1 | Introduction

The loss of any part of a limb or limb as a whole is a great cost for any human. The hand is one of the most precious tools humans have and thus a loss of this would prove to be a great loss of functionality and independence. In an effort to restore some of that ability and autonomy, many patients are provided with prosthetics.
In recent years, prosthetics have become exceedingly good in performance, however, lack of functionality and discomfort of prosthetics are causing patients to reject the provided prosthesis. [**Reiber2010**]. Commercial available prosthetics range from passive cosmetic prosthetics to functional low degree of freedom (DOF) cable-driven prosthetics and switch controlled myoelectric prosthetics.
In recent years several complex multi DOF prosthetic hands have been developed. Examples of this are the the Vincent hand by Vincent Systems, iLimb hands from Touch Bionics, the Bebionic hands from RSL Steeper and the Michelangelo hand from Otto Bock [**Belter2013**]. Despite the efforts to advance and improve the functionality of prosthetics, a critical bottleneck still exist: the ability to properly control the prosthetic [**Hwang2017**]. The general challenge for users is to be able to consistently produce distinguishable muscle patterns, for the prosthetic control system to recognize. [**Powell2014**]
Most commercially available myoelectric controlled prosthetics rely on switch control which is a robust control scheme, but is slow and non-biologic in movement. In the research area of myoelectric prosthetics newer control schemes have been developed. These control schemes are classification and regression. Classification have been used for many years in research but is to date only used in one commercially available prosthetic. When using classification as a control scheme the classifier attempts to classify similar patterns in electromyography (EMG) signals between previously acquired data and new data [**Mendez2017**]. The regression control scheme determine the output signal for a input based on a regression line. This provide a continuous output value contrary to classification which provide a single value. [**Hahne2014**]
Both types of control schemes have become exceedingly good at correctly estimating muscle patterns. [**Hahne2014**, **Bruun2017**, **Englehart2003**, **Scheme2015**] However, there still exist a challenge for the users to be able to consistently produce distinguishable muscle patterns [**Powell2014**]. In resent years many advancements have been made in research on system training. System training is the training of the control system to recognize the input signals from the user [**Fougner2012**]. This area focus on the design of the hardware and software side of the system in EMG prosthetics. Jiang et al. [**Jiang2012**] argue that a change should be made in the focus of research on myoelectric prosthetics in relation to improving control. The awareness in the research area show a very single-minded approach to possible improvements of control, and thus mainly system training have been researched. Jiang et al. [**Jiang2012**] discuss that the awareness of possible other practical implementation have been underestimated. One such implementation which have been addressed in only a few studies is user training [**Fang2017**, **Powell2014**, **Pan2017**]. Contrary to system training, user training focus on the user's ability to control a prosthetic [**Fougner2012**]. User training differ from regular use of a prosthetic in that the training is part of the initial period, where the system is being adjusted to the individual user. Here different types of feedback can be used to inform the user on how well it performs movement or how well the system recognizes the users performed movements. [**Powell2014**, **Simon2013**]
In a 2014 study Powell et al. [**Powell2014**] provided the user with real-time visual feedback of a virtual prosthetic. This type of feedback is similar to the visual feedback a prosthesis user would receive using a normal prosthesis, albeit without the sensory feedback of the weight of the prosthesis. Pan et al. [**Pan2017**] provided a visual feedback of an arrow to be moved on a 2D plane. The arrow was controlled by two DOF's; one controlled the horizontal position of the arrow, while the other could rotate the arrow [**Pan2017**]. Fang et al. [**Fang2017**] provided real-time visual feedback of subjects performed movement in relation to the classes defined in the system. The feedback visualized a map of clusters of different classes which subjects could match the position of a cursor to. When subjects could match the cursor to the centroid of a cluster the performed movement corresponded the best with the class of that movement. [**Fang2017**] All studies observed an improvement in user performance after being exposed to focused

user training with visual feedback.

Studies investigating the effect of user training shows promising results, however, as Jiang et al. [**Jiang2012**] discuss the myoelectric prosthetic research area might have been too focused on system training in resent years and could overall benefit from an expansion of research interests to include previously underestimated implementations or completely new approaches.
A 2013 study by Scheme et al. [**Scheme2013**] proposed a novel approach of utilizing confidence-based rejection to improve system training of myoelectric control. Here a classification control scheme was provided with confidence scores to assist in acceptance or rejection of the class output. The confidence scores were calculated from a modification of Bayes' theorem. Scheme et al. [**Scheme2013**] showed a significant improvement in performance with the use of the rejection-capable system when compared to the normal classification scheme. A similar approach could be used in user training by providing the confidence scores of the classification to the user as a form of visual feedback.
Thus, this study propose a novel method of providing users with feedback containing confidence scores for different classes in a classification control scheme during user training. Contrary to current feedback methods in user training this approach could enable users to better understand how the classification works based on their performed movements. During user training this could improve the way users perform specific movements in order to enable the system to better recognize and classify movements correctly.

2 | Background

2.1 Anatomy of the human lower forearm

This project will use six movements for control of a virtual interface and visual feedback. The movements are extension, flexion, radial and ulnar deviation, closed and opened hand as well as rest. The movements are shown in figure 2.1. These movements are either individually or in combination most often used during many daily life tasks. Thus training users to improve performance of these movements for use in a myoelectric prosthesis control scheme, would prove to cover most functions a prosthesis should cover for use in daily life tasks.

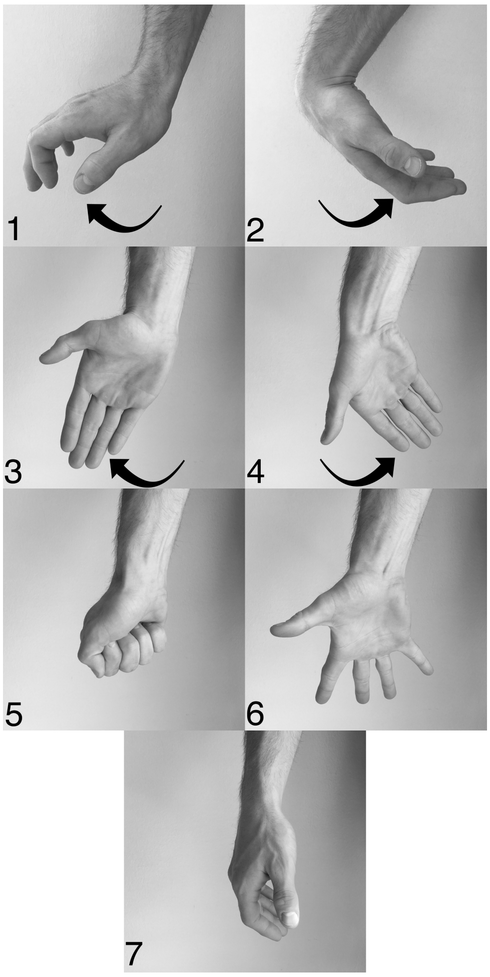


Figure 2.1: The figure shows the six hand movements used in this study as well as rest. The movements are: 1) extension, 2) flexion, 3) radial deviation, 4) ulnar deviation, 5) closed hand, 6) opened hand and 7) rest.

The human arm has five DOFs being covered by the movements extension, flexion, abduction, adduction and rotation at the shoulder, flexion at the elbow and supination and pronation of the lower forearm.

The human hand is a very versatile and dexterous apparatus possessing a total of 25 active DOFs. These DOFs are expressed at the fingers, wrist and palm, where the fingers account for a total of 21 DOFs and the wrist account for two DOFs. Each finger posses four DOFs, while the thumb has five, also being able to do opposition movement to each other finger. At the wrist the hand can flex and extend as well as perform radial and ulnar deviations. The last two DOFs exist in the palm of the hand where the joints between the fourth and fifth metacarpal bones and the hamate bone of the carpal bones in the wrist give the hand movement along two additional axes which is used when doing opposition of the thumb and the fifth finger. [**Martini2012**]

Many of these DOFs and corresponding movements are active when using the hand. The chosen movements for this project will involve most of the DOFs in the hand, as can be seen on figure 2.1. Movements at the wrist (extension/flexion, radial/ulnar deviation) are DOFs of the hand. Even though many individual movements of joints and DOFs of the hand and fingers are active during opening and closing of the hand. During the rest of the project these two movements will be described as being movement in one DOF.

2.1.1 Muscles of the lower forearm

To perform movements of the hand and fingers and at the wrist, muscles in the lower arm are active. All movements relevant for this study are controlled by muscles in the lower forearm, thus it is relevant to gain knowledge on the muscles in the lower forearm. When performing actions at the wrist the following muscles are active: flexor carpi radialis, flexor carpi ulnaris, palmaris longus, extensor carpi radialis longus, extensor carpi radialis brevis, extensor carpi ulnaris. The flexor and extensor muscles are naturally responsible for performing flexion and extension respectively at the wrist. They are however also responsible for performing radial and ulnar deviation, where the flexor carpi radialis and extensor carpi radialis brevis muscles, which are antoganistic muscles when doing flexion and extension, will work together when performing radial deviation. The extensor carpi radialis longus muscle is also responsible for performing radial deviation. The flexor and extensor carpi ulnaris muscles are responsible when performing ulnar deviation. The palmaris longus muscle is only active during flexion at the wrist. [**Martini2012**]

Several more muscles are further specified to perform movements of the fingers but many of these are also active during extension/flexion and radial/ulnar deviations at the wrist. Muscles responsible when opening the hand by extending the fingers are: extensor digitorum, extensor pollicis brevis, extensor pollicis longus, extensor indicis and the extensor digiti minimi muscle. Contrary, the muscles responsible for closing the hand by flexing the fingers are: flexor digitorum superficialis, flexor digitorum profundus and the flextor pollicis longus. [**Martini2012**]

2.2 Electromyography

This project will utilize the method of electromyography to record the muscle activation of the lower arm muscles in relation to the gestures presented in section 2.1 on anatomy. To develop theoretical background knowledge, a short introduction of the essentials of the signal will be presented.

Electromyography is the recording of muscle activity based on the amount of electrical stimulation. The amount of activity is found by measuring the electric potential, an action potential causing a muscle contraction. The process of planning and executing a voluntary movement starts at the motor cortex in the brain, where a nerve impulse is send and travel through the spinal cord to the lower motor neuron. As seen in figure 2.2 the path from alpha motor neuron through the axon to the motor endplates is what makes up a motor unit. The alpha motor neuron originates from the spinal cord along the axon to the muscle it controls. The axon branches out to multiple muscle fibers through motor endplates innervating the muscle fibers.

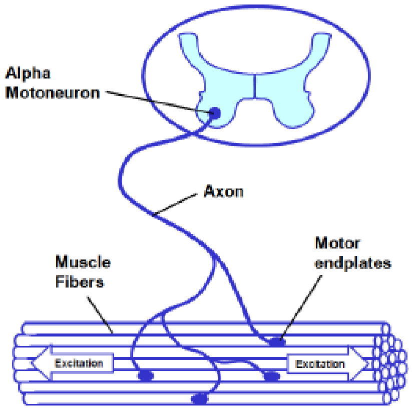


Figure 2.2: The figure illustrates the neural pathway from the alpha motor neuron to the innervated muscle fibers, making up a motor unit.[**Konrad2005**]

The essentials of understanding the application EMG is the excitation of muscle cells. The excitability of the muscle fibers play a crucial role in a muscle contraction. The mechanisms of a contraction can be understood through a series of events. First the muscle cell membrane is at a resting potential between -80 to -90 mV, due to an equilibrium of NA^+ and K^+ through the intracellular and extracellular side of the membrane, maintained by an ion pump. The before mentioned alpha motor neuron reaches the motor endplates where a transmitter substance is released. The substance alters the membrane characteristics and allows a greater flow of Na^+ into the cell. This causes a membrane depolarization, changing the membrane potential. If a threshold between -55 mV to -50 mV is reached excitation in the form of an action potential is formed, travelling in both directions of the muscle fiber, as seen on figure 2.2. The membrane potential is quickly restored with a great outflow of Na^+, resulting in a repolarization. The action potential from each of the activated muscle fibers summates spatially and temporally forming a motor unit action potential (MUAP). The spread of the MUAP over the muscle membrane is recorded with EMG. The number of recruited motor units is a way of controlling the force of a muscle contraction depending on the force needed. Like the recruitment of motor units, the frequency of activation can be modulated for generating a specific amount of force. However, as different muscles recruit and activate muscle fibers at different frequencies, the amplitude and frequency visible from an EMG recording is necessary correlated with the generated muscle force. In EMG it is the sum of activity of activated motor units that is recorded. [**Cram2012**] In the scenario of this project multiple EMG electrodes will record signals from many muscles in the lower forearm. This will result in some muscles being very active during some movements as they contract, while other muscles will be inactive, as described in section 2.1.1 on antoganistic muscles. On an EMG recording this will be visible as contracting muscles will show high activity, while others will show little to no activity. An illustration of this can be seen on figure **??**. The number of motor units innervating muscle fibers depend on the muscle characteristics and the purpose is serves. A low innervation ratio between motor units and muscle fibers gives the opportunity of fine motor tasks, while a low ratio is ideal for tasks demanding strength. Furthermore the motor units are recruited in an asynchronous pattern. This further facilitates the possibility of smooth muscle movements. [**Martini2012**, **Cram2012**]

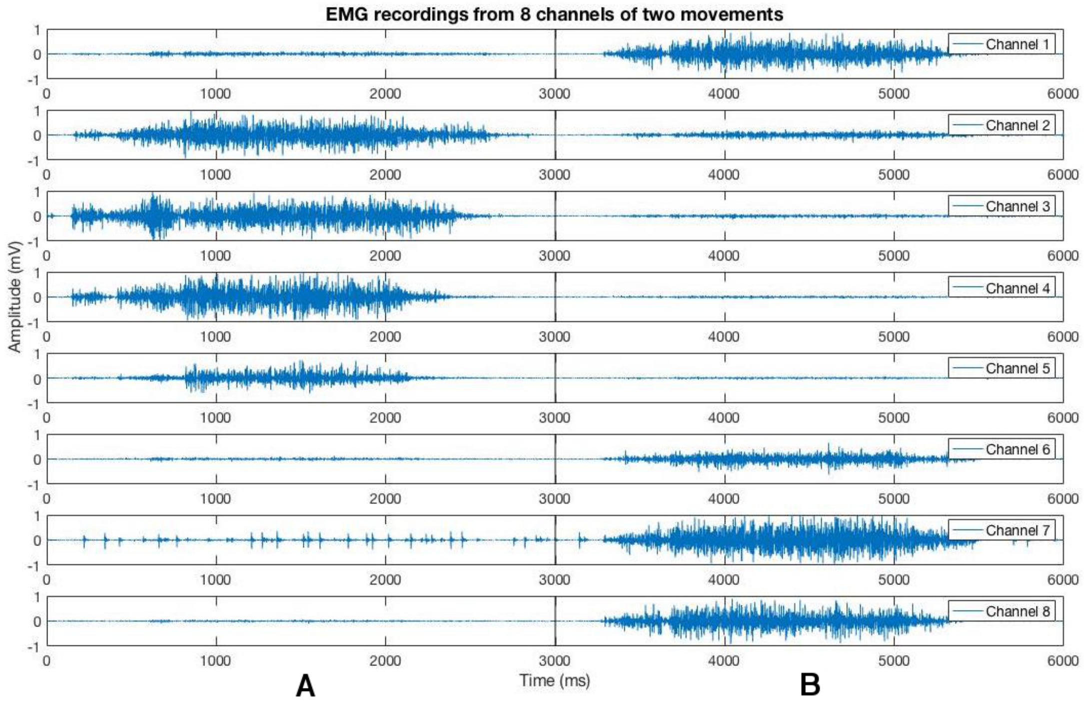


Figure 2.3: Illustration of the activity in an EMG recording of two movements: extension and flexion. Left side A) shows the activity recorded by EMG electrode channels during extension of the hand. Right side B) shown activity during flexion of the hand.

Recording EMG can be done either through the most often used surface EMG (sEMG) or by intra vascular EMG (iEMG). In iEMG a needle is inserted into the muscle measuring the MUAP directly on site. The more often used sEMG uses electrodes measuring the MUAP on the skin surface and will be used to acquire EMG signals in this project. [**Cram2012**]

2.3 Data acquisition

For a myoelectric prosthetic control system to be able to recognize hand movements it needs to be given prior information on how the movements looks like represented as a EMG-signal - this is also called training the control system. Thus, EMG data needs to be acquired from the user and used to train the control system. The following section describes which types of EMG acquisition techniques that are commonly used.

As presented earlier in section 2.2 the source of the EMG signal is motor unit action potentials. The energy generated in action potentials is of a very small size and is measured in microvolts. Very sensitive recording equipments is therefore key in doing electromyography. It is essential to consider the type of electrode intended to use. Electrodes come in various different sizes and shapes and are therefore very depended on the intended measurement site. Typically electrodes made of silver-impregnated plastic are used. They present desired characteristics by being disposable, relatively low price and by having low impedance with the skin. Most electrodes are covered with an adhesive compound in order for them to stick to the skin. These can either be 'dry' or covered with different types of gel, in order to reduce impedance and thereby noise, getting a more accurate EMG recording. Dry electrodes do not use gel, but instead rely on the skin to sweat and thereby decreasing the skin impedance. Dry electrodes should prove better to patients with sensitive skin. Different skin conditions may also effect the electrode-skin impedance. Make-up, scale or much hair increase the impedance, thus the recording site should be prepared by removing hair or cleaning the skin with alcohol wipes. [**Cram2012**] The following section will introduce the choice of acquisition device used in this project.

2.3.1 Myo armband

In this project the Myo armband (MYB) from Thalmic Labs will be used for EMG data acquisition. MYB is an electrode armband with eight dry stainless steel electrode-pairs around the inside of the armband, as depicted in figure 2.4. The advantage of dry electrodes is that they do not need to be disposed after usage as conventional gelled EMG-electrodes [**Cram2012**]. In addition, MYB can communicate wireless to a computer via Bluetooth 4.0 [**Myoarmband2013**]. Thus, it is an easy and non-time-consuming device to use both during pilot-testing and for the final experiment. In the following section more information about the MYB will be presented.

MYB records EMG data in a unitless 8-bit resolution. As usual when recording EMG the higher the performed contraction is, the higher the values in the output will be. To avoid interference from power lines a 50 Hz notch filter is built-in in the MYB from the manufacturer. However, the MYB is not able to make any further filtering, therefore this will be implemented later during signal processing described further in section 2.4. The MYB has a 200 Hz sample rate, and thus samples with a lower resolution than the EMG spectrum consists of, which is between 10-500 Hz [**Cram2012**]. Using MYB will likely result in an aliased EMG signal and confinement in using features representing the frequency information the signal. In section 2.4.2, different techniques to counter the negative effect of low-resolution EMG will be proposed for the implementation. Besides the EMG sensors the MYB can provide position and orientation information, using its three inertial measurement units consisting of a three axis gyroscope, a three axis magnetometer and a three axis accelerometer. This inertial information is sampled at 50 Hz. [**Myoarmband2013**]
When initiating the wearing of the armband there is two calibration phases the user must follow before the armband is ready to use - the warm-up phase and the sync phase. During the warm-up phase the armband is ensuring as strong electrical connection with the muscles in the forearm as possible. This is mainly provided by light sweating on the skin under the electrodes, which improve the connection similar to electrode gel [**Cram2012**]. During the sync phase, the armband determines its orientation in space, position and on which arm it is placed. The MYB works most optimal when fitted tightly on the thickest part of the forearm. For users with smaller forearms a set of clips can be added for the armband to get a constrained grip. [**Myoarmband2013**]

Figure 2.4: MYB from Thalmic Labs. The number on each channel indicates which channel corresponds to which column in the digital output, when acquiring data from MYB.

2.4 Data processing

In order to use the acquired EMG-signal in myoelectric prosthesis control, it first has to be processed, this is refereed to as pre-processing. Since the acquisition and most processing is done in the MYB before Bluetooth transmission, further processing of the signal is moderate. In myoelectric prosthesis control features are extracted for use in control, instead of using the entire EMG-signal. Hereby the amount of information is reduced resulting in faster computational speed. The following two sections will briefly describe theory behind filtering and feature extraction in relation to this project.

2.4.1 Filtering

Filtering is a cornerstone in preparing an EMG-signal for any kind of use. The frequency spectrum of EMG is 10 Hz to 500 Hz and most electrodes has a working range of 0 Hz to 500 Hz [**DeLuca2010**]. According to the Nyquist theorem, to achieve a loss-less representation of the signal the sampling frequency must be at least twice the maximum frequency of interest of the original signal [**Pozzo2004**]. Besides sampling with twice the maximum frequency, EMG is sensitive to artifacts of movement and electrical interference. Due to these circumstances, filters are often implemented to remove these unwanted contributors [**DeLuca2010**]. General practice in filtering the EMG-signal will include implementing a notch filter with very narrow width and steep slope, at frequencies 49-51 Hz or 59-61 Hz depending on the power supply. The intend is to remove any electrical interference noise. In the low frequency spectrum several recommendations (5 Hz, 10 Hz and 20 Hz) has been made for optimal corner frequency of a high pass filter, to remove noise. A low pass filter is also typically used to remove any noise and unwanted signal above 500 Hz [**Cram2012**].

This project will utilize a MYB for data acquisition and as mentioned in section 2.3.1 the MYB has a sample rate of 200 Hz. In relation to this project a sampling frequency of at least twice the maximum of the recorded signal is not possible, since muscles of the forearm have a maximum frequency of 400-500

Hz [**Cram2012**]. This would require a sample rate of at least 1000 Hz, which cannot be achieved due to limitations in the MYB. The effect of the low sample rate of the MYB is aliasing in the recording, causing a frequency component not originally in the EMG signal. To account for this it would be resourceful to implement a low pass filter to act as an anti-aliasing filter.

2.4.2 Feature extraction

The raw EMG-signal is not itself used for myoelectric prosthesis control, but features that are extracted from it. Thereby reducing the amount of information limiting it to its most useful properties, resulting in faster computational speed.
There are numerous feature components from an EMG signal which can be extracted either from the time-domain, frequency-domain, or time-frequency domain. Most used are features from the time- and frequency-domain. Time-domain features can be categorized in five different types based on their mathematical properties: energy information, complexity information, frequency information, prediction modelling and time-dependency. Extracting features from the frequency-domain requires a frequency transformation, calculating the spectral properties of the recorded signal, which takes up longer processing time than simply using time-domain features. Time-domain features are often chosen based on their quick and easy implementation as they do not require any transformation before extraction and are calculated based on the raw EMG-signal. In addition, it is important not to choose redundant features for the classifier which would be to chose features providing similar information. [**Phiny2012**]

Extracting features for real-time prosthesis control is done by taking segments of the continuous signal, called windows. Calculation on extracting features are done in these discreet windows. This is done instead of using the instantaneous value due to the signals random nature. These windows are often overlapped to create a dense information stream for extraction. The relationship between window and overlap length is significant, when trying to determine the best representation. The window length is a matter of getting enough samples to do the calculation, but too long a window will result in delays slowing the control. Overlapping the window is a way to faster acquire windows by reusing a determined last segment of the prior window. Smith et al. [**Smith2014**] found that the optimal window length in a classification control scheme that enables best performance ranges from 150-250 ms.

2.5 Classification

For a myoelectric prosthesis to be able to distinguish between movements it needs to perform, a control scheme is needed to categorize the movements. The control scheme is trained by being given information about the EMG signal represented as the features extracted from the raw EMG. If the features between each movement are well separated the control scheme is able to recognize each distinct movement. For this purpose classification control schemes are commonly used. A classifier categorizes each movement as a class, and based on the input features it gives one class as output at a time. Using a classifier thus limits the user to only performing movements which have been defined as classes. However, if trained properly a classifier can reach a low error rate for the trained movements [**Scheme2013**]. A frequently used classification control scheme for myoelectric prosthetic control is the Linear Discriminant Analysis classifier (LDA) [**Englehart2003**, **Scheme2015**, **Fang2017**, **Scheme2013**]. The advantage of LDA is that whilst having a low computation time it still enables robust control. An assumption about the LDA is that the input needs to be Gaussian distributed, which the EMG probability properties has shown to adhere to [**Duda2000**, **Nazarpour2013**]. The following section provides further theoretical information about the LDA classifier.

2.5.1 Linear Discriminant Analysis

LDA is a supervised classification method used to separate classes of data by linear decision boundaries. Each decision boundary is a hyperplane from which the shortest distance to each class feature value and class mean is maximized for each class. A decision boundary is defined as a linear combination of the feature values x and is given as [**Duda2000**]:

$$g(x) = w^t x + w_0 \tag{2.1}$$

where w is a weight vector deciding the orientation of $g(x)$, and w_0 is a bias deciding the position of the hyperplane in relation to the origin. If $w_0 > 0$ the origin is on the positive side of the decision boundary, and if $w_0 < 0$ the origin is on the negative side. In the case of $w_0 = 0$ the decision boundary passes through origin. The distance from origin to the boundary is given as $\frac{w_0}{||w||}$. The position of the decision boundary is necessary to know to when separating features into regions. [**Duda2000**]

In a two category case the decision rule for deciding classes is to decide class w_1 if $g(x) > 0$ and class w_2 if $g(x) < 0$. $g(x) = 0$ then defines the decision boundary that separates the features into two decision regions R_1 for w_1 and R_2 for w_2. The normal vector w is orthogonal to any vector on the hyperplane, which is used to calculate the distance r from feature values (x) to the decision boundary [**Duda2000**]:

$$r = \frac{g(x)}{||w||} \tag{2.2}$$

The distance from origin and boundary to feature value (x) is needed to decide in which region the feature value belongs. [**Duda2000**] These distances are illustrated in figure 2.5.

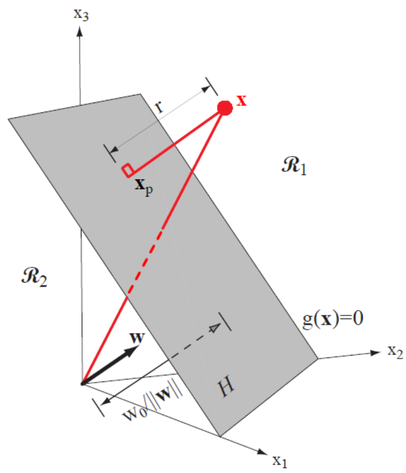


Figure 2.5: A geometric illustration of the linear decision boundary $g(x)$ that separates the feature space into two decision regions R_1 and R_2. x is the feature value, and x_p is the point on the decision boundary in which x is orthogonal projected on vector w. The distances from origin and boundary to feature value x is marked red. [**Duda2000**]

When feature values are to be classified into more than two classes more decision boundaries are needed. This is a multicategory case in which c numbers of boundaries are defined. When defining linear boundaries in this case any number can be chosen, but to minimize ambiguous decision regions the boundaries are defined by [**Duda2000**]:

$$g_i(x) = w^t x_i + w_{i0} \qquad i = 1, ..., c, \tag{2.3}$$

This equation follows the notation of the two-category case, with the addition of i numbers of boundaries, feature values and biases. This type of classifier is called a linear machine, dividing the feature space into c regions. A liner machine will be adopted as classification method in this project. Regions R_i and R_j, that are connected is divided by a boundary hyperplane H_{ij} defined by [**Duda2000**]:

$$g_i(x) = g_j(x) \tag{2.4}$$

Often regions are contiguous and will have a single boundary to separate several regions. [**Duda2000**] Illustrations of this case can be seen on figure 2.6.

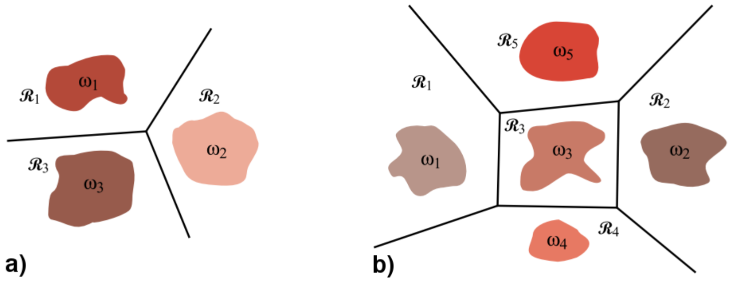


Figure 2.6: A three class (a) and five class (b) case each respectively separated by one decision boundary linear machine. [**Duda2000**]

When the decision boundaries $g_i(w)$ have been calculated as in equation (2.3), the input feature values can be decided upon which class they belong to by calculating the distance to the decision boundary as in equation (2.2).

2.5.2 Classification confidence scores

An additional reason for using LDA as control scheme is because it enables the calculation of confidence scores for the classes, which will be used in the user training described in section 2.7 to improve the prosthesis control. Based on the classification of feature values by the linear machine, confidence scores for the classes can be evaluated by computing the posterior probability of each class. Calculating the posterior probability is possible by knowing the likelihood $P(x|w_j)$ and the prior probability $P(w)$. The posterior probability for a class is a value between 0 and 1, and is calculated as follows:

$$P(w_j|x) = \frac{P(x|w_j)P(w)}{P(x)} \tag{2.5}$$

where w_j represents a class and x represents a feature value. The posterior probability is given as the product of the class conditional probability, $P(x|w_j)$ and the prior probability $P(w)$ divided by a normalization term $P(x)$ that guaranties that the posterior probabilities for all classes sums to one. $P(x|w_j)$ is the probability of obtaining a feature value when selecting samples randomly from a class. $P(w)$ is the probability of a sample from a specific class appears in its correct class, before it have actually appeared. Summation of posterior probabilities for all classes will equal 1.

2.6 Linear regression methods

Classification can be used together with regression methods to provide a combination of the two control scheme. The output from the LDA classifier can be set to only decide on which movement is performed, and not at which contraction level the muscles used in the given movement are performing. Thus, the prosthesis can not perform any movement. In statistics linear regression is often used to determine relations between variables. This notion can also be applied for myoelectric prosthetic control. While classification only provides an output on which class is recognized, a linear regression model provides a continuous output value based on the input value. If the regression model is fitted with information on different contraction levels for a given movement, control proportional to the contraction level will be achieved [**Hwang2017**, **Hahne2014**, **Bruun2017**]. In the overall control scheme the classifier can then be used to decide which movement is performed, and a regression model can decide at which contraction level the movement is performed at. Similarly as with the classifier, regressors needs to be trained based on data acquired from the user, where the features extracted from the raw EMG signal is used as input. This procedure is described in the following section.

Different models of linear regression exist to account for different uses. When utilizing regression methods it must be considered which type of input variables are used and what type of relation these variables might have. The appropriate regression must then be applied. Simple linear regression approximate a relation between one dependent variable Y and one independent variable X [**Zar2009**]:

$$Y = \alpha + \beta X + \epsilon \tag{2.6}$$

where Y is the control output for the prosthesis, X is the feature extracted from the EMG signal, β is the regression coefficient in the sampled population, ϵ is the error, and α is the predicted value of Y at $X = 0$. This model can be expanded to estimate relations between one dependent variable and several independent variables. This is called multivariate regression and expands on the equation of simple linear regression, given in equation (2.6) [**Zar2009**]:

$$\hat{Y} = \alpha + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_i X_i + \epsilon_i \tag{2.7}$$

where i in this project would correspond to the number of channels in the MYB [**Zar2009**]. Since this regression model approximates the relation between several independent variables and one dependent variable, this model can be used as a control scheme in myoelectric prosthetics, and . Here the channel-recordings of muscle activity can be considered independent variables, and used to estimate one control output, which would be the dependent variable. [**Bruun2017**]

2.7 User training

As user training in relation to prosthetic control is the main focus of this project an understanding of this concept in relation to receiving a prosthetic device is of great importance. Therefore the following section will cover an introduction to the concept of user training and its importance when preparing a subject to receive a prosthetic device. In addition some of the prior techniques of conducting user training will be presented, facilitating the possibility of assembling a user training protocol based on the most recent and cutting edge results.
When fitting an amputee with a prosthesis, the way the prosthesis is controlled is important. A lot of work lies both ahead and behind fitting a person with a prosthesis. When developing and manufacturing a prosthesis two concepts emerge, one being system training and the other being user training. System training is training the control system to be able to recognize and differ movements based on the

EMG-signal being fed to the system. [**Fougner2012**] User training on the other hand focuses on training the user in performing distinguishable movements which can be recognized by the control system. Here different types of feedback can be used to inform the user on how well the user performs a movement or how well the system recognizes the users performed movements. [**Powell2014**, **Simon2013**]
Only few studies have earlier explored the optimal way of giving visual feedback in user training [**Jiang2012**]. In a 2014 study Powell et al. [**Powell2014**] provided the user with real-time visual feedback of a virtual prosthetic. This type of feedback is similar to the visual feedback a prosthesis user would receive using a normal prosthesis, although without the sensory feedback of the weight of the prosthesis. Pan et al. [**Pan2017**] provided a visual feedback of an arrow to be moved on a 2D plane. The arrow was controlled by two DOF's; one controlled the horizontal position of the arrow, while the other could rotate the arrow [**Pan2017**]. Fang et al. [**Fang2017**] provided real-time visual feedback of subjects performed movement in relation to the classes defined in the system. The feedback visualized a map of clusters of different classes which subjects could match the position of a cursor to. When subjects could match the cursor to the centroid of a cluster the performed movement corresponded the best with the class of that movement. [**Fang2017**] All studies observed an improvement in user performance after being exposed to focused user training with visual feedback.
A 2013 study by Scheme et al. [**Scheme2013**] proposed a novel approach of utilizing confidence-based rejection to improve system training of myoelectric control. Here a classification control scheme was provided with confidence scores to assist in acceptance or rejection of the class output. The confidence scores were calculated from a modification of Bayes' theorem. Scheme et al. [**Scheme2013**] showed a significant improvement in performance with the use of the rejection-capable system when compared to the normal classification scheme. A similar approach could be used in user training by providing the confidence scores of the classification to the user as a form of visual feedback.

2.8 Validating Performance

Measuring the performance of achieved prosthesis control cannot be seen as a trivial task, and different approaches can be used. The achieved performance can be measured by affixing a prosthesis on to the test subject and validate performance hereby. Often, like the current project, the subjects do not consist of actual amputees but instead healthy subject. In these cases the performance validation is done by implementing a virtual test environment where the subjects is to control an object on the computer screen by performing movements. The following section will further elucidate the procedures of such a virtual test for validating prosthetic control.

2.8.1 Modified Fitts' Law

Fitts' law task is a common method of quantifying performance of movements, first proposed by Paul M. Fitts in 1954 [**Fitts1954**]. Originally, the only output of a Fitts' law task was the throughput, as given by equation (2.8). A modified Fitts' law task designed for a virtual 2D and 3D target acquisition task has later been used by [**Kamavuako2014**] and [**Scheme2013**] respectively. Here, four additional metrics were added in an online task, where a virtual computer cursor was used to represent the control output [**Scheme2013**, **Kamavuako2014**]. The four additional metrics, path efficiency, overshoot, stopping distance and completion rate, were made by [**Poulton2013**] and [**Simon2011**]. While the throughput measure from the conventional Fitts' law task is usable, it does not cover all aspects of the control required to complete a task. The additional four measures were added to quantitatively assess performance of naturalness, spontaneity, and compensatory motions during use. The total five proposed performance measures in assessing myoelectric control are [**Scheme2013a**]:

Throughput (TP) which represents the trade-off between speed and accuracy. TP uses the relationship of time taken to reach a certain target in seconds (MT) and the index of difficulty (ID). This forms: [**Scheme2013**, **Fitts1954**]

$$TP = \frac{1}{N} \sum_{i=1}^{N} \frac{ID_i}{MT_i} \tag{2.8}$$

where i is a specific movement and N is the total number of movements. ID relates to the target distance D and width W. The ID for each task, from the origin to a specific target of a certain size is calculated using [**Scheme2013**, **Fitts1954**]:

$$ID = log_2(\frac{D}{W} + 1) \tag{2.9}$$

Path Efficiency (PE) describes the quality of control by making a measure of the straightness of the cursor's path to the target, by making a ratio of the actual path distance versus the optimal path distance. This tests the users ability to continuously control the cursor position. Following the optimal path will result in a PE of 100%. PE is calculated as follows [**Scheme2013**, **Poulton2013**]:

$$PE = \frac{Optimal\ Distance}{Actual\ Distance} \tag{2.10}$$

Overshoot (OS) is the number of times the cursor enters and then leaves the target before the dwell time inside the target is reached, across all target in the task, divided by the total number of targets. OS tests the users ability to control the velocity of the cursor accurately. A perfect OS-score of zero is reached if the cursor dwells within the target boundaries on the first try for all targets, and is calculated as the following [**Scheme2013**, **Poulton2013**]:

$$OS = \frac{Total\ Number\ of\ Overshoots}{Total\ Number\ of\ Targets} \tag{2.11}$$

Stopping Distance (SD) describes the users ability to rest and thereby perform no movement. The SD measure is the distance moved during the dwell time across all targets, and is given as [**Scheme2013**]:

$$SD = \sum_{i=1}^{N} (Distance\ Inside\ Target)_i \tag{2.12}$$

where i is a reached target and N is the total number of reached targets.

Completion Rate (CR) describes the percentage of targets reached within the total allowed time. This gives a general idea of the user's performance, and is calculated as [**Scheme2013**, **Simon2011**]:

$$CR = \frac{Number\ of\ Reached\ Targets}{Total\ Number\ of\ Targets} \tag{2.13}$$

2.9 Data separability

Besides evaluating the user performance in real-time, it would be beneficial to evaluate the clustering of feature values used to fit the classifier and regressors off-line. This will provide information on how the clusters between classes separates and how the feature values within clusters bundle. By doing this between sessions it can be evaluated how distinguishable the clusters are to judge how well the classifier and regressors will discriminate between the different clusters. For this purpose a Principle Component Analysis (PCA) will be utilized to reduce the dimensionality of the feature space, from which is subsequently can be used to calculate the distance between clusters and distances from feature values to centroids within clusters. The following section provides theoretical information on the PCA procedure.

2.9.1 Principal Component Analysis

PCA is used to express a set of possibly correlated variables into uncorrelated components, called principal components (PC). A dataset of many variables can thus be expressed in a reduced dimensionality hyperspace using less variables that are the most defining for the given dataset. Each principal component is orthogonal on the former and are uncorrelated and have zero covariance. They each define the largest variance in an axis, such that the first PC describes the direction of the maximum variance of the dataset. Each following PC describes the next highest variance of the dataset, with the constraint that it is orthogonal and has zero covariance with any of the former PCs. [**Semmlow2014**] PCA is the orthogonal projection of data onto a lower dimension linear space. A PC is found by minimizing the variance by projecting the feature values, the red dots in figure 2.7, onto the line describing the highest variance in the data set (purple line) as seen on figure 2.7. The PC (purple line) is found by minimizing the mean square distance between the data points. [**Semmlow2014**]

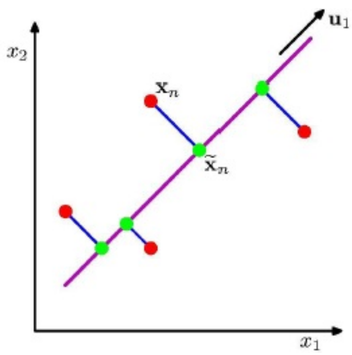


Figure 2.7: Projection of feature values (red dots) onto PC axes (purple).

The algebraic method of calculating the PCs can be done by using Singular Value Decomposition (SVD). The first step is to compute the squared cross product matrix of variances and covariances among every pair of the variables in the data set, where the diagonals are the variances and the off-diagonals are the covariances, as done in the following equation [**Semmlow2014**]:

$$S = X'X \tag{2.14}$$

Where S is the cross product and X is the feature set matrix. When finding the PCs it includes an eigen-analysis of S. The eigenvalues of are solutions to the following equation [**Semmlow2014**]:

$$|S - \lambda I| = 0 \tag{2.15}$$

Where λ is the variance of each PC and I is the identity matrix. After solving for λ the eigenvectors can be solved through the following equation [**Semmlow2014**]:

$$det|S - \lambda I|b_i = 0 \tag{2.16}$$

Where b_i is used to calculate the eigenvectors as in [**Semmlow2014**]:

$$u_i = \frac{b_i}{\sqrt{b_i' b_i}} \tag{2.17}$$

Where u_i is the i^{th} number of eigenvectors. The number eigenvectors equal the dimension size of the original feature space. The SVD orders the eigenvalues by size, so that $\lambda_1 > \lambda_2 \ldots > \lambda_i$. The scores for each PC is equal to the corresponding eigenvalue for that exact axis. The eigenvalues describe how much of the variance is accounted for by the associated PC. Summation of all eigenvalues accounts for the total variance of the data set; this is called the trace. To find how much the each PC accounts for, the eigenvalue of that PC is divided by the total variance: $\%\ of\ total\ variance\ = \frac{\lambda_i}{Trace}$. Deciding how many PCs the feature space should be reduced to, by setting a threshold of how much of the total variance should be preserved. [**Semmlow2014**]

2.9.2 Distance measure

After reducing the dimensionality of the original feature set the clusters can be analysed. For the purpose of measuring distances between and within clusters the centroid of each cluster must be calculated, as in equation (2.18):

$$C = \left[\frac{[x_1 + x_2 + \ ... \ + x_i], [y_1 + y_2 + \ ... \ + y_i], \ ... \ , [k_1 + k_2 + \ ... \ k_i]}{k} \right] \tag{2.18}$$

Where C is the centroid, i is the number of feature point in a dimension and k is the number of dimensions. To calculate the distance between centroids of clusters the Euclidean distance (ED) is computed. The ED is the length of the a line segment connecting points, in this case in form of two centroids p and q. The calculation of ED in a k-dimensional space is as in equation (2.19):

$$ED(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 \ + \ ... \ + (p_k - q_k)^2} \tag{2.19}$$

When calculating the distance from feature values in a cluster to their corresponding centroid the ED is computed likewise. To get a general impression of the distance from the feature values constituting the cluster to the centroid of the cluster the average of the distances is calculated.

from feature values of a cluster to their centroid, the average distance from all feature values to their centroid is calculated.

2.10 Statistical analysis

When evaluating the scores obtained in the performance test a statistical analysis is used. For this project a Friedman's test will be utilized due to a small sample population, and because it is assumed that the probability distribution of the performance scores is unknown [**Zar2009**]. Friedman's test is commonly used to test if different treatments have similar or different effects in a subject population. In the case of this project it is used to test whether performance in prosthesis control is similar or different across sessions in a subject population and if performance scores differ or bear resemblance between two subject populations. In the following section, theory on how the Friedman's test is calculated will be presented.

2.10.1 Friedman's test

The aim for the Friedman's test is to calculate the Friedman's F value to compare it with its corresponding critical F value to finally decide if the null-hypothesis or an alternative hypothesis should be accepted. The null-hypothesis expresses a relationship between measurements (the means of measurements are equal) and the alternative hypothesis expresses no relationship between measurements (the means od measurements are unequal). When testing a subject population of a given number of measurements the measurements must first be arranged in columns, where each column corresponds to a certain measurement, and each row corresponds to one subject; this row is also called a block. Each block must then be ranked separately, where the smallest number is ranked 1. If numbers in a row are equal they get the mean of the rank they would have received. The sum R of each column is then calculated, and the number of measurements k and number of subjects n are used to calculate the Friedman's F value in the following equation [**Zar2009**]:

$$F = \left[\frac{12}{n \cdot k \cdot (k+1)}\right] \sum_{i=1}^{k} R_i^2 - 3 \cdot n(k+1) \tag{2.20}$$

The critical value of F is then determined by looking in a table of critical values for Friedman's test using the values for k, n and a significance level α. α needs to be chosen when looking up the critical values, where a level of 0.05 typically is selected; it is not excessively high to seize too many type 1 errors (rejecting a true null hypothesis) and not excessively low to seize too many type 2 errors (retaining a false null hypothesis). The F value and critical F value are then compared in order to decide whether to retain or reject the null-hypothesis. If the calculated F value is larger than the critical F value the null-hypothesis is rejected and vice versa. [**Zar2009**]
Another method of deciding on which hypothesis to be accepted is to evaluate the probability value (p-value), which can be returned by using statistical software. A significance level of 0.05 is again usually chosen. If the p-value is under 0.05 the null-hypothesis is rejected and vice versa. [**Zar2009**]

When comparing multiple groups of measurements the incident of rejecting a true null-hypothesis increase, and thus ignoring type 1 errors between pairs of means in the measurement groups. Several methods exist to counteract this multiple comparison problem. In this project the Bonferroni correction will be utilized for this purpose.

2.10.2 Bonferroni correction

A total number of $\frac{k(k-1)}{2}$ pairs can be coupled in multiple comparison testing, thus a total number of $\frac{k(k-1)}{2}$ hypotheses can be defined. The Bonferroni correction counteracts the incorrect rejection of a null-hypothesis by lowering the significance level for each tested hypothesis by a scale m, where m is the total number of hypotheses. Thus, the new significance level tested for each individual hypothesis is $\frac{\alpha}{m}$. The Bonferroni correction then rejects the null-hypothesis when p-value $< \frac{\alpha}{m}$.

3 | Methods

3.1 Study Design

This experiment focused on training the user to improve prosthetic control on a fixed pattern recognition-based control system. The novel approach in this study was to provide the user with visual feedback on how well the system recognized the performed movements during user training, by showing the confidence levels of the movements the control system recognized. The following section will lay an overview of the implementation of the different stages of this experiment.

To test if myoelectric prosthetic control could be improved by using visual confidence level feedback the following research hypothesis was made.

Exposing subjects to user training, in which confidence levels of movement recognition is used as feedback, will show statistically significant improvement in performance in a classification-based myoelectric prosthetic control scheme compared to a control group.

To test the hypothesis XX subjects of age XX, std $\pm$ XX were recruited, X right handed and X left handed, and randomly assigned to either a control group or test group. The subjects enrolled were assessed to meet inclusion criteria presented in the experimental protocol for test subjects in section 6. The experimental protocol was handed out to possible test participants before enrolment. The experiment was designed as a three session investigation. In each session both groups had data data acquired, received user training and did a performance test. It was important that during the experiment the subject was seated on a chair, with the dominant arm wearing the MYB hanging relaxed laterally down the torso. A graphical illustration of the stages of the experiment design can be seen on figure 3.1. Essential for the experiment was the difference in user training highlighted in step 3 where the groups received two different kinds of visual feedback. The sections to come will further elaborate on the implementation of each element in the experiment.

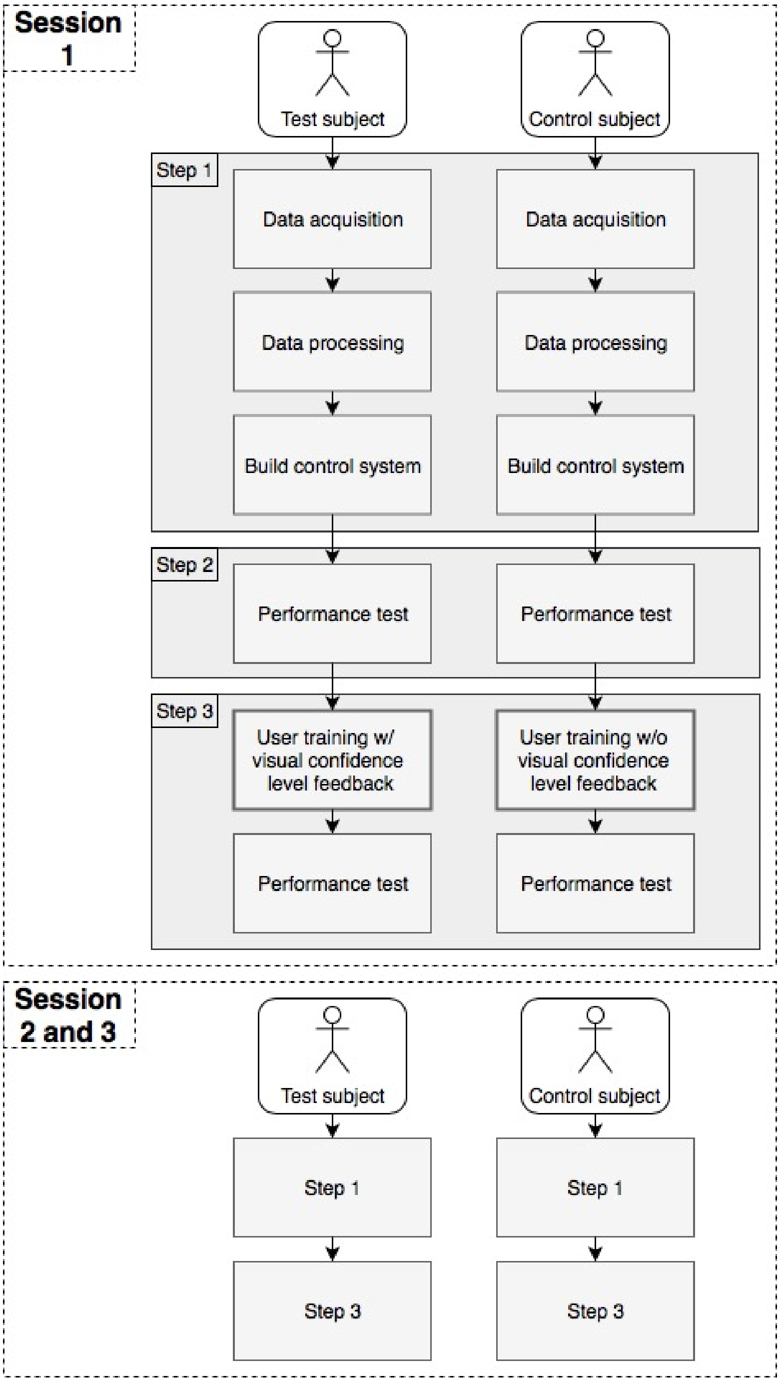


Figure 3.1: Graphical illustration of the experiment showing the steps of each session for the test and control group. Highlighted is user training in step 3 which is the only procedure that varies between the two groups, and thus the area of research interest in the experiment.

3.2 Data Acquisition

This section will clarify the method of acquiring data in this project. For data acquisition the MYB was used to record EMG signals from muscles in the lower forearm described in section 2.1. The recordings was made on test subjects instructed to perform six different hand gestures as introduced in the protocol, section 6.

For acquiring data a Graphical User Interface (GUI) was designed using MATLAB. In the GUI it was possible to change settings for different types of recordings. The first type of recording was a baseline measurement. This recording was made in order to be able to remove signal artefacts. This was done by subtracting the baseline from the EMG signal when the the EMG signal reached higher than the baseline. When the EMG signal was below the baseline, it was set as 0.
The second recording type was a Maximum Voluntary Contraction (MVC) which was a 15 second recording of the subject's maximum contraction of one movement that could be kept constant for 15 seconds. The mean MVC across all channels was set as a reference value for the following recordings.
The third type of recording was of EMG signals used to train the control system. The recordings of EMG signals were based on fractions of the MVC, which could be set using a menu in the GUI. As stated in the protocol, section 6, three contraction levels was used: 20%, 40% and 60%. The level of contraction defined the height of the plateau of a trapezoid trajectory which would be plotted in a window in the GUI. When doing EMG recordings the subjects must perform the instructed movement to control the height of a cursor in the trapezoid plot to best match the trajectory of the trapezoid. The cursor height was calculated as the mean EMG signal across channels normalized based on the MVC. The subject only controlled the height (EMG intensity) of the cursor as the cursor would automatically move forward along the x-axis in relation with time. The recording time was 15 seconds: 2.5 seconds rest at the initiation and ending, 2.5 seconds on the trajectory incline and decline and 5 seconds on the plateau. This approach provided data from a performed movement in both the transition and steady state phase. This data acquisition method was applied since the use of dynamically changing force data in training a classification-based control scheme has shown to improve performance and tolerance to proportional control [**Scheme2015**]. During recordings the investigators evaluated whether the subject followed the trajectory well enough. Furthermore to evaluate the training data, the investigators observed a spider-plot during the acquisition. The spider-plot showed the amplitude output for each channel in the MYB. If the activation pattern of the channels changed dramatically, it was a sign of fluctuations in muscle activation, and thus the subject did not perform the instructed movement. If this was observed the recording was discarded and a new was be acquired. An illustration of the data acquisition GUI is shown in figure 3.2.

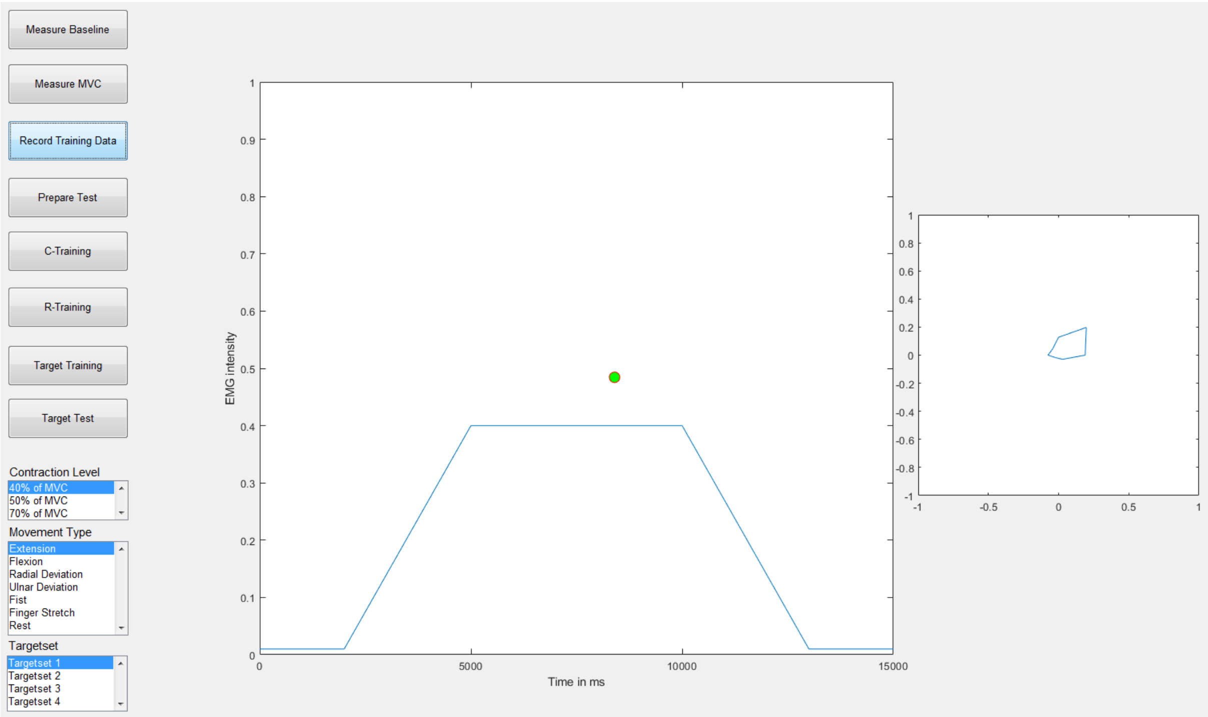


Figure 3.2: The implemented data acquisition interface. On the left is different buttons shown, where only "Measure Baseline", "Measure MVC" and "Record Training Data" is used in the data acquisition. The "Contraction Level" menu forms the trapezoidal trajectory and "Movement Type" saves the performed movement the correct label. In the center is the trapezoidal trajectory and the cursor representing the EMG signal. On the right is the spider-plot used to evaluate the quality of the performed movement.

3.3 Data processing

The following two sections will cover the implementation of the filter used to prepare the EMG-signal and the extraction of features to represent the signal. Choices behind implemented methods builds on background knowledge acquired in section 2.4.

3.3.1 Filtering of signal

As earlier mentioned in section 2.4.1, due to the MYB specifications limiting the sample rate to 200 Hz and due to movement artefacts in the low-frequency spectrum, it would be resourceful to implement a bandpass filter to avoid a biased signal. In the interest of representing the signal with its true properties a second order Butterworth bandpass filter has been implemented with cut-off frequencies at 10 Hz and 90 Hz. A filter steeper than second order was deselected due to a chosen trade-off between filter performance and computational performance, which is of great importance when doing real-time control. In figure 3.3 is the result of the bandpass filter implementation shown. The unfiltered signal shows frequency components in low-frequency spectrum around 0-10 Hz and indicating frequency components above above 100 Hz. Both ends of the spectrum have been attenuated limiting impact of artefacts and possible aliasing. Furthermore is the presence of the build-in 50 Hz notch filter elucidated as explained in section 2.3.1.

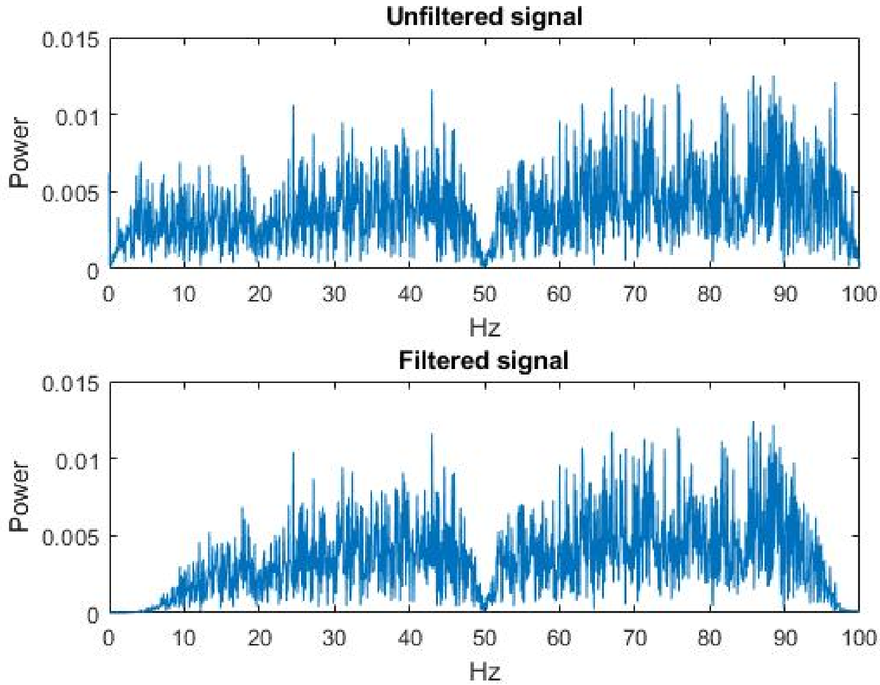


Figure 3.3: Frequency spectrum of a randomly selected EMG-recording showing the difference before an after implementing the bandpass filter. The unfiltered signal shows frequency components in low-frequency spectrum around 0-10 Hz and indicating frequency components above 100 Hz. The filtered signal shows reduction in the signal outside the cut-off frequencies.

3.3.2 Feature extraction

In section 2.4.2 it was stated that when extracting features for real-time prosthesis control, features are extracted from segments of the signal called windows. This projects will for the feature extraction utilize a window size of 200 ms and a 50 % overlap for all channels. This thereby gives the possibility of calculating and updating the feature values ten times a second, thus minimizing the update delay in the real-time user training and performance test.

The features chosen to represent the information of movements contained in the signal is primarily based on recommendations from [**Donovan2017**] where they found the optimal features for a real-time classification control scheme using the MYB. Donovan et al. [**Donovan2017**] used so called space domain features along with the MYB and got a five percent higher accuracy than by using the well known Hudgins time domain features. A total of seven features, Mean Absolute Value (MAV), Mean Mean Absolute Value (MMAV), Scaled Mean Absolute Value (SMAV), Correlation Coefficient (CC), Mean Absolute Difference Normalized (MADN), Mean Absolute Difference Raw (MADR) and Scaled Mean Absolute Difference Raw (SMADR) were derived and the following section will explain the extraction of each. This project will use SMAV, CC, MADN and SMADR for the final classification to reduce redundancy, but all seven will be explained because some features are a combination of others. [**Donovan2017**] Furthermore it has been chosen to extract the time domain feature of waveform length (WL) to represent frequency related information of the signal. The extraction of this feature will lastly be explained as well.

MAV is a feature that primarily is affected by the force produces when making a contraction. MAV is extracted for each window and calculated for each of the i^{th} channel. The extraction is expressed as:

$$MAV_i = \frac{\sum_{n=1}^{ws} |x_i[n]|}{ws} \tag{3.1}$$

where ws is the window size, the number of raw data points in that exact window. $x_i[n]$ is the n^{th} raw data points from the i^{th} channel.

The mean MAV across all channels, MMAV, is used to remove dependency of movement intensity. MMAV is calculated by using the MAV of all channels for the current window, and is done as following:

$$MMAV = \frac{\sum_{i=1}^{8} MAV_i}{8} \tag{3.2}$$

MMAV can be used to scale the MAV feature creating the SMAV feature. This feature should represent a non-dimensional relationship between channels. SMAV is simply calculated as:

$$SMAV_i = \frac{MAV_i}{MMAV} \tag{3.3}$$

As each of the eight EMG sensors in the MYB are located around the arm, they acquire signals from a mixture of sources. Also individual sources may affect multiple sensors depending on their size. Due to this a source measured by multiple sensors will effect their acquired signal correlation. An idea is therefore to calculate the correlation coefficient between each channel and its neighboring channel.

$$CC_i = \frac{\sum_{n=1}^{ws} X_i[n]X_{i+1}[n]}{\sum_{n=1}^{ws} X_i[n]^2} = \frac{\sum_{n=1}^{ws} X_i[n]X_{i+1}[n]}{ws} \tag{3.4}$$

$X_i[n]$ is the n^{th} data point from channel i. When calculating CC the data from each window is normalized by subtracting its mean value from each raw data point, and afterwards divided by their standard deviation.

Calculating CC can prove rather demanding in computational power due to the series og multiplication operations. Therefore Donovan et al. [**Donovan2017**] proposed introducing a mean absolute difference-based feature of lower computational complexity which still characterizes the spatial relationship between channels. The MAD feature is normalized in the same way as CC, making up the MADN feature calculated as:

$$MADN_i = \frac{\sum_{n=1}^{ws} |X_i[n] - X_{i+1}[n]|}{\sum_{n=1}^{ws} X_i[n]^2} = \frac{\sum_{n=1}^{ws} |X_i[n] - X_{i+1}[n]|}{ws} \tag{3.5}$$

If the normalization of the signal proves too demanding the feature can be calculated on the raw EMG-signal without the normalization. This makes up the MADR feature, calculated as:

$$MADR_i = \frac{\sum_{n=1}^{ws} |x_i[n] - x_{i+1}[n]|}{ws} \tag{3.6}$$

As the SMAV feature the MAD feature can be scaled by MMAV to remove movement intensity dependency. SMADR is calculated for each channel by:

$$SMADR_i = \frac{MADR_i}{MMAV} \tag{3.7}$$

As stated in the beginning some of these features introduce redundancy, subsequently the features of SMAV, CC, MADN and SMADR are the ones used for classification. [**Donovan2017**]

To further improve the decision foundation of the classifier it was proposed to include the time domain feature of WL calculated by:

$$WL_i = \sum_{n=1}^{N-1} |x_{i+1}[n] - x_i[n]| \tag{3.8}$$

WL is a measure of the signal complexity by calculating the cumulative length for each channel [**Phiny2012**].

3.4 Building the Control System

Following the data acquisition and processing, the training data obtained was used for movement classification. The features extracted for each of the seven movements were used for building the classification. The first subsection will cover the implementation of the classification and its output. An explanation of how the classification output for user training and control is used can be found in section 3.5 and section 3.6. Furthermore to be able to read out the intensity of movements a regression based calculation is made. The implementation of movement intensity control will be explained in the adjacent section.

3.4.1 Movement Classification

For classifying movements in this project the use of Linear Discriminant Analysis will be used as presented in section 2.5. The classifier is fed with the prior acquired training data in order to train the control system. The acquired training data were assembled in to matrices for each of the seven movements with one of the five features, containing the feature values for each of the eight channels. An example of one of these matrices can be seen in equation (3.9). This matrix contains the feature values n, for the feature CC for all three intensities of extension across all eight channels.

$$AllIntCC\_Ex = \begin{bmatrix} \begin{bmatrix} CCExtension40_{1,1}, CCExtension40_{1,2} \cdots CCExtension40_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ CCExtension40_{n,1}, CCExtension40_{n,2} \cdots CCExtension40_{n,8} \end{bmatrix} \\ \begin{bmatrix} CCExtension50_{1,1}, CCExtension50_{1,2} \cdots CCExtension50_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ CCExtension50_{n,1}, CCExtension50_{n,2} \cdots CCExtension50_{n,8} \end{bmatrix} \\ \begin{bmatrix} CCExtension70_{1,1}, CCExtension70_{1,2} \cdots CCExtension70_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ CCExtension70_{n,1}, CCExtension70_{n,2} \cdots CCExtension70_{n,8} \end{bmatrix} \end{bmatrix} \tag{3.9}$$

As seen the matrix consists of three smaller matrices, one for each of the intensity as explained in section 3.2. The AllIntCC_Ex matrix is just one of multiple as mentioned, to assemble a total training matrix to feed train the classifier all matrices are assembled into one big training matrix. This matrix can be seen below in equation (3.10).

$$\begin{bmatrix} AllIntCC\_Ex, AllIntSMAV\_Ex, AllIntSMADR\_Ex, AllIntMADN\_Ex, AllIntWL\_Ex \\ AllIntCC\_Fl, AllIntSMAV\_Fl, AllIntSMADR\_Fl, AllIntMADN\_Fl, AllIntWL\_Fl \\ AllIntCC\_Rd, AllIntSMAV\_Rd, AllIntSMADR\_Rd, AllIntMADN\_Rd, AllIntWL\_Rd \\ AllIntCC\_Ud, AllIntSMAV\_Ud, AllIntSMADR\_Ud, AllIntMADN\_Ud, AllIntWL\_Ud \\ AllIntCC\_Ch, AllIntSMAV\_Ch, AllIntSMADR\_Ch, AllIntMADN\_Ch, AllIntWL\_Ch \\ AllIntCC\_Oh, AllIntSMAV\_Oh, AllIntSMADR\_Oh, AllIntMADN\_Oh, AllIntWL\_Oh \\ AllIntCC\_Re, AllIntSMAV\_Re, AllIntSMADR\_Re, AllIntMADN\_Re, AllIntWL\_Re \end{bmatrix} \tag{3.10}$$

The classifier is trained, by fitting the matrix presented in equation (3.10) with labels for each of the movements, making it a supervised learning method. The classifier thereby forms seven classes, one for each of the movement, with discriminative decision boundaries separating them. For calculating the real-time use of classification outcome and confidence scores in user training and performance test as intended, the predict function in MATLAB has been used. The function is continuously evaluating each sample to the different movement clusters in the five dimensional feature space, as theory explains in section 2.5. Hereby the sample is assigned to the cluster (movement) it is most likely to belong to based on the training data. The Predict function also calculates the probability membership for the sample to all classes giving an idea of how sure the classifier is and thereby indicating the correctness of the movement performed.

3.4.2 Movement Intensity Control

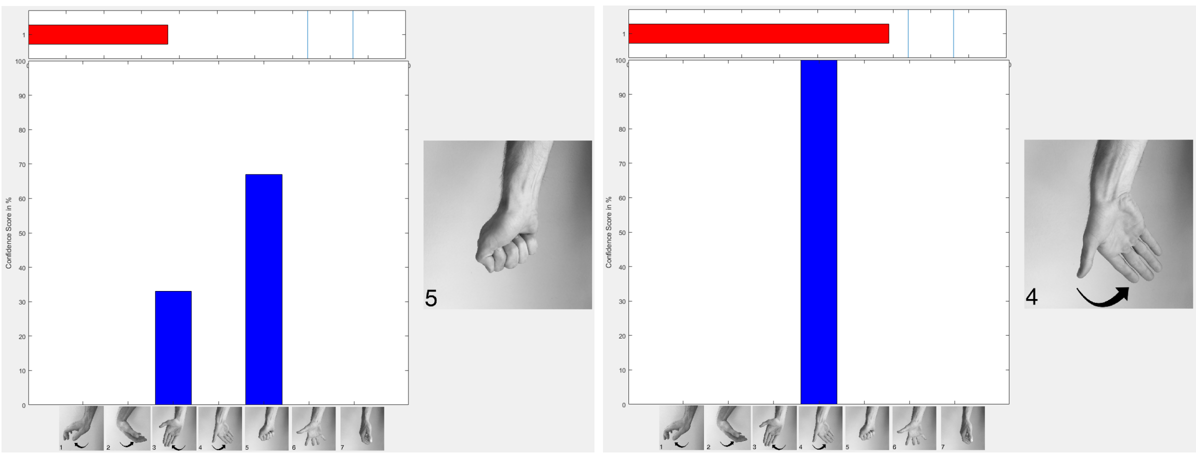
To get a way of measuring the intensity of a performed movement a regression based intensity control has been implemented.

We have six trained regressors - none for rest Classifier choses which classifier to activate

$$AllIntMAVEx = \begin{bmatrix} \begin{bmatrix} MAVExtension40_{1,1}, MAVExtension40_{1,2} \cdots MAVExtension40_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ MAVExtension40_{n,1}, MAVExtension40_{n,2} \cdots MAVExtension40_{n,8} \end{bmatrix} \\ \begin{bmatrix} MAVExtension50_{1,1}, MAVExtension50_{1,2} \cdots MAVExtension50_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ MAVExtension50_{n,1}, MAVExtension50_{n,2} \cdots MAVExtension50_{n,8} \end{bmatrix} \\ \begin{bmatrix} MAVExtension70_{1,1}, MAVExtension70_{1,2} \cdots MAVExtension70_{1,8} \\ \vdots \quad \ddots \quad \vdots \\ MAVExtension70_{n,1}, MAVExtension70_{n,2} \cdots MAVExtension70_{n,8} \end{bmatrix} \end{bmatrix} \quad (3.11)$$

3.5 User training

User training was the essential element of investigation in this experiment, with the movement recognition feedback as the area of interest. This section provides information on how the visual feedback was presented to the subjects in the two experiment groups during the user training, and what the objective for the subjects was.
The user training interface contained the following feedback: an illustration of the movement needed to be performed, a horizontal bar visualizing the contraction level and a vertical bar plot visualizing which movement is being recognized by the control system, as shown in figure 3.4. The difference in feedback given between subject group, lied in the information given in the vertical recognition bar plot.



(a) Test group user training interface. (b) Control group user training interface.

Figure 3.4: Illustration of the user training interface for the test group (a) and the control group (b). The vertical bar plot indicates which movement is being recognized and the horizontal bar plot indicates contraction level. The two vertical lines in the contraction level bar plot illustrates the contraction level interval the subject must reach. The large picture of a movement indicates which movement needs to be performed. The difference between the feedback the two subject groups receive is the information given in the vertical recognition bar plot. The control group only sees a full bar of the movement the control system recognizes the most, whereas the test groups receives the exact recognition probabilities of all movements.

The illustration of the movement needed to be performed was shown for 30 seconds, after which an illus-

tration indicating rest was shown for 7 seconds followed by a countdown from 3 to 1 seconds indicating the time left of the resting period. Thus, the subject needed to perform a movement for 30 seconds and rest for 10 seconds before another movement needs to be performed. The subjects needed to perform all movements in four different contraction level intervals of their maximal intensity, starting with the highest interval: 75-85 %, 55-65 %, 35-45 % and 15-25 %, visualized by the two vertical lines in the horizontal contraction level bar plot. The subjects needed to perform all movements in the same contraction level interval before moving to a new interval. The instructed movements were trained in a random order.
The horizontal bar showed the contraction level. This was calculated as the mean of the latest three intensity outputs as computed in section 3.4.2, regardless of the movement being recognized. This resulted in a 400 ms delay in the visualization of the horizontal bar at the initiation of the training of a movement, due to the windowing used in feature extraction as mentioned in section 3.3.2. However, the delay was not noticeable and the averaging of the intensity output resulted in a smooth visualization of the vertical bar.
The vertical bar plot showed which movement(s) the control system recognized. For a movement to be recognized as an active movement, the subjects had to perform the movement with more than 15 % contraction intensity. The test group received information on the exact probabilities for the movements that were recognized. Thus, more bars could appear at the same time as seen in figure 3.4 (a). The purpose of this feedback was for the subjects to adapt to how the control system recognized the instructed movement. It gave the subjects the possibility of noting which movements that also were recognized when performing the instructed movement. When the instructed movement was not recognized with a 100 % certainty the subject could use the information to slightly correct the performed movement until the control system recognized the instructed movement with a 100 % certainty. This bar plot was calculated as the mean of the recognition certainties calculated from the latest three feature inputs, which resulted in a smooth visualization of certainties for the movements in the bar plot.
The control group only received information on which movement was recognized the most, represented as a single full bar at the recognized movement as seen in figure 3.4 (b). Thus, the control group was not informed on the exact probabilities of which movements the control system recognized. This bar plot was calculated as the movement with the highest certainty out of the mean of the recognition certainties calculated from the latest three feature inputs.
To motivate the subjects and to train the transition to and from resting position a task was included in the user training. The subjects were instructed in performing the instructed movement with a 100 % certainty inside the instructed contraction level interval. When this was reached the horizontal bar would turn from red to green. The subjects were instructed in withholding the green colour for one second, after which the horizontal bar would turn blue and a light sound was played. After this task was reached the subjects was instructed in returning to rest and perform the task again. The objective for the subject was then to make the horizontal bar blue as many times as possible during an instructed movement of an instructed contraction level. The number of times the horizontal bar got blue during an instructed movement in an instructed contraction level was saved for later data analysis.

To summarize, the overall objective for the subjects during the user training was to adapt to how the control system recognized the performed movement. The user training was implemented for the subjects to possibly improve their ability to use the control system. Their ability to use the control system was then evaluated in the modified Fitts' Law task.

3.6 Fitts' Law

As stated in section 2.8 on the theory behind Fitts Law, this project will utilize a modified Fitts' Law task consisting of a virtual target reaching test to evaluate the progress of subjects going through user training. Additionally the proposed additional performance metrics, throughput, path efficiency, overshoot, stopping distance and completion rate will be used in this project. The following section describes how the Fitts' Law task has been implemented in this project.

3.6.1 Virtual target reaching test

The virtual targets reaching test is implemented into the same GUI used for data acquisition and user training, first mentioned in section 3.2. Different functions have been build into the GUI to enable switching between different usages by the press of a button. When switching to the function of the target reaching test the subject is met with the interface shown in figure 3.5. Here the subjects control the position of the cursor by performing movements shown by the borders of the grid area. Thus, extension of the hand will move the cursor to the right of the grid, and flexion will move the cursor to the left. Similarly, radial and ulnar deviation moves the cursor up and down respectively. This approach is used to improve the intuitiveness of the control where the direction of the cursor relate to the directions subjects will perform movements of the hand to control the cursor, when placed as instructed in the protocol, appendix 6 Subjects can control the size of the cursors red area by opening and closing the hand, where an open hand will increase the area and a closed hand will decrease the area. Each target is presented by an area with a center and an outer circle. Targets exist in to sizes to facilitate use of the index of difficulty for the throughput metric. The target reaching test consists of reaching a total of 16 targets which each appear for 15 seconds at positions around the center of the grid area. The order of appearance is fixed for each trial but the same across subjects, thus individual subjects will experience the targets as appearing in a new order with each trial.

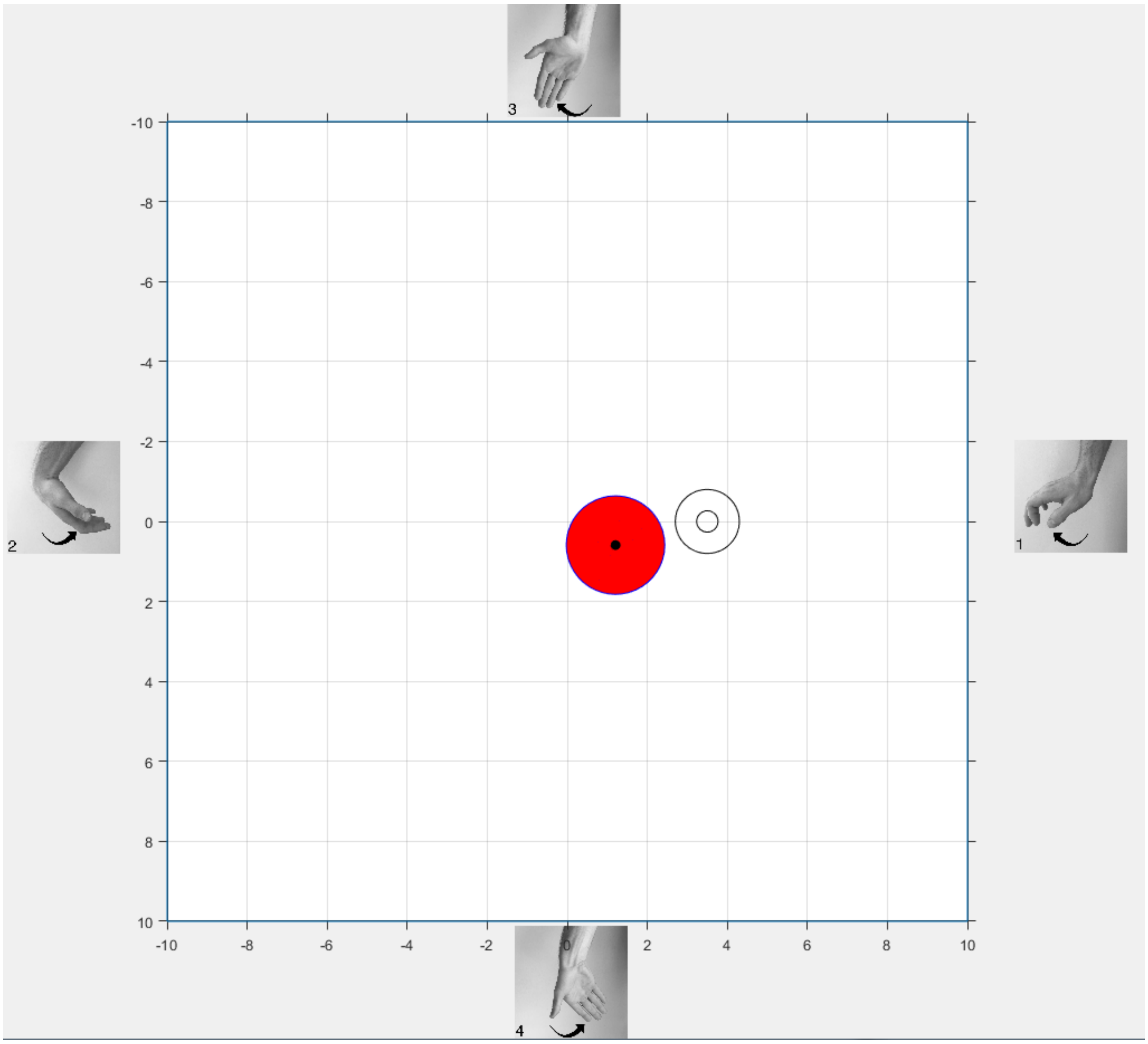


Figure 3.5: The implemented interface for the Fitts' Law task. The grid is the area in which the subject must reach targets with the controlled cursor. The cursor is the red circle area with the black dot in the center. Targets are shown as a circle area with an outer bigger circle. At each border of the grid area a hand movement tis shown. Performing shown movement will move the cursor in the direction of the picture.

Subject must reach the targets inner circle with the cursor dot and expand or decrease the red area of the cursor to reach a size close to that of the target. A moderate size threshold have been implemented to make it possible to reach targets, without a 100% accuracy of control. If a subject reach a target, the cursor will change color from red to blue, and a bell chime will sound to indicate that the target was reached. The cursor position will be reset to the center of the grid area and the color of the cursor will revert back to red when a new target appears. If a target is not reached within 15 seconds the current target will disappear, a new target will show and the cursor position will be set to the center of the grid area. The approach of resetting the cursor position after each target is to equalize the path for every subject.

Behind the scene of the target reaching test several metrics are recorded. A trace of the cursor movement throughout the whole test is recorded to decide the subjects path deviation from the optimal path to calculate the path efficiency and distances to targets, stated in section 2.8.1. An example of a cursor trace is shown in figure 3.6.

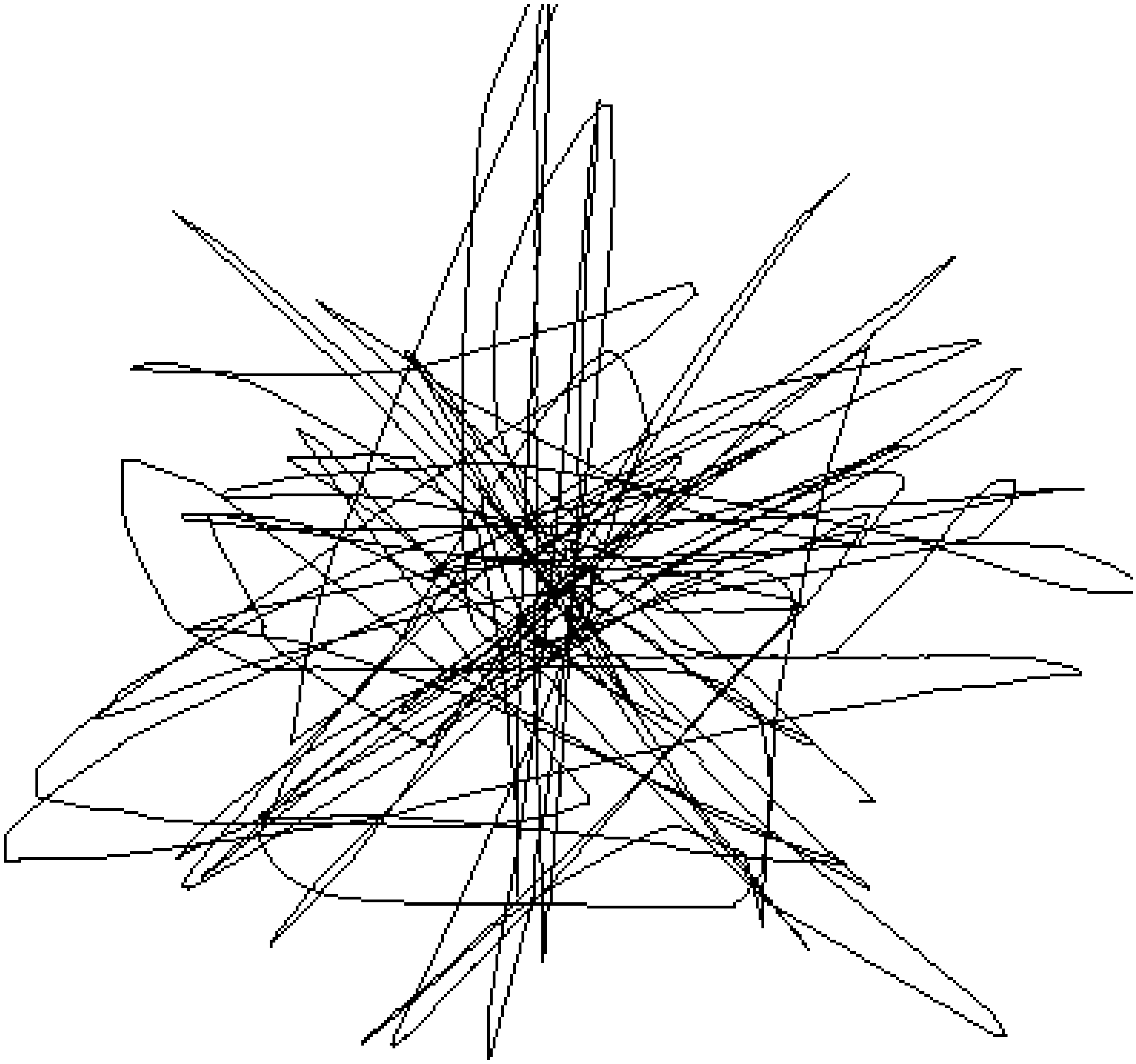


Figure 3.6: The trace of the cursor movement after a target reaching test. This information is not visible or shown to subjects.

The number of times a target is reached and exited without completing the dwell time, is recorded and used to calculate subjects overshoot, as stated in section 2.8.1. Similarly to tracking the travelled distance of the cursor inside the grid area, the travelled distance inside of each target is also recorded to calculate the stopping distance. The number or reached targets is recorded. The total time used to complete the task of reaching the 16 targets is know as 16 $targets\ *\ 15s = 240s = 4\ min$.

Following the completion of recording target reaching test data from all subjects the performance metrics introduced in section **??** are calculated and presented in the Results: chapter 4.

4 | Results

5 | Discussion

6 | Conclusion

Experiment protocol for test subjects

Title of the project
Using confidence levels of movement recognition in user training to improve prosthesis control

Details on investigators
All investigators are 2nd semester biomedical engineering master students at Aalborg University.

Purpose
The purpose of this experiment is to train the subject in getting better at controlling a functional prosthesis. The subject will be training seven hand movements that is used in activating the prosthesis. In the experiment the prosthesis will be represented on a computer screen, where the subject will receive visual feedback on how the prosthesis interpret the hand movement performed by the subject. By receiving this visual feedback it is hypothesized that the user will get better at controlling the prosthesis over time.

Background
Electromyography (EMG), or muscle signals, is widely used for controlling functional lower arm prosthetics for transradial amputees. The ideal purpose of a functional prosthesis is to behave as functional as possible compared to a biological arm. Functional prosthetics that rely on pattern recognition-based control are becoming exceedingly good in performance in a clinical environment, due to highly optimized system control. However, still only one commercially available pattern recognition-based prosthesis exist. Users reject these functional prosthetics usually due to functionality issues when utilizing them in daily life tasks outside the clinical environment. Many improvements have been made in the area of system control, but another approach of improving the prosthetic control is by training the user. User training has only been explored scarcely in the research literature, thus, new techniques to improve the user's ability to control a prosthesis are yet untouched. This experiment will focus on training the user to improve prosthetic control on a fixed pattern recognition-based control system. The novel approach in this study is to provide the user with information on how well the system recognizes the performed movement during user training.

Research hypothesis
Exposing subjects to user training, in which confidence levels of movement recognition is used as feedback, will show statistically significant improvement in performance in a classification-based myoelectric prosthetic control scheme, when compared to subjects who have not had the same feedback during user training.

Session time
The experiment consist of three sessions, which are spread over three consecutive days; one session per day. Each session is estimated to have a total duration of 30-60 minutes.

Inclusion criteria
The subject needs to be:

- able bodied.
- above 18 years old.
- able to read, understand and speak Danish and/or English.
- assessed by the investigators to understand and perform the instructions given during the experiment.

Exclusion criteria
The subject must not have:

- diseases that might influence subject performance.

Experiment procedure

The experiment consists of three sessions containing different procedures as illustrated on figure 1. The concept and chronology of each procedure is described below the illustration. During the experiment it is important that the subject is placed sitting on a chair, with the arm wearing the Myo armband (MYB) hanging relaxed down by the side of the body, as shown in figure 7 on page 42 illustrating the experiment setup.

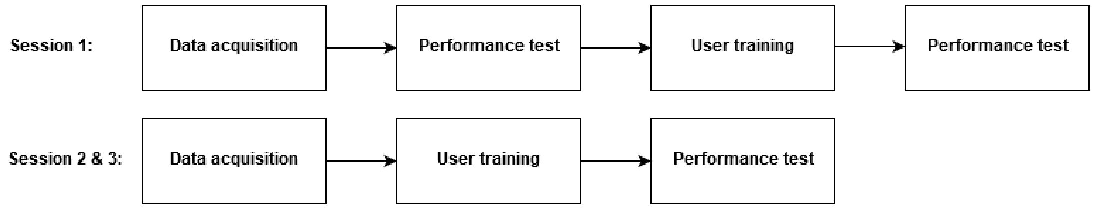


Figure 1: Pipeline for the three sessions in the experiment and what procedures each session contains.

Data acquisition
For the myoelectric control system to be able to identify a performed movement as the movement that is actually performed, it needs information about how the movement looks when represented as a EMG signal. Thus, EMG data needs to be acquired from the forearm of the subject while the subject performs the movements that is used in the experiment, see figure 6 on page 41. This data is fed to the control system to train the system to recognize each movement. In this experiment EMG data will be acquired from the subject with an EMG-electrode armband: MYB from Thalmic Labs. The chronology of this procedure is as follows:

1. Apply MYB on dominant forearm at the thickest part.
2. Synchronize MYB by performing wrist extension until three distinct vibrations are felt from the MYB.
3. Perform 15 seconds of maximum voluntary contraction (MVC) of instructed movement. The MVC is a contraction the subject is able to withhold in a constant intensity for the 15 seconds. Following the MVC the subject will be given a 30 seconds resting period to avoid muscle fatigue.
4. Perform three 15 seconds contraction trails of respectively 40%, 50% and 70% of MVC. During these contractions the subject will control a green marker representing the EMG signal and try to follow a trapezoidal trajectory as precise as possible. The trapezoidal trajectory consists of two 2.5 second transition phases and one 5 second plateau phase. Between each trial the subject will be given a 10 seconds resting period to avoid muscle fatigue.
5. Repeat step 3-4 until training data from all four wrist movements has been recorded.

An illustration of the interface used for data acquisition is shown in figure 2.

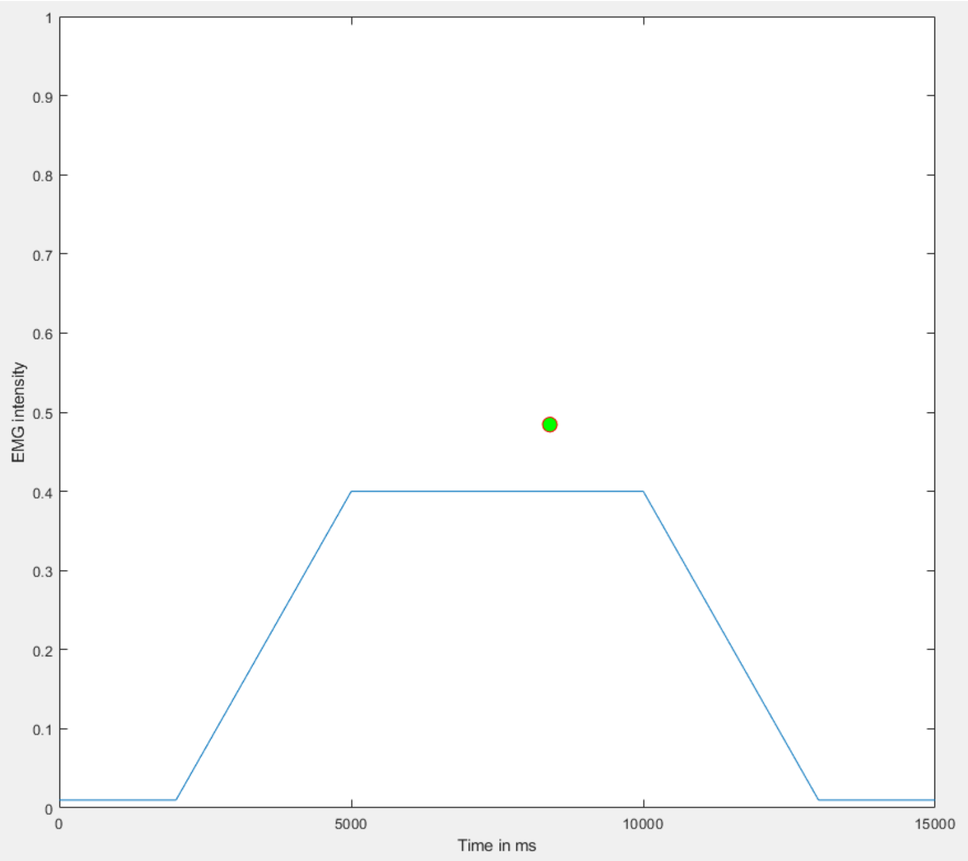


Figure 2: Illustration of the data acquisition interface showing the trapezoidal trajectory and the green marker representing the EMG signal.

User training

The purpose of user training is for the subject to train the movements used in the performance test. During the user training the subject will train one movement at a time at different contraction levels. When training a movement, visual feedback in form of confidence levels on how well the control system recognizes movements, is shown in percentage in a bar plot. In addition, the level of contraction is shown in a horizontal bar above the other bar plot. When performing the instructed movement at the instructed level of contraction the horizontal bar plot will appear green; if it is outside the instructed level or if the system does not recognize the performed movement, it appears red. The aim for the subject is to reach and withhold the instructed contraction level with 100 % confidence for each movement. When the subject withholds the contraction level inside the instructed contraction level for 1 seconds with a 100 % confidence the colour of the horizontal bar will appear blue. This indicates that the subject is performing well. After it has appeared blue, the subject must return to rest and perform the movement again and try to reach the instructed contraction level with a 100 % confidence. An additional aim for the subject is to make the horizontal bar plot appear blue as many times as possible. The chronology of this procedure is as follows:

1. Perform instructed movement at 75-85 % contraction level for 30 seconds followed by 10 seconds rest.
2. Perform step 1 for the remaining movements.
3. Repeat step 1-2 at 55-65 % contraction level.

4. Repeat step 1-6 at 35-45 % contraction level.
5. Repeat step 1-6 at 15-25 % contraction level.

An illustration of the interface used for user training is shown in figure 3.

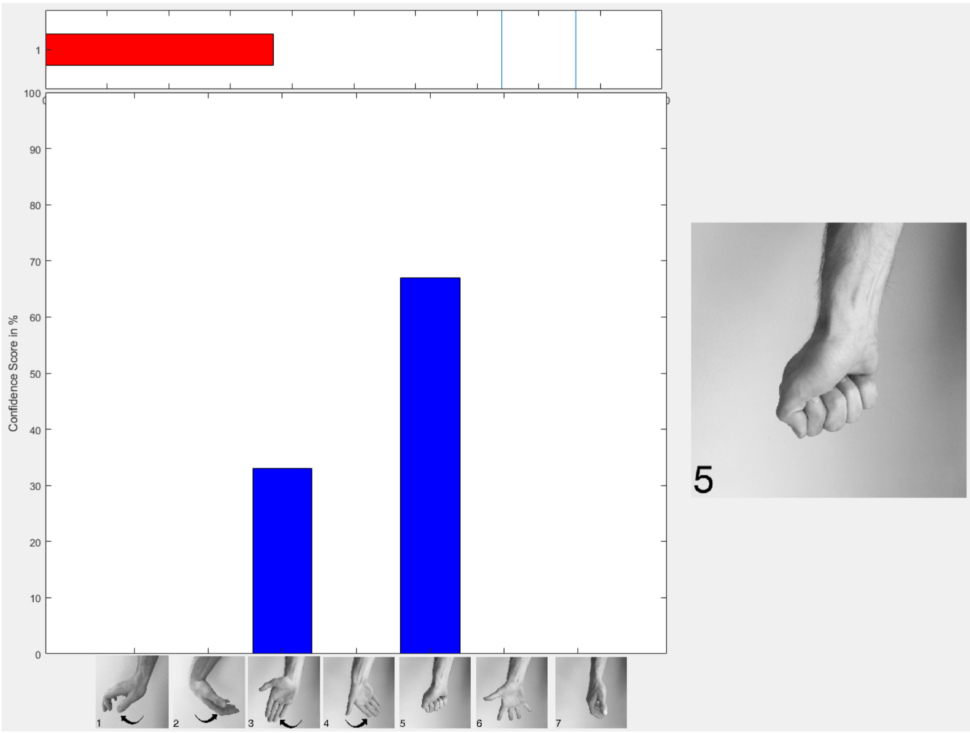


Figure 3: Illustration of the user training interface showing the bar plot indicating the confidence level of movement recognition and horizontal bar plot indicating contraction level. The picture on the right side of the bar plot indicates which movement needs to be performed.

User Training

The purpose of user training is for the subject to train the movements used in the performance test. During the user training the subject will train one movement at a time at different contraction levels. When training a movement, visual feedback in form of which movement the control system recognizes, is shown in a bar plot. In addition, the level of contraction is shown in a horizontal bar above the other bar plot. When performing the instructed movement at the instructed level of contraction the horizontal bar plot will appear green; if it is outside the instructed level or the control system does not recognize the performed movement, it appears red. The aim for the subject is to reach and withhold the instructed contraction level for the instructed movement while the control system recognizes it. When the subject withholds the contraction level inside the instructed contraction level for 1 seconds while the control system recognizes it the colour of the horizontal bar will appear blue. This indicates that the subject is performing well. After it has appeared blue, the subject must return to rest and perform the movement again and try to reach the instructed contraction level while the recognition of the control system matches the performed movement. An additional aim for the subject is to make the horizontal bar plot appear blue as many times as possible. The chronology of this procedure is as follows:

1. Perform instructed movement at 75-85 % contraction level for 30 seconds followed by 10 seconds rest.

2. Perform step 1 for the remaining movements.
3. Repeat step 1-2 at 55-65 % contraction level.
4. Repeat step 1-6 at 35-45 % contraction level.
5. Repeat step 1-6 at 15-25 % contraction level.

An illustration of the interface used for user training is shown in figure 4.

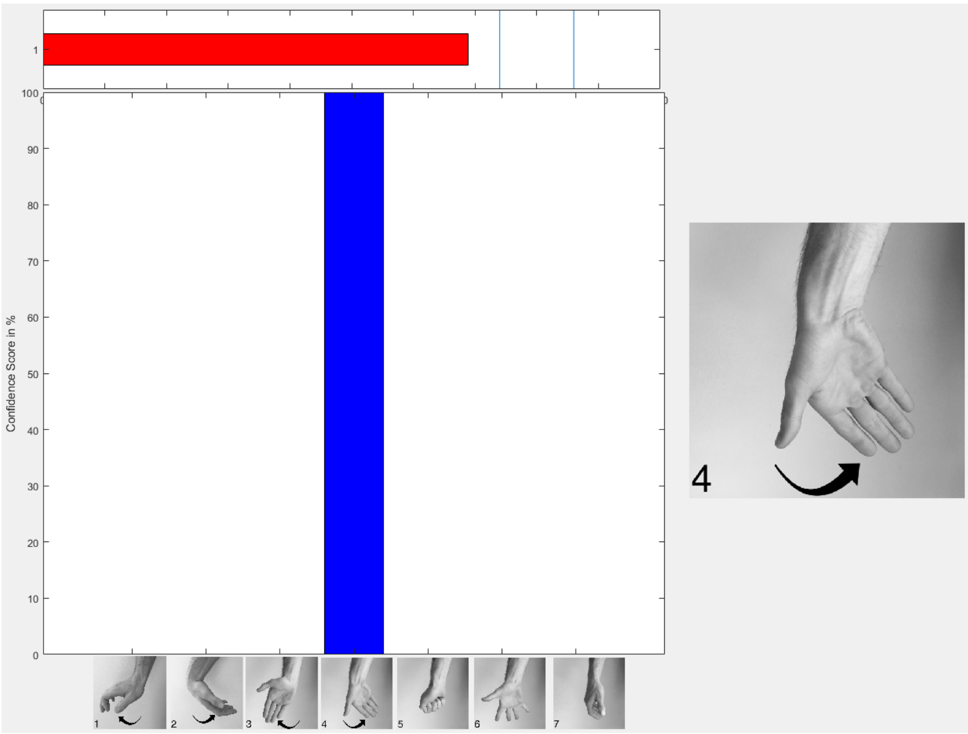


Figure 4: Illustration of the user training interface showing the bar plot indicating which movement is being recognized and the horizontal bar plot indicating contraction level. The picture on the right side of the bar plot indicates which movement needs to be performed.

Performance test

The purpose of the performance test is to assess the subject's ability to control a prosthesis. Instead of doing a test with a real prosthesis a virtual alternative has been developed for this experiment. The prosthesis is represented as a red circular cursor with a black dot inside in a Cartesian coordinate system, which the subject can move as well as expand and shrink in size by performing the trained movements. The following bullets describe which movement corresponds to which action in the coordinate system:

- Extension moves the cursor right.
- Flexion moves the cursor left.
- Radial deviation moves the cursor up.
- Ulnar deviation moves the cursor down.
- Closed hand shrinks the cursor.

- Opened hand expands the cursor.
- Rest keeps the cursor still.

The performance test consists of a target reaching test, where the subject must reach 16 targets of different sizes and locations. A target consists of a circle with a smaller circle inside. Only one target will be visible at a time. For the subject to reach a target and make a new appear, the subject must center the black dot of the cursor in the small circle of the target and expand/shrink the cursor to fit the size of the outer circle of the target. The cursor will appear green, when located at the correct position. The subject must dwell the cursor in a target for 1 seconds for it to be reached. When the cursor has dwelled for 1 second, it will appear blue for 1 second to indicate that the target has been reached. If a target is not reached within 15 seconds a new target will appear. When a new target appears the cursor will reset its position the origin. The aim for the subject is to reach as many target as possible as quickly as possible. The subject is only able to perform one movement at a time, as trained in the user training. Thus, no simultaneously performed movements will be recognized by the control system. The chronology of this procedure is as follows:

1. Use 2 minutes to get acquainted with the test.
2. Reach the visible target.

An illustration of the interface used for the performance test is shown in figure 5.

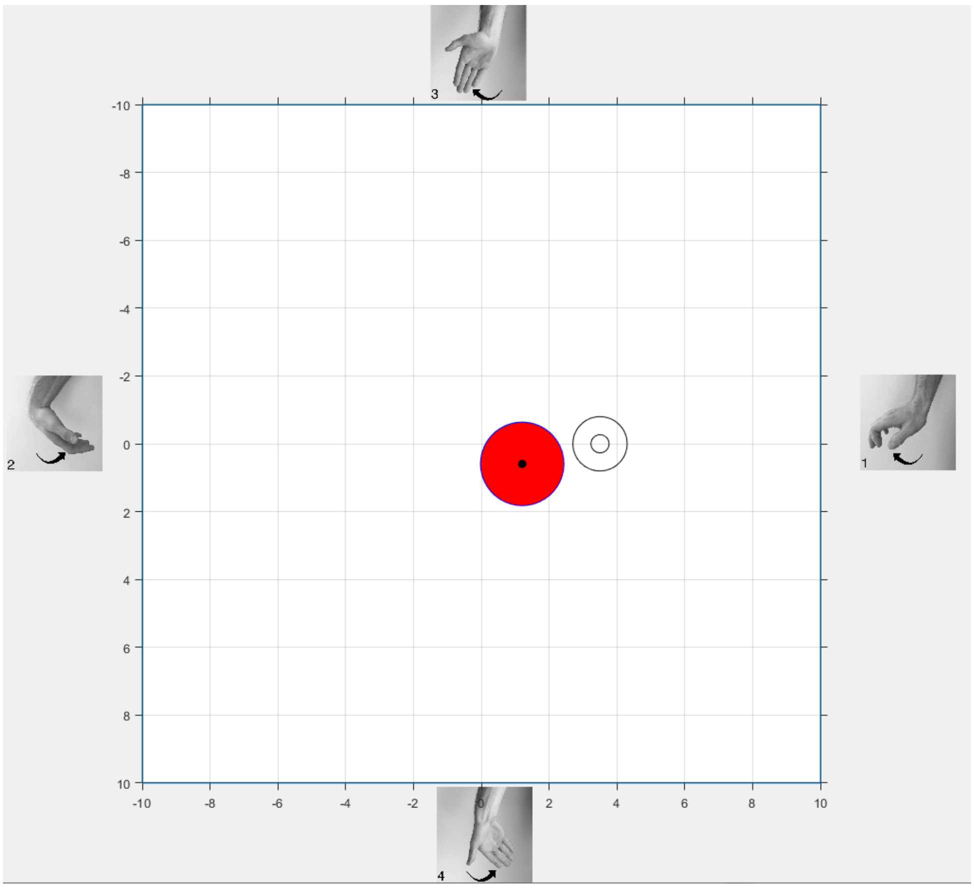


Figure 5: Illustration of the performance test interface showing a target and the cursor representing the prosthesis output. The pictures on the axes indicate which movement must be performed to move the cursor in a certain direction.

Movements used in the experiment

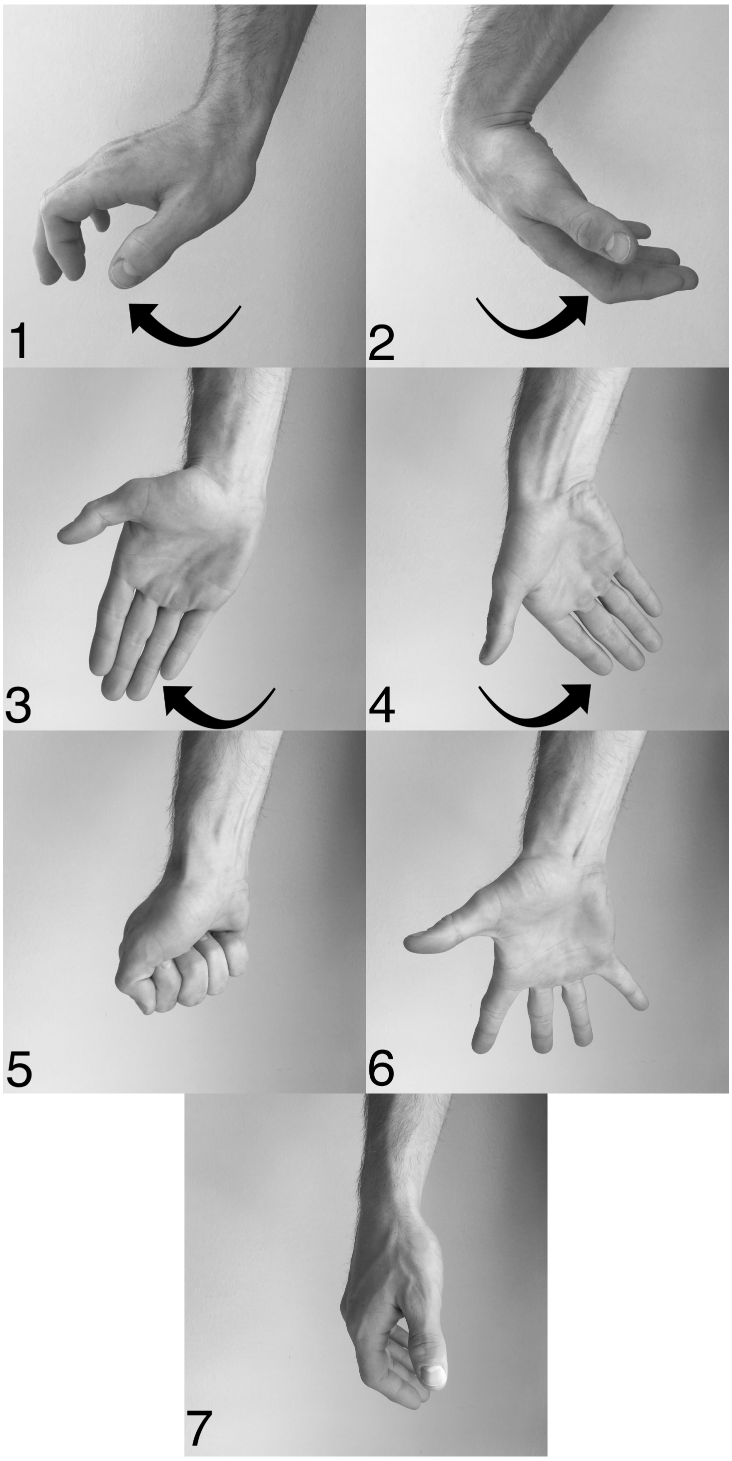


Figure 6: Illustration of the movements used in the experiment. 1: extension, 2: flexion, 3: radial deviation, 4: ulnar deviation, 5: closed hand, 6: opened hand and 7: rest.

Experiment setup

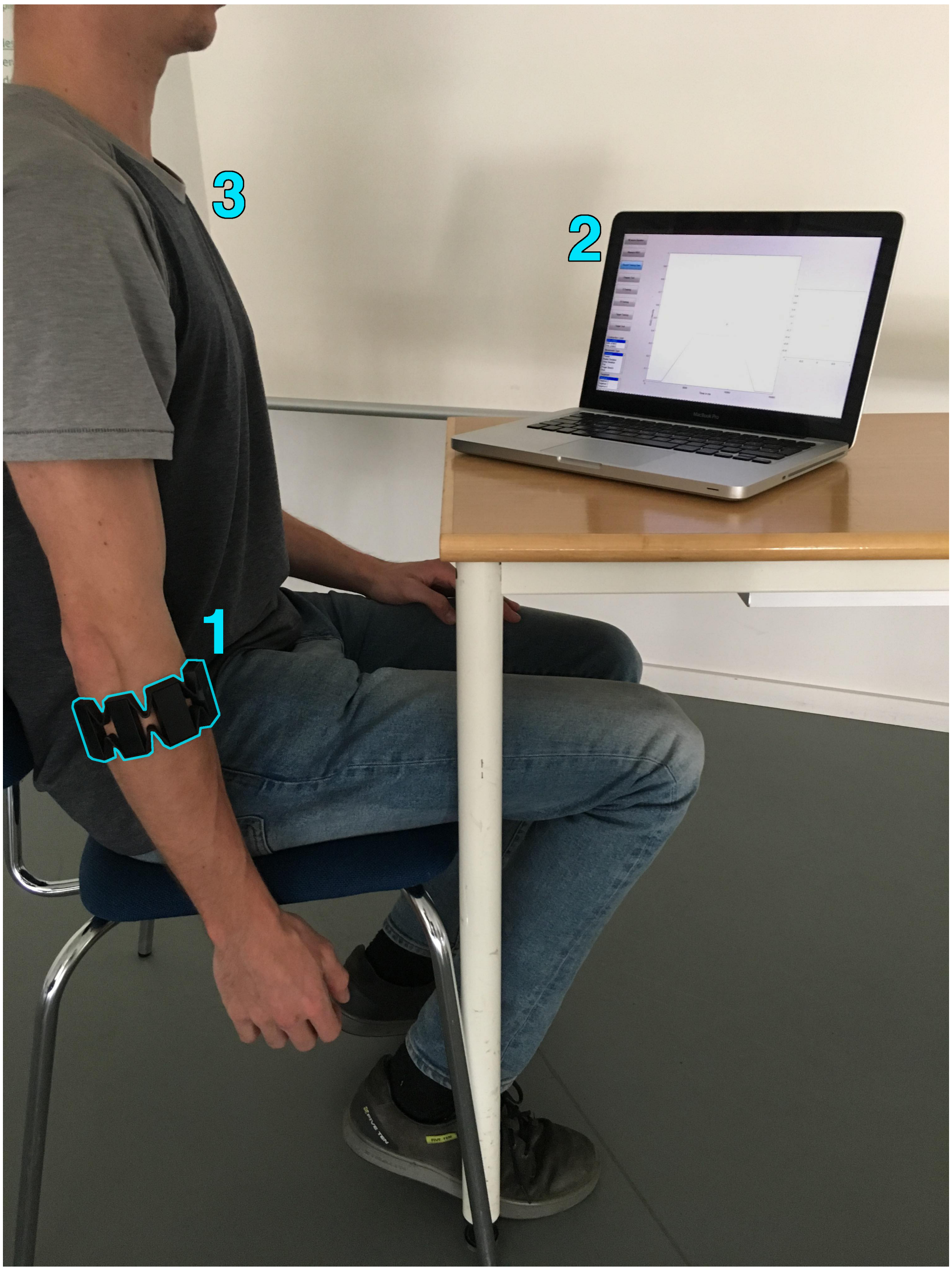


Figure 7: Illustration of the experimental setup; 1: MYB, 2: computer with interface and 3: subject. The subject is seated facing the computer screen with the arm wearing the MYB hanging relaxed down the side of the torso.

Experiment protocol for investigators

Subject name:

Session number:

This protocol functions as a checklist for the investigators in the experiment "Using confidence levels of movement recognition in user training to improve prosthesis control". The checklist is used to ensure all steps in the experiment is performed correctly and that no steps will be neglected. The experiment consists of 3 session of 3-4 procedures in each session, as shown in figure 8. The same procedures (data acquisition, user training and performance test) occur in all sessions and needs to be performed similarly each session. A checklist for each procedure is described in the sections below figure 8.

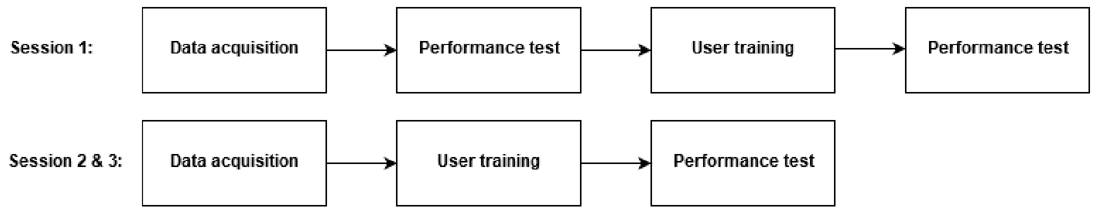


Figure 8: Pipeline for the three sessions in the experiment and what procedures each session contains.

The instruction of the of aim the respective procedures and content and functions in the interfaces is based on the information written in the experiment protocol for test subjects. It is expected that the subject has read the experiment protocol handed out prior the experiment, but the information regarding the respective procedures is retold to verify that the subject has understood the following procedure.

Data acquisition

- ☐ Disinfect MYB with alco-swabs.
- ☐ Disinfect MYB application area of subject's dominant forearm with alco-swabs.
- ☐ Instruct subject to stand in anatomical standard position.
- ☐ Mark with a permanent marker the size of the main channel (channel with LED) of the MYB on the most lateral position of the thickest circumference of the subject's dominant forearm.
- ☐ Instruct subject in applying MYB with the main channel (channel with LED) located on the marked position. The MYB must be worn so that the LED is located as distally as possible. Add clips to tighten the MYB if necessary.
- ☐ Ensure that the main electrode-channel is placed correctly.
- ☐ Instruct subject to sit on a chair facing the screen showing the interface, with the arm wearing the MYB hanging relaxed lateral to the torso.
- ☐ Connect MYB in armband manager.
- ☐ Instruct subject in synchronizing MYB by performing extension until three distinct vibrations are felt from the MYB.
- ☐ Instruct subject in the movements about to be performed in the data acquisition.
- ☐ Instruct subject in performing an MVC; that the contraction must be steady during the 15 seconds.
- ☐ Record MVC for one movement. Observe spider plot meanwhile. If the activation pattern for the channels alters too much during the recording is to be discarded and a new must be acquired.
  - ☐ Extension
  - ☐ Flexion

    - ☐ Radial deviation
    - ☐ Ulnar deviation
    - ☐ Closed hand
    - ☐ Opened hand

- ☐ Instruct the subject in tracing the trapezoidal trajectory with the green cursor in different contraction levels of the MVC.

- ☐ Record contraction levels of MVC for one movement. Observe spider plot meanwhile. If the activation pattern for the channels alters too much during the recording is to be discarded a new must be acquired.

    - ☐ Extension: ☐ 40 %, ☐ 50 %, ☐ 70 %
    - ☐ Flexion: ☐ 40 %, ☐ 50 %, ☐ 70 %
    - ☐ Radial deviation: ☐ 40 %, ☐ 50 %, ☐ 70 %
    - ☐ Ulnar deviation: ☐ 40 %, ☐ 50 %, ☐ 70 %
    - ☐ Closed hand: ☐ 40 %, ☐ 50 %, ☐ 70 %
    - ☐ Opened hand: ☐ 40 %, ☐ 50 %, ☐ 70 %

- ☐ Build regressors for each movement and build classifier trained with all movements.

User training

- ☐ Instruct subject in aim of the user training, and explain the content and functions of the interface.
- ☐ Initiate user training.

Performance test

- ☐ Instruct subject in aim of the performance test, and explain the content and functions of the interface.
- ☐ Initiate performance test.
- ☐ Save all training data and performance measures in folder named after name of subject, session number and which experiment group the subject belongs to.

Comments: