable.
2. Linear-phase filtering can be implemented by FIR, meaning that there is no phase shift across the whole frequency.
3. When correcting frequency response errors, the FIR's precision is superior to IIR.

$$H(z) = B(z) \tag{C.4}$$

The IIR has the standard transfer function template, i.e. coefficients in both numerator and denominator (equation C.5). Its main disadvantage is its non-linear phase characteristics, which complicates implementation. On the other hand, IIR filters can meet certain frequency criteria with lower filter order (number of coefficients) than FIR filters, thus saving the computation time.

$$H(z) = \frac{Y(z)}{X(z)} \tag{C.5}$$

Those differences are needed to keep in mind, if one needs to decide the purpose of applied filter. Based on the purpose, the correct filter type, which fulfils the needs, can be then chosen.

## C.2 Filter Stability

A stable filter produces limited response when limited input signal is provided. If this is not the case, the filter is considered unstable and there is not much use for such filter design.

A filter is considered to be stable if its impulse response h(n) goes to zero as n approaches infinity (equation C.6).

$$\lim_{n\to\infty} h(n) = 0 \tag{C.6}$$

Standard stability inspection method is to print all poles of the transfer function onto z plane and see if they all lay in the unit circle. If all poles are inside the unit circle, the filter is considered stable. Otherwise, poles laying outside the unit circle cause instability. See figure C.2 for illustration. (Stanford 2016)

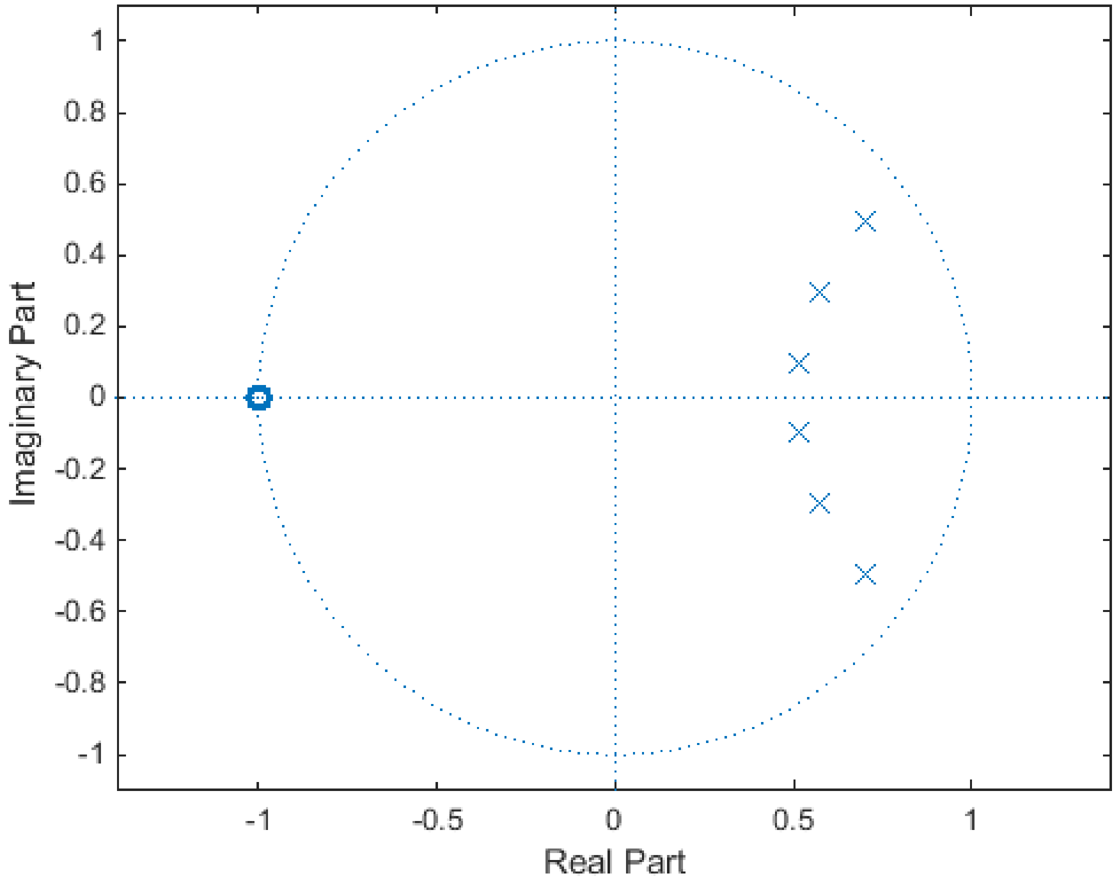


**Figure C.2:** *Poles of sixth order Butterworth printed on the unit circle on z-plane in MATLAB. All poles are in the circle, which means stability of the filter.*

## C.3 MATLAB Implementation

MATLAB possesses a lot of assistive routines when designing a digital filter. Applying Z-transform and deriving at filtered output is carried out by routine described in C.7, where y is the output, x is the input, and b with a are the desired coefficients.

$$y = filter(b, a, x) \tag{C.7}$$

For this to work though, the coefficients b(n) and a(n) must be already known. That is where the actual filter design comes to play, because different types of filters have different ways in obtaining the coefficients, which results in different behaviour of the filter. MATLAB has predefined functions for obtaining the coefficients through applying the most known filters (Butterworth, Chebyshev ect.). Those are generating the coefficients from specified data, which usually are filter order(n), normalized cut-off frequency(Wn), and filter type (lowpass, highpass, bandstop, bandpass). See the syntax for applying a Butterworth filter in routine C.8.

$$[b, a] = butter(n, Wn,' type') \tag{C.8}$$

Note that applying the routine C.7 applies the desired filter, but also shifts the phase of the filtered signal away from the original signal. Now this can be avoided by using routine C.9. This way the filtered signal is going to be in the same phase as the original, however, it doubles the filter order.

$$y = filtfilt(b, a, x) \tag{C.9}$$

If it is desired to investigate stability of a filter, MATLAB offers a simple function which you can see in routine C.10. Provided the filter coefficients, the function returns true (1) if the filter is stable, or false (0) if the filter is unstable. (Dekker 2004) (miniDSP 2016)

$$flag = isstable(b, a) \tag{C.10}$$

# Appendix D

# Signal Classification Methods

Classification is a method that takes an input in the form of different features and tries to determine which class the given input belongs to. This is useful in a variety of scenarios, and does not only apply to EMG features, but could for instance also be applied in an image processing application.

Since all features are numbers they can be associated and plotted. This could for instance mean that if there are two features, mean and peak count for a one channel EMG, they can be plotted in a 2D plot. If another feature was added it would be plotted in a 3D plot. If another channel was added, the number of dimensions of the plot would increase with the number channels multiplied by the number of features used, and vice versa. Thus the number of dimensions is the product of the number of channels and features.

Different methods can be used to classify an input. They do all have one thing in common though. They all require pre-recorded features (training data) for comparisons. If the input is very similar to a group of features which all belong to the same class, the input is determined to also belong to that class.

This chapter discusses some different methods and explains how they work and their implementation.

## D.1 K nearest neighbour

The K nearest neighbour classifier may be one of the most simple classifiers. It is based on the distance between two features and can be calculated by using the Pythagoras theorem. This will then be done for all points and a distance matrix can be obtained that contains the distance from the input point to all other points. The K nearest points are then found and depending on what class is represented the most in those K samples, the input is classified as.

Figure D.1 illustrates how the nearest points are found end then the class is identified based on a majority vote.

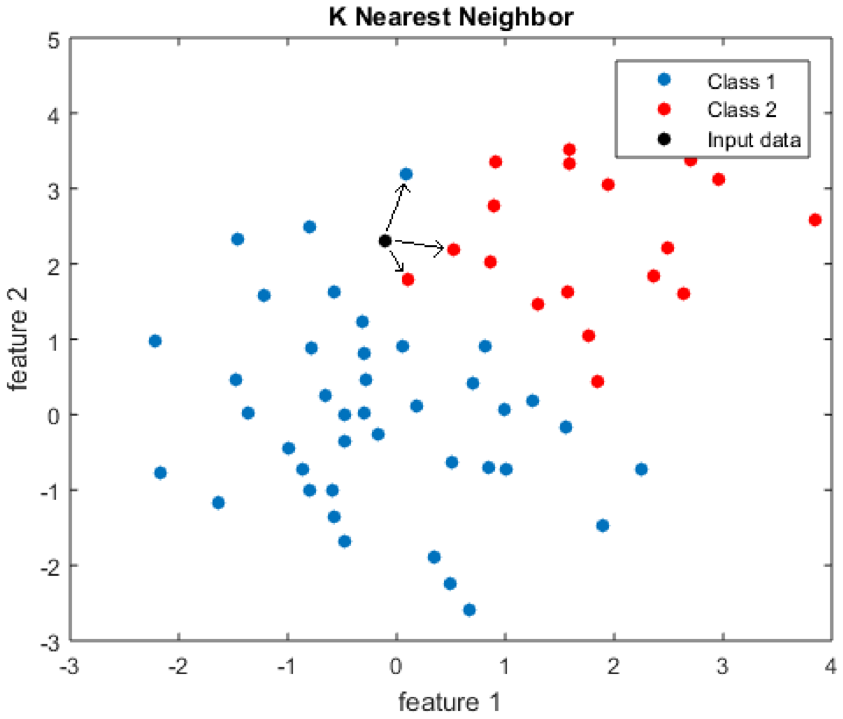


**Figure D.1:** *Example of K nearest neighbour with $K = 3$. The majority vote favours red and thus the input is classified as class 2.*

## D.2 Linear Discriminant

Another simple classification method is the Linear Discriminant. This method tries to separate the classes by simply adding a line between the different classes in the training data. The input is assigned to a class depending on what side of the separation line it stands. Figure D.2 illustrates this principle.

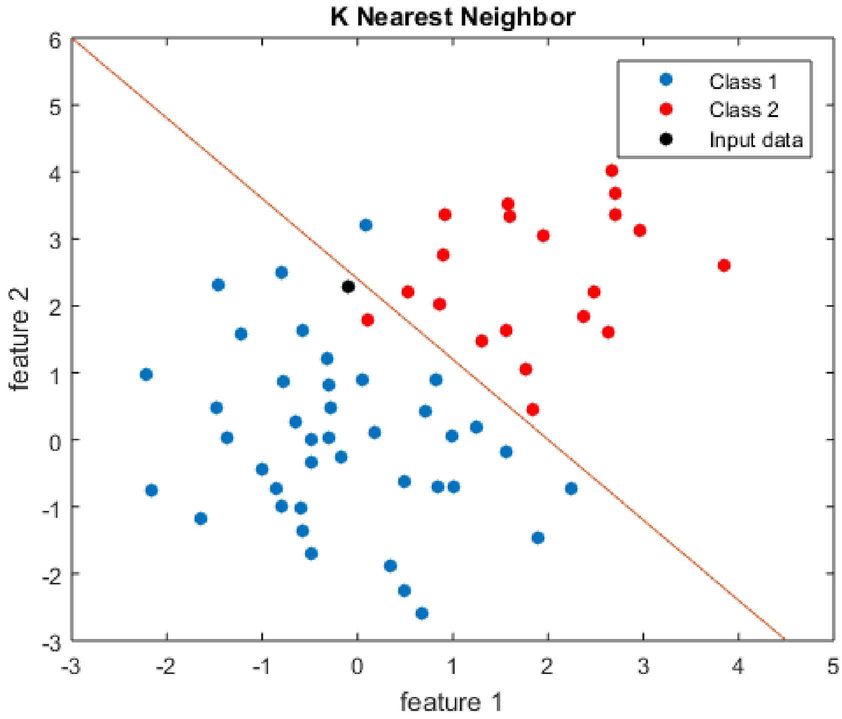


**Figure D.2:** *Example of Linear Discriminant. The input data lies on the blue side and thus is classified as class 1.*

One thing to be aware of when designing a system that relies on a linear discriminant classifier, is that over-fitting can become a problem. It is possible to have a classifier that is able to separate 100% of all class 1 on the left and red on the right, but this introduces the over-fitting problem where the classifier is only reliable in that single scenario and not in the general case.

## D.3 Naive Bayes Classifier

The third and last classifier mentioned here is the Naive Bayes classifier which is a probabilistic classifier and can be seen in equation D.1. Where $w_j$ is a class and $x$ is a random feature that is to be classified.

$$P(w_j \mid x) = \frac{p(x \mid w_j) \times P(w_j)}{p(x)} = \frac{likelihood \times prior}{evidence} \tag{D.1}$$

Where:

1. $\mathbf{P(w_j \mid x)}$ **(Posterior)**

   The probability that it is class $w_j$ given random feature $x$.

2. $\mathbf{P(x \mid w_j)}$ **(Likelihood)**

   The probability of observing feature $x$ assuming class $w_j$.

3. $\mathbf{P(w_j)}$ **(Prior)**

   The probability of observing class $w_j$. Ex. if two class a equally likely to occur, this will be 0.5.

4. $\mathbf{P(x)}$ **(Evidence)**

The probability of observing feature $x$. The evidence is used to normalise so that $\sum P(w_j \mid x) = 1$ and can therefore often be left as it is not the actual percentage probability that is interesting, but rather which class is more probable.

An example for this could be that a system has to classify whether it is seeing oranges or bananas. All incoming fruit is measured on their yellowness and their length. The system has data on 1000 pieces of fruit and it is distributed as seen in table D.1

| Type | Feature 1 | | Feature 2 | | Total |
|---|---|---|---|---|---|
| | long | not long | yellow | not yellow | |
| Orange | 0 | 600 | 200 | 400 | 600 |
| Banana | 200 | 200 | 300 | 100 | 400 |
| Total | 200 | 800 | 500 | 500 | 1000 |

**Table D.1:** *Features of oranges and banana.*

The prior ($P(w_j)$) for these two classes can then be calculated:

$$P(orange) = \frac{600}{1000} = 0.6$$
$$P(banana) = \frac{400}{1000} = 0.4$$

And the evidence:

$$p(long) = \frac{200}{1000} = 0.2$$
$$p(yellow) = \frac{500}{1000} = 0.5$$

And finally the likelihood can be calculated.

$$\begin{aligned} P(long \mid orange) &= \frac{0}{600} = 0 \\ P(long \mid banana) &= \frac{200}{400} = 0.5 \\ P(yellow \mid orange) &= \frac{200}{600} = \frac{1}{3} \\ P(yellow \mid banana) &= \frac{300}{400} = 0.75 \end{aligned}$$

If a random fruit is then given, the posterior can now be calculated. If the fruit is both long and yellow it will be classified as (note that the evidence is left out as it is a simple comparison of the two probabilities and thus the normalisation is redundant):

$$\begin{aligned} P(orange \mid long, sweet) &= P(long \mid orange)P(yellow \mid orange) \times p(orange) \\ &= 0 \times \frac{1}{3} \times 0.6 = 0 \\ P(banana \mid long, sweet) &= P(long \mid banana)P(yellow \mid banana) \times p(banana) \\ &= 0.5 \times 0.75 \times 0.4 = 0.15 \end{aligned}$$

And since $0 < 0.15$ the random fruit is classified as a banana.

As it can be seen in the figures D.1 and D.2 the K nearest neighbour and the Linear Discriminant can give different results even with the same training data. This indicates that both methods can give errors (i.e. wrong classification) and thus having good training data, that is clearly separated, is desirable.

# Appendix E

# Feature Extraction

This appendix gives the mathematical description of the different features used feature extraction for a signal.

## E.1 Simple Square Integral

Simple square integral (SSI) uses the energy of the EMG signal as a feature. It square all the EMG signal amplitude values and then sum them up. This parameter is named energy index, which can be expressed as:

$$SSI = \sum_{i=1}^{N} x_i^2 \tag{E.1}$$

## E.2 Absolute Mean

Mean absolute value (MAV) is an intuitive feature extraction method, which often is used, in context of EMG signal analysis. When working with surface EMG signals for prosthetic limb control, the first method to extract features is MAV, also known as an onset index.

MAV feature is an average of absolute value of the EMG signal amplitude in a segment, and is therefore a time based features, which is be defined as:

$$MAV = \frac{1}{N} \sum_{i-1}^{N} |x_i| \tag{E.2}$$

## E.3 Zero Crossing

Zero Crossing (ZC) is an extraction method which aims at extracting frequency information, from the the time domain, for a given EMG signal. This information is extracted by counting, the number of times the amplitude value of the EMG signal crosses zero amplitude. To make sure low voltage fluctuations or background noises are not included in the counting, a Threshold condition is implemented. The calculation is defined as:

$$ZC = \sum_{i=1}^{N-1} [sgn(x_i \times x_{i+1}) \cap |x_i \times x_{i+1}| \geq threshold] \tag{E.3}$$

$$sgn = \begin{cases} 1, & \text{if } x \geq threshold \\ 0, & \text{otherwise} \end{cases} \tag{E.4}$$

## E.4 Waveform Length

Waveform length (WFL) is a method, where the focus is measuring the complexity of the EMG signal. By looking at the cumulative length of the EMG waveform over the time segment, it is possible to calculate the complexity. This feature can be calculated by:

$$WFL = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \tag{E.5}$$

## E.5 Wilson Amplitude

The feature Wilson amplitude is similar to Zero crossing, since it measures frequency information, form the time domain. It is defined as the amount of times the change of EMG signal between two following samples exceeds defined threshold. Mathematically, this feature can be described like this:

$$WAMPl = \sum_{i=1}^{N-1} [f(|x_n - x_{n+1}|)] \tag{E.6}$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq threshold \\ 0, & \text{otherwise} \end{cases} \tag{E.7}$$

## E.6 Slope Sign Change Modified

An extraction method, which similarly to ZC and WAMPL is used to describe frequency of the signal. Slope sign change (SSC) represents the number of times the EMG signal slope changes sign. Mathematically, Slope sign change can be expressed like this:

$$SSC = \sum_{i=2}^{N-1} f[(x_i - x_{i-1}) \times (x_i - x_{i+1})] \tag{E.8}$$

The modified version applied in this project works differently. It simply differentiates the signal and and count zero crossings of the differentiated signal. This means that the modified version counts every place where the signal has a horizontal tangent, thus not only the local maximums and minimums but also the inflection points.
From the equation it is apparent that Slope Sign Change investigates changes between positive and negative slopes among three sequent samples. Of course, the registered changes are compared to estimated threshold in order to eliminate background noise.

$$f(x) = \begin{cases} 1, & \text{if } x \geq threshold \\ 0, & \text{otherwise} \end{cases} \tag{E.9}$$

## E.7 Myopulse Percentage Rate

Myopulse is defined as, the number of times the absolute amplitude value of an EMG signal exceeds a defined threshold. The myopulse percentage rate (MPR) is the average of the myopulse percentage rate. The mathematical description of MPR is defined as the following:

$$MPR = \frac{1}{N}\sum_{i=1}^{N}[f(x_i)] \tag{E.10}$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq threshold \\ 0, & \text{otherwise} \end{cases} \tag{E.11}$$

## E.8 Root Mean Square

Root mean square (RMS), is a time domain feature, which analyses the square root of the average squared data. RMS is defined as the following:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{E.12}$$

## E.9 Standard Deviation

Standard deviation (STD) shows how spread the data is. A higher number means that the data is more spread, while a lower number represents that the data is closer to the mean. STD is defined by the following equation:

$$STD = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2} \tag{E.13}$$

## E.10 Median

Median (MD) is a feature based on statistic, from the time domain, analysing the power index of a give EMG signal. Median is defined in the following equation for even number of observations:

$$MD = \frac{x_{\frac{n}{2}} + x_{\frac{n+2}{2}}}{2} \tag{E.14}$$

while, the following is for odd numbers of observations.

$$MD = x_{\frac{n+1}{2}} \tag{E.15}$$

## E.11 Variance

Variance (VAR) is a time domain feature, which statistically analyses the power index of a given EMG signal. Variance is defined in the following equation:

$$VAR = \frac{1}{N-1} \sum_{i=1}^{N} x_i^2 \tag{E.16}$$

## E.12 Log Detector

The feature log detector(LOG), is a non linear detector which estimates the muscle force. The mathematical description is the following:

$$LOG = e^{\frac{1}{N} \sum\limits_{i=1}^{N} log|x_i|} \tag{E.17}$$