

Control of a robotic arm using a low-cost BCI

Steuerung eines Roboterarmes mithilfe eines low-cost BCI

Bachelor-Thesis von Daniel Alte aus Frankfurt am Main

Tag der Einreichung:

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Dr. Guilherme Maeda
3. Gutachten: Rudolf Lioutikov



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Control of a robotic arm using a low-cost BCI
Steuerung eines Roboterarmes mithilfe eines low-cost BCI

Vorgelegte Bachelor-Thesis von Daniel Alte aus Frankfurt am Main

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Dr. Guilherme Maeda
3. Gutachten: Rudolf Lioutikov

Tag der Einreichung:

This work is dedicated to my family, especially my parents and my girlfriend.

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 8. April 2016

(Daniel Alte)

Abstract

Electromyography (EMG) measurements recently became accessible and inexpensive due to new hardware developed for general Brain Computer Interface (BCI) applications. Low-cost devices for EMG measurements, therefore, enable potential deployment of intelligent hand prosthetics for a wider range of people, for example, children in developing countries who cannot afford the required and constant replacement of actuated prosthetics. However, such inexpensive devices use passive surface electrodes and the limited processing power only allows for very basic signal processing. As a consequence, EMG signals are characterized by noise, drift, and low repeatability. This thesis investigates the use of machine learning methods, particularly classification methods, to validate the feasibility of a low-cost BCI device for controlling a robotic arm. Part of this study regards the experimental search for the location of the surface muscles of the forearm, that have to be found and measured for different movements. This thesis investigates preprocessing methods for the raw measurements with a bandpass filter and root mean square (RMS) over a time window. Finally, this thesis compares classification results with linear support vector machines, K nearest neighbors and a Gaussian classifier. In our findings, the support vector machine and the k nearest neighbor output the best results. Although low-cost electromyography devices measure less reliable signals, they are capable of classifying different hand movements.

Zusammenfassung

Das messen von Elektromyographie (EMG) wurde in letzter Zeit, aufgrund neuer Hardware für allgemeine Brain Computer Interface (BCI) Anwendungen, zugänglich und kostengünstig. Kostengünstige Geräte ermöglichen daher den potentiellen Einsatz von Hand Prothesen für ein breiteres Spektrum der Menschen. Zum Beispiel für Kinder in Entwicklungsländern, welche sich keinen ständigen Austausch der Prothesen leisten können. Da kostengünstige Geräte passive Elektroden nutzen und eine begrenzte Rechenleistung haben, erlauben sie nur eine grundlegende Signalverarbeitung. Aus diesem Grund zeichnen sich die EMG signale durch Rauschen, Driften und einer niedrigen Wiederholbarkeit aus. Diese Thesis untersucht den Nutzen von Methoden des maschinellen Lernens. Insbesondere Methoden zum klassifizieren, um die Durchführbarkeit der Kontrolle eines Roboterarms mithilfe eines kostengünstigen BCI Gerätes zu überprüfen. Teil dieser Thesis ist die experimentelle Suche des genauen Orts von oberflächlichen Muskeln des Unterarms, welche für verschiedene Bewegungen gemessen werden müssen. Diese Arbeit untersucht einen Bandpassfilter und das Quadratische Mittel über ein Zeitfenster zum vorverarbeiten der Signale. Letzendlich werden die Ergebnisse der Klassifizierungen von dem K-Nächste-Nachbarn-Klassifizierer, der Support Vector Machine und einem Gausschen Klassifizierer miteinander verglichen. Die Ergebnisse zeigen, dass die Support Vector Machine und der K-Nächste-Nachbarn-Klassifizierer die besten Resultate liefern. Obwohl preiswerte Elektrographie Maschinen weniger zuverlässige Signale messen, sind sie fähig unterschiedliche Handbewegungen zu klassifizieren.

Acknowledgments

First of all, I would like to thank my first supervisor Dr. Guilherme Maeda for the time he spent on giving me advises and motivating me. Thank you not only for your aid in the thesis, but also for a your emotional support to master this challenge and achieving my personal goals. Also I would like to thank to Rudolf Lioutikov for supporting me in the beginning phase of the thesis. Furthermore, I would like to thank professor Jan Peters for giving me the opportunity write my thesis at IAS. Last but not least I want to thank to Thomas Hesse, Christos Votskos and Natalie Faber for many interesting and eye-opening conversations, which motivated me to keep up. Special thanks to the residents of the *Dungeon*!

Contents

1	Introduction	2
1.1	Motivation	2
2	Background	3
2.1	Biology of EMG Signals	3
2.2	EMG Signals	5
2.3	EMG Signals for Robotics Applications	5
3	Methods and Experimental Setup	8
3.1	Processing the Data	8
3.2	K Nearest Neighbor	9
3.3	Support Vector Machines	9
3.4	Gaussian Classifier	11
3.5	Hardware	12
3.6	Software	12
3.7	Bipolar Setting	13
3.8	Electrode Placement	13
4	Experiment	15
4.1	Features	15
4.2	Finding Muscle Locations	15
4.3	Forearm Setup	19
4.4	Results	23
4.5	Discussion	29
5	Conclusion and Future Work	32
	Bibliography	33

Figures and Tables

List of Figures

1.1	An open-source 3D printed prosthetic hand mechanically actuated.	2
2.1	Organization of a neuron [1].	3
2.2	Action potential: 2.2a illustrates the direction of the resting potential. The inside of the axon is charged negative, thus the voltage between the inner and the outer axon amounts $-70mV$. 2.2b illustrates the transmission of an action potential within an axon.(the graphic has to be interpreted from top to bottom)(The transmission ensues from the left to the right) 2.2c illustrates the current Potential depending on the time. $V_s = \text{threshold}$, $V_{RP} = \text{resting potential}$ [1]the course of an action potential.	4
2.3	The transmitting neuron releases neurotransmitters to trigger an action potential in the receiving neuron. [1]	4
2.4	Organization of a skeletal muscle [1].	5
2.5	A EMG signal is the composition of the Motor Unit Action Potentials [2].	6
2.6	Two different measurement procedures.	6
2.7	RAW EMG signal of a muscle contraction [3].	6
3.1	This figure illustrates how the original signal will be approximated using Gaussian functions.	9
3.2	This figure illustrates the hyperplane which separates the two classes plus(+) and minus(-). It is an example for the two dimensional case. The circled data points are the support vectors[4]. The hyperplane is described by $\mathbf{w}^T \mathbf{x} + b = 0$, and the width of the margin is $d = \frac{2m}{\ \mathbf{w}\ }$	10
3.3	The 32 bit <i>OpenBCI</i> board [5].	12
3.4	Setup	14
4.1	Anatomy of the forearm muscles [6].	17
4.2	forearmsetuptop	18
4.3	The four different hand movements measured and classified during the experiment. The bottom row shows the associated robot motions.	19
4.4	This figure shows the different processing and classification combinations. Each of the 12 paths is one combination. Each signal is filtered by a stopband with the cutoff frequencies $45Hz$ and $50Hz$. KNN has been applied for $K \in \{1, 2, 4, 8\}$	20
4.5	This figure shows the $\Phi(\mathbf{x})$ functions without the weights multiplied. The x-axis represents the number of samples and the y-axis the voltage in μV . The number of $\Phi(\mathbf{x})$ is $d = 20$. Each $\Phi(\mathbf{x})$ is a Gaussian distribution with $\sigma^2 = 0.05$. The expectations are evenly distributed over the x-axis.	20
4.6	This figure compares the RAW EMG signal compared to the fitted curve $\mathbf{w}^T \Phi(\mathbf{x})$	21
4.7	Different noise levels depending on the location.	22
4.8	This figure illustrates the raw measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of $250Hz$, which gives a duration of 3 seconds. <i>note:</i> Each color represents a trial.	24
4.9	This figure illustrates the bandpass filtered measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of $250Hz$, which gives a duration of 3 seconds. <i>note:</i> Each color represents a trial. Cutoff frequencies of the bandpass filter: $0.2Hz$ and $124Hz$	25
4.10	This figure illustrates the bandpass filtered measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of $250Hz$, which gives a duration of 3 seconds. <i>note:</i> Each color represents a trial. Cutoff frequencies of the bandpass filter: $10Hz$ and $124Hz$	26

4.11	This figure illustrates the bandpass filtered and RMS measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of 250Hz, which gives a duration of 3 seconds. <i>note:</i> Each color represents a trial. Cutoff frequencies of the bandpass filter: 10Hz and 124Hz. Window length of the RMS: 30	27
4.12	This figure illustrates the accuracies for the different used classifiers with the two preprocessing steps <i>bandpass</i> and <i>bandpass + weight space</i> . Filter parameter: 0.2Hz , 124Hz.	28
4.13	This figure illustrates the accuracies for the different used classifiers with the two preprocessing steps <i>bandpass</i> and <i>bandpass + weight space</i> . Filter parameter: 10Hz , 124Hz.	28
4.14	This figure illustrates the pipeline of the preprocessing steps and the training of the classifier.	29
4.15	Confusion matrices for SVM , GC , KNN (K=1) , KNN (K=2) , KNN (K=4) and KNN (K=8) on the bandpass filtered + RMS dataset. pred. = predicted class, act. = actual class	30
4.16	Precisions for SVM , KNN and GC with RMS.	31
4.17	This figure illustrates the final preprocessing steps. SVM , KNN (K=1) and KNN (K=2) have been empirically determined as the best classifiers.	31

List of Tables

3.1	This table illustrates the data format of one message. Byte 1 is the start byte, Byte 33 the stop byte. The bytes 3-26 are the bytes, which represent the EMG data. Each channel is encoded by three bytes.	13
4.1	This table shows the relation between the muscles and their movements. The muscles can be found in Figure 4.1	16
4.2	This table shows the empirical determined relation between the muscles and their movements. The muscles can be found in Figure 4.1. Flexor Carpi Radialis and Flexor Carpi Ulnaris could not be measured in this setup.	16

Abbreviations, Symbols and Operators

List of Abbreviations

Notation	Description
EMG	Electromyography
GC	Gaussian classifier
KNN	K nearest neighbor
rms	Root mean square
SEMG	Surface Electromyography
SVM	Support vector machines

1 Introduction

1.1 Motivation

Recently, it became increasingly popular to create three dimensional printed prosthetic hand. 3D models and assembly manuals for the massive deployment of low cost prosthetics can now be found on the Internet. Websites such as the e-NABLE [7] provide manuals for the manufacturing of limb prosthetics, which work mechanically. One of the designs is shown in Figure 1.1. Flexing the wrist or the elbow triggers the prosthetic hand to close by means of a cable construction. The hand allows the user to use the prosthetic device for simple applications, for example, for holding objects or riding a bicycle. Finer movements like stretching the index finger to press a button or making finger gestures are not possible.



Figure 1.1: An open-source 3D printed prosthetic hand mechanically actuated.

Therefore another kind of controlling a prosthetic limb is required. Electromyography (EMG) is a promising technology to control prosthetics or robotic arms [8]. Surface electromyography (SEMG) is a non-invasive method and therefore can be easily applied without any medical intervention [3]. In the case of an amputee, EMG allows to use measurement of the activation signal of the same muscle that was once used to moves the lost limb. This arguably allows for a more natural and intuitive control of a prosthetic robotic device—when compared to, for example, flexing the wrist to close the hand. One technological barrier that hinders the wide use of EMG signals to control prosthetic devices is that reliable and clean signals require expensive hardware [9].

Recently, devices like the *OpenBCI* have been introduced [5]. It is a low-cost brain-computer interface (BCI), which can also be used to measure EMG signals. Notably, the introduction of the ADS1299 [10] by Texas Instruments in 2012, an integrated circuit specific for biopotential measurements that integrates operational amplifiers and digital to analog converter, has allowed the design of low-cost and portable BCI devices [10]. The *OpenBCI* is a low-cost portable BCI device (approx. 600 USD) based on the ADS1299 that can be used to measure EMG signals and is at the order of magnitude cheaper than conventional EMG machines [11]. Associated with the new generation of open-source 3D-printed prosthetics, low-cost methods to measure EMG signals can have a profound impact in the quality of life of those who cannot access the expensive prosthetics. The use of such low-cost devices however, comes at the price of low quality signals, often with poor signal-to-noise ratios and large drifts. It is not clear if the measurements from such new generation of low-cost BCI devices is useful for the control of actuated prosthetics. The research field of machine learning, however, offers a number of classification methods that may alleviate the classification problem. This thesis aims to evaluate the feasibility of such low-cost EMG devices by measuring and classifying signals representative of movements that are useful for controlling a robotic arm for an amputee. Especially for people who lost their hand, EMG could be used to control a hand prosthetic hand. The necessary muscles to control the hand are located in the forearm and therefore, can be accessed via EMG measurements. This thesis, therefore, focus on the measurement and classification of the hand and forearm movements.

If the EMG signals are amenable to classification, we can endow actuated prosthetics with intelligent and adaptive control methods. In this thesis, we illustrate the use of a KUKA LWR7 lightweight arm as a proxy of an advanced prosthetic device. Not only disabled people could benefit from low-cost BCI systems. A possible scenario would be to control a robotic arm over a long distance and interact with the robot's environment. For example interaction in radioactive contaminated areas could be possible, which could help minimizing catastrophes. The control via EMG offers a more natural user interface, since natural movements can be transfered and applied to the robot via muscle activation.

2 Background

2.1 Biology of EMG Signals

In order to understand the nature of EMG signals an overview of the biological mechanism of muscle contraction is given. Every process in the human body including muscle contraction is controlled via neurons. These cells consist of one input and one output. If the amplitude of the input signal exceeds a certain threshold the cell transfers the input. The frequency of the input encodes the information [1].

Incoming signals are received at the dendrites and passed on to the soma (see Figure 2.1). At the axon hillock they are summed to one signal. Action potentials (APs) arise from these summed signals. The AP will be passed on via the axon to the synapses, from where it will cause the release of neurotransmitters which have an effect on the dendrites of a following neuron. Figure 2.1 shows this organization [1].

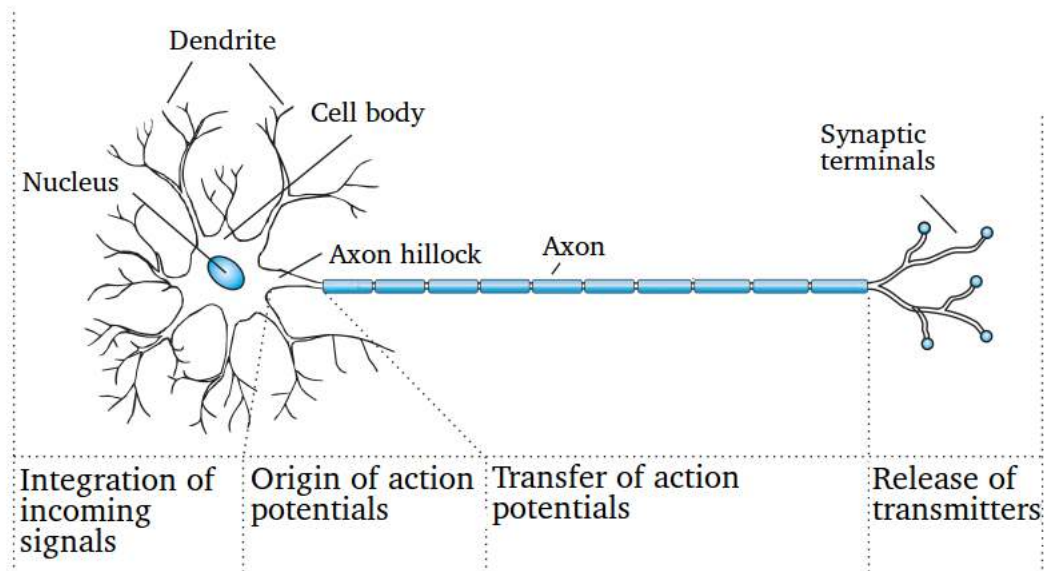


Figure 2.1: Organization of a neuron [1].

The default state of a neuron is the resting potential. It occurs when the cell is not stimulated. The inner charge of a neuron is negative in relation to the outside of the cell. As shown in Figure 2.2a the potential amounts $-70mV$. To keep the resting potential stable positive charged molecules are transported outside.

The incoming signals received at the dendrites charge the inside of a neuron with positive charge. Now the current potential exceeds the resting potential. This process is called depolarization (equation 2.1.)

$$V_c \geq V_{rest}, \quad V_c = \text{Current Potential} \quad (2.1)$$

If the current potential at the axon hillock (Figure 2.1) reaches the threshold $-60mV$ an action potential is triggered. The inside of the axon will be filled with positive charged molecules thus the current potential increases. Figure 2.2b shows that the entering of positive charged molecules is wandering through the axon. Figure 2.3 illustrates the synapse which is located in the end of the axon. A synapse of a transmitting neuron is connected to a dendrite of a receiving neuron. If the action potential reaches the synapse, synaptic vesicles containing neurotransmitters merge with the membrane of the synapse. This causes the release of the neurotransmitters into the synaptic gap. Neurotransmitters cause the entering of positive charged molecules into the dendrite of the adjacent neuron.

Figure 2.2c shows the course of an action potential depending on the time. It is triggered at $-60mV$, reaches its maximum at $+30mV$ and the depolarization gets compensated to $-80mV$ until the resting potential is restored. To compensate the depolarization the cell pumps positive charged molecules out of the cell.

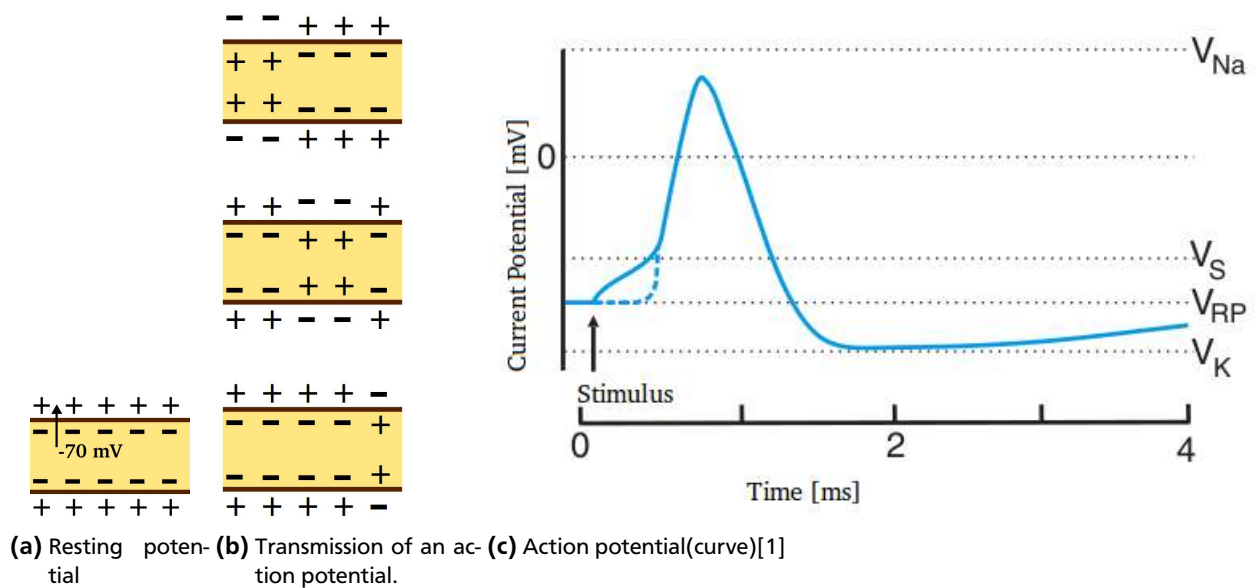


Figure 2.2: Action potential:

2.2a illustrates the direction of the resting potential. The inside of the axon is charged negative, thus the voltage between the inner and the outer axon amounts -70 mV .

2.2b illustrates the transmission of an action potential within an axon. (the graphic has to be interpreted from top to bottom) (The transmission ensues from the left to the right)

2.2c illustrates the current Potential depending on the time. V_S = threshold, V_{RP} = resting potential [1] the course of an action potential.

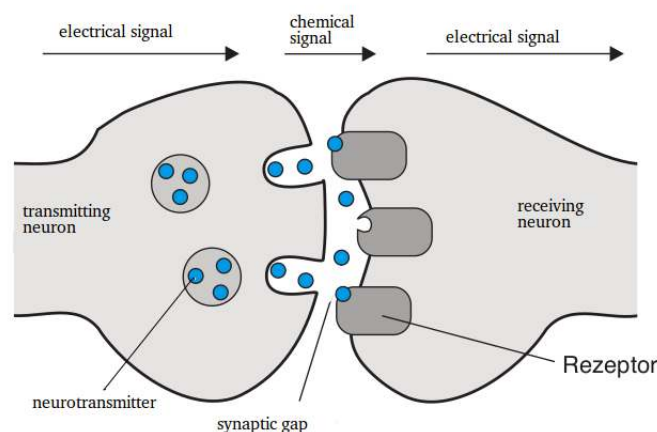


Figure 2.3: The transmitting neuron releases neurotransmitters to trigger an action potential in the receiving neuron. [1]

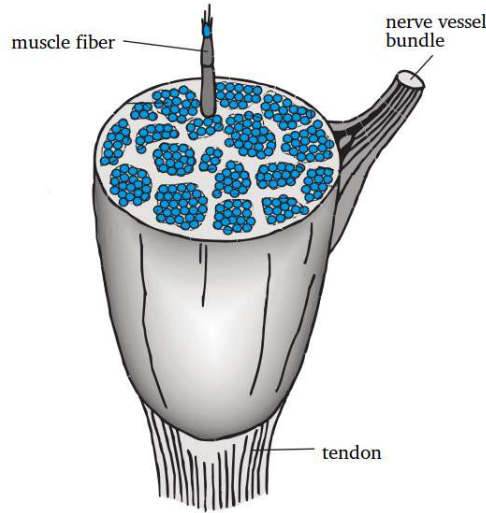


Figure 2.4: Organization of a skeletal muscle [1].

A skeletal muscle (Figure 2.4) consists of many muscle fibers. Each fiber is connected to a α -motor neuron. This neuron is responsible for the contraction of a muscle fiber. If a human wants to contract a muscle the neurons in the brain send signals through a chain of neurons to the α -motor neuron [1].

Similar to the synaptic transmission from a neuron to another, the α -motor neuron depolarizes the muscle fiber which leads to the contraction of the muscle fiber. The contraction of all fibers leads to the contraction of the muscle [1].

2.2 EMG Signals

A surface electrode is placed on the skin at the location of the muscle. Figure 2.5 illustrates measurements made by a surface electrode. The depolarizations of all muscle fibers located next to the electrode are summed to a single signal where each is called **Motor Unit Action Potential (MUAP)**. Due to the high density of the muscle fibers it is not possible to measure an action potential of a single α -motor neuron.

$$\text{Potential}_{\text{Electrode}} = \sum \text{MUAP} \quad (2.2)$$

Electromyography is used to obtain the activity of a muscle by measuring the voltage between the muscle itself and a neutral reference point (Figure 2.6a). Electrodes consisting of a high conducting material are placed on the muscle and the reference point. The voltage between electrode and the reference point is amplified by a high gain low noise amplifier.

In Figure 2.6a the setting for a monopolar measurement is illustrated. The monopolar measurement is one simple way to measure EMG signals. Monopolar measurements are more likely affected by artifacts [12].

$$V_{\text{Monopolar}} = \text{Amplification} \cdot (\text{Potential}_{\text{Signal}} - \text{Potential}_{\text{Reference}}) \quad (2.3)$$

In practice, bipolar measurement as shown in Figure 2.6b methods are favored. The bipolar method presupposes two electrodes per muscle, where the difference between both electrodes placed on the muscle is amplified. Bipolar measurements provide more stable measurements than the monopolar method [12].

A common EMG signal of a muscle contraction is shown in Figure 2.7. EMG signals have a magnitude at the order of microvolts and therefore are extremely sensitive to noise. Active electrodes are useful to amplify and filter EMG signals, however, they also increase the cost. The technology to amplify signals of such magnitude using low cost electronics is still evolving, but devices such as the openBCI may be already useful for discrete classification of movements.

2.3 EMG Signals for Robotics Applications

Classification of EMG signals has been emphasized in many works that attempt to control robot hands. Support vector machines (SVMs) has shown to achieve high accuracies based on offline training of hand movements. This was demonstrated for an online classification of EMG signals, which was used to control a robotic arm in [9, 8]. This thesis follows

Pictorial Outline of Decomposition

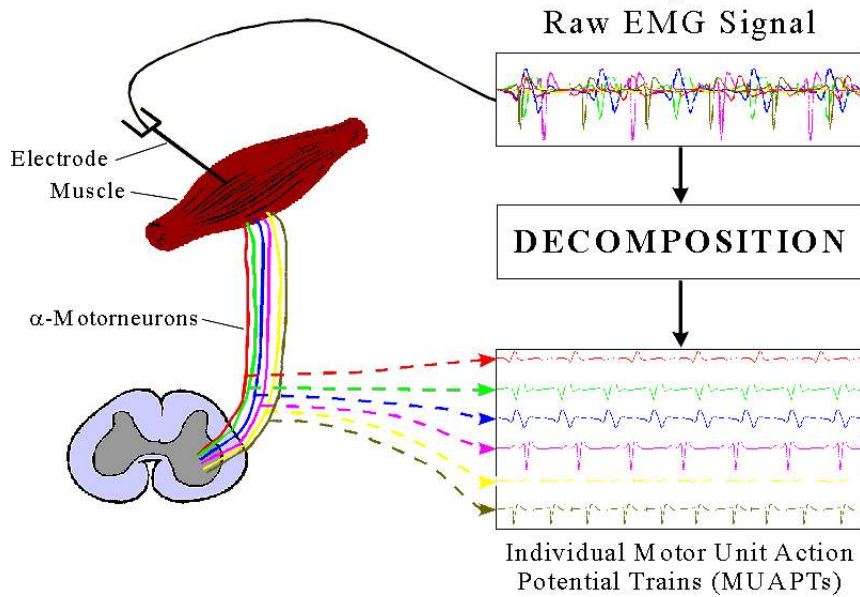


Figure 2.5: A EMG signal is the composition of the Motor Unit Action Potentials [2].

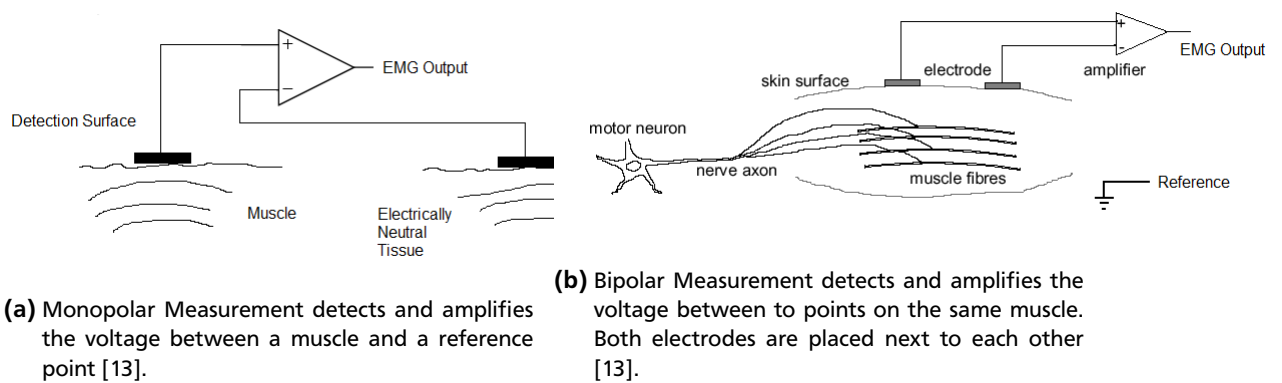


Figure 2.6: Two different measurement procedures.

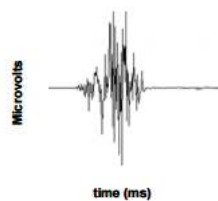


Figure 2.7: RAW EMG signal of a muscle contraction [3].

the approach presented in [8]. Namely, we are concerned with the problem of classifying the signals that are produced by a selected number of hand gestures from a pure data driven point of view. That is, no explicitly modeling of the muscles and forces involved is considered. However, our goal is to achieve a reasonable classification rate with a low-cost BCI, such as the *OpenBCI*. As opposed to [9], the *OpenBCI* hardware uses passive electrodes.

3 Methods and Experimental Setup

3.1 Processing the Data

Bandpass

EMG data has frequency spectrum between 10Hz and 250Hz [3] providing some initial values to design filters. It is common practice to use Butterworth filters [14] with EMG set to a bandpass configuration. The low frequency cut is useful to eliminate the slow varying drift of the signal, which is typical in EMG. High frequency cut is set to clean the signal from high frequency noise. Also, due to the magnitude of the signals, interference from power supplies with a frequency range between 50Hz and 60Hz can be noticeable, and in such cases a second bandpass (notch) filter is added.

Root mean square (RMS)

Calculating the root mean square over a time window is a common method in the processing of EMG signals [3]. In order to calculate the RMS, a discrete signal $S(b)$ will be split in J segments with $J = \frac{N}{wl}$, where wl is the time window of the RMS. The following equation represents the RMS for one segment

$$S(b)_{\text{rms}(j)} = \sqrt{\text{mean}[\mathbf{j} \cdot \mathbf{j}]} = \sqrt{\frac{\mathbf{j} \cdot \mathbf{j}}{wl}}, \quad (3.1)$$

where \mathbf{j} is one segment with the dimension wl and $S(b)_{\text{rms}(j)}$ is the RMS value of this segment. All segments will be concatenated to get the RMS approached signal over the time window wl [15, 4].

Regression with Radial basis functions (RBFs)

Measurements of EMG signals have a high sampling frequency (usually at the order of hundreds of Hertz) and are often corrupted by high levels of noise. Regression methods, particularly with radial basis functions are useful to both reduce the dimension of the data, which now has the same number of the features, and also to filter noise due the "clean-up" effect. Let d be the number of features defined at each time step n $\Phi_d(x_n)$, the measured data can then be represented as

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + \dots + w_D\Phi_D(x) \quad (3.2)$$

Based on the method of least square, the squared error between the observation t_n and the approximated data

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \Phi(\mathbf{x}_n)\}^2 \quad (3.3)$$

has to be minimized. N is the initial dimension of the data, (x_n, t_n) is the n -th data point of the data, \mathbf{w} is a vector of weights and $\Phi(x)$ is a vector of non-linear functions. Derivation of the error in respect to \mathbf{w} provides

$$\begin{aligned} E'(\mathbf{w}) &= \sum_{n=1}^N \{t_n - \mathbf{w}^T \Phi(\mathbf{x}_n)\} \{\Phi(\mathbf{x}_n)^T\} \\ E'(\mathbf{w}) &= \sum_{n=1}^N t_n \Phi(\mathbf{x}_n)^T - \sum_{n=1}^N \{\mathbf{w}^T \Phi(\mathbf{x}_n)\}. \end{aligned} \quad (3.4)$$

Setting this equation to zero and solving for \mathbf{w} gives

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}. \quad (3.5)$$

$(\Phi^T \Phi)^{-1} \Phi^T$ is called *Moore-Penrose pseudo-inverse* [16]. It is used to invert the non-quadratic $N \times M$ Matrix Φ , which is called *Design Matrix*.

$$\Phi = \begin{pmatrix} \Phi_1(1) & \Phi_2(1) & \dots & \Phi_D(1) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1(N) & \Phi_2(N) & \dots & \Phi_D(N) \end{pmatrix} \quad (3.6)$$

Each column in Φ represents Gaussian distribution as illustrated in Figure 3.1. A dataset with N data points has been represented by D weighted Gaussian distributions.

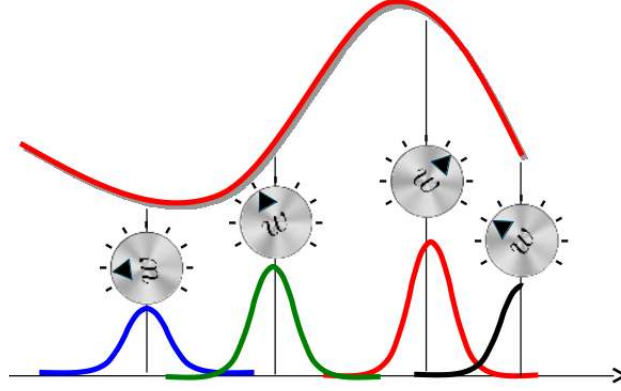


Figure 3.1: This figure illustrates how the original signal will be approximated using Gaussian functions.

3.2 K Nearest Neighbor

The first classifier used in this thesis is the K-Nearest Neighbor (KNN). It is the most simple machine learning algorithm to implement. It is a non-parametric classifier, which has the characteristic that no calculations have to be done ahead of the test phase. The computational cost in the training phase $\Theta(d)$ is mainly related to data storage. This reduces the duration of the training and makes it possible to add new data to the training set. For small data sets also a fast classification is possible, which makes KNN suitable for online classification [4].

Since each data is represented as a vector, the Euclidean distance between two data can be calculated. Basically the distance between each vector of the labeled training data and the test data vector is calculated. Subsequently the number of classes of the K shortest distances will be compared. Given a training set \mathcal{D} and one test case x , the following derivations of the KNN classifier are indicated. The Euclidean distance between the vector x and each column in the training set \mathcal{D} is calculated. The K smallest distances are considered. If the test and training belong to the same class the indicator function (3.8) is 1. Dividing the sum of the correct classifications through K , delivers the probability for class c .

$$p(y = c|x, \mathcal{D}, \mathcal{K}) = \frac{1}{K} \sum_{i \in N_K(x, \mathcal{D})} \mathbb{I}(y_i = c) \quad (3.7)$$

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases} \quad (3.8)$$

The equation gives the likelihood if the training data \mathcal{D} belongs to one assumed class c , thus the likelihoods for all combinations have to be calculated. Afterwards the class belonging to the maximum likelihood is the predicted class.

$$c^* = \operatorname{argmax}_{c \in \mathcal{C}} \{p(y = c|x, \mathcal{D}, \mathcal{K})\}, \quad (3.9)$$

where c^* is the predicted class [4]. When $K=1$ this algorithm is called simply nearest neighbor (NN). In this case, the decision is based on a simple point. In noisy data, NN tends to perform poorly as it overfits. On the other hand, excessively large values of K can smooth the decision boundaries and adversely affect the resolution of the classification.

3.3 Support Vector Machines

In its simplest form, a Support Vector Machine (SVM) is a binary classifier, which can distinguish between two linear separable classes. It can also be used as a non-linear classifier, if a kernel function is applied to the data before training and testing. Since the support vectors are computed during the training step, which represents the slowest part of the algorithm, SVMs can potentially classify faster than KNNs for time-critical applications.

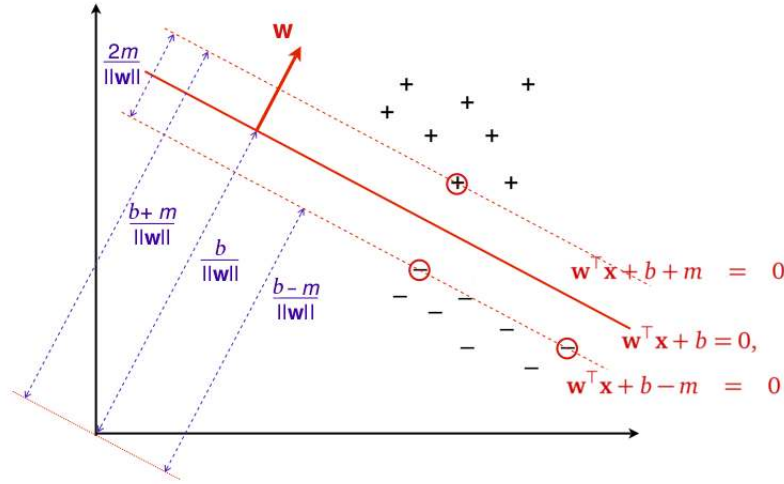


Figure 3.2: This figure illustrates the hyperplane which separates the two classes plus(+) and minus(-). It is an example for the two dimensional case. The circled data points are the support vectors[4]. The hyperplane is described by $\mathbf{w}^T \mathbf{x} + b = 0$, and the width of the margin is $d = \frac{2m}{\|\mathbf{w}\|}$.

The idea behind support vector machines is to find a decision boundary which separates two classes such that the margin of the boundary is as wide as possible. The hyperplane

$$\mathbf{w}^T \mathbf{x} + b = 0, \quad (3.10)$$

where \mathbf{w} is the normal and b is the *bias term*, serves as decision boundary. The width of the margin is the distance between

$$\begin{aligned} \mathbf{w}^T \mathbf{x} + b + m &= 0 \quad \text{and} \\ \mathbf{w}^T \mathbf{x} + b - m &= 0. \end{aligned} \quad (3.11)$$

It is defined by

$$d = \frac{2m}{\|\mathbf{w}\|}. \quad (3.12)$$

To maximize the decision boundary, $\|\mathbf{w}\|$ has to be minimized, thus $\frac{1}{2}\|\mathbf{w}\|^2$ also can be minimized [4]. This leads to following quadratic, constrained optimization problem

$$\begin{aligned} (\mathbf{w}^*, b^*) &= \underset{\mathbf{w}, b}{\operatorname{argmax}} \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } & y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, 1 \leq i \leq n. \end{aligned} \quad (3.13)$$

It can be solved with the Lagrange function

$$\begin{aligned} \Lambda(\mathbf{w}, b, \alpha_1, \dots, \alpha_n) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1) \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i) + \sum_{i=1}^n \alpha_i y_i b + \sum_{i=1}^n \alpha_i \\ &= \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \mathbf{w} \cdot \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) + b \left(\sum_{i=1}^n \alpha_i y_i \right) + \sum_{i=1}^n \alpha_i. \end{aligned} \quad (3.14)$$

To maximize the Lagrange function the gradient has to be calculated with respect to \mathbf{w} and b .

$$\nabla_{\mathbf{w}, b} \Lambda(\mathbf{w}, b, \alpha_1, \dots, \alpha_n) = 0 \quad (3.15)$$

Solving the resulting equation system

$$\begin{aligned}\frac{\partial \Lambda}{\partial \mathbf{w}} &= 0 \\ \frac{\partial \Lambda}{\partial b} &= 0\end{aligned}\tag{3.16}$$

$$\begin{aligned}\mathbf{w} - \sum_{i=1}^n \alpha_i y_i x_i &= 0 \\ \sum_{i=1}^n \alpha_i y_i &= 0\end{aligned}\tag{3.17}$$

gives $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i x_i$ and $\sum_{i=1}^n \alpha_i y_i = 0$ for a maximal $\Lambda(\mathbf{w}, b, \alpha_1, \dots, \alpha_n)$. Subsequently the Lagrange function will be substituted with the output of the equation system.

$$\begin{aligned}\Lambda(\alpha_1, \dots, \alpha_n) &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{i=1}^n \alpha_i y_i x_i - \sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{i=1}^n \alpha_i y_i x_i + b \sum_{i=1}^n 0 + \sum_{i=1}^n \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{i=1}^n \alpha_i y_i x_i + \sum_{i=1}^n \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + \sum_{i=1}^n \alpha_i\end{aligned}\tag{3.18}$$

Λ is now independent from \mathbf{w} and b . We calculate the α , that maximizes

$$\begin{aligned}\alpha_1^*, \dots, \alpha_n^* &= \underset{\alpha_1^*, \dots, \alpha_n^*}{\operatorname{argmax}} -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + \sum_{i=1}^n \alpha_i \\ &\text{subject to } \alpha_i \geq 0, 1 \leq i \leq n \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0.\end{aligned}\tag{3.19}$$

With $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i x_i$ the decision boundary is determined.

Natively, SVMs can only distinguish between two classes. They can be extended to classify between multiple classes using the *one versus rest* scheme [4]. In short, given that z binary classifiers will be trained, the first classifier separates C_1 from C_2, \dots, C_z , the second one separates C_2 from the rest, and so on [4]. This approach will be used in this thesis to classify different movements of the hand.

3.4 Gaussian Classifier

Gaussian classifiers require a small amount of training data to estimate the parameters and can provide soft bounds between classes. Gaussian classifiers are robust to noise [17].

Assume each class being modeled Gaussian distributions. The Gaussian classifier determines to which class c_i one test \mathbf{x} belongs. The general form of a Gaussian distribution with the dimension d is

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1}(\mathbf{x} - \mu)\right),\tag{3.20}$$

where \mathbf{x} is a n -dimensional vector, μ the n -dimensional center of the Gaussian and $\Sigma = (\sigma_{ij})$ the covariance matrix. $P(\mathbf{x}|\mu, \Sigma)$ is the probability, that \mathbf{x} belongs to the distribution. The covariance matrix is a squared matrix, which contains all combinations of covariances between the elements of \mathbf{x} . As a consequence the diagonal of Σ are all variances of \mathbf{x} . Since μ and Σ are specified for each class, $P(\mathbf{x}|\mu_i, \Sigma_i) = P(\mathbf{x}|c_i)$ applies. Using the Bayes' theorem provides

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})},\tag{3.21}$$

which is the probability c_i given \mathbf{x} , i.e. the probability that training \mathbf{x} belongs to c_i . Since we just need to know which c_i \mathbf{x} belongs to, the proportionality of

$$\begin{aligned} P(c_i|\mathbf{x}) &\propto P(\mathbf{x}|c_i)P(c_i) \\ &\propto P(\mathbf{x}|\mu_i, \Sigma_i)P(c_i) \\ &\propto \frac{1}{(2\pi)^{d/2} \det(\Sigma_i)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^\top \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right) P(c_i) \end{aligned} \quad (3.22)$$

$$(3.23)$$

suffices for classification. To classify we select the class that maximizes

$$c_{\text{predicted}} = \underset{c}{\operatorname{argmax}} (P(\mathbf{x}|\mu_i, \Sigma_i)P(C)). \quad (3.24)$$

In the thesis, the data of different channels are concatenated to provide a single vector of measurements. This approach allow us to capture also the correlation between channels. This differs from methods based on naive Bayes classifiers where the probability of each channel could be given independently.

3.5 Hardware

The hardware used in this thesis is the *OpenBCI* 32bit (see Figure 3.3) board. It is a 8-channel neural interface with a 32-bit processor. At the core of the device is the integrated circuit ADS1299. This circuit is a low-noise, 8-Channel, 24-Bit analog front-end, specifically designed for biopotential measurements [10]. The OpenBCI firmware is preflashed to the board. It is designed for the measurement of EEG, EMG and EKG. The board communicates wirelessly to a computer via the the OpenBCI programmable USB dongle, which is based on the RFDuino radio module [5]. As electrodes the “standard gold cup electrodes” provided by the manufacturer are used. To increase the conductivity a conductive paste “Ten20” also provided by the manufacturer is applied on the electrodes [5]. In this thesis a computer with a dual core CPU, 4 GB RAM, running on Windows 7 received the signals from the board via the Bluetooth dongle.

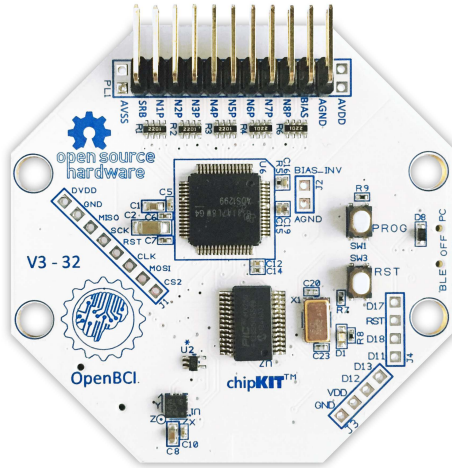


Figure 3.3: The 32 bit *OpenBCI* board [5].

The board has for each channel two pins. One N-pin and one P-pin, which can be connected to a differential amplifier, leading to bipolar measurements. The connection depends on the setting of the multiplexer of the circuit. According to the tutorial on the manufacturer’s website, the multiplexer is set in this settings default [5].

3.6 Software

The software was implemented in *MATLAB*, which provides an interface for serial connections. Two tasks had to be implemented. First, receiving and saving of the data and second preprocessing and classifying it. To start the data stream the PC has to send the character ‘b’. To end the stream, the serial devices requires the character ‘s’. The information

Byte 1	Byte 2	Bytes3-5	Bytes6-8	Bytes9-11	Bytes12-14	Bytes15-17	Bytes18-20	Bytes21-23	Bytes24-26	Bytes27-32	Byte33	
0xA0	Sample Number	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	accelerometer data	0xC0	

Table 3.1: This table illustrates the data format of one message. Byte 1 is the start byte, Byte 33 the stop byte. The bytes 3-26 are the bytes, which represent the EMG data. Each channel is encoded by three bytes.

is taken from the official documentation on the manufacturer's website [5]. While receiving the data from the board, the program checks if the data is complete. In order to do the checking, the data format of a message is needed (see Table 3.1). The first and the last byte are start and stop bytes. To check whether a message is complete the number of received bytes has to be 33 before a new start byte is detected. If the message is not complete an exception will be thrown. To extract the information of the message the three bytes of each channel have to be shifted and added up. The unit of the variable *newInt* is count. The next is to multiply *newInt* by a scale factor of 0.02235 microVolts per count. This supplies a value with the unit microVolts. Given $h = (\text{time} * 250\text{Hz})$ the values will be saved in a $h \times 8$ matrix, where *time* is the duration of the measurement.

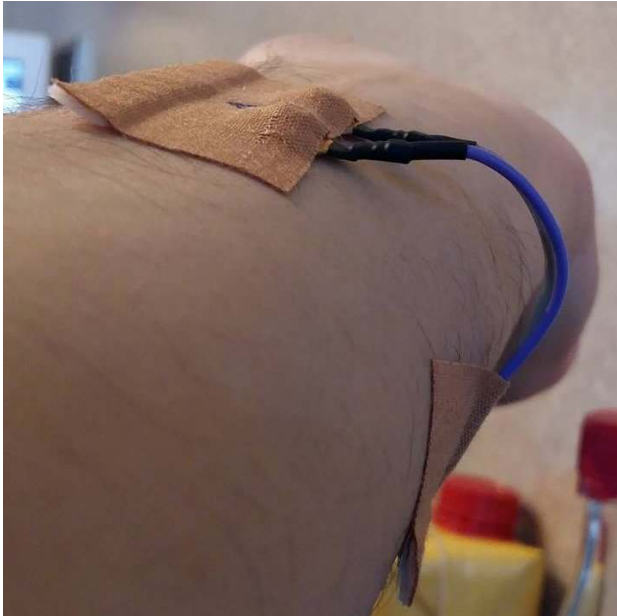
To receive all messages in time and avoid issues with buffer overflow, each message has to be received and saved with a sampling rate higher than 250Hz (that is, in less than $\frac{1}{250}s$). This is a key problem for preprocessing or plotting the data in real time. Multi-core processing could potentially alleviate this problem. The manufacturer of the board supplies a *Python* framework for the openBCI serial communication to implement a live plot using the *multiprocessing* library. However, we decided to not follow with the provided framework due to the constraints in the sampling time.

3.7 Bipolar Setting

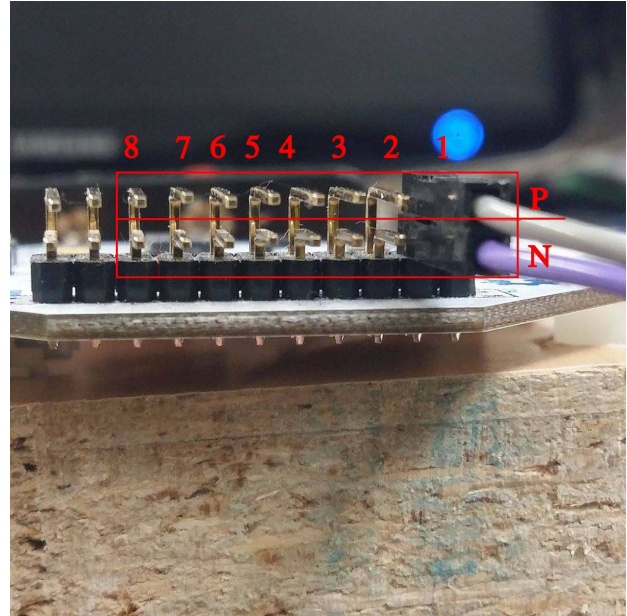
For the bipolar setting two electrodes per muscle are placed along its fibers (p. 9 [3]). One is the signal, the other the reference electrode. As shown in Figure 3.4b the signal electrodes are attached to the lower pin and the reference electrodes to the upper pin. Each electrode is fixed with two pieces of patch. One for the electrode itself and one to fix the cable to the skin. This avoids movements of the electrodes, thus the distance between electrode and skin remains constant.(Figure 3.4a) Although bipolar measurement should ideally avoid artifacts caused by 50 Hz power supplies the participant is placed at least 50 cm from these artifact sources. [12] The electrodes where connected to the xN and xP channel, where x is the number of the channel.

3.8 Electrode Placement

Before applying the electrodes to the skin, it was cleaned with antiseptic solution to remove the fat layer off the skin, which leads to a better conductivity(see [3] page 15). In this thesis, the standard gold cup electrodes have been used and applied as recommended by the manufacturer [5]. Each electrode was fixed with two pieces of patch. One to fix the electrode and another one to prevent the wire from shaking and unclamping (see Figure 3.4a).



(a) Attachment of two electrodes for measuring one muscle.



(b) Channel one. (plugged bipolar. The channel order goes from the right to the left) The upper row are the xP (signal) pins and the lower pins are the xN (reference) pins

Figure 3.4: Setup

4 Experiment

In the previous chapters the preparation of the measurement of EMG signals and the theoretical methods for classification were explained. This chapter aims at validating and comparing the classifiers using real EMG data obtained from the openBCI system.

4.1 Features

In this thesis, three feature spaces N , D and R were used to learn different movements. N is the originally dimension of the signal (in feature space N , each y value of the EMG signal is handled as a feature), D the weight space (see chapter 3.1) and R the dimension of the RMS signal with a window length of l . Equation (4.1) relates the size of each of them. Let n_{ch} be the number of measured channels and t the time of the measurement in seconds. $n_{weights}$ is the number of weights, which can be freely chosen.

$$\begin{aligned} N &= 250 \cdot t \cdot n_{ch} \\ D &= n_{weights} \cdot n_{ch} \\ R &= \frac{N}{l} \cdot n_{ch} \end{aligned} \tag{4.1}$$

4.2 Finding Muscle Locations

Amputees who lost their hands still have forearm muscles, which are partly responsible for finger and hand wrist movements. Thus, these muscles are suitable to activate control actions on a prosthetic limb. Ideally, the patient must be able to use the same muscles he/she used to contract before the loss of the hand, leading to a natural interaction with the prosthetic device. Passive surface electrodes are the most cost-effective and the fastest option to get EMG signals.

However, Surface EMG (SEMG) is limited in the sense that is only possible to measure superficial muscles. SEMG signals are, therefore, corrupted by the skin and its physical condition. Table 4.1 shows the superficial muscles which are responsible for hand movements [18]. The Flexor Digitorum Sublimis is, however, an exception as it is partly covered by the Brachio Radialis and Pronator Teres and the tendons of Palmaris Longus and Flexor Carpi Radialis (see Figure 4.1). Since the tendons of the Palmaris Longus and Flexor Carpi Radialis do not affect EMG signals, electrodes placed over these areas are useful to measure the activity of the Flexor Digitorum Sublimis muscle. Patch *no. 1* in Figure 4.2a shows the position of them. The first challenge was to find the given muscles.

In order to find the location of the muscles, the participant's arm had to be compared to an anatomic figure of the forearm muscles such as the one in Figure 4.1. Empirically, we found that the most effective method for finding a muscle is to find the tendon next to the hand wrist, and then to flex the muscle by executing the pronounced movements. This makes it possible to feel the shape of the muscle by moving along the tendon until the muscle begins. After the supposed muscle has been found, two electrodes have been attached according to the bipolar setting on the center of the muscle along the fibers. Subsequently the muscle was measured for different movements. A strong indication that the right muscle has been found is given when the expected movements matched with the measured ones.

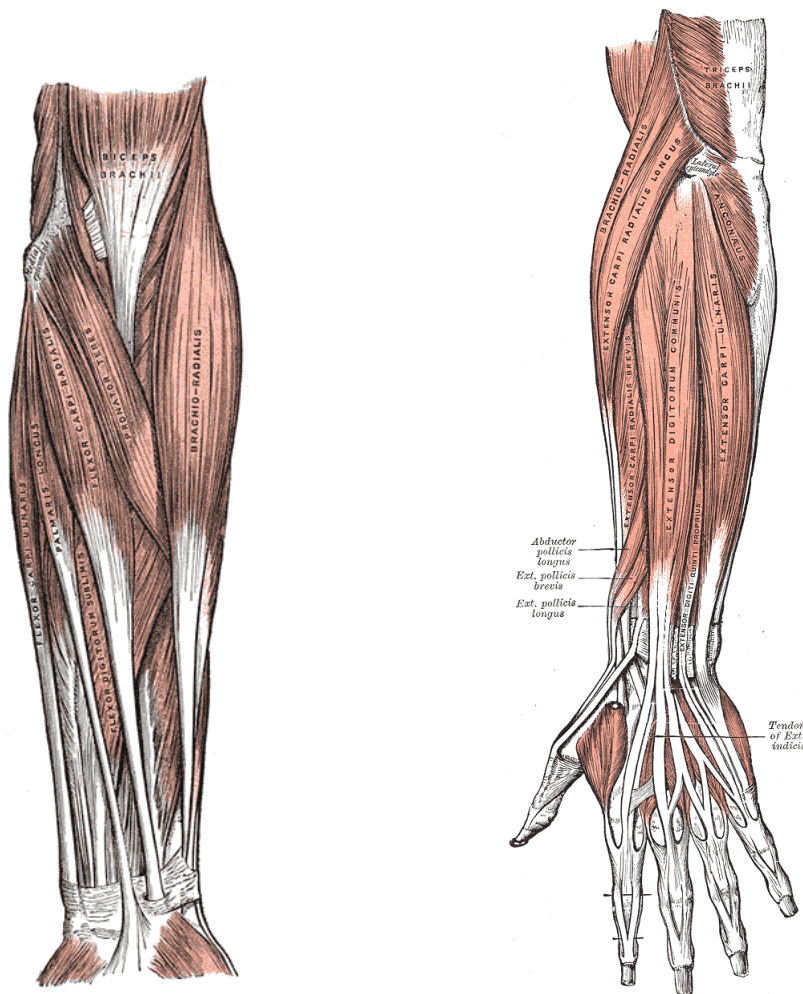
If the measured correlations between muscle and movements comply with correlations of Table 4.1, it is likely that the right muscle is found. It seemed to be the most convenient approach to find useful muscles. Table 4.2 shows the empirically determined correlations between muscles and movements. Table 4.1 compared to table 4.2 shows that not every muscle could be measured with the given instruments. Flexor carpi ulnaris could not be measured. On the other hand the signals should be amenable to classification, since no movement has the same muscle combination.

muscle/function	elbow: flexion	radioulnar joint: pronation (pronation rotating inwards)	handwrist: palmarflexion	handwrist: ulnar abduction	handwrist: radial abduction	spans aponeurosis palmaris	second phalanx of fingers 2-5: palmarflexion	fingers 2-5: dorsal extension	handwrist: dorsal extension
flexor carpi radialis	X	X	X		X				
palmaris longus	X	X	X			X			
pronator teres	X	X							
flexor digitorum superficial			X				X		
flexor carpi ulnaris			X	X					
extensor digitorum communis								X	X

Table 4.1: This table shows the relation between the muscles and their movements. The muscles can be found in Figure 4.1

muscle/function	elbow: flexion	radioulnar joint: pronation (pronation = rotating inwards)	handwrist: palmarflexion	handwrist: ulnar abduction	handwrist: radial abduction	spans aponeurosis palmaris	second phalanx of fingers 2-5: palmarflexion	fingers 2-5: dorsal extension	handwrist: dorsal extension
Flexor Carpi Radialis	-	-	-	-	-	-	-	-	-
Palmaris Longus							X		
Pronator Teres		X							
Flexor Digitorum Superficial							X		
Flexor Carpi Ulnaris	-	-	-	-	-	-	-	-	-
Extensor Digitorum Communis								X	X

Table 4.2: This table shows the empirical determined relation between the muscles and their movements. The muscles can be found in Figure 4.1. Flexor Carpi Radialis and Flexor Carpi Ulnaris could not be measured in this setup.

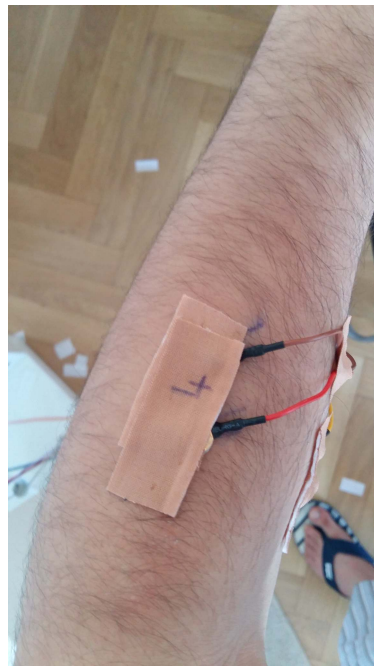


(a) superficial muscles. bottom view of the forearm (b) superficial muscles. top view of the forearm

Figure 4.1: Anatomy of the forearm muscles [6].



(a) The electrode placement. Bottom view.



(b) The electrode placement. Top view.

Figure 4.2: This figure shows the electrode placement:

1. Flexor Digitorum Superficial
2. Palmaris Longus
3. Pronator Teres
4. Extensor Digitorum communis

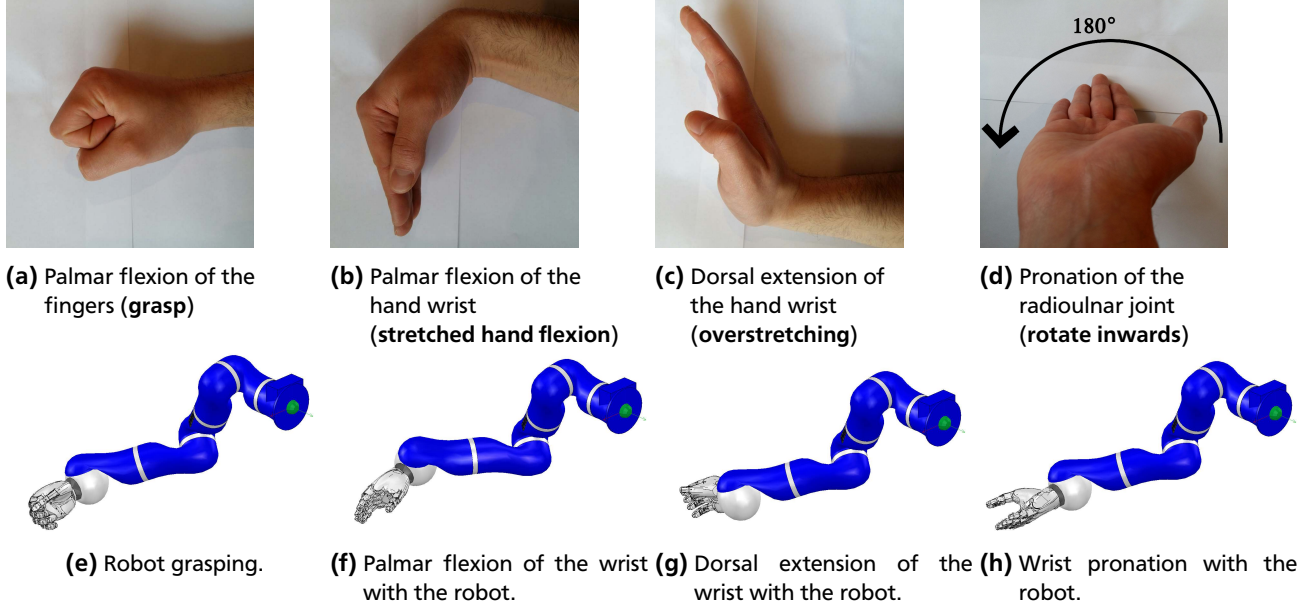


Figure 4.3: The four different hand movements measured and classified during the experiment. The bottom row shows the associated robot motions.

4.3 Forearm Setup

Based on the results of the previous experiment, the muscles Flexor Digitorum Superficial, Palmaris Longus, Pronator Teres and Extensor Digitorum communis have been measured simultaneously, thus $n_{ch} = 4$. Each muscle was measured in the bipolar setting (see Chapter 3.7).

To minimize interference, the participant was sit in a room with minimal electronic equipment (see layout of the experiment location in Figure 4.7a). The participant was located in room L2, sitting on a chair. The *OpenBCI* board was placed next to the participant. During the experiment four types of movement were measured: **grasp**, **stretched hand flexion**, **overstretching** and **rotate inwards**. Figure 4.3 illustrates the different movements. Each movement was measured 20 times. The duration was set to $t = 3$. Outliers were sorted out by visual inspection after each measurement.

After the measurements the training data consisted of one 3000×20 matrix for each movement. Due to the small training number the data has been classified with different classifiers using leave-one-out-cross-validation. The cross validation was applied for the different combinations, preprocessing steps and classifier shown in Figure 4.4.

The $N = 750$ dimensional feature space per channel has been reduced to a $d = 20$ dimensional weight space per channel using the linear regression on RBFs as described in Chapter 2.1. As *radial basis functions* Gaussian PDFs have been used. The centers of these PDFs were evenly distributed over the time axis and the variance σ^2 of each Gaussian was set to 0.05.(see Figure 4.5). Figure 4.6 illustrates how the sum of Gaussians fits the bandpass filtered EMG signal of the grasping movement by the Flexor Digitorum Superficial. Now we have two different feature spaces for each movement. Each classifier will classify the N dimensional signal and the d dimensional signal, where the d is the dimension of a vector of weights.

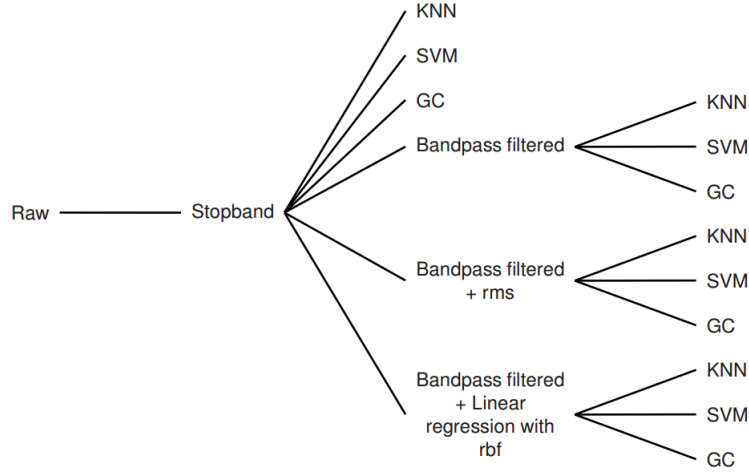


Figure 4.4: This figure shows the different processing and classification combinations. Each of the 12 paths is one combination. Each signal is filtered by a stopband with the cutoff frequencies 45Hz and 50Hz. KNN has been applied for $K \in \{1, 2, 4, 8\}$.

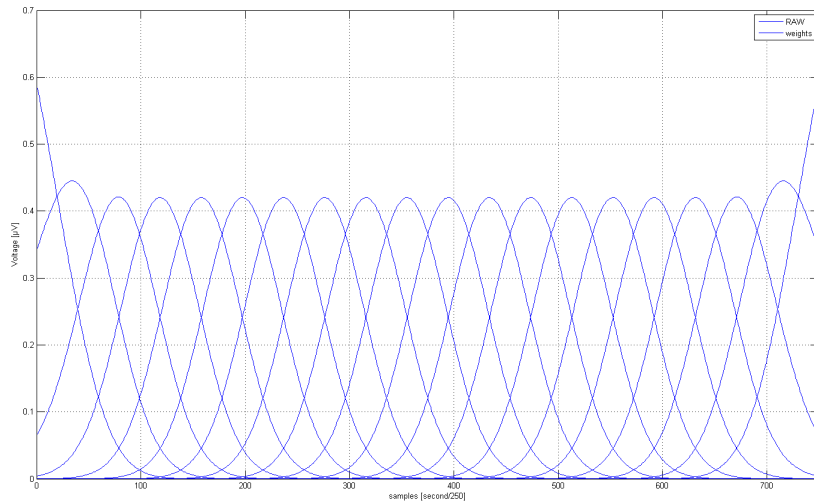


Figure 4.5: This figure shows the $\Phi(x)$ functions without the weights multiplied. The x-axis represents the number of samples and the y-axis the voltage in μV . The number of $\Phi(x)$ is $d = 20$. Each $\Phi(x)$ is a Gaussian distribution with $\sigma^2 = 0.05$. The expectations are evenly distributed over the x-axis.

4.3.1 Sources of Interference

During the experiments it was observed that the measured signal had great sensitivity to external noise. As shown in Figure 4.7b, when the participant had contact in any kind of plugged electric cable, the signal was overlaid with a 50Hz noise. Using the advantages of having a wireless device allowed us to change the location of the participant. Figure 4.7 shows how the signal behaved depending on the location. It shows that it is not possible to measure clean data under realistic conditions. The noise has to be filtered out.

It was also noticed that the monopolar setting was extremely sensitive to movements of the cables. This problem was greatly decreased with a bipolar setting.

An issue which occurred in every measurement was that the electrodes began to detach over time, thus they had to be pressed against the skin. To avoid the electrodes from falling completely of the skin, they were fixed with an extra piece

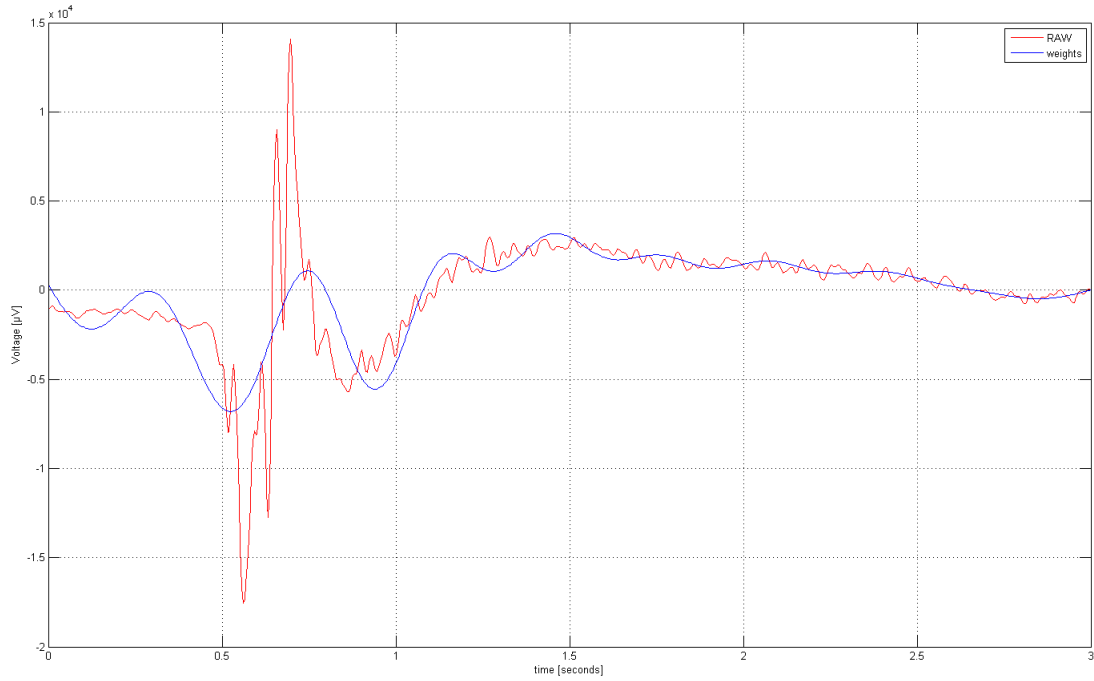
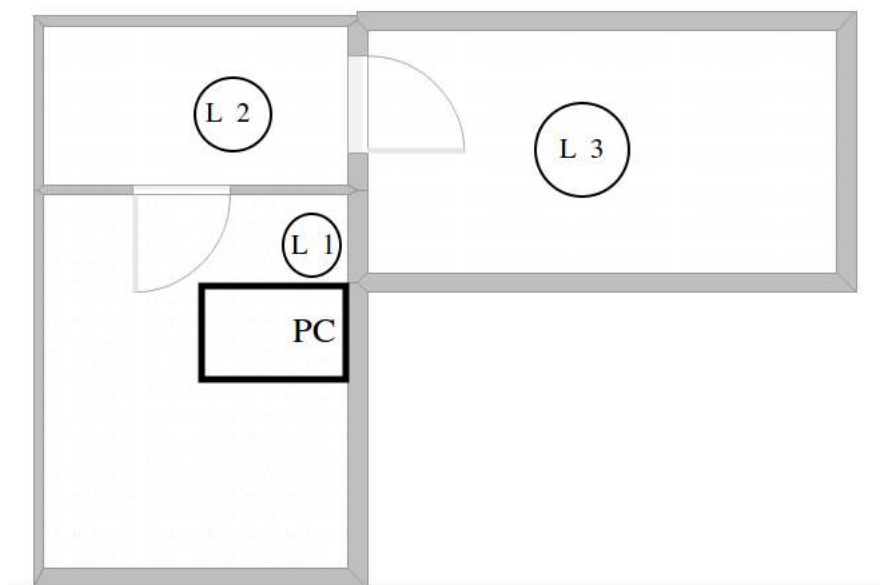


Figure 4.6: This figure compares the RAW EMG signal compared to the fitted curve $\mathbf{w}^\top \Phi(\mathbf{x})$

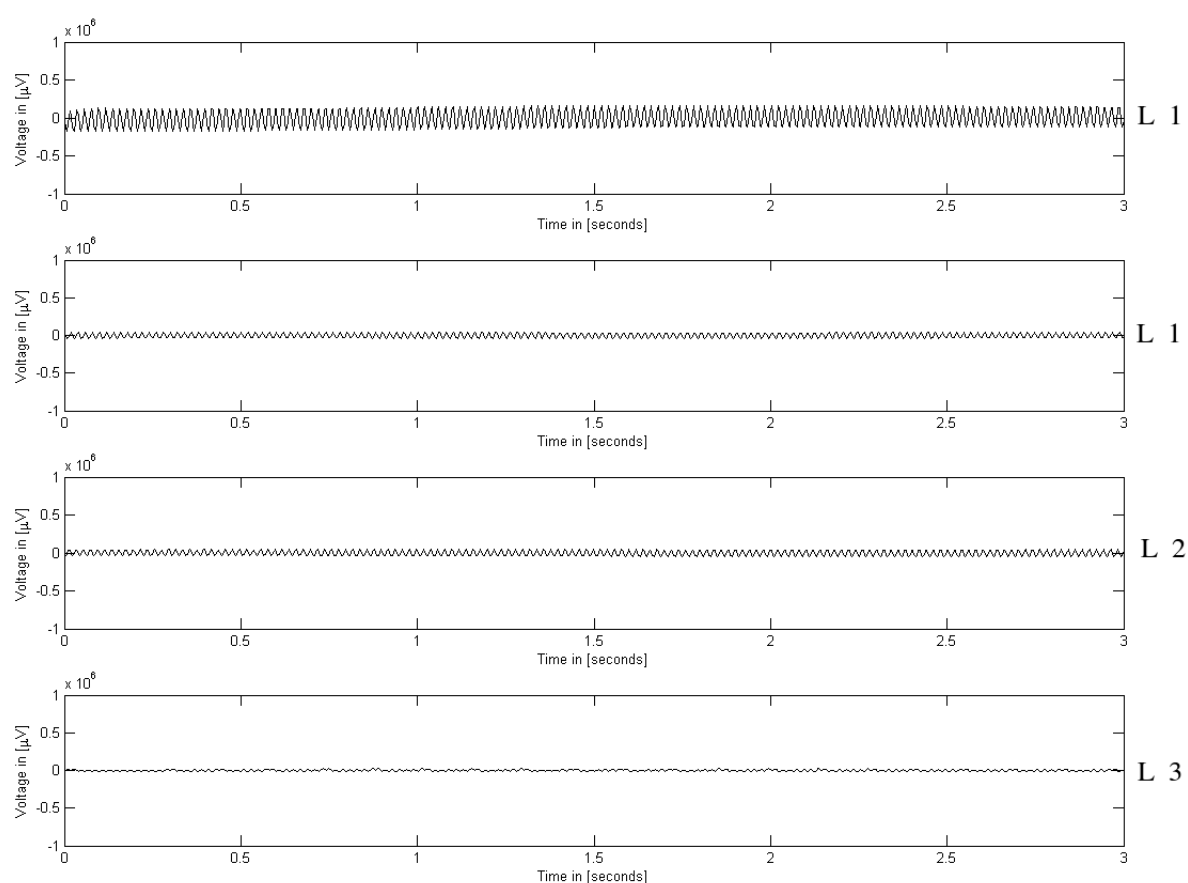
of patch on the wire.

Two filters have been applied. At first a butterworth band-stop filter with the cutoff frequencies $45Hz$ and $55Hz$ to eliminate the power supply frequency of $50Hz$. Subsequently a butterworth bandpass has been applied. The ideal parameters for the cutoff frequencies have been determined empirically and will be discussed in the next chapter.

A commonly used preprocessing step of EMG signals is the computation of the root mean square over a time window [3]. The ideal parameter for the window length has been determined empirically and will be discussed in the next chapters.



(a) This figure illustrates the three different locations. L1 is in a room, which contains power supplies. L2 is in the room next to L1 and contains no power supplies. L3 is outside of the flat in the staircase where power supplies were removed.



(b) Each graph shows the noise of different locations. They first two graphs have been measured at the same location. The upper one has been measured while the participant was in contact with a plugged electric cable. Flexor Digitorum Superficial has been measured. The participant did not move.

Figure 4.7: Different noise levels depending on the location.

4.4 Results

The graphs in Figure 4.8 show the raw signals of the complete dataset measured in the forearm setup. The columns represent the different movements in the following order (left to right): **grasp, stretched hand flexion, overstretching, rotate inwards**. The rows are the muscles in the following order (top to down): **Flexor Digitorum Superficial, Palmari Longus, Pronator Teres, Extensor Digitorum communis**. Each graph shows the voltage in μV depending on the number of samples. Each color represents one of 20 trials. One can see that the signals are aligned well with respect to the x-axis. However they are not aligned with respect to the y-axis. Also, during some trials, for example the blue curve on the subplot of the third row, first column, tend to drift. Despite the observed disadvantages of the raw dataset, the accuracy for this dataset is the highest (see Figure 4.12). The reason will be discussed in the discussion section. Figure 4.9 illustrates the bandpass filtered dataset with the cutoff frequencies $0.2Hz$ and $124Hz$. Compared to the raw dataset the trials are now aligned with respect to the y-axis. The DC frequency components and low frequencies of the drift were removed. The accuracy of this dataset is less than the accuracy of the raw data (see Figure 4.12).

Figure 4.10 shows the raw data after a bandpass set between cutoff frequencies of $10Hz$ and $124Hz$. This frequency range represents the actual EMG spectrum of the signal. As it will be shown, however, classification based on this measurement had a significant lower accuracy than the two the datasets mentioned before [3] (see Figure 4.13). Note, however, that undesirable effects caused by movement artifacts, such as the relative displacement of the electrode on the skin, are removed with the cutoff of $10Hz$, and therefore this setting better extracts the actual EMG signal from the raw data.

Applying RMS over a window with a length of 30 provides the signals illustrated in Figure 4.11. The signals are now positive. The curves have been rectified. Also the accuracy is higher than the exclusively bandpass filtered signals. Given the preprocessed measurements, the accuracy of different classifiers can now be evaluated. In the following Figures 4.12 and 4.13, the accuracy of the raw data (shown as the blue bar) is compared to the different filter options. Figure 4.12 shows the accuracies for the bandpass filtered dataset with the cutoff frequencies of $0.2Hz$ and $60Hz$. The classification on the raw data is by far higher for all used classifiers. KNN outperforms SVM and GC for $K \in \{1, 2, 4, 8\}$ on classifying the raw data. Figure 4.13 shows the accuracies for the bandpass filtered dataset with the cutoff frequencies of $10Hz$ and $60Hz$. The classification on the raw data is by far higher for all used classifiers. The accuracy of the bandpass filtered data is poor, while the RMS accuracy remains high.

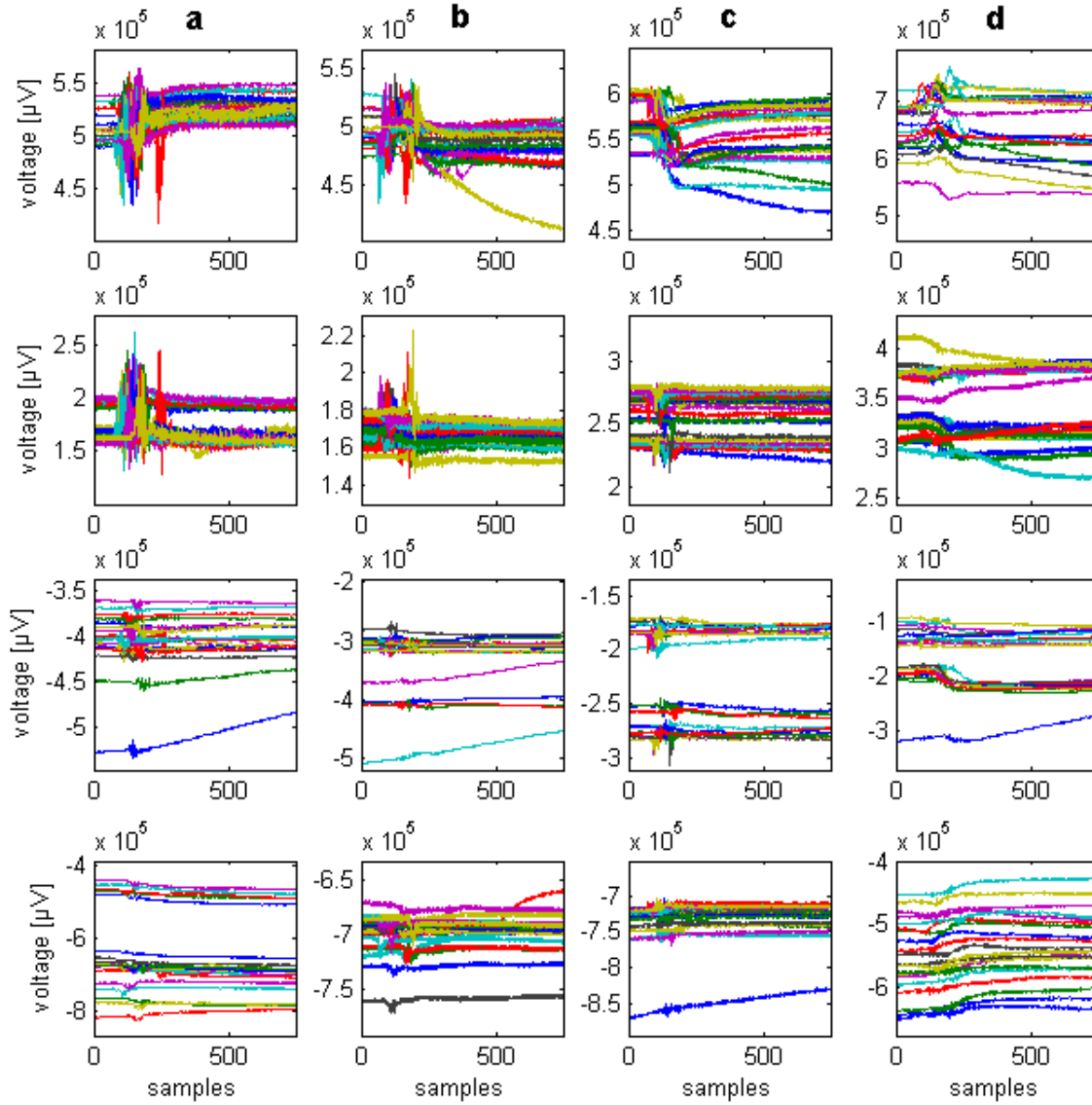


Figure 4.8: This figure illustrates the **raw** measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of 250Hz, which gives a duration of 3 seconds. *note:* Each color represents a trial.

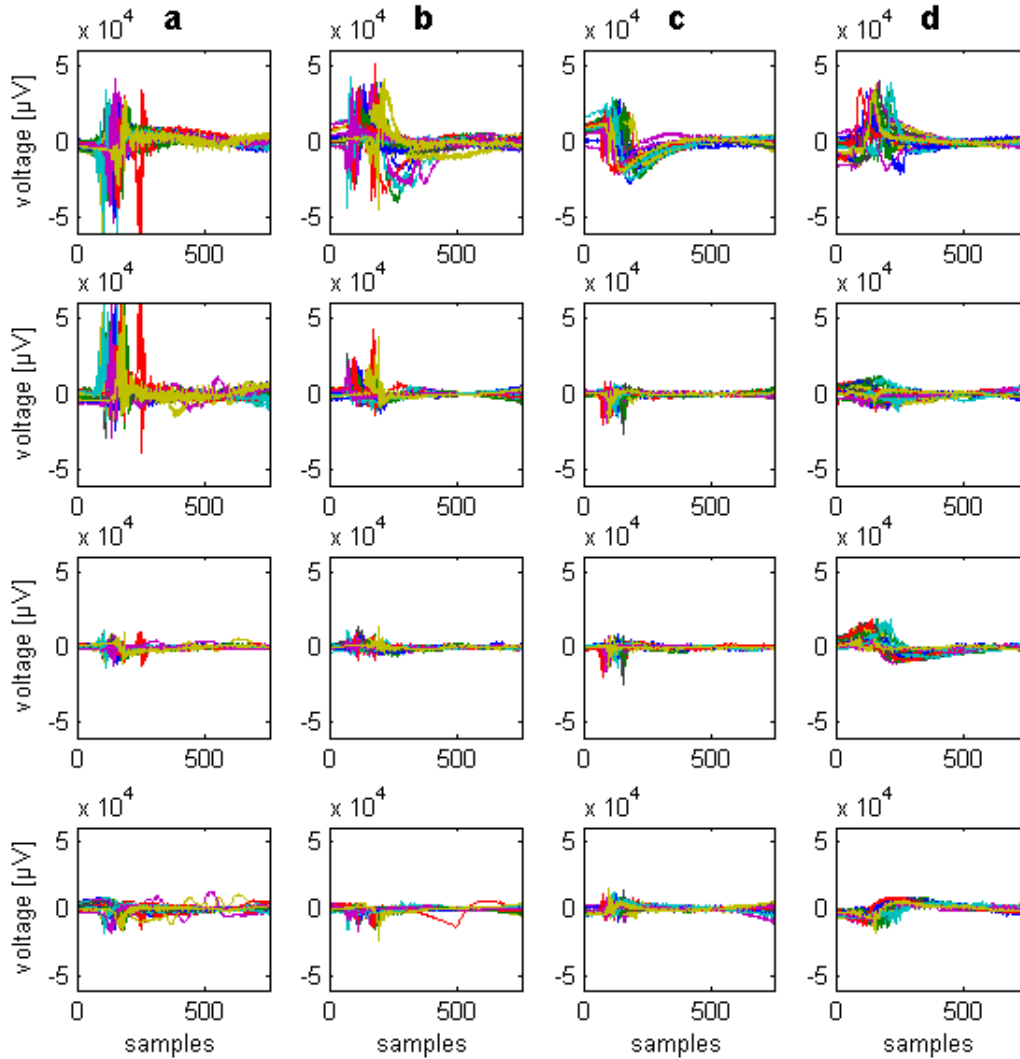


Figure 4.9: This figure illustrates the **bandpass filtered** measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of 250Hz , which gives a duration of 3 seconds. *note:* Each color represents a trial. Cutoff frequencies of the bandpass filter: 0.2Hz and 124Hz

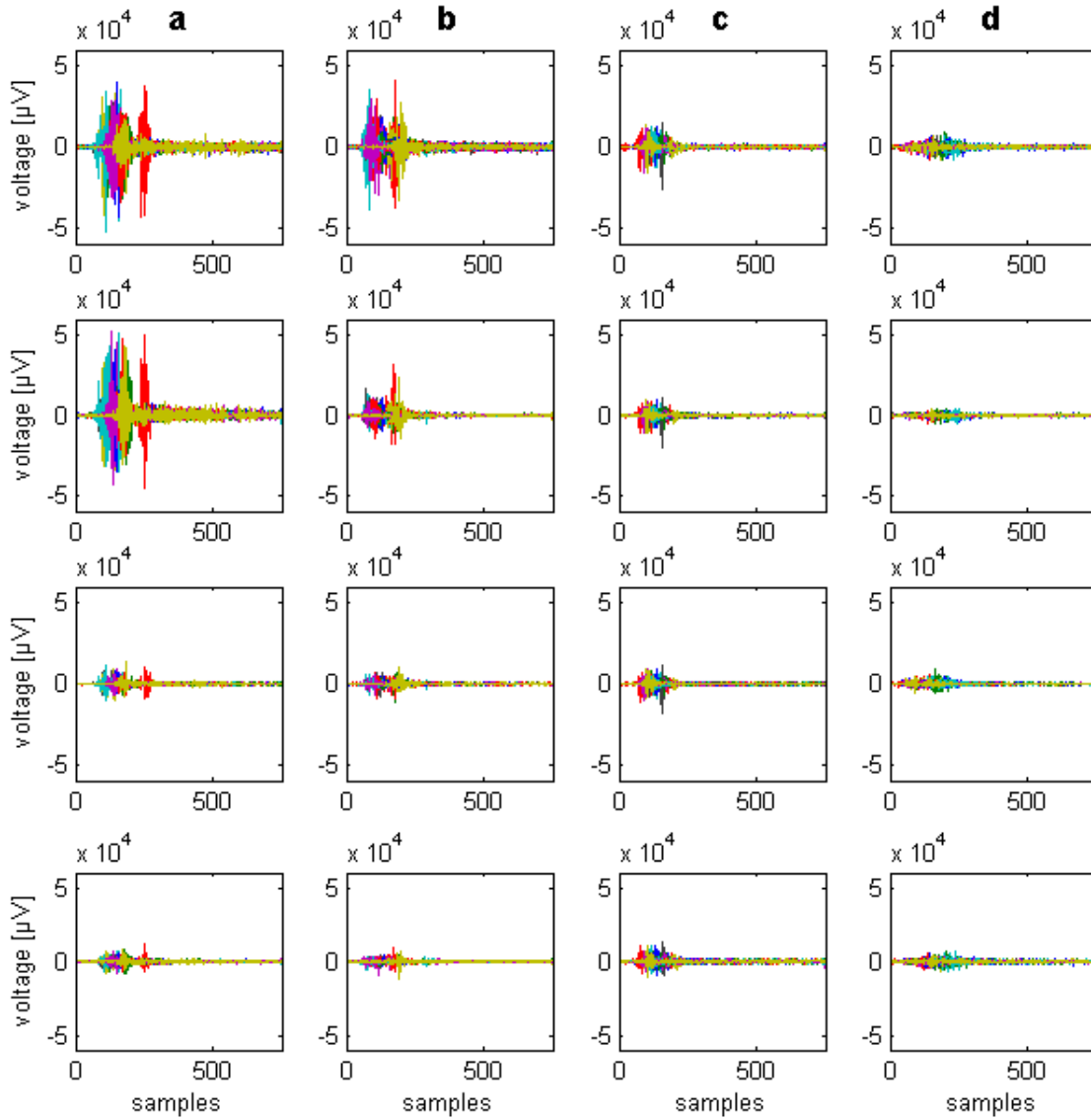


Figure 4.10: This figure illustrates the **bandpass filtered** measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of $250Hz$, which gives a duration of 3 seconds. *note:* Each color represents a trial. Cutoff frequencies of the bandpass filter: $10Hz$ and $124Hz$

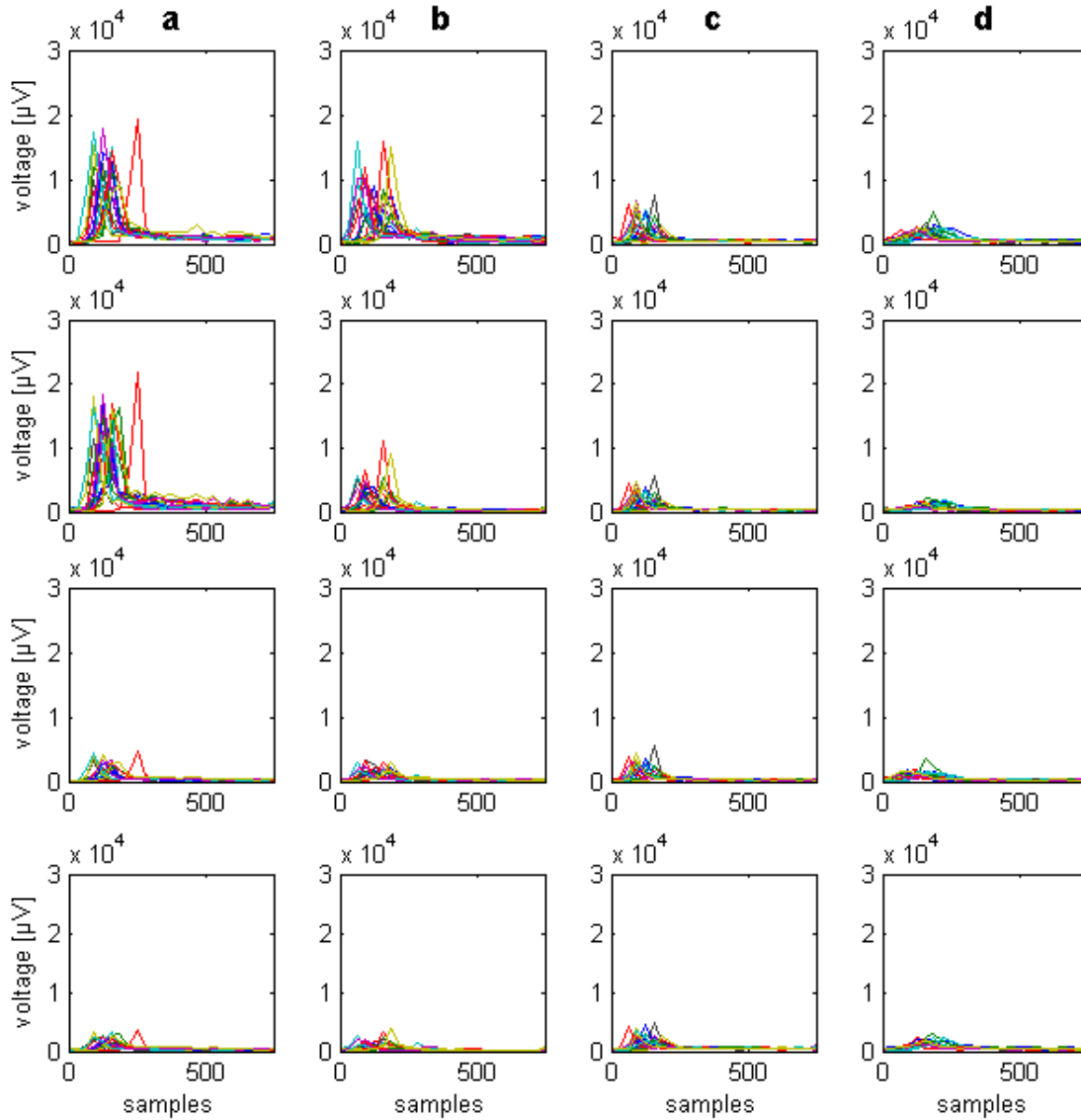


Figure 4.11: This figure illustrates the **bandpass filtered** and **RMS** measurements of the forearm setup. The rows represent the muscles in ascending order (see Figure 4.2). The columns represent the movements in alphabetically order (see Figure 4.3). Each graph shows the voltage depending on the samples. The number of sample is 750 with a sample rate of 250Hz , which gives a duration of 3 seconds. *note:* Each color represents a trial. Cutoff frequencies of the bandpass filter: 10Hz and 124Hz . Window length of the RMS: 30

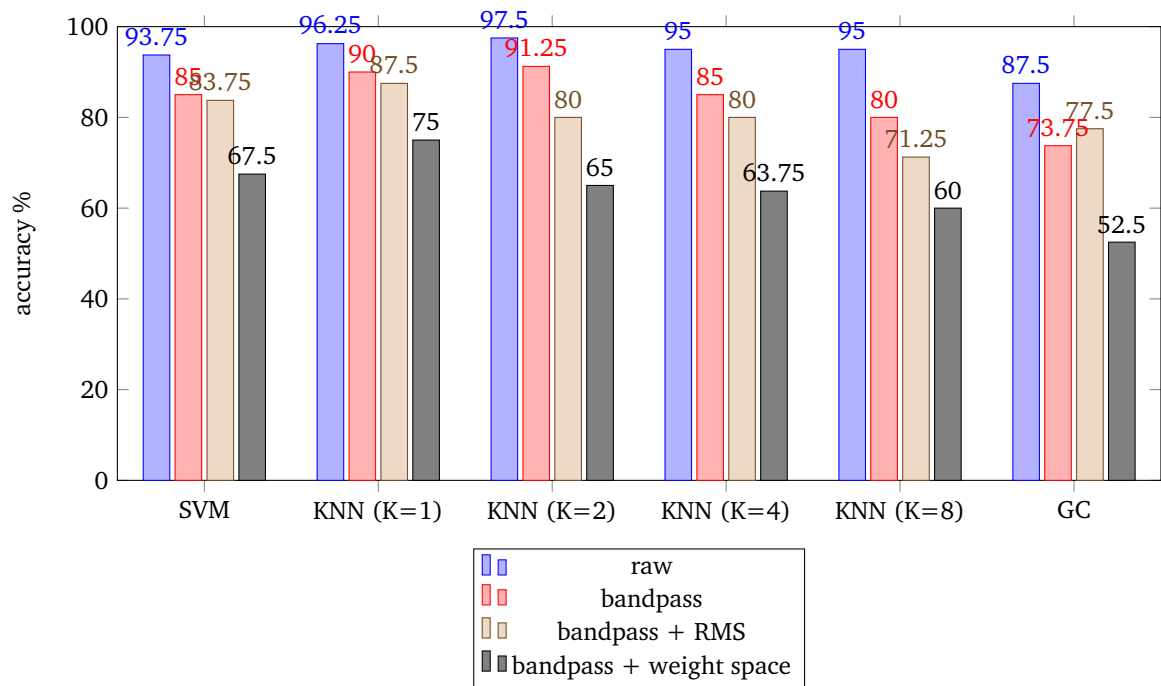


Figure 4.12: This figure illustrates the accuracies for the different used classifiers with the two preprocessing steps *bandpass* and *bandpass + weight space*. Filter parameter: 0.2Hz , 124Hz.

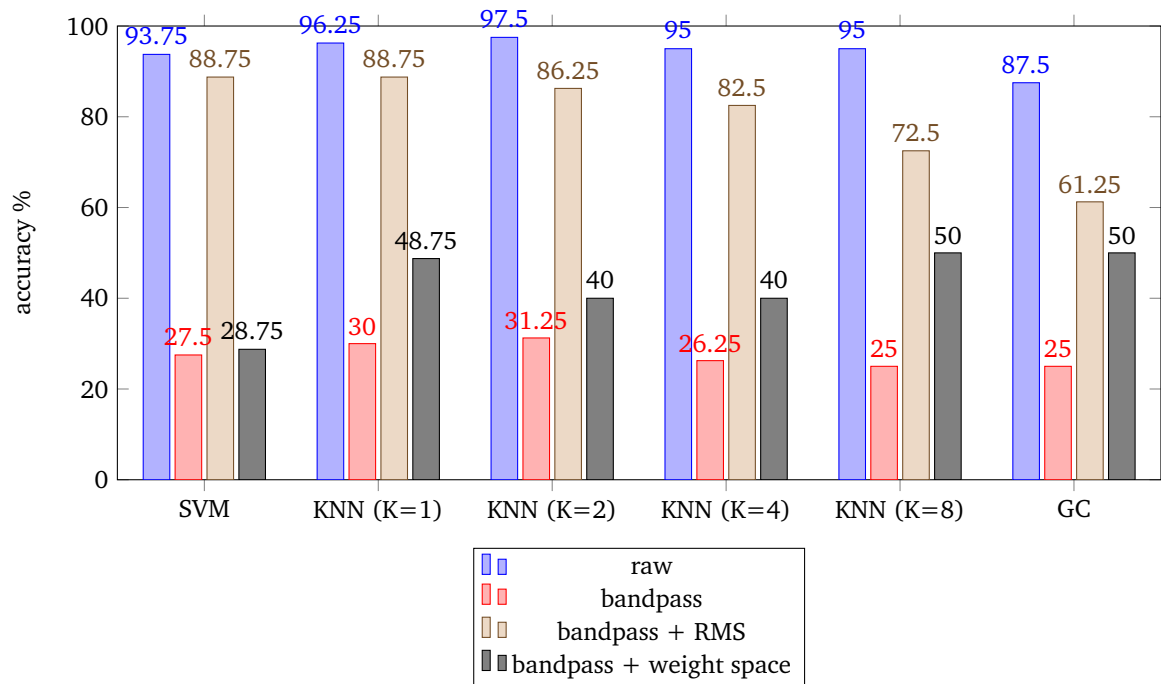


Figure 4.13: This figure illustrates the accuracies for the different used classifiers with the two preprocessing steps *bandpass* and *bandpass + weight space*. Filter parameter: 10Hz , 124Hz.

4.5 Discussion

In the previous section we observed correlations that will be discussed in the following section. We first discuss the results under the preprocessing steps of **bandpass** and **bandpass + RMS**. The accuracies for the bandpass filtered signal is greater than 73.75% for all classifiers, when the lower cutoff frequency is set to 0.2Hz. When the low cutoff frequency is increased to 10Hz (theoretically the lowest frequency of EMG signals), the highest accuracy amounts 31.25%. Considering the existence of four different classes, guessing a class would give an accuracy of 25%, thus the classifiers for the bandpass filtered signal are just above the uniform guess. Due to a frequency spectrum of 10Hz to 250Hz for EMG signals, this leads to the conclusion that the classifiers did not reach that high accuracies by classifying EMG signals. This result indicates that lower frequency phenomena (less than 10 Hz) is also important for improving classification rates. These low frequency signals could be related to the changes in amplitude that occur when the contact between the electrode and the skin vary as the arm moves.

Most of the classifiable information is between 0.2Hz and 10Hz. The bandpass filtered data cannot be used as data to classify on. Comparing it to the **bandpass + RMS** data shows, that the **bandpass + RMS** data is not significantly affected by the change of the lower cutoff frequency, thus the pipeline shown in Figure 4.14 is the optimal preprocessing for the EMG signals measured in this thesis.

Although classifying movement artifacts outputs high accuracies, it does not reflect the intended objective to classify EMG signals. Also there is no evidence that these movement artifacts are repeatable measurements. However EMG signals are scientifically proven.

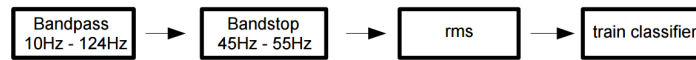


Figure 4.14: This figure illustrates the pipeline of the preprocessing steps and the training of the classifier.

The raw data provides high classification rates under LOOCV tests. Note, however, that the training data was collected in a single batch where movements were made in repetitive sequences. Therefore, when classifying on raw data, drifts and offsets were also taken into account. These features are, however, not repeatable. That is, if new tests are made on a second batch or on a second day it is unlikely that the same offset and drift will occur. On the other hand, detrended data where drifts and DC components are filtered out may provide a long term trained model. This makes the use of raw data difficult in practice.

It is interesting to note from Figure 4.12, however, that the nearest neighbor ($K=1$) is slightly worse than KNN with $K=2$. This indicates that the classifier is being affected by noise. The SVM achieves comparable performance in raw data with KNN with four and eight neighbors. Considering that SVMs are computationally intense during training but not during classification, the advantages of SVM in relation to KNN seem to be: no parameter tuning K , and low computational cost during classification.

Due to the issues of drift and shifts in data that is observed in raw data, the **bandpass filtered + RMS** will be used for more detailed analysis. Therefore it is relevant to review the confusion matrices of this data. Let **grasp** = c_1 , **stretched hand flexion** = c_2 , **overstretching** = c_3 and **rotate inwards** = c_4 . For the KNN K was set 1,2,4 and 8. The SVM used the one-vs-rest tactic to classify multiple classes (p. 83 [4]). Figure 4.15 shows the confusion matrices for the six options.

On the basis of the confusion matrices the precisions shown in Figure 4.15 have been calculated[19]. Observing the accuracies in Figure 4.13 and the precisions in Figure 4.16 one can see, that SVM, KNN ($K=1$) and KNN ($K=2$) output the highest accuracies and average precisions. In this thesis the SVM, KNN ($K=1$) and KNN ($K=2$) have been empirically determined as the best of the applied classifiers for classifying EMG signals. As preprocessing steps a bandstop, a bandpass and the calculation provided satisfying results. Figure 4.17 clarifies the results.

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	20	0	0	0
act. c_2	0	18	0	2
act. c_3	0	0	15	5
act. c_4	0	1	1	18

(a) Confusion matrix of "SVM BandPass + RMS"

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	17	2	0	1
act. c_2	1	18	1	0
act. c_3	0	1	17	2
act. c_4	0	0	1	19

(c) Confusion matrix of "KNN BandPass + RMS($K = 1$)"

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	16	3	0	1
act. c_2	1	15	2	2
act. c_3	0	2	15	3
act. c_4	0	0	0	20

(e) Confusion matrix of "KNN BandPass + RMS($K = 4$)"

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	20	0	0	0
act. c_2	13	7	0	0
act. c_3	2	0	18	0
act. c_4	9	1	6	4

(b) Confusion matrix of "GC BandPass + RMS"

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	18	1	0	1
act. c_2	1	18	1	0
act. c_3	0	3	15	2
act. c_4	0	0	2	18

(d) Confusion matrix of "KNN BandPass + RMS($K = 2$)"

	pred. c_1	pred. c_2	pred. c_3	pred. c_4
act. c_1	15	4	0	1
act. c_2	1	11	6	2
act. c_3	0	1	12	7
act. c_4	0	0	0	20

(f) Confusion matrix of "KNN BandPass + RMS($K = 8$)"

Figure 4.15: Confusion matrices for SVM, GC, KNN ($K=1$), KNN ($K=2$), KNN ($K=4$) and KNN ($K=8$) on the bandpass filtered + RMS dataset.

pred. = predicted class, act. = actual class

c_1	1	c_1	1
c_2	0.9	c_2	0.35
c_3	0.75	c_3	0.9
c_4	0.9	c_4	0.2
average	0.8875	average	0.6125

(a) Precisions of "SVM BandPass + RMS"

c_1	0.85	c_1	0.9
c_2	0.9	c_2	0.9
c_3	0.85	c_3	0.75
c_4	0.95	c_4	0.9
average	0.8875	average	0.8625

(b) Precisions of "GC BandPass + RMS"

(c) Precisions of "KNN BandPass + RMS($K = 1$)"

c_1	0.8	c_1	0.75
c_2	0.75	c_2	0.55
c_3	0.75	c_3	0.6
c_4	1	c_4	1
average	0.825	average	0.725

(d) Precisions of "KNN BandPass + RMS($K = 2$)"

(e) Precisions of "KNN BandPass + RMS($K = 4$)"

(f) Precisions of "KNN BandPass + RMS($K = 8$)"

Figure 4.16: Precisions for SVM, KNN and GC with RMS.

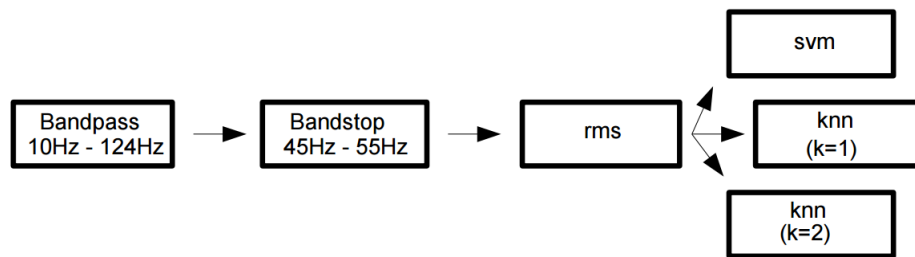


Figure 4.17: This figure illustrates the final preprocessing steps. SVM, KNN ($K=1$) and KNN ($K=2$) have been empirically determined as the best classifiers.

5 Conclusion and Future Work

Low-cost electromyography (EMG) devices like the *OpenBCI* board recently became popular and inexpensive. Also the massive deployment of 3D printers make the availability of cheap prosthetic devices inexpensive. Due to their low-cost both low budget BCIs and three dimensional printed prosthetic hands could be used as an inexpensive alternative to actuated prosthetics.

This thesis investigated whether and which machine learning classification methods provide useful classification rates given noisy and corrupted measurements from the low-cost *OpenBCI* board. The first step compared different measurement methods of EMG (chapter 2.3) signals. We presented the different preprocessing steps usually found in the EMG literature; namely, bandpass filtering, root mean square over a time window, and regression with radial basis functions (Chapter 3.1). For the classification of the EMG signals this thesis investigated K Nearest Neighbor (Chapter 3.2), Support Vector Machines (Chapter 3.3) and Gaussian classifiers (Chapter 3.4).

Using the *OpenBCI* board we have applied the methods of Chapter 3. It was necessary to investigate different approaches for the electrode placement. The locations of the surface muscles of the forearm had to be found and measured for different movements. Up to four muscles could be measured simultaneously (Chapter 4.3). Four different movements with the proposed methods were classified. Subsequently those classifiers were compared with respect to different preprocessing methods. The combinations **bandpass filtered + RMS + SVM**, **bandpass filtered + RMS + KNN (K = 1)** and **bandpass filtered + RMS + KNN (K = 2)** achieved accuracies from 86.25% to 88.75%, the most successful classifiers of this thesis (Chapter 4.5). Classification directly on raw data was not considered relevant as it relies on drift of signal and offset as features that have no repeatability over trials.

As future work, both the quality of the measurement and classification methods could be improved. For example, more sophisticated analysis and preprocessing methods could be investigated. To improve the quality of the signals an impedance test could be integrated into the routine of a measurement [3]. Also the use of active electrodes could increase the quality of the measured EMG signals [9]. For realistic online use, implementation of the code in C/C++ is desirable. By reducing the computational time, online learning could be considered. This is an important feature when using low-cost devices as the signals tend to show low repeatability over trials, and therefore, an adaptive method to estimate the new parameters becomes essential. SVM using kernels could be considered. Since the training set was small, a bigger training set allows to use cross-validation to improve parameters. In that case, measurements should be repeatable over long period of time. This could be achieved by elaborating a way to place the electrodes at the same locations over multiple measurement sessions, which was, in fact, one of the main difficulties regarding the use of the hardware.

Bibliography

- [1] C. W. Clauss, Cornelia, *Humanbiologie kompakt*. Springer, 2009.
- [2] N. muscular research center, “Physiological introduction to the motor unit.”
- [3] P. Konrad, “The ABC of EMG,” *A practical introduction to kinesiological electromyography*, vol. 1, 2005.
- [4] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.
- [5] Conor Russomanno, “Openbci.” <http://www.openbci.com>, 2015. Online; accessed December 2015.
- [6] H. Gray, *Anatomy of the human body*. Lea & Febiger, 1918.
- [7] “Enabling the future.” <http://enablingthefuture.org/>. Online; Last accessed: 28 February 2016.
- [8] P. Shenoy, K. J. Miller, B. Crawford, and R. P. Rao, “Online electromyographic control of a robotic prosthesis,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 3, pp. 1128–1135, 2008.
- [9] S. Bitzer and P. Van Der Smagt, “Learning EMG control of a robotic hand: towards active prostheses,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2819–2823, IEEE, 2006.
- [10] Texas Instruments, “Ads1299.” <http://www.ti.com/lit/ds/symlink/ads1299.pdf>, 2015. Online; accessed December 2015.
- [11] bio-medical, “Biomed.” <http://bio-medical.com/products/home-units/emg-incontinence-devices.html>, 2015. Online; accessed December 2015.
- [12] J. Ohashi, “Difference in changes of surface emg during low-level static contraction between monopolar and bipolar lead,” *Applied Human Science*, vol. 14, no. 2, pp. 79–88, 1995.
- [13] NR Sign, “Monopolar bipolar.” <http://www.nrsign.com/monopolar-vs-bipolar-emg-readings/>, 2015. Online; accessed December 2015.
- [14] R. G. Mello, L. F. Oliveira, and J. Nadal, “Digital butterworth filter for subtracting noise from low magnitude surface electromyogram,” *Computer methods and programs in biomedicine*, vol. 87, no. 1, pp. 28–35, 2007.
- [15] I.-D. Nicolae, P.-M. Nicolae, and M.-Ş. Nicolae, “Tunning the parameters for the fft analysis of waveforms acquired from a power plant,” *Acta Electrotehnica*, vol. 56, no. 3, 2015.
- [16] C. M. Bishop, “Pattern recognition,” *Machine Learning*, 2006.
- [17] T. Elomaa and J. Rousu, “On decision boundaries of naive bayes in continuous domains,” in *Knowledge Discovery in Databases: PKDD 2003*, pp. 144–155, Springer, 2003.
- [18] H. Lippert, *Lehrbuch Anatomie: 184 Tabellen*. Elsevier, Urban&FischerVerlag, 2006.
- [19] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [20] W. Weißgerber, *Elektrotechnik für Ingenieure 2*. Springer, 2013.