VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ACTIVE PROSTHETIC HAND

ACTIVE PROSTHETIC HAND

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Maximilian Brenner, BSC.**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Vratislav Harabiš, Ph.D.**

SUPERVISOR

BRNO 2019

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Master's Thesis

Master's study field **Electrical, Electronic, Communication, Control and Biomedical Technology**
Department of Biomedical Engineering

Student: Maximilian Brenner, BSC. ***ID:*** 211004
Year of study: 1 ***Academic year:*** 2018/19

TITLE OF THESIS:

Active prostetic hand

INSTRUCTION:

1) Provide a brief overview of electromyography controlled active prosthesis of the upper hand. 2) Write a survey in the field of EMG sensing and EMG signal processing methods for prosthesis control. 3) Design, test and evaluate an appropriate technique for acquisition of an EMG signal from the upper hand. 4) Design and develop methods for EMG signal processing and recognition of movement of the hand. 5) The processed signal will be used to control a model or a prototype of the upper hand prosthesis. 6) The implemented methods must be validated using appropriate measurements and results must be discussed.

REFERENCE:

[1] MERLETTI, Roberto. a Philip PARKER. Electromyography: physiology, engineering, and noninvasive applications. Hoboken, NJ: IEEE/John Wiley, c2004. ISBN 9780471675808.
[2] DAMELIN, Steven B. a Willard. MILLER. The mathematics of signal processing. New York: Cambridge University Press, 2012. Cambridge texts in applied mathematics, 48. ISBN 9781107601048.

Assigment deadline: 4. 2. 2019 ***Submission deadline:*** 17. 5. 2019

Head of thesis: Ing. Vratislav Harabiš, Ph.D.

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií
Ústav biomedicínského inženýrství
-1-

Subject Council chairman

Abstract

BACKGROUND: Based on mainly vascular diseases and traumatic injuries, around 40,000 upper limb amputations are performed annually worldwide. The affected persons are strongly impaired in their physical abilities by such an intervention. Through myoelectric prostheses, affected persons are able to recover some of their abilities.

METHODS: In order to control such prostheses, a system is to be developed by which electromyographic (EMG) measurements on the upper extremities can be carried out. The data obtained in this way should then be processed to recognize different gestures. These EMG measurements are to be performed by means of a suitable microcontroller and afterwards processed and classified by adequate software. Finally, a model or prototype of a hand is to be created, which is controlled by means of the acquired data.

RESULTS: The signals from the upper extremities were picked up by four MyoWare sensors and transmitted to a computer via an Arduino Uno microcontroller. The Signals were processed in quantized time windows using Matlab. By means of a neural network, the gestures were recognized and displayed both graphically and by a prosthesis. The achieved recognition rate was up to 87% across all gestures.

CONCLUSION: With an increasing number of gestures to be detected, the functionality of a neural network exceeds that of any fuzzy logic concerning classification accuracy. The recognition rates fluctuated between the individual gestures. This indicates that further fine tuning is needed to better train the classification software. However, it demonstrated that relatively cheap hardware can be used to create a control system for upper extremity prostheses.

Keywords

Upper limb prosthesis, EMG control, signal acquisition, neural network, gesture recognition

Prohlášení

Prohlašuji, že svou závěrečnou práci na téma „Active prosthetic hand“ jsem vypracoval samostatně pod vedením vedoucího magisterský práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

I declare that my final thesis on "Active prosthetic hand" was elaborated independently under the guidance of the master’s thesis supervisor and using specialized literature and other information sources, all of which are quoted in the work and listed in the literature at the end of the thesis.

As the author of this final thesis I further declare that in connection with the creation of this final thesis I did not infringe the copyrights of third parties, in particular I did not interfere illegally with the foreign copyrights of personality and I am fully aware of the consequences of the Copyright Act No. 121/2000 Coll., including possible criminal law consequences resulting from the provisions of Part Two, Title VI. Part 4 of the Criminal Code No. 40/2009 Coll.

M. Bremer

Brno, May 2019 ....

Author's signature

Poděkování - Acknowledgments

I would like to thank my supervisor Ing. Vratislav Harabiš, Ph.D. for his support during the creation of this work. Through his accommodating and helpful nature, this work could be carried out successfully. In addition, I would also like to thank my family and friends for their help and support without which the realization of this work would not have been possible.

Table of Content

List of figures

List of tables

1. Introduction and Background

The following chapter explains the basics of the project as well as its background and gives a short introduction into the topic of prostheses and myoelectric sensing.

In the beginning, a brief description of the project is given. The reasons and the distribution of amputations are discussed, followed by an introduction of different prostheses, in particular myoelectric ones.

Afterwards, the further scope of this project as well as the emerging tasks are described in more detail.

1.1. Preamble

Many people worldwide suffer from the loss of a limb. Therefore, it is important to advance the development of intelligent prostheses in order to give these people a better life. With the help of myoelectric prostheses, affected persons can recover parts of their limb functionality. A significant step to achieve this is to develop cost-effective alternatives compared to conventional prostheses which can cost a substantial amount of money. Therefore, this work deals with the development of a low cost concept alternative.

With the device which is going to be developed it shell be possible to obtain electrical muscle signals from the upper extremities and to convert these into movement of a prosthesis. The muscle signals are to be taken simultaneously at several positions of the arm. To record these signals, a suitable microcontroller should be used so that they can be processed and filtered afterwards. The resulting cleaned signals are then to be used to control, for example, a generatively fabricated prosthesis or computer model. For this, the individual signal patterns must be classified and assigned to specific gestures. The project is completed by means of an analysis regarding the success rate in the detection of different movement patterns.

With the help of this work it should be possible in the future to build cheap myoelectric prostheses by means of simple and easy to acquire components.

1.2. Characteristics of limb amputations

An amputation can be defined as the "*Removal of part or all of a body part that is enclosed by skin. Amputation can occur at an accident site, the scene of an animal attack, or a battlefield. Amputation is also performed as a surgical procedure. It is typically performed to prevent the spread of gangrene as a complication of frostbite, injury, diabetes, arteriosclerosis, or any other illness that impairs blood circulation. It is also performed to prevent the spread of bone cancer and to curtail loss of blood and infection in a person who has suffered severe, irreparable damage to a limb.*" [1]

Around 1.5 ‰ of the total world's population are affected by such an amputation of a limb. This corresponds to around 10 million people who suffer from the loss of a body part whereby 30% of those amputations affect the upper extremities. Almost 80% of these 3 million arm amputees are people living in developing countries. [2]

The cause of such measures is usually an arterial circulatory disorder. The affected part has then to be removed because otherwise the life of the patient could be at risk due to dying tissue. Another reason for an amputation of a limb may be a traumatic injury at which the affected part of the body cannot be rescued. The third main reason for amputations is tissue damage due to malignant ulcers.

Finally, in exceptional cases, an amputation can be the outcome of a punishment. However, such punishments are only practiced in a few countries such as Saudi Arabia, United Arab Emirates or Iran and are very rare. [3]

The reasons for amputations differ greatly between different age groups as well as between industrialized nations and developing countries. For example, in industrialized countries about 80% of all amputations are due to vascular diseases, whereby in developing countries only 20% are caused by these. On the other hand, traumas with about 20-30% prevalence are much more common as cause in developing countries compared to industrialized nations with 5%. These divergences are especially high when considering infections. For these, the ratio is 3-5% in first world countries, compared to around 20% in developing countries. [4]

Due to the higher life expectancy in higher developed countries, people are usually older when undergoing amputations and the cause are lifestyle-related illnesses.

In contrast, people from low-income countries are mostly affected by amputation reasons that are not due to lifestyle-related diseases. Due to lower occupational safety, riots and poorer healthcare, traumatic amputations are more common. [5]

The following table shows the cause-to-age-dependency relationship in industrialized countries.

Table 1: Etiology dependency due to age

Age at amputation (years)	Arterial occlusive diseases	Trauma	Tumor
0-20	<1%	90%	5-10%
20-60	30%	60%	5-10%
60+	90%	5%	5-10%

Such an amputation of a limb can severely affect a person's autonomy, depending on the height of amputation and the lost body part. In some cases an amputation can be equivalent to a severe disability. In addition, such an intervention has a strong impact on the psyche of those affected. Therefore, prostheses need to be used to recover parts of these body functions and to help the ones affected to have a normal life. [3]

1.3. Common upper limb amputations

There are several heights at which an amputation of the upper limb is normally carried out. Those positions start at the fingers and go all the way up to the shoulder. The most common ones are listed below [6] [7]:

- **Fingers/ Metacarpal**: Amputation of finger segments or parts of the metacarpal bones.
- **Wrist disarticulation**: Surgery at the wrist whereby both, radius and ulna are not affected. Leaves a relative long residual limb which is suitable for mounting of aids.
- **Transradial**: Transradial is also known as "below the elbow" whereby the amputation takes place through the radius and ulna. The length of the residual limb is important to allow control over pro- and supination.
- **Elbow disarticulation:** The amputation is carried out in such a way, that the entire humerus is maintained. The surgery is through the elbow joint and the lower arm is removed.
- **Transhumeral:** Transhumeral amputation is also known as "above the elbow" whereby the amputation takes place through the humerus.

- **Shoulder disarticulation:** Shoulder disarticulation describes an amputation at the height of the shoulder. The scapula remains, but the clavicle may or may not be removed.
- **Forequarter (intrascapulothoracic) amputation:** During this amputation, the humerus, scapula and clavicle are removed.

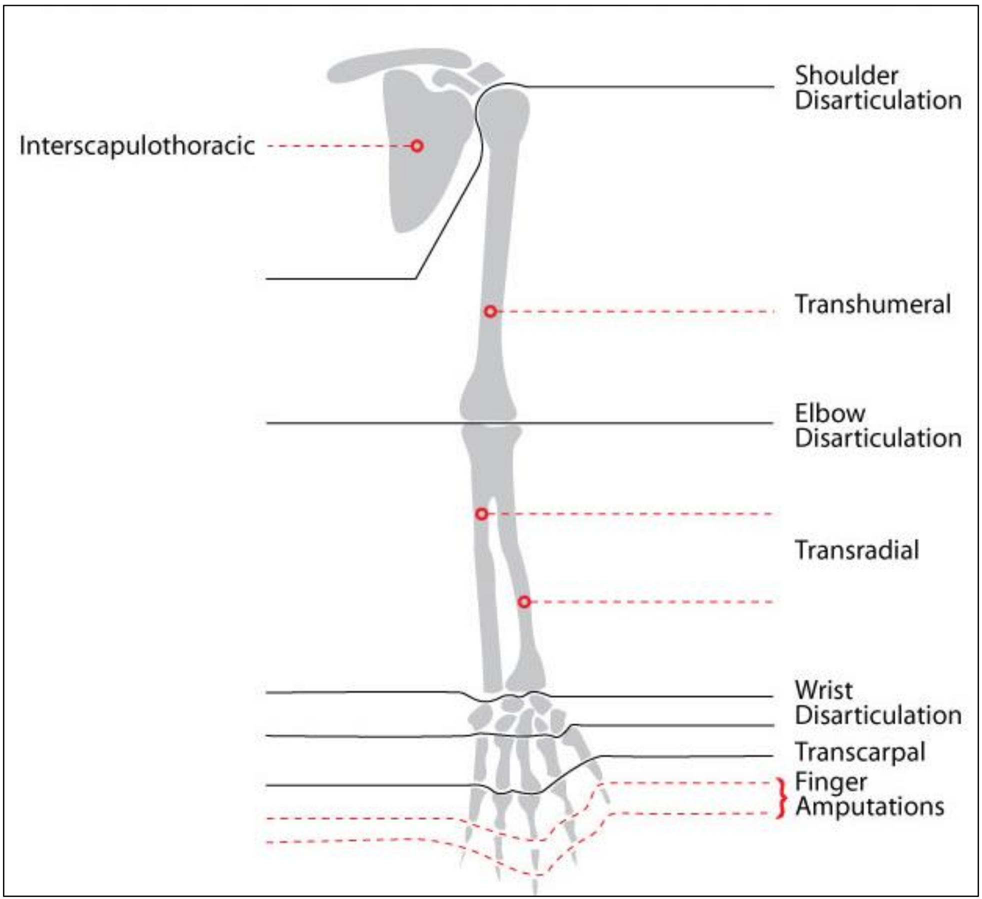

Figure 1: Different upper limb amputation levels [8]

1.4. Prostheses

A prosthesis can be defined as "*an artificial device to replace or augment a missing or impaired part of the body*". [9] Prostheses of the upper extremities can be attached to different places and replace different parts of the limb. This ranges from fingers to the hand, wrist, forearm, elbow, upper arm and shoulder. [10]

Nowadays there are many different types of prostheses. These range from cosmetic embellishments over simple passive mechanical aids, such as hooks or the like, to actively driven prostheses, which can at least partially restore the function of the missing limb. The development and different kinds of such upper limb prostheses are explained in more detail below.

1.4.1. History of prostheses

The idea of artificially replacing lost limbs has existed for thousands of years. There are prostheses that are over 3000 years old, such as the so-called "Cairo Toe" which was found at an Egyptian mummy and was supposed to replace a lost right big toe. Prostheses like this one were made from natural raw materials such as leather, wood and flax. In Figure 2, the Cairo Toe can be seen. [4]

Figure 2: The Cairo Toe from around 950 B.C. [11]

In 300 B.C., the first known prosthetic leg, the so called “Capua leg” was crafted by the Romans. It was made out of iron and bronze and had a wooden core.

During the dark ages, prostheses such as hand hooks and peg legs appeared which allow for walking or holding shields. Those were mainly built from iron and steel. [12]

During the Renaissance, anesthesia and wound management made great progress, making amputations safer compared to before. With new amputation options, the proliferation of prostheses increased and inventions like the tourniquet helped to stop heavy bleedings during the amputation process. There were prostheses like the “Knight Götz von Berlichingen iron hands” (1504), which could be moved and manipulated due to spring loaded mechanisms inside the hand. During this time, mainly iron, steel, copper and wood were used for prostheses. [13]

At the time of the American Civil War as well as during the two world wars, prosthetics experienced big boosts from the multitude of wounded soldiers. In addition, new materials such as cosmetic rubber were invented, which supplement the former prostheses made of wood and leather. This resulted in attachments like brushes and hooks.

In the years after the Second World War, many new materials were developed that made wood and leather unnecessary. These included, among others, resins, polycarbonates, plastics, carbon fiber and laminates. Their use made prostheses lighter and more durable.

Since then, the used material compositions have been further refined in recent years and now allow high-performance prostheses that have higher stabilities and comfort despite lower weights. In addition, sensors and actuators can be partially embedded in prostheses, which enable active control by means of microprocessors. These prostheses are complemented by new, generative manufacturing processes such as 3D printing, with which simple prostheses can be produced very cheap and uncomplicated. [14]

1.4.2. Classification of prostheses

Upper limb prostheses can be divided into two main parts. The first one is the socket which is the interface between the actual prosthesis and the residual limb. Connected to this is the second part, the actual prosthesis, which replaces the missing limb. At the distal end of the prosthesis is the terminal device which can be for example a mechanical hand or a hook. In addition, prostheses can be subdivided into active and passive ones.

Passive prostheses are prostheses that have no moving parts. These are mostly used for aesthetic purposes. There are also terminal devices designed for special tasks, such as gardening or sports.

Active prostheses, in contrast, are intended to support more productivity and functionality. These have moving parts that are powered either by the body itself or by external energy. In addition, there are hybrid combinations of these two active types, which are driven partly by the body, partly by actuators. [6] This subdivision is shown in Figure 3. [7]

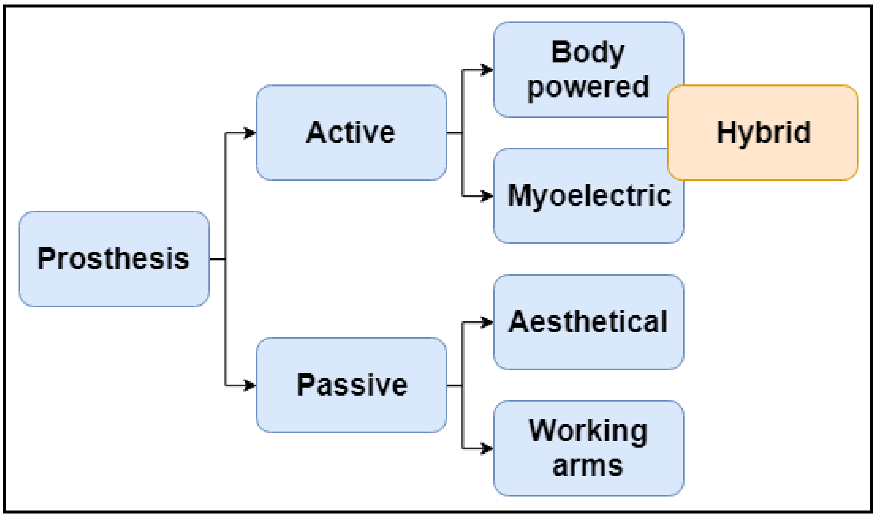

Figure 3: Subdivision of prostheses

Prostheses can fulfill two different tasks, which may be fundamentally different. On the one hand, prostheses are supposed to restore functions that have been lost due to the loss of the limb. On the other hand, they are used to optically restore the "normal state" of the body. However, this often results in function and appearance competing with each other. Prostheses that visually resemble a natural limb are often limited in functionality, whereas functional prostheses are often not visually pleasing.

Therefore, there are many patients who have several different prostheses, for example one that visually looks similar compared to a natural limb and one that is as functional as possible. [10]

In general prostheses intended for below-the-elbow amputations are much easier to construct and to control compared to those where the amputation was above the elbow. If the shoulder is also affected by the amputation, the complexity of prostheses needed increases again. [7]

1.4.3. Passive prosthesis

The cosmetic use of prostheses is quite important because especially the upper extremities are frequently used in social interactions, such as gestures or during communication. Visually inconspicuous appearance can thereby help to avoid psychological stress due to being “different”. This is especially the case if not only the forearm is affected by the amputation, but also the upper arm. [15]

The best representation of a natural hand is provided by passive cosmetic prostheses. These have no noticeable harness and can be adapted to the patient by means of shape and color. Thus, for example, skin color and anatomical features such as moles or even arm hair can be imitated. Aesthetic prostheses can also be used for simple bimanual tasks like fixating paper when writing, to stabilize objects which are held in the intact hand, or to keep a door open. [16] [7]

Generally, these are very light and have a high wearing comfort. The low weight is because they have no motors and only a few mechanical components. [7] Such passive prostheses are shown in the following figure.

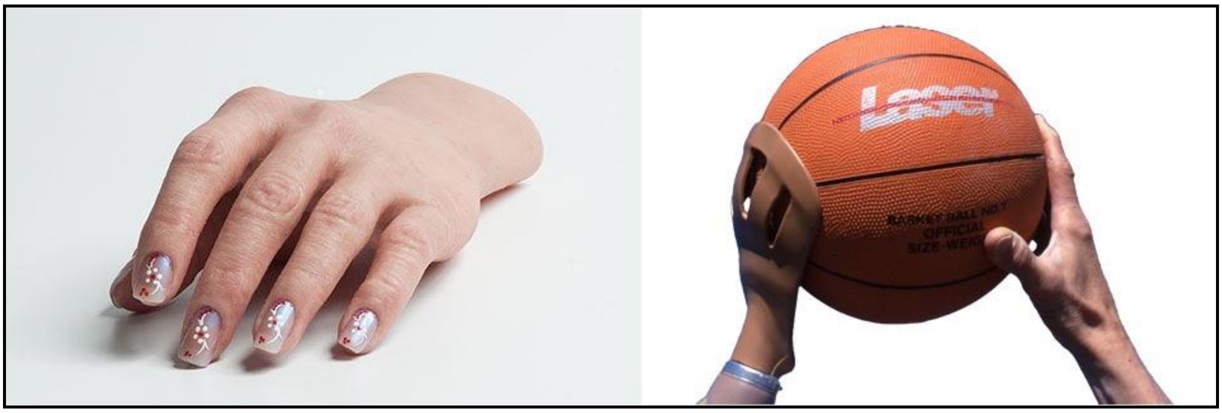

Figure 4: Examples of passive prostheses [17] [18]

1.4.4. Active prosthesis

The much more frequently occurring prostheses are the active ones. These have moving parts that are driven either by the body or by their own energy source. With body powered prostheses the control is usually done by the movement of a muscle near the amputated limb. The energy of this movement is transmitted via metal cables to the prostheses and there converted to perform, for example, an opening or closing of a gripper. With myoelectric prostheses, the action potentials of muscles are monitored and used to control the movement the prosthesis.

Active prostheses can take many forms such as hands, moving hooks or special shapes for specific tasks and activities. In the case of hooks, these usually have a movable and a stationary part which allows objects to be gripped. The moving part is usually adjusted by steel cables or electric actuators.

This kind of prostheses is usually heavier than passive prostheses because they are designed for higher loads. As a result, they are often made of heavier but more durable materials such as metal, hardened plastic or compounds.

The most important task of active prostheses is to restore limb functionality to those affected. This is because the upper extremities and especially the hand are of great importance for manipulating objects. With the help of such prostheses it is possible to grasp objects and to handle the activities of everyday life. This can be anything from simple activities like dressing or putting on robes, to holding cutlery. Most active prostheses allow one or two specific actions to be performed. However, there are also prostheses that are even more versatile and allow several different actions. [10]

Body powered prostheses:

Body-powered prostheses are often referred to as "cable controlled" because they require steel cables as well as harnesses during operation. Usually, these harnesses are constructed in such a manner that a strap passes over the scapula and attaches to a cable pull which in turn operates the prosthesis. Since body powered prostheses are directly linked to e.g. shoulder movement, such prostheses have a high level of feedback based on the control cable’s tension. [6] [18]

Other advantages of these body powered prostheses, compared to actively driven ones, are that they are in most cases lighter, quieter and more resistant. Since they have no electronical parts, they are in most cases waterproof and easy to clean. Their simple design allows affected persons to faster learn how to operate them and they also cost significantly less compared to actively driven ones.

A disadvantage of these body powered prostheses is that they need the harness to operate the terminal device. This meant that the affected persons must have a certain strength and freedom of movement in order to be able to utilize such devices. This can be very difficult, especially when working overhead. In addition, these prostheses are often optically less appealing compared to electrically driven ones due to the harness. [6] [7]

Prostheses with such a mechanical power transmission are more popular than the electric ones. 90% of people who use an active prosthesis use a body powered one. This is largely due to their lightweight, durable construction and the better haptic feedback generated by the cables. In less developed countries, durability, no need for regular services and their lower costs are a major selling factor. [6] [10]

Such body powered prostheses can be seen in Figure 5.

Figure 5: Body powered prostheses [19] [20]

Externally powered (myoelectric) prostheses:

The second large group of active prostheses is the group of the externally powered prostheses. These are mainly electrically powered and are often called myoelectric or switch-controlled prostheses. [6]

Such prostheses have external energy storage, which in turn supplies the built-in actuators. Generally, the energy is stored in form of accumulators. These devices can be controlled by multiple inputs such as electromyography (EMG) signals, the current measurements and feedback of the motors, as well as dedicated switches. Such physical switches are particularly useful when a high amputation has been performed. This is because in such cases usually many different motors for the different joints are needed and have to be controlled individually. However, myoelectric prostheses are the most widely used externally-powered prostheses, especially in cases of low amputation heights.

Myoelectric prostheses are based on measuring the electric excitation of muscles. Electrodes are attached to the muscles which measure the electrical signals from skeletal muscle contractions. The changes in electromagnetic fields, which arise when a muscle is flexed, is picked up by surface electrodes and forwarded to a microcontroller. In most cases, the electrodes are attached to two antagonistic muscles, such as the wrist extensor and the wrist flexor. In this case one muscle is used for one direction of movement of the prosthesis. For example, tensing one muscle opens a gripper and tensing the counterpart closes the gripper. This is also referred to as a simple two-site direct control system. In order to avoid involuntary movements, thresholds are set for the EMG signals.

Only when a certain threshold has been exceeded, the prosthesis begins to move. In many cases there is also a functionality which controls the speed of the movement. Slightly exceeding the threshold results in a slow movement and a greater divergence in a faster one. Thus, the speed of the prosthesis is proportional to how much the limit is exceeded. This allows the users to control the speed and gripping power. [7]

One of the advantages of such a myoelectric prosthesis is that it allows greater gripping forces to be achieved compared to body-powered devices. In some cases, this can be beneficial when holding objects for a longer time. In addition, no harnesses are needed for controlling purposes. This allows controlling multiple axes and joints simultaneously. The absence of a harness also allows the prostheses to look more like a real limb and therefore provides an aesthetic advantage over body powered devices.

However, there are also reasons why body-powered prostheses are 10 times more popular than myoelectric ones. This is mainly due to the higher purchase price of such devices. In addition, they are less robust, due to the built-in electronics only partially waterproof and they usually need to be recharged on a daily basis to be functional. Since there is no mechanical connection between the terminal device and the remaining limb, the haptic feedback is worse. It is sometimes harder for those affected to properly assess and apply the required gripping force. Therefore, a lot of training and education is necessary, especially when several actuators have to be controlled.

Furthermore, due to the complex design, these prostheses break more easily and must be serviced more often. Finally, the electrodes used are another disadvantage of these devices as it may happen that they move or lose contact. In these cases, prostheses cannot be operated properly. Constant contact with electrodes also may cause skin irritation or an unpleasant feeling if the prosthesis is not properly adjusted. Nevertheless, these prostheses are constantly evolving and could be more widespread in the future. [10] [7]

Figure 6: Active electrically controlled prostheses [21] [22]

Hybrid:

As mentioned above, there are also devices that consist of a combination of body-powered and myoelectric components. An example of such a hybrid prosthesis is a myoelectric terminal device with a body-powered elbow joint. This combination allows utilizing the benefits of both types. One can achieve high gripping forces whilst keeping the prosthesis lightweight. In addition, this approach can ease the control of the prosthesis, if the person concerned does not cope with the sole control by means of muscle signals. [10] [6]

1.5. Problem statement and approach

As briefly described in the beginning, a myoelectric prosthesis is to be developed, which is controlled by means of electric activity signals of muscles of the upper extremities. These signals are to be picked up by means of surface electrodes. All muscles of the forearm and the upper arm are available for this purpose.

A microcontroller, such as the Arduino Uno together with a MyoWare Muscle Sensor should be used for data collection. These are designed for biomedical applications as they have the required resolution and processor power to record muscle signals without significantly falsifying them. [23] [24] Such a commercial microcontroller, like the Arduino, should be used as they are readily available and sufficiently tests as many EMG sensing projects are based on such microcontrollers.

The recorded muscle signal should then be processed to access which muscles were moved. The classification should be done by means of a database which has been recorded in advance. For this analysis, the data can be forwarded to an external computer, which handles the processing of the signals.

Thresholds for the individual muscles and gestures are to be determined as well as where the best position for attaching the electrodes is. By determining the correct electrode positions, the quality of the signals shell be improved and thus the reliability of the classification in total. Additionally, it should be tested how many electrodes are needed to distinguish between individual gestures reliably.

The classification process itself shell be performed by an algorithm, for example a neural network. Such classification algorithms are further described in Chapter 3 and shell be carried out in a Matlab or LabView. To implement algorithms like neural networks, readily available libraries shell be used like the Deep Learning Toolbox from MathWorks. [25]

Since the control is supposed to be a quasi-real-time application, a certain delay should not be exceeded, so that a tensioning of a muscle is followed by a reaction of the prosthesis in a timely manner. This time dependency shell be analyzed and broken down.

The presentation of the recognized gestures may initially be done in a Matlab script. In this script, the model of a human hand shell be shown which replicates the gestures of the actual limb. Toolboxes such as the Robot Toolbox from MathWorks or other modelling programs can be used for this purpose.

Based on the graphical output of the script, it can then be recognized whether a movement has been identified correctly. [26] [27]

Right now, systems like the one presented in [26], allow differentiating between few gestures by means of support vector machines. [26] used EMG signal recognition based on 3 channels to distinguish 5 different gestures. This should serve as a starting point for this work. Similar results shell be recreated and serve as a reverence value. With this kind of setup around, 85% classification accuracy could be reached.

Other projects like the one presented in [28] could classify 15 different gestures with a reliability of around 95%. Similar results should also be achieved with the algorithms developed for this work.

By improving the position of the electrodes and implementing alternative classification algorithms, like fuzzy logics or neural networks, the reliability should be improved. The recognition rate shell then be compared to the one of former solutions.

Compared to other solutions, the number of EMG Sensors may be adapted for better gesture recognition. The achieved classification algorithms are to be designed in such a way that they deliver consistent results as far as possible, even if they are applied to different test persons as it turned out that this was a big problem for similar systems.

Once this works reliably for several different gestures, a real mechanical prosthesis is to be created, which is powered by several motors. To produce this prosthesis, generative manufacturing processes such as 3D printing may be considered. The reliability of the whole system has then to be tested. For this, the recognized gestures shell be compared to the real ones and it shell be calculated how many were correctly classified by the algorithm.

2. EMG control acquisition

The following chapter explains what muscle signals are and how they can be recorded. In the beginning, the anatomy of the human arm will be examined, in particular the anatomy of muscles. Afterwards it is examined what myoelectric signals are, as well as how an electric circuit looks like to record and process those.

2.1. Muscles of the upper limb

Skeletal muscles are part of the musculature responsible for active voluntary body movements and thus part of one of the three main muscle types. Just like the heart muscles, skeletal muscles belong to the group of striated muscles and are also referred to as voluntary muscles. Apart some exceptions, those muscles are connected to bony structures by tendons. They often exist in pairs, whereby the first muscle is the primary mover and the second one is its antagonist. For example, the biceps and triceps are such a pair of antagonists. When one of them contracts, the other one relaxes to allow the movement and vice versa.

Skeletal muscles have a complex structure. They are composed of fascicles which are bundles of elongated muscle fibers. The muscle fibers themselves are consisting of bundles of myofibrils. Myofibrils themselves are composed of myosin and actin filaments. These two filaments are stacked in regularly repeating arrays and are responsible for the muscle contraction itself by sliding against each other. Those myosin and action arrays are called sarcomeres. Through this sliding action, the muscles can be shortened and thus contracted. Motor neurons which control the contraction are connected to bundles of muscle fibers and are together called a motor unit. In places where finer movements have to be achieved, only few muscle fibers are connected to one neuron. In places where a lot of strength is required, one motor neuron is in control of lots of muscle fibers. [29]

The most important muscles of the arm are listed below. The numbering scheme follows Figure 7.

- 1: Musculus pectoralis major
- 2: Musculus deltoideus
- 3: Musculus bizeps brachii
- 4: Musculus trizeps brachii
- 5: Musculus brachioradialis
- 6: Musculus flexores digiti

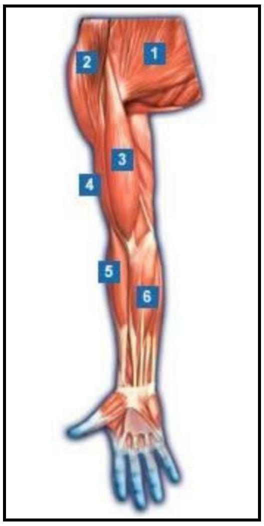

Figure 7: Most important muscles of the upper limbs [30]

Motor neurons are located inside the brainstem and the spinal cord and are connected to the muscle via axons which can transfer excitation signal over long distances. The activation of the muscle fibers is done by electrical potentials of cell membranes. This means that the voltage inside of a cell is usually 60 to 90mV lower compared to its surrounding. By opening and closing of ion channels membranes can allow movement of ions and thus create an electromagnetic field signal. This signal will travel along the axons as a wave to the end of the motor neuron.

The place where the motor neurons connect to the muscle fibers is called neuromuscular junction. This is the place, where the fibers start to respond to the signal of the motor neuron and thus start to contract. The neurons release acetylcholine at the junction which itself creates an excitation in the muscle fibers. Ca^{2+} is freed and allows the sarcomeres to be shortened. With help from released ATP, the sarcomeres can return to their normal position to allow the contraction to end. The structure of skeletal muscles is shown in Figure 8. [31]

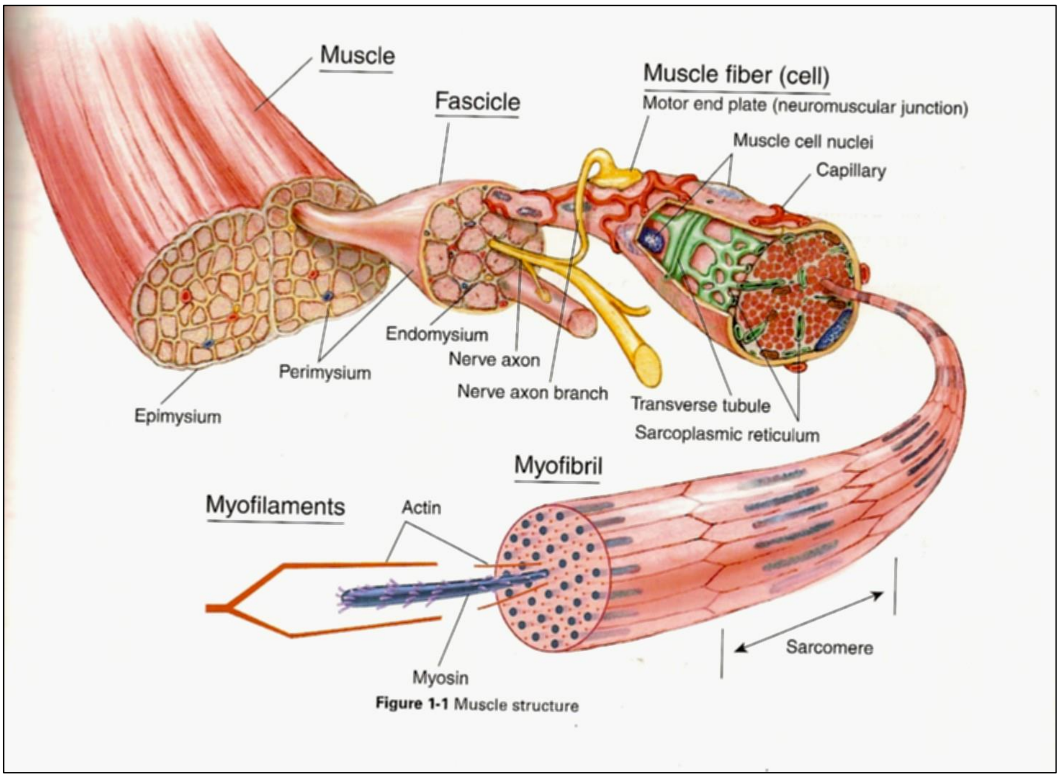

Figure 8: Structure of a skeletal muscle [32]

2.2. EMG signals

As described in the section above, action potentials are created during the contraction of skeletal muscles. Those action potentials can be measured and are the basis of EMG signals. EMG signals are used for analysis and clinical diagnosis in biomedical applications such as management and rehabilitation of motor disabilities.

The electrical currents generated during the flexion process can be measured using electrodes on top or inside the muscle. EMG signals are quite complicated as they are dependent on the anatomy and the physiological properties of the muscle. Impurities of these signals are quite common and accumulate whilst traveling through the body. Also, an EMG signal is the sum of multiple motor units firing at the same time and thus there can be interactions between these different signals. As the intervals at which the action potentials of specific motor unit occur are random, the EMG signal may be either positive or negative at a given time.

The motor unit action potential itself is the combination of the muscle fibers action potentials belonging to a single motor unit. It can be described with the formula below.

$$x(n) = \sum_{r=0}^{N-1} h(r)e(n-r) + w(n)$$

In this discrete formula, *x(n)* is the resulting EMG signal, *e(n)* the firing impulse of sample *n, h(r)* the motor unit action potential, *w(n)* the additive white Gaussian noise and *N* the number of motor unit firings. [33] The composition of an EMG signal can be seen in Figure 9.

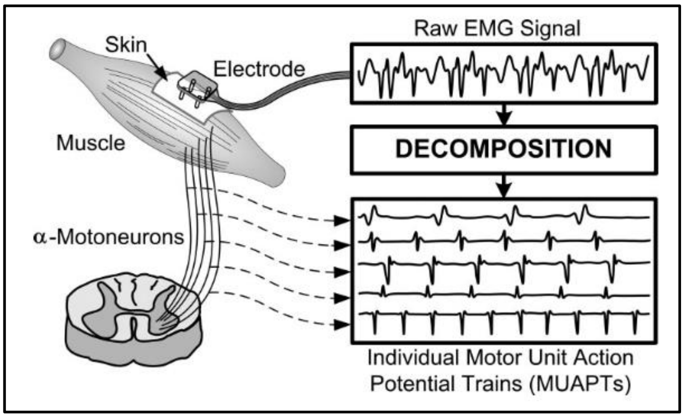

Figure 9: Composition of an EMG signal [34]

EMG signals have distinct properties which differentiate them from other body signals. The most important properties are its frequency range and its amplitude. Motor units have a high dynamic range of amplitudes which results in combined amplitudes of 0 to 10mV (peak-to-peak) or 0 to 1.5 mV (root mean square). The frequencies are between 20 and 500Hz whereby the dominant frequencies are in the range of 50-200 Hz as shown in Figure 10. [33] [35]

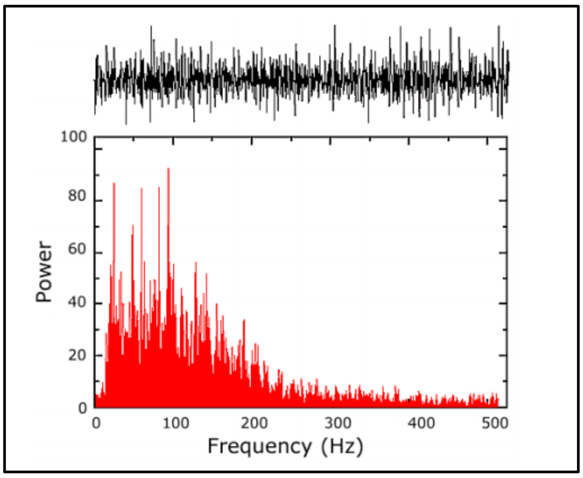

Figure 10: Power of individual frequencies, measured at the Tibialis Anterior muscle during isometric contraction [35]

2.2.1. Recording of EMG signals

EMG signals are picked up by electrodes which are either placed on the skin above the muscle or inside the muscle itself. Both variants have their pros and cons. When using intramuscular sensors, the environment and the sensors as well have to be sterilized. As it is an invasive procedure it carries the risk of transmitting a disease or triggering an infection. On the other hand, once the electrode is places it does not cause discomfort and the signals are not distorted by the tissues between the muscle and the skin surface. This leads to a higher signal to noise ratio. Also, high spatial resolution is possible. On the other hand surface electrodes can be repositioned if the position is not suitable and do not need invasive procedures. Thus they are also suitable for patients with a needle aversion. For short measurements those surface electrodes are also more versatile. [34]

It is very important to place electrodes correctly and to use electrodes suitable a specific task, as their selection will influence the obtained signals. To get the highest signal strength, the electrode has to be placed on the muscle belly in the direction of the muscle fibers.

The most commonly used electrodes are Ag/AgCl as they are not polarizable and allow immediate current flow. In most cases they are attached using a conductive gel to reduce impedance of the skin. The placement of the electrodes relative to each other and the size of the electrodes are also important. The further away electrodes are compared to each other, the higher the measuring depth is. The bigger the electrodes are, the lower the spatial resolution as increased size leads to an averaging effect. On the other hand, the skin impedance is reduced which leads to less noise and better frequency response. [36] [37]

Usually $2n + 1$ electrodes are used; two for each channel n and one reference electrode which is located on electrically unrelated tissue.

After the signal is picked up, it is usually amplified as its amplitudes are quite small. For the first stage of amplification, a differential amplifier is commonly used. Additional stages of amplification may follow afterwards. [33] [35]

A differential amplifier is used to eliminate the common mode currents of the signal. To do this, the signal is picked up at 3 locations, two detection electrodes and one reference electrode. The reference electrode defines the neutral ground that the other two electrodes share. Any signal that is common to these electrodes will be removed. The signals they don't share will then be amplified. It is essential to have high accuracy electronics as this step strongly influences the shape of the resulting signal. Common Mode Rejection Ratios of 90dB and more are considered as sufficient. The differential amplifiers impedance shell be as large as possible to prevent attenuation and distortion of the signal.

Afterwards an amplifier is used to further increase the system's signal amplitudes. Typical values for the total amplification are 1000 up to 20000. A low pass filter shell be applied to eliminate high noise frequencies. Cut off frequencies of around 1000Hz are appropriate as it is two times the highest expected EMG frequency according to the Nyquist theorem.

The 50Hz frequency interferences of the mains power line can be eliminated with a band stop filter. Furthermore, a rectification of the signal can be applied to flip the negative signal parts and makes them positive. This eases the application of an integrator low pass filter to get the envelope of the signal if needed. Finally, an analog to digital converter is applied to transform the continuous signal to a discrete one, so that a computer or microcontroller can work with the EMG signal. A resolution of 10 bits is a typical value for such applications.

An important part of every electric circuitry in medical applications is the galvanic isolation of the patient from mains power. This is needed to eliminate the risk of electrocution due to malfunction of the system. Another way of handling this problem is to only use low voltage power sources and to abstain from mains power. [38] [35]

This filtering and amplification process is shown in Figure 11**.**

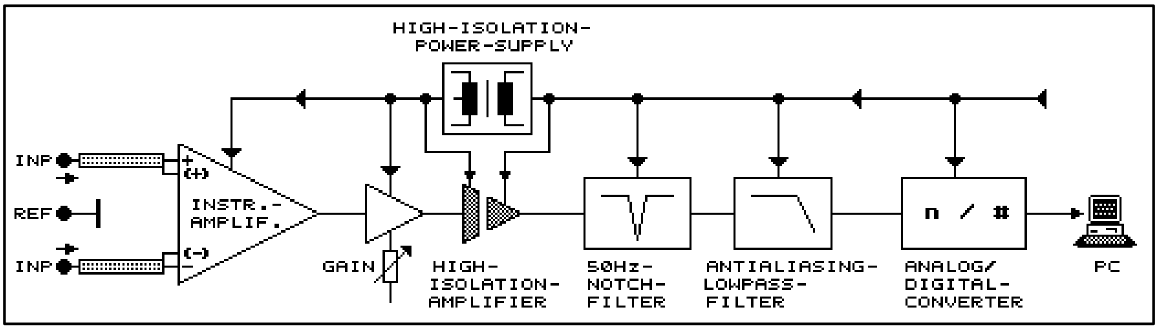

Figure 11: Electronics for recording EMG signals [38]

2.2.2. Interferences

When measuring such signals, there are also many factors that can decrease the quality of the signal. These interferences are also called artifacts or noise. Some of them can be avoided or reduced. For example, this can be done by applying the electrodes at the correct position and using a reference electrode as briefly discussed before. The most important sources of noise are:

- **Inherent noise in the equipment:** This kind of noise can't really be avoided as it is inherent to the acquisition system itself. It's the noise the acquisition system itself produces during capturing and processing. It has a frequency range of 0 Hz to several thousand Hz. It cannot be completely removed, but it can be reduced by high quality equipment and intelligent circuit design.
- **Movement artifact:** This noise is due to movement of the electrode when the muscles contract as well as from the movement of the cables connecting the electrode to the amplifier. This kind of artifact is usually in the range of 1 to 20Hz. Unfortunately their amplitudes are in the range of the EMG signals' amplitudes so it can highly distort the signal. Recessed electrodes can minimize movement artifacts significantly by reducing the skin impedance as well as proper design of the electronic circuit.
- **Electromagnetic ambient noise:** Electromagnetic noise can appear in EMG signals due to the fact that every electromagnetic device generates noise. The human body is at all times inundated by such electromagnetic radiation which is then picked up by the electrodes. Such noise can be up to 3 times higher than the EMG signal itself. The most common ambient noise is the one from the mains power supply with 50Hz. If the frequency of the ambient noise is known, the noise and its harmonics can be filtered by means of band stop filters.
- **Cross talk:** Crosstalk describes unwanted signals from muscle groups near the muscle which is actually under investigation. It can be reduced by placing the electrodes in such a way that the signals of other muscles are attenuated as much as possible before reaching the electrode.
- **Inherent instability of the signal:** EMG signals are affected by the rate at which the motor units fire. These fire randomly with a frequency of 0 to 20Hz and thus create quasi random amplitudes in the EMG signal.
- **Electrocardiographic (ECG) Artifacts:** As the heart is also a muscle, it produces artifacts which highly influence EMG measurements. This is especially the case when measuring with surface electrodes near the shoulder and trunk region. This noise can be removed by either applying a high pass filter which lets frequencies of 100Hz and higher pass, or by applying an electrode along the heart's axis and using common-mode rejection. [37] [33] [35]

The aim is to have the highest signal to noise ratio possible to ease further processing.

3. Classification of the movement

Picking up the signals from the muscles is only the first step of many to control a myoelectric prosthesis. The picked-up signals are still raw and unprocessed and need further processing to be able to be used for controlling of such devices. Furthermore, the individual processed signals have to be classified to detect which gestures were performed and thus how the prosthesis itself should move.

This chapter is concerned with the different techniques how recorded signals can be enhanced and classified so that the prosthesis moves as it is supposed to. The following steps are suited for filtered and mostly noise free signals. As described above, low pass filters, band stop filters as well as rectifiers may be used to achieve such noise free signals. These can be implemented either in hardware, software or a combination of both. A benefit of an implementation in hardware is that it causes nearly no delay to filter the signal. On the other hand, the implementation is in most cases easier in software, as frequencies can be selected more specific, and filters can be adjusted if the results are not appealing.

The time dependencies of the feature extraction and classification algorithms will also be concerned, as the whole system should be able to act with as little as possible delay.

As the following signal processing steps can be quite intensive in computational resources, these might be performed on a more capable device like a computer or laptop.

3.1. Signal analysis and feature extraction

The first step after the signal is pre-filtered will be to determine whether or not muscle activity is present at all. This is because most of the time there will be no muscle activity and the prosthesis will be in an idle state. In this case, there will be only minor signals, such as random noise or artefacts from heart activity which couldn't be filtered. Thus, a threshold can be applied to distinguish if there is muscle activity from one of the muscles under investigation. While the signal is beneath a given threshold, it does not have to be analyzed and the processing device can save resources.

In case that there is activity after a period of non-activity, an interrupt signal can be used to start with the feature extraction. [39]

Furthermore, a discrete time window can be assigned in which the signal is then analyzed. Typically, the longer the window in which the signal is analyzed, the better the result. The downside is that this reduces the real time capability of the system and thus it should be tried to reduce the delay and window length to a minimum. In similar projects a window length of around 250ms was found to be sufficient. [26] [40] [41] Those sample frames can then be analyzed in the frequency and time domain.

There are also other important parameters which can be used to distinguish between noise and muscle signals if normal thresholding cannot be applied. Such parameters are for example the root mean square or the mean absolute value. These can also be used during the classification process to tell different muscle signals apart due to their specific properties. [42]

3.1.1. Typical parameters used to describe signals

The following list is a short summary of commonly used parameters to distinguish or detect muscle activity in an EMG signal.

- **Root Mean Square (RMS):** The root mean square is defined as the arithmetic mean of a set of squared values. In discrete signal processing it can be described with the formula below, whereby N is the length of the signal frame and x_n are the individual signal values inside the sample. In electronics it can link the power of alternating current to the one of direct current and in prosthesis control it can be interpreted as amount of muscle activity. [42] [43]

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2}$$

- **Modified Mean Absolute Value (MMAV):** The mean absolute value can be calculated similar to the RMS. One advantage is that it can be adapted to get the modified mean absolute value whereby each individual signal value x_n can be weighted to smooth the results. The formula is given below whereby ω_n.is the weight of the individual values x_n. [42]

$$MMAV = \frac{1}{N}\sum_{n=1}^{N} \omega_n |x_n|$$

- **Mean and Median Frequency (MNF and MDF):** Mean and median frequency are often used to describe the process of muscle fatigue as they can be used as indicator for it. As MNF and MDF are in the frequency domain, they can show better performance compared to other characteristic parameters. The Fourier Transform is used to obtain the power spectrum of the signal and transform it from the time domain into the frequency domain. MNF is the average frequency of the signal and MDF is the frequency which divides the power spectrum into two regions with the same amplitude. The formulas for MNF and MDF can be seen below whereby f_n is the frequency of the power spectrum and P_n the power spectrum itself. [42] [44]

$$MNF = \frac{\sum_{n=1}^{N} f_n P_n}{\sum_{n=1}^{N} P_n}$$

$$MDF = \frac{1}{2}\sum_{n=1}^{N} P_n$$

- **Variance (VAR) and Standard Deviation (SD):** The variance is the squared deviation of a variable from its mean value. It can give information about how far a signal is spread around its mean value. With the formula below the variance can be calculated. The standard deviation is the square root of the variance. [42] [28] [4]

$$VAR = \frac{1}{N-1}\sum_{n=1}^{N} x_n^2$$

- **Peak amplitude:** The peak amplitude is an indicator for the maximum value of a signal and can be used to distinguish between signals which have the same RMS but a different shape**.** [28]

3.1.2. Fourier Transformation

When measuring a signal, the values gathered are mostly in the so-called time domain. This means that each sample is interconnected to one specific moment in time. Such a signal in the time domain can be seen for example on the screen of an oscilloscope which displays the measured signal in real time. On the other hand, a signal can also be displayed in its spectral domain. The spectral domain shows how a signal is composed of individual oscillations that add up and together compose the complete signal. This is due to the fact that every waveform can be generated by adding up sine waves. This time and spectral domain relationship can be seen in Figure 12. At a) the two individual sinusoids are displayed which together compose the signal. At b) the overlay of the two signals can be seen in the time domain and at c) how those two signals are represented in the frequency domain.

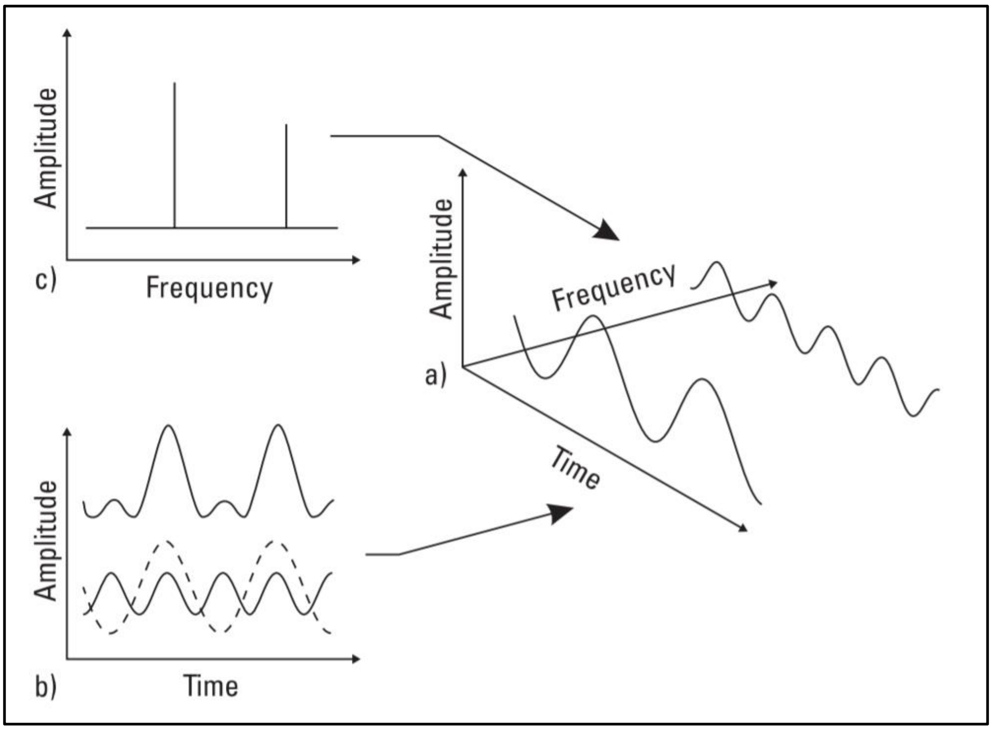

Figure 12: Relationship between b) time domain and c) frequency domain [45]

In case that the signal is not continuous but discrete, one has to work with the formula for the Discrete Fourier Transformation to get the information about which frequencies compose the signal. As the signal of the sensor will be polled and processed in timely discrete periods (e.g. 1000HZ), the Discrete Fourier Transformation has to be applied. Its formula can be seen below.

$$F(j\omega) = \sum_{k=0}^{N-1} \mathrm{f}|k| e^{-j\omega kT}$$

In this equation, N is the number of samples, $f|k|$ the individual samples, T the sample time and $j\omega$ is the frequency response.

During a Fourier Transformation a signal is compared with sinusoids of various frequencies to get the corresponding magnitude and phase shift for each frequency. The magnitude spectrum is an indicator for how present a specific frequency is in a signal. If the magnitude spectrum is high, that means that the signal under investigation composes a high share of this signal. The phase shift on the other hand adds the information on whether or not there is an offset of the individual frequency and how big the offset is compared to the origin. [46]

This information can help to either filter for specific frequencies or to distinguish between actual muscle signals and noise. As some noise includes all frequencies, those can be easily filtered because the frequency range of muscle signals is known.

When time samples are investigated, there are some flaws when applying Fourier Transformations as some information describing the original signal may be lost in the process. For example, a longer time window may improve the resolution of frequencies, but information about the exact time when events happened during the time window is lost. When using a short time window, the time resolutions stays quite high, but the frequency resolution is compromised. Wavelet analysis can help to solve this problem.

3.1.3. Wavelet Transformation

The Wavelet transformations works in a similar way like the Fourier Transformation. The difference is that the Wavelet Transformation compares the signal to so-called “wavelets” to gain coefficients showing the similarity between those and the signal. These wavelets are finite in length and can have different shapes; they can be symmetric or asymmetric, regular or irregular. Such wavelets can be seen in Figure 13.

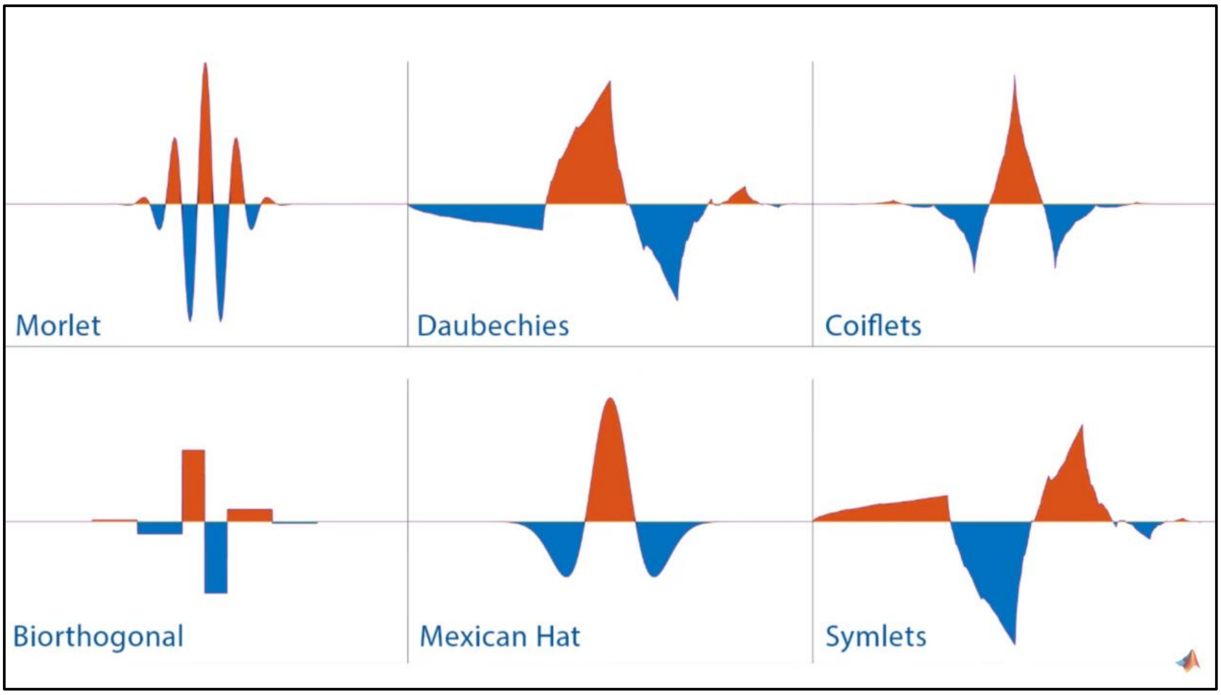

Figure 13: Different shapes of frequently used wavelets [47]

What differentiates the Fourier Transformation from the Wavelet Transformation is that the wavelets are localized in the time domain as well as the frequency domain. This is because the wavelets have a limited time duration and frequency spectrum. Thus, Wavelet Transformation is very well suited for processing non-stationary signals whose spectrum changes with time. Also, the Fourier Transform may not present abrupt changes sufficiently. [48]

By changing the size and the position of the so called mother wavelet, a wavelet family containing the dilated and translated sub-wavelets can be created. This process is called scaling and shifting.

Scaling means the compression or stretching of a wavelet in time. A scale factor larger than 1 means that the wavelet is stretched so that it correlates with lower frequencies. A scale factor between 0 and 1 means that the wavelet is shrunk so it correlates with high frequency components of the signal.

Shifting means the change of the onset of the wavelet in comparison to the signal. When a wavelet center is shifted over a signal artefact and the artefacts frequency correlates to the length of the wavelet, the correlation between those two is large. The Wavelet Transformation itself then computes the inner product of a signal with a wavelet family.

In case of a Continuous Wavelet Transform (CWT), the number of coefficients can be much higher compared to the original length of the signal. For example, if the signal has a length of 1000 samples and the wavelet family would consist of 20 different wavelets, there would be 20000 coefficients. This would allow a deep level of analysis but would also need high computational power at the same time. The formula for the CWT of a function $x(t)$ can be seen below. The variable a represents the scale factor, b is the translation and ψ is the used mother wavelet. [40] [47] [48] [49]

$$y(a,b) = \frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right)dt$$

Because of the high number of coefficients, CWT is rarely used in real time applications and the Discrete Wavelet Transformation (DWT) is used instead as it uses fewer coefficients. When using dyadic scaling and shifting, it eliminates redundant coefficients. In this case, the number of output coefficients is the same as the number of input samples.

The dyadic WT is performed by passing a signal through a series of high- and low-pass filters which then give the coefficients of the transformation. The filters used have to be quadrature mirror filters which means that their magnitude response is mirrored around $\pi/2$ respectively to each other. After passing through one level of the filter, half of the signal samples are removed. This is because in the resulting signals half the frequencies have also been removed and thus, according to the Nyquist theorem, only have the samples are needed to faithfully represent the signal.

The signal from the low-pass filter is then further processed by passing it through a new high-pass and low-pass filter combination. This is done for each level of the filter bank and can be seen in Figure 14. After each filter the signal is down sampled by a factor of 2.

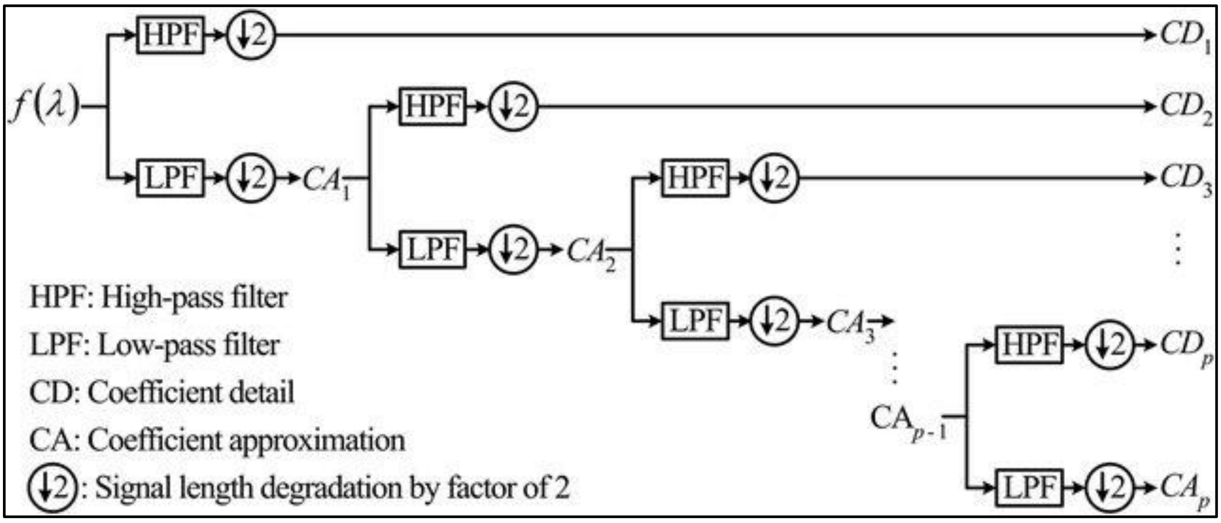

Figure 14: Composition of a dyadic filter bank for DWT [50]

These individual signal bands can then be processed to gain information or to reduce noise. This can be done by e.g. removing all signals of the highest frequency band which are below a given threshold. By inverting the decomposition procedure the modified input signal can be reconstructed. [33] [47]

In most cases, the Discrete Wavelet Transformation is used for signal compression, to decrease noise and for peak detection. The low number of coefficients results in high computational performance.

3.2. Movement pattern classification

When only one joint of a myoelectric prosthesis has to be controlled, the easiest way is to use the EMG signal from two antagonist muscles. In this two-site direct control scheme, the EMG signal from one muscle indicates that the prosthesis should move in one direction. When the antagonist muscle is contracted, the prosthesis does the opposite. These muscles could be for example the wrist flexor and wrist extensor.

With such a simple control scheme it is easy to distinguish between a few possible movements. But the control gets much more complex when more joints and terminal device movements have to be controlled. In this case new strategies have to be implemented to allow the control of multiple axis and joints. These strategies can be multiple quick contraction of one muscle, the combination of multiple muscles at the same time or in a sequential manner. For example a contraction of the wrist extensor and flexor in a short time window could cycle through the different joints of a prosthesis which can be controlled. [7]

This combination of multiple movements in a sequence allows control over multiple axis but can be quite hard to remember if it exceeds a certain limit. When combining more than only the EMG signals from two antagonist muscles the functionality of the prosthesis can be widely improved. Such multi-channel EMG signals can then be analyzed to recognize individual movements and thereby control a prosthesis accordingly. This classification can be carried out in multiple ways.

3.2.1. Thresholding

The simplest way to classify the combination of multiple EMG channels is to threshold the individual channels. In this case the rectified signal is compared to a reference amplitude threshold. By doing this, one could achieve a control as stated in Table 2.

Table 2: Possible movements due to exceeding thresholds in individual channels

Action	Activity channel 1	Activity channel 2	Activity channel 3
Idle	0	0	0
Close terminal device	0	0	1
Open terminal device	0	1	0
Wrist pronation	0	1	1
Wrist supination	1	0	0
Wrist extension	1	0	1
Wrist flexion	1	1	0
Go into defined position	1	1	1

By simply combining the individual channels, a high number of different movements can be achieved.

The threshold to distinguish whether or not a channel is active can be either static or dynamic and the thresholding applied can be hard or soft. A possibility for dynamic thresholding could be to monitor the average amplitude in the samples before and to use this value as a reference. This would decrease the problem of electrode movement which leads to a shift of idle potential. Furthermore, a combination of static and dynamic thresholding can be applied so that a dynamic threshold is calculated but with a second hardcoded limit as backup to decide if a muscle is active or not.

There are also other possibilities how the thresholding techniques for classification of movements can be adapted. For example it would also be possible to set a threshold for the time the signal has to be high before it is considered an active signal. This can minimize the amount of false detections. [7] [33]

However since such a strict distinction between active and not active cannot be achieved in every situation and for every movement, there are other techniques how EMG signals can be classified.

3.2.2. Fuzzy logic

One real time classification technique which doesn’t rely on the strict distinction of the muscles into active and non-active is fuzzy logics. Fuzzy logic is a part of artificial intelligence and is a method of clustering whereby data can belong to one or more clusters. Instead of calculating definitive outputs, the system returns a probability of a state. By doing this, computers are able to calculate with uncertainties, as uncertain values don’t have to be 1 or 0 but can be somewhere in between. [51]

Fuzzy logic systems are based on reasoning and can be fed with knowledge to help it build up a rule base for decision making. The working principle of a fuzzy system can be seen below.

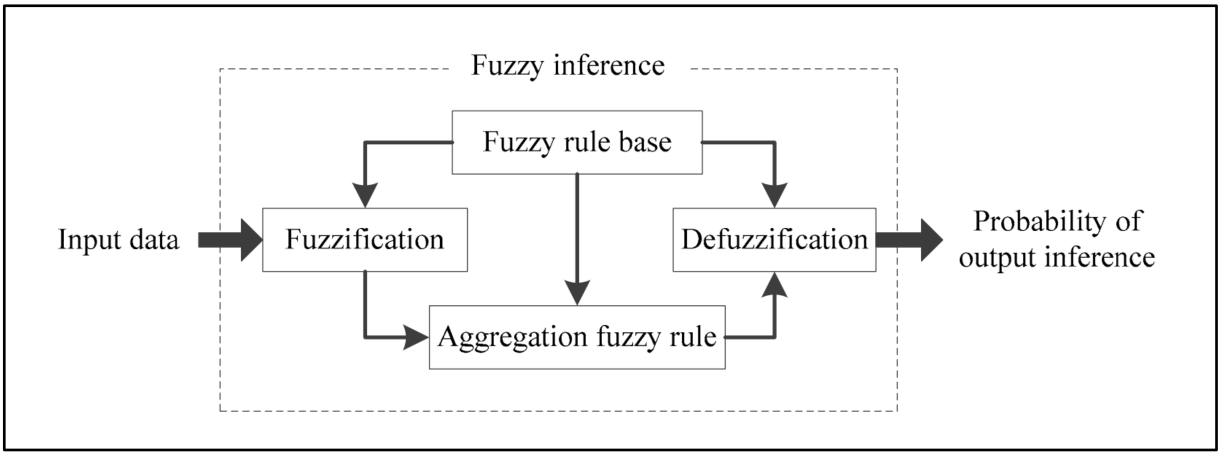

Figure 15: Working principle of a fuzzy system [52]

In this diagram, crisp values get injected into the system as input. This data is then fuzzified which means that it is translated into probabilities of belonging into a specific class. The fuzzified values are translated into fuzzy sets during this step. These sets are then injected into the inference block, where logical rules are applied to it to get a fuzzy output set. The fuzzy output is then defuzzified to get a crisp system output value which can afterwards be applied to a device or prosthesis. The rules which are applied are based on a knowledge database which increases over time and in which knowledge can be injected. [51] In practice, the inference rules are on a "IF…AND…THEN" basis. When a certain combination of probability values are exceeded, the fuzzy system will inference a rule.

To give a frequently used example, one can look at a temperature controller of a shower. The crisp input of the system is the current temperature of the water. This exact data is then fuzzified into a fuzzy set "temperature". Inside this set are multiple members, for example "cold", "warm" and "hot" as the temperature could be described in such dimensions. Depending on the value of the input it belongs to some members more than to others. A temperature corresponding to the black line in Figure 16 would correlate to "cold" with 0.8, to "warm" with 0.1 and to "hot" with 0.0.

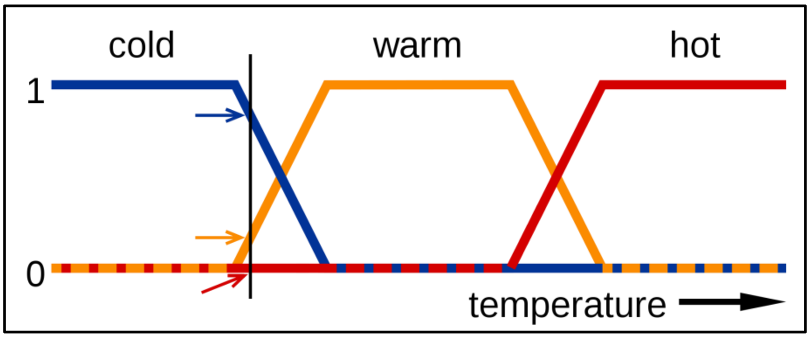

Figure 16: Fuzzy temperature set [53]

Depending on those values the inference rule block would then decide which actions to take. If the rules state that a temperature in the "warm" range would be better, it would take actions to achieve this goal. By following the "IF…AND…THEN" decision process it would reason that if the water is cold and not warm enough, then turn on hot water. There can be many such rules which together comprise the rule base.

These output actions are also in a fuzzy state which means that the output could look like "turn hot water on" with 0.7, "turn down cold water" with 0.6 and "turn on space heater" with 0.4.

This fuzzy output set would then be defuzzified which would lead to crisp values like turning down the cold water by half a turn and turning on the hot water by ¾ of a turn. [51]

Logics like this can also be applied to a prosthesis control where the input signals can be translated to fuzzy values like “little activity”, “medium activity” and “high activity”.

Fuzzy logic systems can be used very well in biomedical signal classification as it can handle slightly deviating signals very well. As muscle signals are never exactly the same and change over time due to muscle fatigue and other factors, these applications are very well suited for this kind of classification. Also, fuzzy systems can detect interconnections between input factors which may be very well hidden in the raw data. Thus it can also help with understanding the data. [33] The downside is that it can need plenty of data till it works in a meaningful manner. [51]

3.2.3. Support Vector Machines

Support vector machines (SVM) are another alternative to classify data based on machine learning. They are founded on supervised learning and can, after a learning phase, distinguish data into multidimensional pre-selected groups. The algorithm itself separates the data into groups and tries to fit a linear decision boundary, also called hyperplane between the extreme points of each dataset. Thereby it tries to find the border which best separates the individual groups apart.

To distinguish between the classes, properties of these are mapped against each other on a multidimensional scale. Between the different classes, a boarder is drawn using the data points which are nearest to the estimated boarder. Those data points are called support vectors and are used to define the specific position and slope of the hyperplane. This is done by choosing the hyperplane which leaves the maximum margin between two classes.

The margin is defined as the closest distance between the hyperplane and the closest members of the individual groups. Such segregation into two groups using a linear border can be seen in Figure 17. In this case the margin of z_2 is bigger compared to z_1 so it is a better solution for the problem and will be chosen to separate the groups. [54] [55]

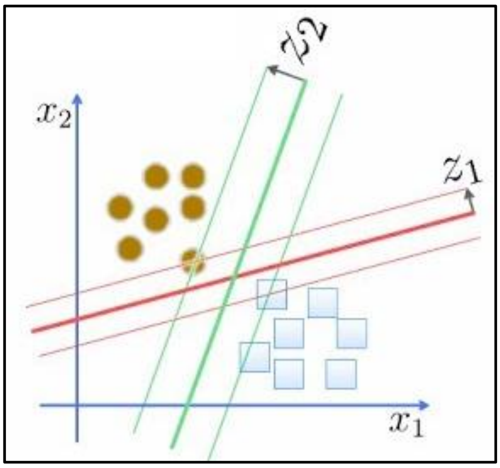

Figure 17: Two different possibilities to place a hyperplane [54]

Finding the maximum margin between the multidimensional data is a nonlinear constraint optimization task which is solved by using complex conditions and Lagrange multiplications.

As the algorithm has to be trained, it is necessary to carry out multiple runs where the gesture is known and the corresponding muscle signal is recorded. The muscle signals have to be normalized and turned into a one feature vector. This data then has to be processed to fit the model.

In some case a linear hyperplane cannot be inserted into the data sets because it wouldn't be able to separate two groups without intersecting one. In these cases the data has to be transformed into higher dimensional feature space using so called kernels. The most widely used kernels are the Polynomial Kernel, the Radial Basis Function Kernel and the Sigmoid Kernel. Unfortunately it is very complex to choose the right kernel for a specific task as each kernel has its own strengths and weaknesses. Such a higher dimensional kernel transformation can be seen in Figure 18. In this case, by going from two dimensional space to three dimensional, a hyperplane could be fitted.

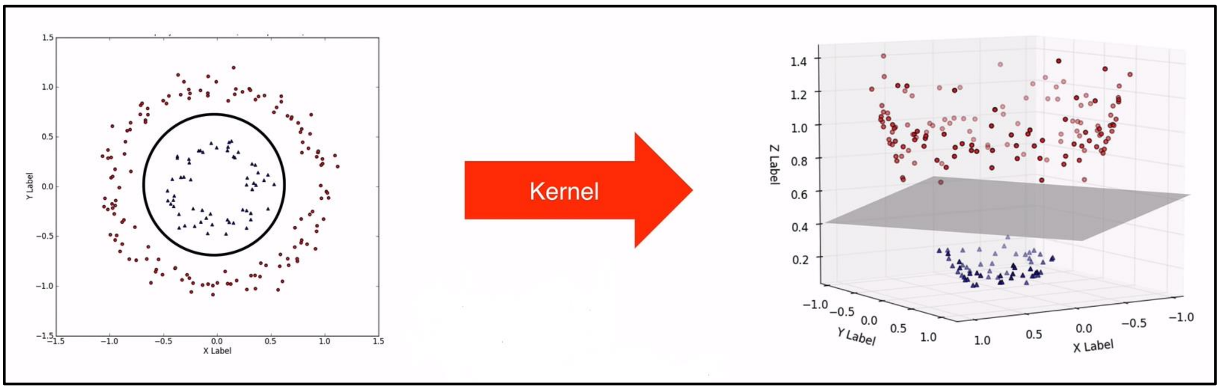

Figure 18: Projection from 2D into 3D [56]

After a kernel is chosen, one hast to tune its parameters to get good classification performance.

An advantage of support vector machines compared to other classification methods is that they require less input data for training purpose and in some cases it is easier to optimize them compared to e.g. neural networks. They work really well with small data sets and are well suited for high dimensional data.

A downside is that they have to map the data into multidimensional feature space and this can be very computational insensitive which might infringe its real time capability. Also, finding the right parameters for the model can be quite complex. [54] [55] [56] [57]

3.2.4. Neural Networks

The third widely used method to classify EMG signals is to use Neural Networks (NN). NN consist of multiple layers. Using mostly either Fourier analysis or wavelet transformation, specific features can be extracted from signals which serve as input for the neural network classification.

The first layer is the input layer where the different features are ingested into the system. The input layer has multiple nodes, so called neurons, which are all connected to the nodes of the layer behind it. Past the input layer are multiple hidden layers which further process the signal according to the inner nodes and their interconnections. The weighted sum of the whole previous layer is the input for each neuron. Inside each node, linear and nonlinear functions may be applied. Finally there is an output layer which then gives the result of the neural network. In the case of an EMG signal the input features could be specific frequencies, pauses in the signal or arbitrary looking artefacts. The output would be the classification of the movement. [58]

The inner nodes are weighted linear or nonlinear combinations of the neurons of the layer before. These weightings are built by teaching the NN. This teaching process is also called back propagation. It works by giving the network a specific signal and telling the network which signal it is. The neural network then recalculates how the weights would have to be to come to the same results. A strength of neural networks is that EMG signals which don't display relevant muscle activity can also be fed into the system to teach it how a relevant EMG signal does not look like. As neural networks are now being used more frequently, implementations in software are readily available. [57] [33]

Such a neural network layout can be seen in Figure 19.

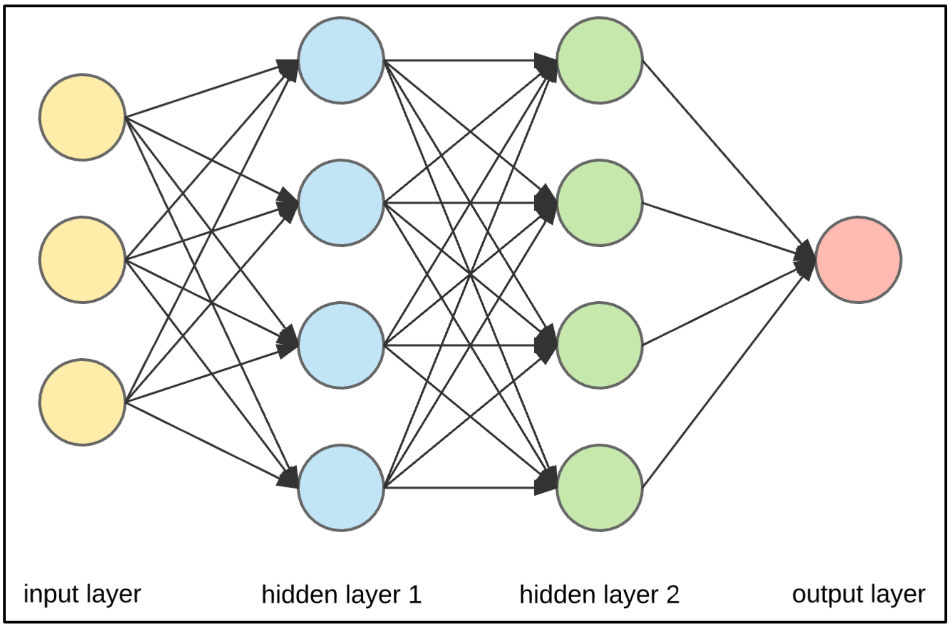

Figure 19: Layout of a Neural Network [57]

One downside of neural networks is that human knowledge cannot be injected into this classification system and thus the learning process takes longer and needs more input data compared to other classification techniques. Also it is not possible to understand or research why a NN outputs a specific value as the inner proceedings of a NN are unknown. The initial state is known, but through the learning process the weightings are changed. [58]

Some EMG prostheses are based on this kind of classification, as neural networks allow real time signal processing and controlling. They can be highly adapted to individual patients and thus reducing the failure rate. [33]

As all these systems have their strengths and weaknesses, it is also possible to use a combination to overcome those individual weaknesses. For example it is possible to use fuzzy logics to cluster features and then use neural networks for classification of those clusters. This joint use of several classification algorithms can certainly reduce the error rate. [33]

4. Practical implementation

In the following chapter, the individual steps that were necessary for the implementation of the project are explained in more detail. Different solution concepts are discussed as well as the hardware and software which was suitable to collect, transmit, process and visualize the data.

4.1. Concepts for implementation

Several different possible solutions were appropriate to be used for the implementation of the given task. These possible options were related to both, hardware and software choices, and are explained below.

4.1.1. Possible processing techniques

With regard to implementation, there were several possibilities on which devices the program could be executed. The choice was between the required resources and the resulting performance. Several different concepts based on microcontrollers or computers were considered. These different possibilities are described in more detail below and then explained on which bases the decision was made.

In all cases it was assumed that the muscle signals are recorded by a sensor which has an analog voltage as output. Thus, the analog voltage must subsequently be digitized and processed.

Microcontroller:

Considering processing devices, there were multiple options to choose from.

The first possibility was to use a microcontroller to measure the muscle activity and to evaluate the signals. Such microcontrollers could be e.g. Arduinos from the open source company of the same name or the MSP430 from Texas Instruments. Such development boards usually have a small energy-efficient processor as well as various connectors to communicate with other sensors and devices. For example, the Arduino Uno is based on an ATMEL ATmega328 chip, runs at a frequency of 16 MHz and can communicate with other devices via UART, SPI and I2C. In addition, the general purpose input and output (GPIO) pins can be used to control various other peripherals or to read analog and digital signals thanks to the built in 10bit analog to digital converter. Some of these GPIO pins can also be used as pulse width modulation pins to control servo motors. [24] [59]

This capability is particularly advantageous in terms of controlling prostheses, as such tasks require a control unit that controls the individual motors of a prosthesis anyway.

A further advantage of these development boards is that they can usually be programmed via various integrated development environments and don’t require dedicated accesses and bit manipulations of individual registers. This facilitates the handling and programming of these microcontrollers. In addition, such boards are widespread, easily accessible and there are large amounts of already created code examples and documentation to further facilitate programming.

A disadvantage, however, is usually the performance of the processors used. Since they have to be as small and energy efficient as possible, they often have only limited performance.

This means that only simple mathematical operations can be carried out on these devices. Otherwise it could have a significant impact on the performance of the entire system. This can outweigh by far the advantages, especially for real-time applications such as the EMG analysis discussed in this paper.

Due to the low performance, signals may not be read out on time and processing may take longer, resulting in a high delay in the overall system or limited functionality. [60]

Therefore, further alternatives for the control of the system were investigated, which have better performance to avoid getting into a bottleneck, limiting the overall performance.

In the figure below an Arduino Uno R3 can be seen. The GPIO connectors and the ATmega328 chip are easily recognizable.

Figure 20: Arduino Uno R3 [23]

Raspberry Pi:

Single-board computers, which have more power than the microcontrollers discussed above, were another option. The best-known representatives of this category are the computers of the Raspberry Pi series, whereby the current generation is the Raspberry Pi Model 3B+, which is manufactured by the Raspberry Pi Foundation. It is similar to an Arduino but has components that remind you more of a conventional computer. A Raspberry Pi Model 3B+ has several USB ports for input and output devices, an HDMI and Ethernet port, a quad-core processor with 1.4 GHz from Broadcom and 1 GB of RAM. As the memory space can be increased by an SD card, larger programs and scripts can also be stored.

Since the Raspberry Pi can be installed with a Linux-based operating system, it can be operated like a normal computer by means of a graphical interface. There are several editors available to create your own programs and there is also a wide range of code examples on the Internet.

Furthermore, the Raspberry Pi offers the possibility to control GPIO pins and to communicate with peripheral devices. However, the Raspberry Pi has no analog to digital converter and supports pulse width modulation only poured. This makes it difficult to control servo motors and therefore requires a control unit. [61]

Since the GPIO pins have no analog to digital converter functionality, a dedicated ADC would have to be connected in order to record signals with a Raspberry Pi. For example, the ADS 1X15 product family comprises ADCs that can be controlled via the pins of the Raspberry. These are also compatible in terms of voltage and communication protocols and have a resolution of 12 to 16 bits. [62]

In terms of costs, Raspberries are usually somewhat more expensive than conventional microcontrollers, but are also much more powerful than those. [63]

Figure 21 shows the different connections of a Raspberry Pi.

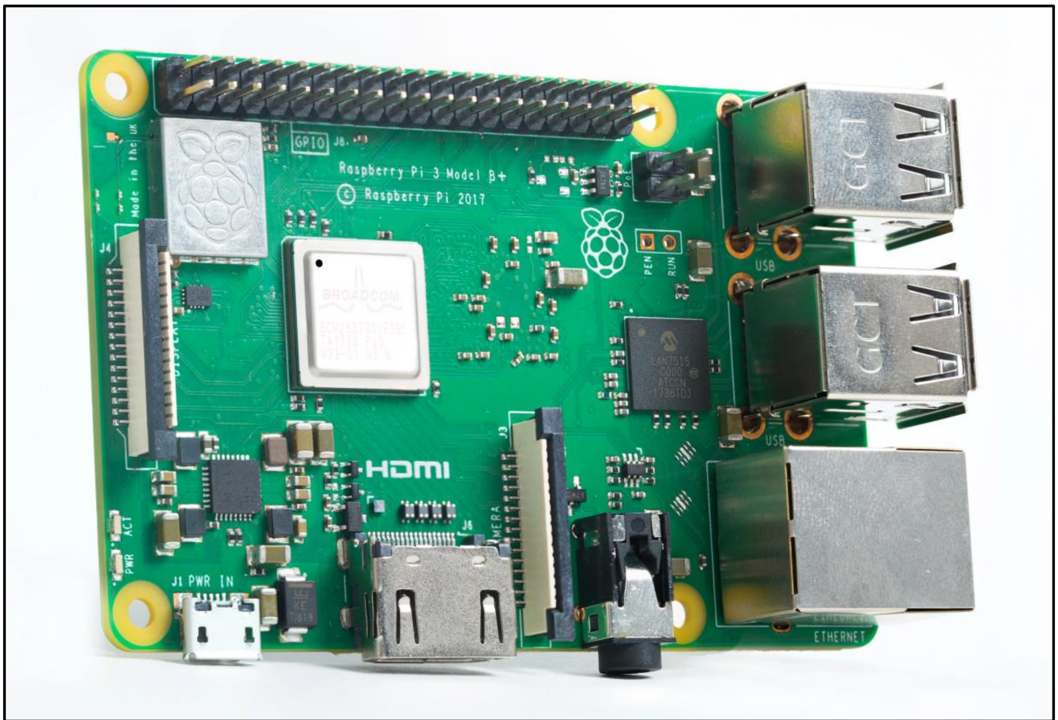

Figure 21: Raspberry Pi 3 Model B+ [63]

Another advantage of solutions using an Arduino or Raspberry Pi is that programs like Matlab have hardware support for these platforms. With these respective packages the devices can be programmed and functions such as filters and repetition instructions can be provided.

Conventional computer with DAQ:

The third way how the signals can be processed is by means of an ordinary computer. The signals are gathered by a DAQ, i.e. a data acquisition tool. This is necessary because most sensors cannot be directly connected to a PC.

The advantage of using a PC is that it is extremely powerful compared to the other alternatives. In addition, a large number of programs are available which can be used for further processing of the signals. For example, Matlab, LabView and other similar programs offer innumerable functions for signal processing and evaluation.

In addition, practically everyone can access a PC, making it easy to share a solution developed in this way with other people.

In this case, an Arduino can be used as a DAQ, to which the sensors are connected. The serial interface of the Arduino allows it to forward the signals it receives to the computer without any problems.

4.1.2. Concepts for signal processing

As already described in Chapter 3, there are several ways to extract information from signals. The different methods have different advantages and disadvantages. These are shown above all in the robustness of the solution with fluctuations of the input signal, the simplicity of the implementation as well as the required hardware resources. Thus, simple solutions could also be implemented using thresholds on devices that only provide relatively low hardware performance. However, their performance might be inferior.

The other methods of classification aside plain thresholding sometimes require considerably more resources. Especially when training a neural network a lot of processor power is needed. In many cases even a dedicated graphics card is recommended to train neural networks for new tasks. [64]

Since better performance was expected from more complex code and processing, the decision was made to use more powerful hardware. Solutions using PC and Raspberry Pi were considered. With regard to classification, both Neural Networks and Fuzzy Logic were narrowed down.

A solution using Arduino and simple thresholds would also have been possible, but this would have had to be connected to an output device anyway in order to display the recognized gestures graphically. For these reasons, the other options were preferred.

4.2. Used hardware

The two concepts, using Raspberry Pi and computers, were followed more closely. The structure and the components used are described in more detail below. In both cases, servo motors were attached to the hardware to simulate control of a prosthesis.

4.2.1. Data acquisition

The first step in data acquisition was to identify the correct positions for the electrodes and to correctly prepare these spots.

In order to be able to recognize gestures correctly, several muscle signals must be tapped. To classify the individual fingers, at least 4 sensors are required. One sensor for the thumb, one sensor each for the middle and index finger and one sensor for ring and little finger together. Theoretically a 5th sensor could have been used to have a dedicated sensor for each finger and thus a dedicated signal. This turned out to not be necessary because the individual fingers could be clearly identified with only four sensors. Meanwhile, it became apparent that 3 sensors were not sufficient to detect the individual fingers apart with certainty. With only 3 sensors, especially the little finger and ring finger could not always be distinguished. Therefore 4 sensors were chosen for the further proceeding.

A total of 9 electrodes were used. Four pairs with two electrodes each as well as an additional electrode, which was used as reference electrode. This reference electrode was placed on the back of the forearm at the level of the elbow, as little muscle activity was to be expected there.

The remaining positions where the electrodes were mounted are shown in Figure 22. The signals of the following muscles were tapped [65]:

- **Thumb:** Extensor Pollicis Brevis and Abductor Pollicis Longus
- **Index finger:** Flexor Carpi Ulnaris
- **Middle finger:** Flexor Digitorum Superficialis
- **Ring and little finger:** Palmaris Longus and Flexor Carpi Ulnaris

When attaching the electrodes, it was important to make sure that the skin was prepared in advance. This included the removal of excessive hair as well as the cleaning of lipid residues with soap or alcohol swabs.

Round self-adhesive silver-silver chloride disposable electrodes from the company TIGA-MED were used which had a diameter of 48mm. [66] Due to the size of the electrodes and the spacing given by the sensors, they had to be cut to size. Care had to be taken that the gel body in the middle was not damaged, as this could have an influence on the signal transmission quality. Alternatively, it would also have been possible to use children's electrodes, which usually have a diameter of about 25 mm. With these, cutting to size would not have been necessary.

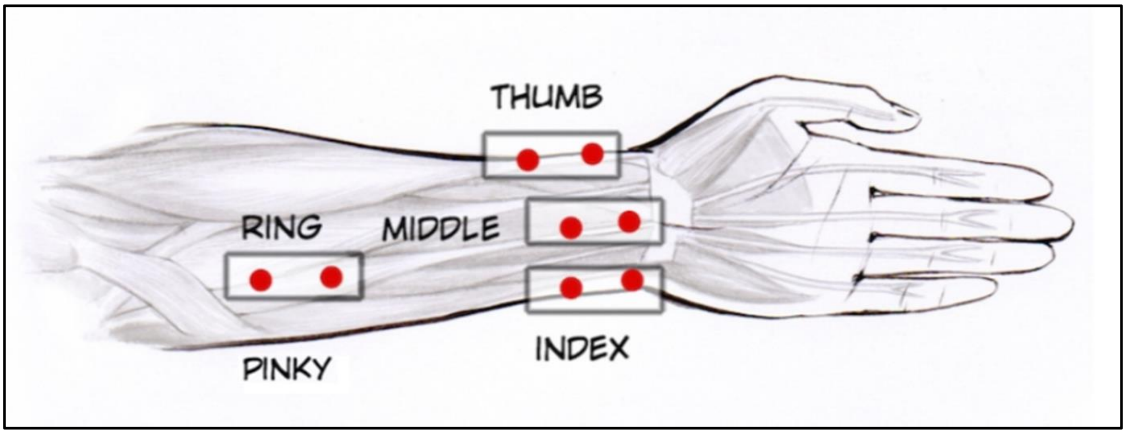

Figure 22: Position of the electrodes [65]

The sensors were then connected to the electrodes themselves to amplify and transmit the muscle signals. These sensors were the MyoWare muscle sensor from Advancer Technologies.

These sensors can be operated with 3.3V or 5V and can output signals in the 0 to V_{Input} range. A processing logic is attached to the sensors, which can generate the signal envelope. If this function is not used, the raw EMG signal can also be picked up. In this case the output signal is centered around the value $V_{Input}/2$.

The MyoWare sensors have an input impedance of 110G Ohm and a common mode rejection ratio of 110. The signal gain can be adjusted by means of a potentiometer. [23]

The output signal of the MyoWare muscle sensors were analog signals, which had to be processed by following devices.

The sensor itselfe is depicted in Figure 23.

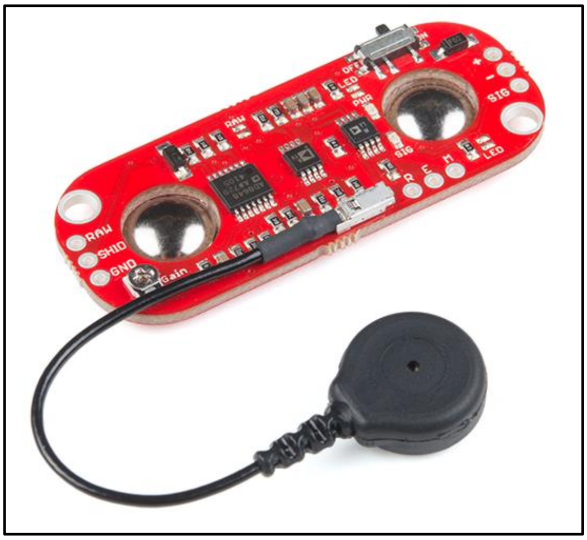

Figure 23: MyoWare Muscle Sensor

In addition, a cable was made with which the reference electrode could be attached further away as the standard cable was to narrow.

In addition, all sensors could be connected to the same reference electrode by using the crafted cable.

4.2.2. Data processing

As described above, there were two approaches to solving the problem. On the one hand by means of a Raspberry Pi in combination with an ADC, and on the other hand by means of a computer and a DAQ. The following two variants were considered:

Conventional computer with an Arduino as DAQ:

In this approach an Arduino Uno R3 was used as DAQ, which in turn was connected to a PC. The 4 sensors were connected to the Arduino with 3 lines each. These were 5V, GND and the data line. The lines were tangled to make the handling easier and to receive less interfering signals.

Via the data lines the analog signals were forwarded to the ADC of the Arduino and converted to digital 10Bit signals. This was done via the pins A0-A3 of the Arduino. The signal lines of the servo motors were connected to pins 11 to 9 as well as 6 and 5. These concrete pins were chosen because the pulse width modulation functionality is given there.

Since the motors required more power than the Arduino could supply, they were supplied by an external power supply. However, the ground lines were connected so that all voltages referred to the same potential. Otherwise it would not have been possible to control the motors.

The Arduino itself was subsequently connected to a computer via an USB cable. This experimental setup can be seen below. However, power cords and the USB cable weren’t drawn in this schematic.

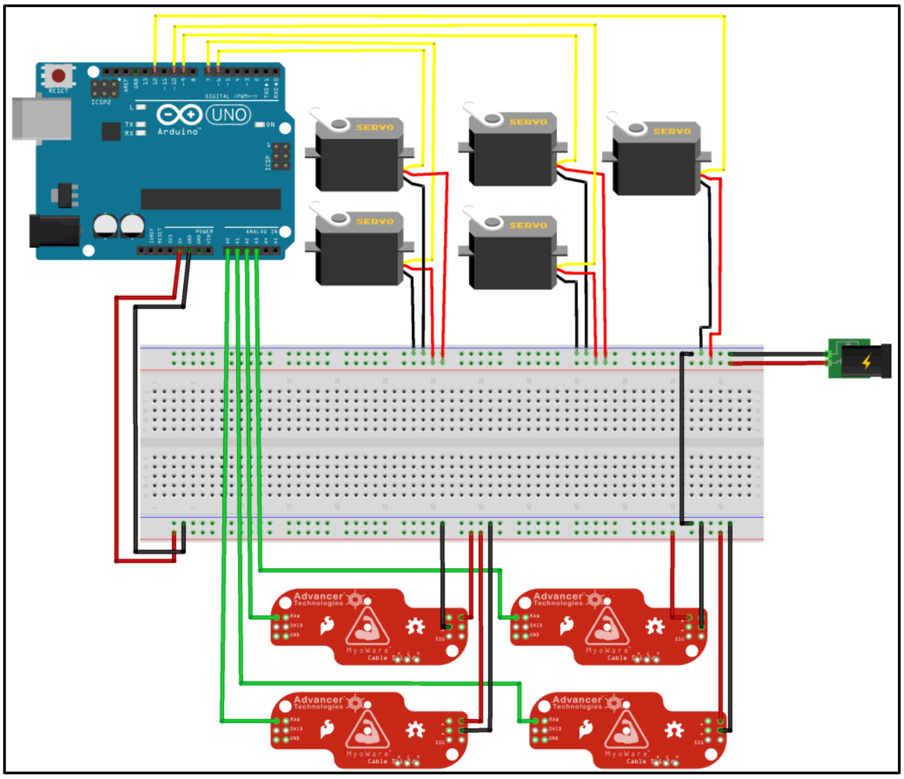

Figure 24: Schematic of first experimental setup

Raspberry Pi 3 Model B+ with ADS1015 ADC:

In the second approach, a Raspberry Pi was used as the central control unit. In addition, an ADS1015 analog to digital converter and a PCA9685 pulse width modulation servo driver were used. I2C was chosen as the communication protocol, as it was supported by all 3 parts. The I2C addresses could be set for both, the ADS1015 and the PCA9685.

The wiring has been carried out as shown in Figure 25. As is the case before, the servomotors were supplied by a more powerful current source, whereby the ground potential was again connected to the logic ground potential.

The used ADS1015 had a resolution of 12bits, whereby 11bits were used for the result and one bit for the sign. With a sampling rate of up to 3300 samples per second, this converter was fast enough for the expected frequencies. [62]

The PCA9685 PWM chip was used to control the individual servos needed to output the recognized gesture. The Raspberry Pi itself would not have been able to do this because it only has pulse width modulation signals generated by software. This can lead to malfunctions and problems when controlling servo motors.

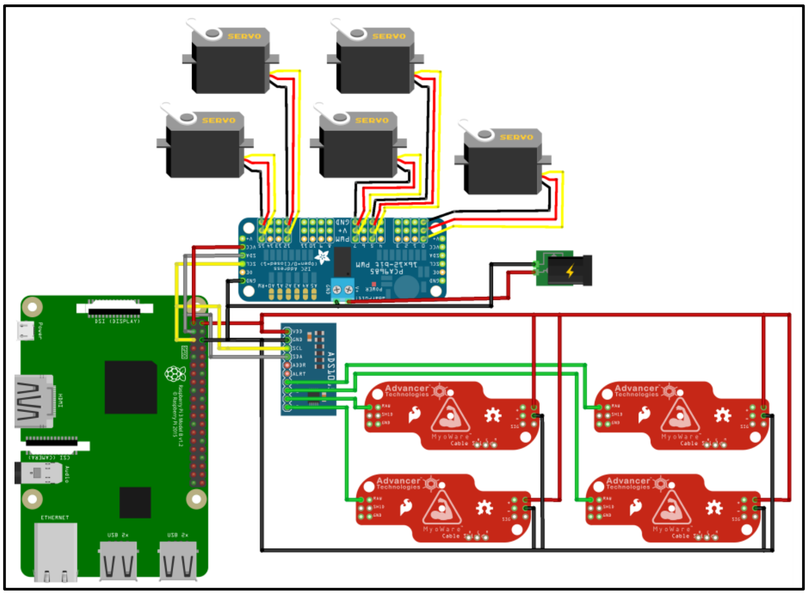

Figure 25: Schematic of second experimental setup

As in the case of the first schematics, the power cables and the connecting cables to other peripherals were not drawn in here either.

4.2.3. Prosthesis implementation

After the gestures have been recognized, they should be displayed afterwards. On the one hand, this should be done as a graphical output on a screen and on the other hand, the recognized gesture should be reproduced by a mechanical prosthesis. Two different prostheses were used for this task.

The purpose was to show in a simple way which gesture was recognized and how myoelectric signals could be used to control such prostheses.

Sain Smart Humanoid Robotic Hand:

This robotic hand was used because it was a ready-made solution and the movements could be easily adjusted.

The prosthesis consisted of 2 metal rails between which 5 "9g Micro Servo" servo motors were mounted. With the exception of the thumb, the fingers were made up of 3 links which, thanks to rotation axes, were movable and could bend. The motors were attached to the tips of the fingers by means of connections and, if necessary, they exerted a pull on them. This pull allowed the fingers to be bent individually.

However, due to its simple construction and weak motorization, this prosthesis was not able to reproduce all gestures correctly. In addition, the motors did not manage to approach all positions from every position, as the torque required for this was lacking. The manufacturer stated a positioning force of up to 1.6kg/cm, but this could not be achieved. Especially when only small movements were made, the servomotors were difficult to handle, especially when they were under load. This could have been due to the power supply or to fluctuations within the production tolerances. [67] [68]

The Sain Smart Humanoid Robotic Hand can be seen in the figure below.

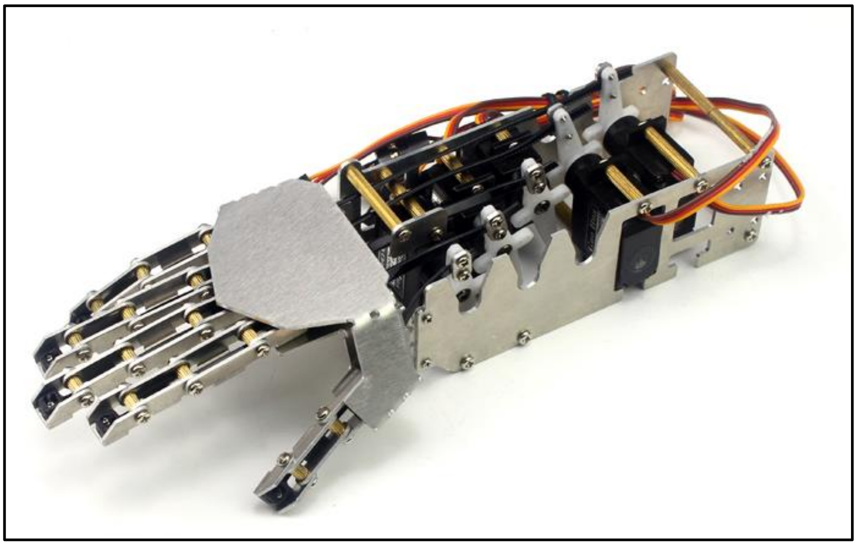

Figure 26: Sain Smart 5-DOF Humanoid Robotic Hand [68]

InMoov prosthetic hand:

The InMoov Hand offers a better way to display movements. It is part of an open source project of the French sculptor and designer Gael Langevin. The InMoov project consists of files for hundreds of 3D printed parts that can be connected to actuators to create a life-size robot. The individual limbs can be controlled and moved by means of motors.

The files for the upper left extremity were downloaded from this open source platform to be able to produce them using a 3D printer. In the 3D printing process, polymers are melted and applied layer by layer by means of a nozzle. These layers usually have a height of 0.1-0.4mm. Thus, small details and a high resolution can be achieved.

By this layer wise generative production, various objects can be produced, which are difficult or expensive to produce with conventional manufacturing methods.

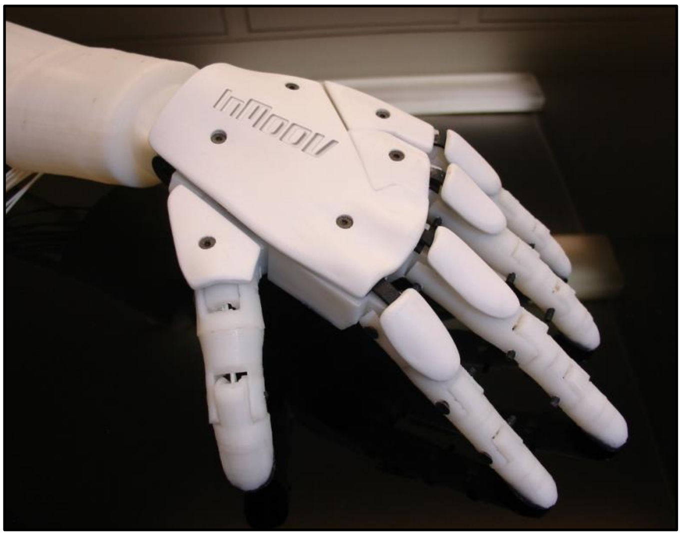

Figure 27: InMoov hand [69]

As with the Sain Smart hand, the fingers can be controlled individually. This is also done by means of 5 servos, which are mounted in the forearm and control the finger via cables. Another servo can be used to rotate the hand around the wrist.

Compared to the Sain Smart hand, however, larger servos such as the MG996R with up to 12kg/cm are used in this robot. In addition, the design went through several design iterations and is continuously adapted and improved. This is to ensure that the hand reacts adequately to signals.

Furthermore, the InMoov hand was life sized in comparison to the other hand. Due to the improved design and the larger dimensions it is anticipated that the gestures can be displayed better and more clearly. [70]

4.3. Software

In the following, the details of the software solution are described in more detail. The flow of information and the processing of the signals are briefly described.

4.3.1. Arduino code

This code refers to the approach where the Arduino was used as a DAQ in combination with a PC. It describes how the signals were recorded and subsequently how the prostheses were controlled.

Data acquisition and streaming:

Several different versions of code were used for data recording. On one hand code was written which collected test data in advance to train the classification algorithms. On the other hand, a program was written that was used during operation of the prosthesis.

The main difference between those programs was that one transmitting a constant data stream. The second program only recorded data for a certain period of time and repeated this process for a defined number of repetitions. Each of these time windows contained exactly one gesture. A portion of the code is shown in Figure 28.

```
while (counter < 250)
{
  if ((micros() - currentMicros) > 1000)
  {
    currentMicros = micros();
    counter++;
    Serial.println(5000);
    Serial.println(analogRead(A0));
    Serial.println(analogRead(A1));
    Serial.println(analogRead(A2));
    Serial.println(analogRead(A3));
  }
}
```

Figure 28: Example of Arduino code

The code, which recorded data for training purposes, took two or four second long signal samples from certain gestures. This process was repeated more than 400 times per gesture. Either the raw EMG signal or the signal envelopes were recorded during this type of measurement.

As can be seen in the code, during normal operation, time windows with 250 samples each were generated and transferred, which were then further processed. Since 1000 samples per second should be measured, i.e. twice the expected highest frequency, a time window had a duration of 0.25 seconds.

The analog signal, which was provided from the sensors was converted to a digital value with a resolution of 10bits, thus a value between 0 and 1023. This was done by means of the analogRead() command.

The baudrate for data transmission was set to 230400, as with lower baudrates, the serial communication was not able to transmit the gathered time frames quickly enough.

By means of the Serial.println() command, the signals were sent via the serial interface and were followed by a tailing '\n' indicating a new line. The data itself was sent in string format. Different other commands like Serial.write(), where the data is sent as bytes, were also tested, but this brought only minor performance improvements. On the receptor side, however, this caused problems with the serial buffer, so the string variant was chosen.

To exactly know which channel was transmitted at a current time, a sync value was used every fifth time. The sync value was chosen with 5000, as this was a value the ADC would never reach and thus was unambiguous. However, this sync-value was only used during real operation. When creating the test files this was omitted and the data was saved directly into a file instead.

The code itself was written in both the Arduino IDE and Matlab and loaded onto the microcontroller. When Matlab was used as the programming environment, it was also possible to program the Arduino using function blocks. This was made possible by the Simulink Support Package for Arduino hardware. [71]

Most of the code was written in Arduino IDE version 1.8.1 and Matlab 2019a.

Prosthesis control:

In order to control the prosthesis with the Arduino, further code was written. This code waited for a command from the computer to be sent via the serial interface. Depending on which character was sent, a corresponding gesture was output. A switch statement was used to identify the received character. A dedicated motor was assigned to each finger. This was done using the servo.attach() and servo.write() functions. The Arduino "servo.h" library was used to enable these commands. The individual fingers were then either stretched out or tilted in. [72]

The corresponding letter and gesture combinations can be found in Table 3.

Table 3: Corresponding gestures and characters

Gesture	Character	Gesture	Character
Idle	A	Fist	B
Scissor	C	Thumb	D
Index finger	E	Middle finger	F
Ring finger	G	Little finger	H
Thumb + index finger	I	Thumb + middle finger	J
Thumb + ring finger	K	Thumb + little finger	L

4.3.2. Matlab code

The functions and code snippets described below are related to those described in 4.3.1. The Matlab code started where the Arduino part ended. The code was divided into several functions that called each other. These are now described in more depth.

Data acquisition processing:

The first part of the Matlab code dealt with data acquisition and processing. The data was sent from the Arduino to the computer via a serial connection. Therefore a serial port object had to be created in Matlab. This was done with the command serial_object=serial('COM PORT', 'baudrate', baudrate). The correct COM port had to be selected, as well as the baudrate specified in the Arduino code. The object created in this way was then used for communication. The most important functions were fgets() and fprintf(). With these data between the devices could be read and sent.

The data were then merged to the respective timeframes. A part of the corresponding code can be found in Figure 29.

```
for i=1:250
    if(str2double(fgets(s))>1024)
        for j=1:4
            Timeframe(j,i) = str2double(fgets(s));
        end
    end
end
```

Figure 29: Matlab code to receive data

The query for a value above 1024 can also be recognized. This is related to the synchronization value, which was transmitted by the Arduino and in this particular case was 5000.

The individual time windows were then processed further. The first step of this processing was to use the detrend() function to subtract the mean value of the voltage in order to refer it to 0V. Then the root mean square, the standard deviation, the peak value and the mean frequency were determined. The values calculated in this way were then saved as shown below and used for the classification algorithms. These 4 parameters were calculated for all 4 channels of the EMG signal.

```
Results(:,1,1) = rms(detrend(Timeframe',0))';
Results(:,1,2) = std(detrend(Timeframe',0))';
Results(:,1,3) = max(abs(detrend(Timeframe',0)))';
Results(:,1,4) = meanfreq(detrend(Timeframe',1),Sample_rate);
```

Figure 30: Generation of classification values

Classification:

In order to classify the signals, two possibilities were considered more closely. The first was to train a neural network to classify the signals. The second possibility was to use fuzzy logic to recognize the gestures.

In order to implement these two types, both the Deep Learning Toolbox and the Fuzzy Logic Toolbox were installed.

For both variants, characteristic values first had to be generated that could be used as the basis for teaching the systems. The training files recorded by the Arduino were used for this and the corresponding parameters were calculated from these.

When training the neural network, these parameters were taken as input. Furthermore a file was generated in which it was specified which gesture the parameters correspond to. The training files as well as the gestures were used as input for the training. A neural network was used which was suited for pattern recognition and classification. This neural network teaching was started with the command "nprtool".

In this tool, the input and target files were selected as well as how many percentages of the input files should be used for training, validation and testing. The percentages were set to 70%, 15% and 15% respectively. The number of hidden neurons which are responsible for the functioning of the neural network was chosen with 50. With this setup, the neural network was trained. The structure of the system can be seen in Figure 31. [73]

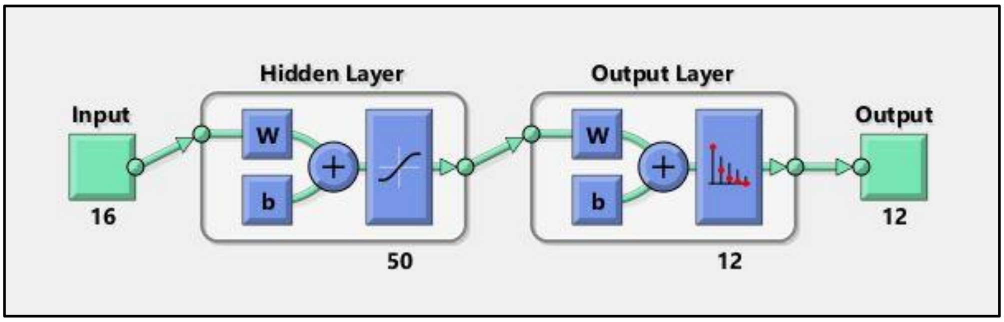

Figure 31: Structure of the neural network

The 16 inputs represent the 4 parameters of the 4 channels and the 12 outputs represent the different gestures.

For the classification with fuzzy logics the same test files were used again. These were deployed to reassign the associated gesture to the input variables. Subsequently, a tool was started via the Fuzzy Logic Toolbox and the command "neuroFuzzyDesigner", with which fuzzy logic systems can be created and taught.

In this tool it was set how many fuzzy membership functions should be created for each variable. For each input and output variable 3 membership functions were created. By using the Neuro Fuzzy Designer the individual weightings of these functions could be determined automatically. With the training finished this fuzzy system could be applied to further input data. [74]

With both variants the output was the recognized gesture. This output was then passed to a function that was responsible for the reproduction of the recognized gesture. If the new gesture was different from the one before, a character which was unique to the gesture was sent to the Arduino which was responsible for controlling the servo motors.

Graphical display:

In addition to the visualization with the Arduino, a graphical user interface was also created. This served basically 2 different tasks, to start the software as well as to display the recognized gestures. Therefore an app was created in which the other functions were located and started from there. The Matlab App Designer was used to create the app. This was done by means of callback functions, among other things. [75]

The output of the gestures was done by a text display as well as an image of the recognized gesture. To create the images, the Design Doll software was used, which is designed to simulate and control human postures. [76] This graphical interface can be seen below.

Figure 32: Graphical user interface

4.3.3. Raspberry Pi code

In the following, the code used with this solution alternative is described in more detail. Many procedures were identical with those of the first solution variant. In order to avoid redundancy in such cases, the exact procedures are only briefly outlined with reference to the previous pages.

The software solution for this variant was created on the text editor “nano” and was implemented in the programming language Python. This language was chosen because Python can be used to write executable programs in a simple way. It also offered the option to access many libraries and code examples. Two of the libraries used for the project were the ADS1015 and PCA9685 libraries which were used to ease communication with those two devices.

With the command adc.read_adc(Channel, Gain) the individual channels of the ADC could be read. The 4 EMG sensors were read out one after the other. The gain for the ADS1015 chip was 2/3. Thus measured values of +/-6.144V could be read in. By a print() request the results could be printed to the console. This was used to debug the program which turned out to be necessary. Also a version of the code was written, where the sensor values were written to a file. [62]

As in Matlab, the analysis of the signal was done by calculating several different characteristic values of time windows. These were the root mean square, the standard deviation and the peak amplitude. The time windows had a length of 250samples as in the first solution and a planned frequency of 1000 samples per second as well.

Even using the numpy library, some commands had to be written from scratch. For example the root mean square was calculated by the command rms = sqrt(mean(square(value))). The calculated values were then compared to thresholds.

The servomotors were again controlled so that the fingers were either stretched out or flexed. This was done with the command pwm.set_pwm(Channel, relative pulse start, relative pulse end). The difference between the two pulse lengths start and end was calculated and set in relation to 4096. From this the pulse width modulation signal and thus the position of the motor was calculated. The control of the motors was done according to the same schema as in Table 3, whereby again a switch instruction was used to select between the individual gestures.

4.4. Description of the test procedure

The test phase was divided into two parts. First, all methods were tested against test files to get comparable results. This was done to reduce the influence of random disturbances. The best method was then selected and its accuracy further tested.

In order to determine the reliability of the best solution, random gestures were made and tested to determine whether the gesture recognized matched the gesture which was done. There was no fixed sequence. This procedure was carried out with 2 other test persons in order to display to what extent the system was adjusted to a specific person through the training data. Each gesture should be presented 20 times to achieve some statistical significance.

Since the measurements of the test files were carried out on the left arm, this arm was also used for the comparison measurements on the test persons. Both a female and a male subject were selected, who also fell into different age groups. This was done to highlight possible deviations in the results.

The signals as well as test files were recorded whilst sitting. The electrodes were placed on the left arm and the arm itself was supported on the thigh. This was intended to reduce distortions caused by movement. This procedure was used for all subjects.

5. Results

In this chapter, the results obtained are presented and explained in more detail. In this context, both discoveries in the development process as well as decisions based on these discoveries are examined more closely.

5.1. Choice of hardware solution

As described in the previous chapter, 2 different hardware solutions were considered in more detail. On the one hand a solution with a computer and a DAQ and on the other hand a Raspberry Pi in combination with an analog to digital converter. Both variants offered advantages and disadvantages. These are discussed below.

5.1.1. General considerations concerning hardware

The advantages of the solution using a conventional computer consisted in the fact that this has by far the largest computational power. The processor unit of a conventional PC is up to 3 times faster as a Raspberry Pi and in many cases a graphics card is additionally available which can also accelerate the program flow. Furthermore, conventional computers usually have more and faster memory, which means that programs, functions and variables can be loaded more quickly.

In addition, there are further advantages with regards to the software. These consist in the fact that many programs exist, with which the necessary tasks which are required during the execution of this work can be accomplished. These programs included Matlab in particular, which was used for the majority of the processing procedures.

However, this variant also had major disadvantages. The first was that it was by far the most expensive solution in terms of total price. Most people have access to a computer, but should this not be the case, this variant would cost about 10 times as much as the one with the Raspberry Pi. In addition there are also the costs for the Arduino as well as those for the software. In the case of Matlab this can cost up tp several thousand Czech crowns.

The second big disadvantage is that this solution is not very portable. As long as you don't use a laptop you are bound to a stationary place with this solution. Even if the graphical output were not used, a computer would still be needed to process the data.

The second variant with the Raspberry Pi also had advantages and disadvantages. In most cases they are exactly contrary compared to the first approach. This means that the Raspberry is extremely mobile compared to a normal computer, especially if the graphical display is not used. In this case, the mechanical hand can be used to display the gestures. Moreover, the price of this solution is much lower than that of the first one.

However, the biggest disadvantage of this solution was the low processor power available. Since the code had to be processed in real time, this was a big problem. The processing power had to be sufficient to avoid data jams and to avoid processing the collected data for too long. This was also true for the other approach, but there the performance was much higher.

5.1.2. Practical implications for the Raspberry Pi

Both hardware variants were investigated, but the Rapsberry Pi variant turned out not to provide the required performance. This was due to the fact that the required sampling frequencies could not be achieved. There were several reasons for this.

The first reason was that the ADS1015 sensor did not reach the required speeds. The sensor has a sampling rate of 3300 samples per second, but it turned out that this refers to all 4 analog to digital converters of the sensor together. However, according to the Nyquist theorem, the 4 signals should be sampled with 1000 samples per second each, since the highest expected frequency is 500Hz. Therefore a sample rate of 4000 would be necessary.

However, this could have been solved by using two ADS1015 sensors. Furthermore the baudrate of the Raspberry Pi was set to 400000, because the standard rate of 100000 was too slow. This theoretically allowed the required speed to be achieved.

Nevertheless, it turned out that with this configuration it took about 1.3 seconds to receive and store 1000 samples. By calculating the individual parameters such as the root mean square, this was only made worse. To process a time window with 250 samples, up to 450ms were needed. Thereby the further processing and control of the program flow has not yet been taken into account. These would have only further slowed down the program and increased the cycle times.

This meant that during at least 45% of the time the muscle signals were not actively monitored. In addition, the time needed to control the servo motors and to recognize which gesture was being performed is not yet included in this numbers.

Since this was too slow to work, the solution was finally abandoned and the computer was used to select the solution.

5.1.3. Practical implications for the conventional computer

In contrast to the other variant, this solution allowed a sample rate of about 1400 samples per second per channel right from the start when no other tasks were pursued. This was limited by the speed of the Arduino's analog to digital converter and the possible baudrates. This conversion rates could have been further increased by directly accessing the corresponding registers and using interrupts. Theoretically up to 15000 conversions should be possible.

The receiving and storing of data also showed that this variant had considerably more processing power. Thus only about 330ms were needed for recording one time frame.

The control of the prosthesis could also be successfully implemented using this configuration. Therefore this variant was chosen. The complete real structure of the implemented setup is shown in Figure 33.

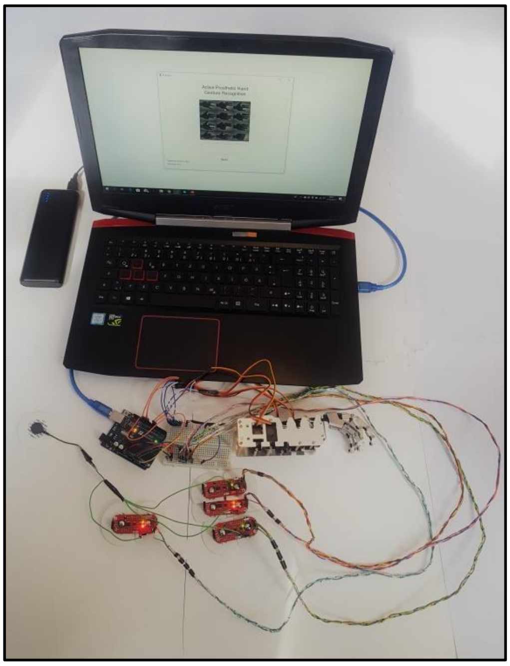

Figure 33: Structure of the real implemented setup

5.2. Software and signal analysis

This chapter describes the decisions and discoveries regarding the software and signal in more detail. This applies in particular to the processing of signals and the classification of gestures.

5.2.1. Findings concerning the signal

Before the signals were classified, the characteristic values had to be determined. Some decisions had to be made regarding data processing. These are discussed below.

Recorded Signal:

Signals as recorded by the MyoWare sensors and the Arduino as DAQ can be seen in the figures below. Figure 1Figure 34 shows the signal envelope of a gesture and Figure 35 the same gesture as raw EMG signals. The color scheme is as follows: thumb: yellow, middle finger: green, index finger: red and the remaining two fingers are shown in blue.

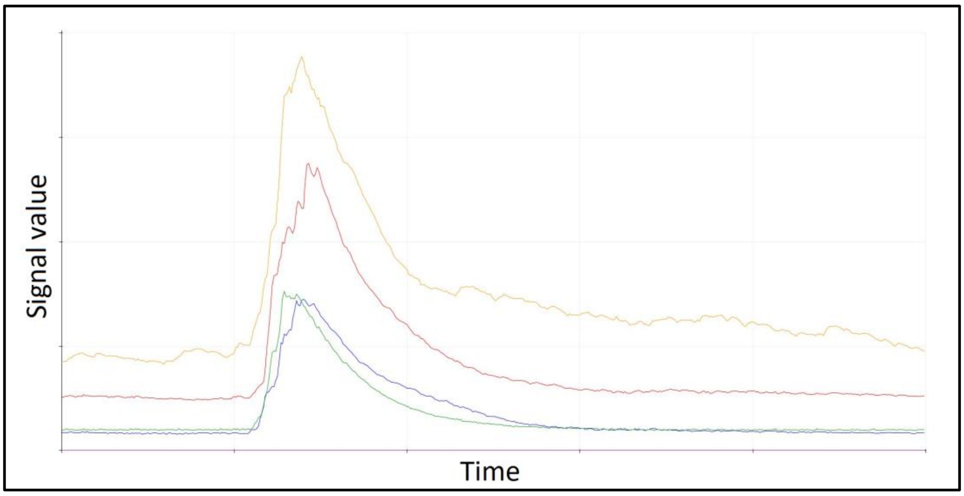

Figure 34: Signal envelope of the fist gesture

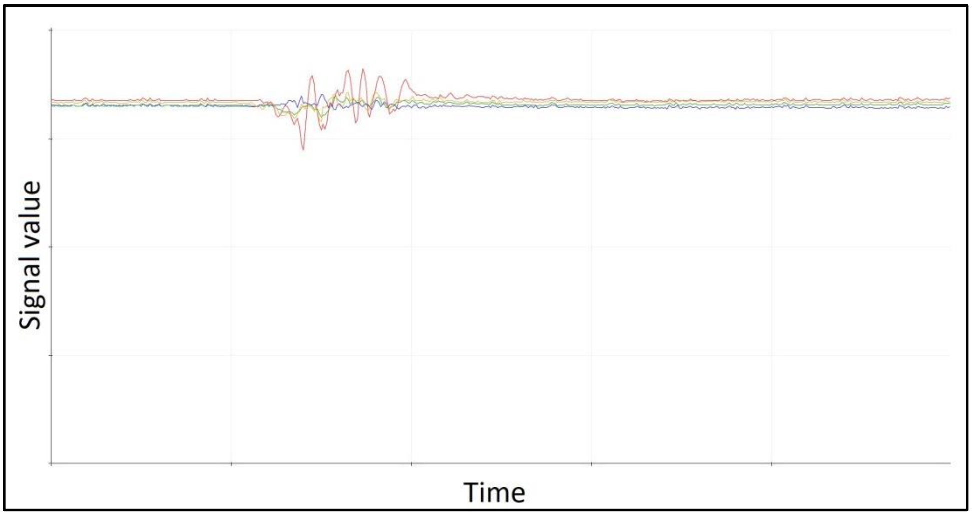

Figure 35: Raw EMG signal of the fist gesture

In the two images it can be seen that in both recording modes offsets occur between the individual channels, although the measured voltages are related to the same reference electrode. Therefore it was necessary to clean the individual channels from their offset. This was done in the software before calculating the individual characteristic values. Apart from the offsets that had to be removed, there were also other artifacts in the signals that made processing more difficult. These are shown in the following two pictures.

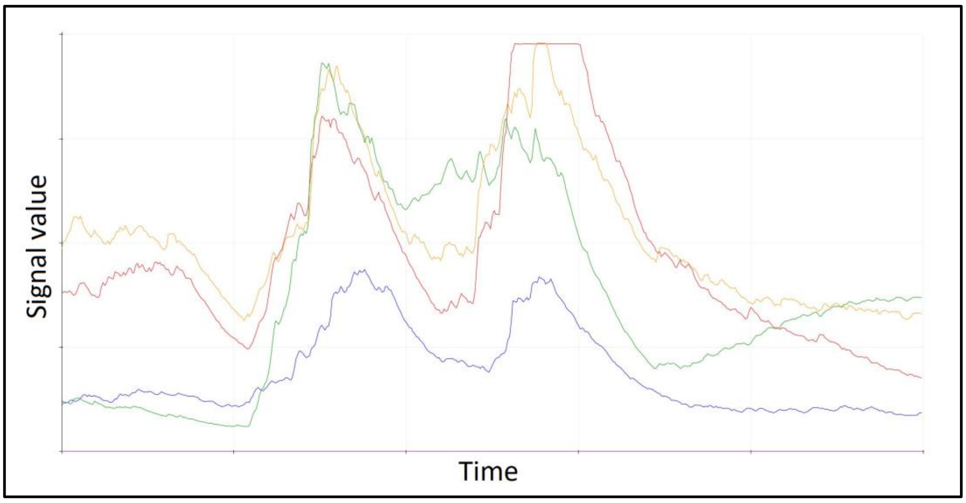

Figure 36: Characteristic signal features

This picture shows how the muscle signals could behave in case of initial movement of the fingers. The signal could get so high that it reached the maximum value of the analog to digital converter. The effect was amplified by the fact that the MyoWare muscle sensors are equipped with a CMOS amplifier. These can be saturated over longer durations or due to very high muscle activity, and react afterwards only delayed to a decrease of the activity. In this case, the maximum value was outputted for a longer time than it actually occurred. This could be prevented by further reducing the gain of the sensor. [23]

Furthermore, a second effect could be observed during data recording of this image. The muscle signals rose within a short time and afterwards weakened again considerably. When the hand returned to the resting position, the signals jumped up again. This is because the antagonists of the muscles were activated and there was also a re-polarization of the affected muscles into the resting state. Its results were the two high peaks which can be seen.

Finally, an effect can be seen below that occurred during the recording of muscle signals. These artefacts could be seen when the arm was moved or when the sensors themselves were moving. Due to their position around the wrist, this could also occur when gestures were performed or when tension occurred through the wires leading to the sensors. Such motion artifacts are shown in Figure 37.

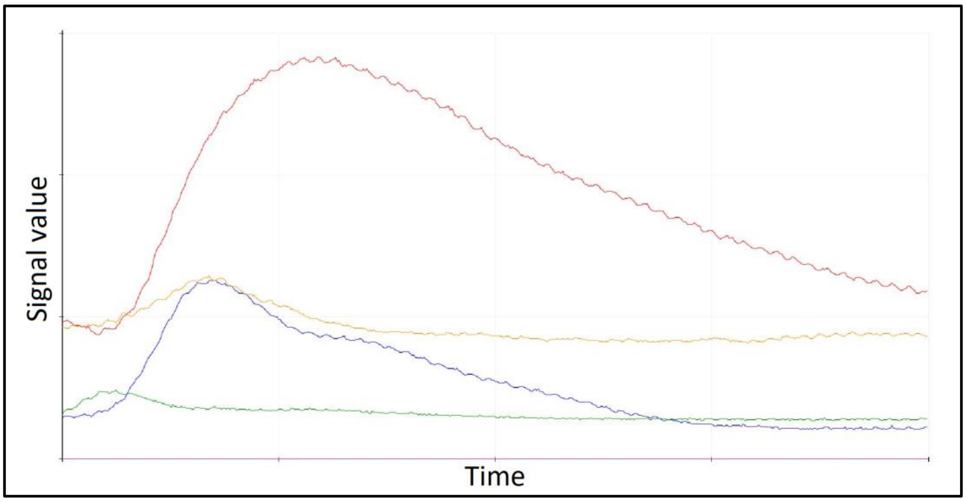

Figure 37: Movement artefacts

Concerning electrode position, no noticeable differences could be observed as long as the electrodes were roughly in the correct position. Although the signal became stronger or weaker depending on the exact position, the signal conditions remained more or less the same. However, there were 2 exceptions. On the one hand the electrode of the middle finger which was partly located above the tendons of the finger muscles. At some positions, the electrode moved heavily, causing disturbances and movement artifacts. Therefore, when positioning the electrode, it had to be ensured that the sensor did not fluctuate too much during gestures.

The second exception was the sensor, which was responsible for the ring or small finger. Depending on its position, the behavior of the signal changed when one of the two fingers was tensed. This could lead to the signals of both fingers to become relatively similar. However, the signals of the other sensors, in particular the sensor responsible for the index finger, could be used to distinguish between such gestures.

Time resolution:

It was important to weigh the real time capability of the solution against the quality of the signals. It was discovered that the longer the time window, the better the individual gestures could be recognized. In addition, this had a positive influence on the calculation speed of the characteristic values, since the time was not proportional to the length of the processed signal. The reason for this is that the fixed time costs were split among a prolonged time window.

For longer time windows, random fluctuations of the signals were less problematic because they had less weighting. In addition, it was more likely that whole gestures were in one time window and that gestures were not spread over several frames.

Shorter time windows had the advantage, however, that the feeling of real time was not affected. Long delays between input and output made the prosthesis feel unnatural. After studying the literature and trying out different time window lengths, the length was set to 250ms as this was a good average. However, window length of up to 1second also worked quite well.

Filtering:

As described in Chapter 3.1, filtering of the signals would be possible to improve the quality of the classification. It could also reduce the influence of interfering factors. However, there are also some arguments against filtering the signals. The most important reason why the idea of using e.g. Fourier or Wavelet transformations has been abandoned is that it represents another source of delay. The conversion of the signals from the time domain to the frequency domain and back again resulted in a delay that further worsened the real-time behavior of the system.

In addition, it turned out that, for example, filtering the 50Hz supply voltage was not necessary, since these only accounted for a small proportion of the frequencies occurring. By omitting filtering, further delays were avoided. The filtering and processing contributed about 100ms delay which, considering a signal length of 250ms, corresponds to a downtime of 40% in which no signals were read in.

Thus the filtering was limited to the most necessary. This consisted of subtracting the offset of the individual muscle signals as well as calculating the absolute of the signal values after subtracting said offset.

If it had not been for reasons of timing issues, filtering would have been highly recommended. However, there are some ways to alleviate the time problem. This includes e.g. bandpass filtering in hardware instead of software by using capacitors and inductors. A further solution would be to accelerate the software solutions by dedicated hardware or corresponding programs, which can also use other hardware such as graphics cards for processing.

Chosen parameters:

Furthermore, the selection of the characteristic parameters used to classify the signals was examined. These parameters were the root mean square, the peak amplitude, the standard deviation and the mean frequency.

During the analysis of the results it turned out that some of these values had considerably more information content than the others. These include in particular the root mean square and the peak amplitude.

The mean frequency showed the strongest fluctuations within the same categories. This is probably due to the short duration of the time windows and the low resolution of the frequencies contained in them. For example, the gesture “Fist” had a mean value of 99.3 with a standard deviation of 13.6. This shows that the results varyed quite widely. Therefore the determination of the mean frequency could be omitted without a greater loss of accuracy.

5.2.2. General considerations concerning classification

The second big decision that had to be made, apart from the hardware, was how to classify and recognize the data. As described above, two approaches were followed. These were neural network or fuzzy logic classification. The design of the individual systems can be found in Chapter 4.3.2.

Both approaches had their advantages and disadvantages, which lay mostly in the convenience of training. This was mostly demonstrated by the amount of training data needed to teach the systems. Both variants had a total of almost 4000 training files available. This was by far enough for the weighting of the individual fuzzy membership functions and also enough to train the neural network for classification. However, it turned out that with more data, the neural network could have achieved even better results. In contrast, the fuzzy logic could be excellently adapted.

The advantage of a neural network was that it automatically adjusted the hidden nodes by itself, making programming much more user friendly. When manually programming a fuzzy logic, this can quickly take on huge dimensions as the number of input and output variables increases. With 3 input variables, each with 3 membership functions, there are already 3^3 rules that have to be programmed.

The use of the "neuroFuzzyDesigner" function therefore had the advantage that the fuzzy logic was taught automatically. This combined the advantages of both approaches. This was also confirmed by the first trials at the beginning of the project in which a fuzzy logic was trained on a few gestures with 2 MyoWare sensors. This worked very well, but with an increasing number of inputs the weakness of this method became apparent. With 16 inputs, each with 3 member functions, the number of rules exceeded to 43 046 721. Even with the automatic derivation of these functions, this represented an enormous computing requirement.

5.3. Statistics of measurements and experiments

Apart from delay times of the two hardware options, many other static values were also determined in order to classify the performance achieved. For example, the recognition rate of the potential software models was analyzed. Such performance is discussed in more detail below. In addition, the overall performance of the achieved solution is subsequently considered.

5.3.1. Performance comparison of the individual systems

The performance of the individual systems was tested using the same test files to see how they performed in direct comparison. It was important to make sure that the same prerequisites were created. Otherwise bad signals could have led to a falsification of the results. Therefore, the muscle signals were stored in advance in a file and later used for testing of both systems. The recognition rates achieved can be seen below. The tests were carried out on the basis of the files of one person and represent values under ideal conditions.

However, it should be noted that the fuzzy logic classification was only applied to a smaller data set that did not contain all gestures and a minor amount of input variables. The reason for this was that due to the sheer size of the classification logic with 16 inputs, the program became unstable and no longer worked properly. Thus, only the root mean squares as well as two of the four peak amplitudes were taken into account for the fuzzy logic.

Table 4: Results of the fuzzy logic

Gesture	Detections
Fist	88%
Thumb	76%
Middle finger	94%
Ring finger	85%
	86%

Table 5: Results of the neuronal network

Gesture	Detections
Idle	91%
Fist	92%
Scissor	87%
Thumb	90%
Index finger	91%
Middle finger	89%
Ring finger	96%
Little finger	91%
Thumb + index finger	90%
Thumb + middle finger	85%
Thumb + ring finger	75%
Thumb + little finger	71%
	87%

The neural network achieved an average detection rate of 87%, with the best single detection rate being 96%. The fuzzy logic reached nearly the same value with 86% with 94% as the peak figure. This shows that the fuzzy logic can achieve similar results as the neural network. It should be noted, however, that due to the smaller range of gestures, this was easier than with a larger selection. On the other hand, the number of input variables was also lower, making it more difficult to distinguish between gestures.

The reason why the fuzzy logic performed so well is that it was also trained by a neural network. Thus, with small amounts of variables, it combines the advantages of both types. This shows that whilst using powerful hardware, the fuzzy logic could deliver even better results than a neural network.

However, since more gestures are to be recognized, the variant with the neural network was chosen for the final evaluation. During this, 2 additional test persons were consulted and live data were used.

5.3.2. Overall obtained performance

As described in Chapter 4.4, the solution was tested on multiple persons. This was intended to help to identify possible influences on the solution, as whether or not it performed the same when applied on different persons. It should also test whether the results are reproducible. Table 6 shows the individual results of the cycles. Each gesture was performed 20 times.

Table 6: Results of the measurements

Gesture	Detections subject 1	Detections subject 2	Detections subject 3	Total
Idle	100%	95%	100%	**98.33%**
Fist	85%	80%	75%	**80%**
Scissor	80%	65%	80%	**75%**
Thumb	90%	80%	95%	**88.33%**
Index finger	90%	80%	90%	**86.66%**
Middle finger	75%	90%	65%	**76.66%**
Ring finger	90%	75%	80%	**81.66%**
Little finger	85%	80%	85%	**83.33%**
Thumb + index finger	65%	75%	65%	**68.33%**
Thumb + middle finger	80%	55%	55%	**63.33%**
Thumb + ring finger	75%	65%	70%	**70%**
Thumb + little finger	70%	80%	65%	**71.66%**
	82.08%	**76.67%**	**77.08%**	**78.61%**

As can be seen from the results, the individual gestures could be recognized to a large extent. However, in 21% of all cases a wrong gesture was recognized.

The gesture with the highest achieved recognition rate was “Idle” followed by “Thumb” and the one with the worst was “Thumb + middle finger”. This could be due to the fact that gesture “Thumb + middle finger“ is very similar to other gestures which include the thumb and that the signal of the middle finger itself isn’t that strong either. This makes it difficult for the algorithm to recognize the differences. As can be seen, gestures that clearly differ from the others achieved the best results. Single fingers thus had better results than combinations of several fingers.

Most of the false statements were made by displaying "Idle" instead of the real gesture. This is because gestures were only accepted if they had more than 75% confidence over two consecutive time windows. This was done so that gestures do not appear arbitrarily while e.g. a fist is being formed.

It can also be seen that the best results could be achieved with subject 1. This is most likely due to the fact that the training data came from this person. As a result, the measured muscle signals were more similar to those used as a reference for learning. In addition, the position of the electrodes was more accurately known and more similar signals were tapped. A further reason why the results were different for the individual test persons is the differences in their anatomy and physiological status.

However, the accuracy difference was only about 5%. This shows that the system can be used quite flexibly.

These results are broadly in line with those achieved in similar projects. In these projects, recognition rates of about 80-90% were achieved. [26] [28] However, it should be noted that more gestures had to be distinguished than had to be done in for example [26]. The increased number of gestures thereby increases the complexity of differentiating between the signal structures as they are more similar than it would be e.g. with only 3 gestures.

Compared to [28], three fewer gestures had to be detected, however [28] used more and different kinds of sensors to achieve the 95% accuracy. In terms of EMG sensors alone, 87% of all gestures were detected in this project whilst using 8 EMG sensors.

As described above, one of the main goals was that the entire system processed the data as quickly as possible and reacted promptly to muscle signals. Therefore the code was optimized in this direction. However, not all delay influences could be eradicated.

The delays that occurred nevertheless were measured and are listed in Table 7.

Table 7: Time needed for one cycle

Cause of delay	Time needed
Recording of signals	330ms
Calculation of the characteristic values	30ms
Classification	35ms
Program logics (GUI, program flow,…)	50ms
	445ms

As can be seen, one cycle needed roughly 445ms. Some delays were expected because the processing of signals is automatically associated those. The largest factor leading to time loss was thereby the data recording and transmition. Without the 250ms that the signal itself needed to record, an additional 80ms were needed. This means that the actual sample rate was only about 750 samples per second. If one includes all delays, only 56% of the time signal samples were taken. The remaining 44% were used for processing. During this time the muscle signals were not monitored.

It should be noted, however, that the exact speed was also dependent on other processes running in the background of the computer. In general, the more background processes, the slower the performance.

However, it turned out that in most cases this did not constitute any problems for classification. The only problems arose when the majority of a signal was not recorded due to delays. In these cases, a correct classification could not be made. However, this was only the case with extremely swift gestures. In most cases these signals would have been filtered out anyway because they did not go past 2 time windows.

5.4. Possibilities to improving detection

Even though in most cases the gestures could be recognized correctly, there are still some possibilities to further improve the performance of the system. The data used to train the neural network is thereby the most important one.

The first suggestion for improvement is to collect more training data in general, so that the individual movements are better known to the classification algorithm in their different execution styles and speeds. This would be especially helpful in the case of the neural network, where the need for training data is very high.

In addition, it would be good not to obtain the training files from just one person, but to generate them from several individuals. This would also contribute to the robustness of the solution. Also, the influence of different anatomy of users would be better balanced. Thereby the difference of signals from different persons with different physiological dispositions would be better compensated.

Another major source of error was the interference picked up by the MyoWare sensors. The most prominent ones were the movement artefacts as shown in Figure 37. Mostly these effects were triggered by a motion of the sensor itself. This could be prevented by sensor solutions as shown in the following picture.

These devices consist of up to 8 myoelectric sensors which are worn as a bracelet around the arm and measure the muscle signals from all sides. Such sensors are better protected against unwanted movements by the radial mounting around the entire arm. In addition, some of these solutions are equipped with Wlan or Bluetooth, which eliminates the need for a cable connection. This further increases mobility with such solutions. [77]

Figure 38: Myo sensor armband

A further step to improve the recognition rate is to bring the training data closer to the real measured data. This is due to the fact that the training data were recorded over longer periods of time than it was the case in real operation. The longer duration recordings were done in order to have an entire event from tensing to relaxing a muscle in one time window. However, this also results in the training data being different from the real data. Therefore it would be advisable to shrink the training data into time windows with a length of 250 samples as well. However, this significantly increases the effort required to generate such data.

The remaining key factor in improving detection is the testing of other parameters. It has been shown that some are more suitable than others. Therefore, it is important to test which combination of parameters can achieve the best outcomes. The selection is not only limited to combinations of the parameter values used. Other values, such as those described in Chapter 3.1.1, can also be tested for the purpose of classification.

In addition, it is possible to not process the parameters as absolute values, but to set them in relation to the other values. This could further reduce the susceptibility to offsets of the recorded signals. These relations could be formed between the same values of the other channels or also other values of the own channel.

Apart from testing different classification parameters, it would be advisable to test the other classification systems apart from neural networks and fuzzy logic as well, and to weigh their performance against the achieved one.

Similar gestures could also be omitted in order to further increase the differences between the remaining gestures. This could avoid confusion between similar gestures.

It would also be advisable to reduce or optimize the hardware resources required to the necessary minimum in order to achieve a smaller, more portable solution. This could be achieved by single chip computers that have more power than e.g. a current Rapsberry Pi does.

6. Summary and discussion

In the following, the individual sections of the work will be recapitulated and then an outlook will be given on how this work could be continued. The required tasks, the results achieved and the implementation are also addressed.

6.1. Assignment and implementation

The work as presented above was created in the course of obtaining a Master's degree. Thereby certain tasks were set, as they were explained in the beginning. These tasks included giving a short summary of electromyography control of upper limb prosthesis as well as researching EMG sensing and processing possibilities to control such prosthesis. Subsequently, muscle signals were to be recorded, processed and used to control a robotic hand or a graphical display to show which gestures were performed. Finally, the results obtained should be evaluated and analyzed with regard to the degree of recognition achieved and potential improvements.

The purpose of this work was to show how to create a classification and recognition system for muscle signals using relatively cheap hardware. In the future, such a system could be used to help people with musculoskeletal disorders, such as muscle weakness, to operate a prosthesis. These artificial limbs would then have the necessary strength and utility to cope with the demands of everyday life. Such smart prostheses can thus help to compensate for disadvantages and support an independent lifestyle.

This is particularly the case when the recognition hardware is combined with prostheses which can be produced relatively cheaply, such as those produced in 3D printing processes. Considering that a large proportion of amputees live in countries of the 3rd world, where expensive sophisticated prostheses can only be afforded in the rarest of cases, this is particularly advantageous.

The tasks given to create the classification and control functioning could be fulfilled and the control of an upper limb prosthesis could be achieved. The muscle signals were recorded using four MyoWare muscle sensors which were attached to the forearm. The individual devices were assigned to specific fingers. The sensors recorded the signals, enhanced them and forwarded these analog values to an Arduino Uno R3, which acted as an analog to digital converter.

The Arduino was also responsible for communication purposes. The digital signals were then sent to a computer, where they were further processed. For the analysis Matlab 2019a was used in combination with several of its toolboxes. Furthermore, test files were created in advance which could be used to teach and train the classification system. The classification itself was performed using a neural network.

6.2. Results and outlook

Compared to the fuzzy logic, superior results were achieved with the neural network. Even though the fuzzy logic had the advantages of both solution variants it still couldn't capitalize on it. This has to do with the fact that the number of rules to be calculated was too big. Even though the training of the fuzzy logic was done through a neural network, this solution proved to be impractical. Thus, the chosen classification algorithm was the neural network.

Finally, the gestures could be displayed in a graphical output or via a prosthesis in real time. Hereby the mechanical hand was controlled by the Arduino and replicated the gestures which were recognized.

With regard to detection, it was shown that the best results could be achieved with the neural network. About 87% of all gestures could be recognized correctly during static testing procedures. The recognition rate of the individual gestures was between 71% and 96%. The best gesture recognized was the ring finger alone and the worst was the “Thumb + little finger“.

However, it turned out that the results varied depending on the test subject. The best detection rates were achieved by the test person, who generated the data for the training files of the algorithm, and thus to whom the whole system was adjusted to. Its results were about 5% better than those of the other subjects.

The results achieved are quite similar to those achieved in other projects. However, this is aggravated by the fact that a larger number of gestures had to be recognized in this project. Due to the similarity of the signals in similar gestures, this is more difficult than when only a few have to be distinguished.

Based on the results obtained, it would nevertheless be advisable to carry out further research in this direction in order to increase the detection rate. Some suggestions on how this could be done are outlined in the previous chapter. In general, it is based on the idea that more training data will improve the recognition rate. This allows the individual variables of the software to be better adapted and thus optimized. In addition, more data increases the chance that the training data itself will be closer to the real signals, which in turn would again increase the potential recognition rate.

Also, other classification techniques can be tried to see if they achieve better results. Examples of such were also explained in more detail in this paper. Furthermore, parameters of the programs can be adapted or the parameters can be replaced by completely different ones.

Finally, a processing and filtering of the signals would also be suitable, if this is possible through an optimization of the software or through the use of even more powerful hardware. It should be noted that the filtering should have as little influence as possible on the time behavior of the system.

If these suggestions are followed, it should be possible to further improve the solution and thus have a reliable option for the myoelectric control of prostheses that could help many people in particularly poor countries.

Bibliography

[1] "Medical Definition of Amputation," WebMD, 12 11 2018. [Online]. Available: https://www.medicinenet.com/script/main/art.asp?articlekey=12537. [Accessed 10 12 2018].

[2] M. LeBlanc, ""Give Hope - Give a Hand" - The LN-4 Prosthetic Hand," 11 09 2008. [Online]. Available: https://web.stanford.edu/class/engr110/2011/LeBlanc-03a.pdf. [Accessed 15 12 2018].

[3] T. Mitterhuber, "MyHandicap," Stiftung MyHandicap gemeinnützige GmbH, 07 2012. [Online]. Available: https://www.myhandicap.de/amputation-prothese/. [Accessed 10 12 2018].

[4] B. Greitemann, L. Brückner, M. Schäfer and R. Baumgartner, Amputation und Prothesenversorgung, Stuttgart, Germany: Georg Thieme Verlag, 2016.

[5] C. Harkins, A. McGarry and A. Buis, "Provision of prosthetic and orthotic services in low-income countries: a review of the literature.," *Prosthetics and Orthotics International,* vol. 37, no. 5, pp. 353-361, 2013.

[6] L. K. Manor, "Analysis of Upper Limb Prosthetic and BMI Data in Atlantic Canada," University of New Brunswick, New Brunswick, Canada, 2014.

[7] National Academies of Sciences, Engineering, and Medicine, The Promise of Assistive Technology to Enhance Activity and Work Participation, Washington DC, USA: THE NATIONAL ACADEMIES PRESS, 2017.

[8] DP Orthotics & Prosthetics, "UPPER EXTREMITY PROSTHETICS," unknown. [Online]. Available: https://dportho.co.za/services/limb-prosthetics/. [Accessed 10 12 2018].

[9] Merriam-Webster, "prosthesis," Merriam-Webste, 18 12 2018. [Online]. Available: https://www.merriam-webster.com/dictionary/prosthesis. [Accessed 23 12 2018].

[10] T. Gale, "Upper Limb Prostheses," Gale Encyclopedia of Nursing and Allied Health, 2006. [Online]. Available: https://www.encyclopedia.com/medicine/encyclopedias-almanacs-transcripts-and-maps/upper-limb-prostheses. [Accessed 10 12 2018].

[11] M. Kačičnik, "Smithsonian," 21 06 2017. [Online]. Available: https://www.smithsonianmag.com/smart-news/study-reveals-secrets-ancient-cairo-toe-180963783/. [Accessed 26 12 2018].

[12] PHYSICAL MEDICINE AND REHABILITATION, "Timeline: Prosthetic Limbs Through the Years," UPMC HealthBeat, 08 03 2015. [Online]. Available: https://share.upmc.com/2015/03/timeline-prosthetic-limbs-years/. [Accessed 10 01 2019].

[13] Dhwty, "16th Century Prosthetic Iron Hand: The Story of Gotz von Berlichingen," Ancient Origins, 23 06 2016. [Online]. Available: https://www.ancient-origins.net/history-famous-people/16th-century-prosthetic-iron-hand-story-gotz-von-berlichingen-006153. [Accessed 26 12 2018].

[14] PHYSICAL MEDICINE AND REHABILITATION , "Timeline: Prosthetic Limbs Through the Years," UPMC HealthBeat, 08 03 2015. [Online]. Available: https://share.upmc.com/2015/03/timeline-

prosthetic-limbs-years/. [Accessed 10 01 2019].

[15] E. Desteli E., I. Y., E. M., S. G. and Cosgun S., "Comparison of upper limb amputees and lower limb amputees: A psychosocial perspective," *European Journal of Trauma and Emergency Surgery,* vol. 40, no. 6, pp. 735-739, 2014.

[16] S. Watve, G. Dodd, R. MacDonald and E. R. Stoppard, "Upper limb prosthetic rehabilitation," *Orthopaedics and Trauma,* vol. 25, no. 2, pp. 135-142, 2010.

[17] touch bionics, "HAND COSMETIC PROSTHESIS / ADULT LIVINGSKIN," unknown. [Online]. Available: http://www.medicalexpo.com/prod/touch-bionics/product-80664-508908.html#product-item_508904. [Accessed 28 12 2018].

[18] Myrdal Orthopedic Technologies Inc., "Rebound Basketball Device," unknown. [Online]. Available: http://www.myrdalorthopedics.com/product/component-room-upper-sports-and-recreation-devices-rebound-basketball-device-749. [Accessed 28 12 2018].

[19] Fillauer LLC, "BODY-POWERED HAND PROSTHESIS / HOOK CLAMP / ADULT," unknown. [Online]. Available: http://www.medicalexpo.com/prod/fillauer/product-74954-769447.html. [Accessed 11 01 2016].

[20] Ottobock, "ody-powered upper limb prostheses," 2013. [Online]. Available: https://www.ottobock-export.com/en/prosthetics/upper-limb/solution-overview/arm-prostheses-body-powered/. [Accessed 28 12 2018].

[21] Ottobock, "bebionic Configurator - Medium/Large Hand," 2016. [Online]. Available: https://professionals.ottobock.ca/en/Prosthetics/Upper-Limb-Prosthetics/bebionic/bebionic-Hands-Wrist-Type/bebionic-Configurator--Medium-Large-Hand/p/bebionicML. [Accessed 3 01 2019].

[22] Ottobock, "AxonHook," 2016. [Online]. Available: https://www.ottobock.at/prothetik/armprothetik/systemuebersicht/axonhook/. [Accessed 3 01 2019].

[23] SparkFun Electronics, "MyoWare Muscle Sensor," Advancer Technologies, 17 03 2016. [Online]. Available: https://www.sparkfun.com/products/13723. [Accessed 07 01 2019].

[24] Arduino, "ARDUINO UNO REV3," Arduino, 2019. [Online]. Available: https://store.arduino.cc/arduino-uno-rev3. [Accessed 05 02 2019].

[25] MathWorks, "Deep Learning Toolbox," MathWorks, Inc, unknown. [Online]. Available: https://www.mathworks.com/products/deep-learning.html. [Accessed 10 01 2019].

[26] Š. Brázdil, "Multichannel EMG signal processing for gesture recognition," 2018. [Online]. Available: https://www.vutbr.cz/en/students/final-thesis?zp_id=110490. [Accessed 8 01 2019].

[27] MathWorks, "Robotics System Toolbox," MathWorks, Inc, unknown. [Online]. Available: https://www.mathworks.com/products/robotics.html. [Accessed 10 01 2019].

[28] J. McIntosh, C. McNeill, M. Fraser, F. Kerber, M. Lochtefeld and A. Krüger, "EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing," in *Proceedings of*

the 34th Annual ACM Conference on Human Factors in Computing Systems, San Jose, USA, 2016.

[29] K. Pirog, "Skeletal muscle structure and function," Institute of Genetic Medicine Newcastle university, unknown. [Online]. Available: https://blogs.ncl.ac.uk/katarzynapirog/skeletal-muscle-structure-and-function/. [Accessed 12 01 2019].

[30] Südwestrundfunk, "Kraftmaschine Mensch," Südwestrundfunk, 23 07 2016. [Online]. Available: https://www.planet-schule.de/sf/php/sendungen.php?sendung=6902. [Accessed 13 01 2019].

[31] OpenStax College, "Anatomy and Physiology," Rice University, 17 10 2018. [Online]. Available: https://courses.lumenlearning.com/nemcc-ap/chapter/skeletal-muscle/. [Accessed 18 01 2019].

[32] Ovij, "How do muscles contract?," Steemit Inc., 09 11 2017. [Online]. Available: How do muscles contract?. [Accessed 18 01 2019].

[33] M. B. I. Raez, M. S. Hussain and F. Mohd-Yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," *Biological Procedures Online,* vol. 8, no. 1, pp. 11-35, 2006.

[34] C. De Luca, A. Adam, R. Wotiz, L. D. Gilmore and S. H. Nawab, "Decomposition of Surface EMG Signals," *Journal of Neurophysiology,* vol. 96, no. 3, pp. 1646-1657, 2006.

[35] C. J. De Luca, "DelSys Surface Electromyography: Detection and Recording," 2002. [Online]. Available: https://www.delsys.com/Attachments_pdf/WP_SEMGintro.pdf. [Accessed 18 01 2019].

[36] S. Grimnes and O. Martinsen, Bioimpedance and Bioelectricity Basics, Oslo, Norway: Elsevier Ltd., 2015.

[37] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali and A. A. A. Bakar, "Surface Electromyography Signal Processing and Classification Techniques," *Sensors (Basel),* vol. 13, no. 9, pp. 12431-12466, 2013.

[38] T. Schaerer, "Elektro-Myographie (EMG) eine kleine Einführung," Elektronik Kompendium, 08 05 2017. [Online]. Available: https://www.elektronik-kompendium.de/public/schaerer/emg1.htm. [Accessed 19 01 2019].

[39] P. Kaur, "Interrupts short and simple: Part 1 - Good programming practices," Embedded.com, 03 10 2012. [Online]. Available: https://www.embedded.com/design/programming-languages-and-tools/4397803/Interrupts-short-and-simple--Part-1---Good-programming-practices. [Accessed 23 01 2019].

[40] S. Mallat, A Wavelet Tour of Signal Processing - The Sparse Way, Burlington, USA: Elsevier, 2009.

[41] A. Phinyomark, R. N. Khushaba and E. Scheme, “Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors.,” *Sensors,* vol. 18, no. 5, p. 1615, 2018.

[42] A. Phinyomark, C. Limsakul and P. Phukpattaranont, "A Novel Feature Extraction for Robust EMG Pattern Recognition," *Journal of Computing,* vol. 1, no. 1, pp. 71-80, 2009.

[43] F. Kerber, M. Puhl and A. Krüger, "User-Independent Real-Time Hand Gesture Recognition Based on Surface Electromyography," in *Proceedings of the 19th International Conference on Human-*

Computer Interaction with Mobile Devices and Services, Vienna, Austria, 2017.

[44] A. Phinyomark, S. Thongpanja, H. Hu, P. Phukpattaranont and C. Limsakul, "The Usefulness of Mean and Median Frequencies in Electromyography Analysis," 17 10 2012. [Online]. Available: https://www.intechopen.com/books/computational-intelligence-in-electromyography-analysis-a-perspective-on-current-applications-and-future-challenges/the-usefulness-of-mean-and-median-frequencies-in-electromyography-analysis. [Accessed 26 01 2019].

[45] Agilent Technologies, "The Fundamentals of Signal Analysis," unknown. [Online]. Available: http://www.modalshop.com/techlibrary/Fundamentals%20of%20DSP.pdf. [Accessed 26 01 2019].

[46] M. Mülle, © 2015, Erlangen, Germany: Springer, Fundamentals of Music Processing.

[47] Matlab, "Youtube," Matlab, 18 08 2016. [Online]. Available: https://www.youtube.com/watch?v=QX1-xGVFqmw. [Accessed 28 01 2019].

[48] National Instruments Corporation, "Introduction to Wavelet Signal Processing (Advanced Signal Processing Toolkit)," National Instruments, 06 2010. [Online]. Available: http://zone.ni.com/reference/en-XX/help/371419D-01/lvasptconcepts/wa_intro/. [Accessed 01 27 2019].

[49] MathWorks, "Continuous Wavelet Transform and Scale-Based Analysis," MathWorks, Inc, unknown. [Online]. Available: https://de.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html#responsive_offcanvas. [Accessed 28 01 2019].

[50] A. Ghiyamat, H. Shafri, G. A. Mahdiraji, R. Ashurov, A. Rashid and S. Mansor, "Impact of discrete wavelet transform on discriminating airborne hyperspectral tropical rainforest tree species," *Journal of Applied Remote Sensing,* vol. 8, no. 1, p. 083556, 2014.

[51] S. Riley, "Fuzzy Logic - Computerphile," Computerfile, 11 06 2014. [Online]. Available: https://www.youtube.com/watch?v=r804UF8Ia4c. [Accessed 28 01 2019].

[52] C.-C. Kuo, C.-H. Liu, H.-C. Chang and K.-J. Lin, "Implementation of a Motor Diagnosis System for Rotor Failure Using Genetic Algorithm and Fuzzy Classification," *Applied Sciences,* vol. 7, no. 1, p. 7010031, 2017.

[53] Creative-Commons, "Datei:Fuzzy logic temperature de.svg," Wikipedia, 01 04 2015. [Online]. Available: https://de.wikipedia.org/wiki/Datei:Fuzzy_logic_temperature_de.svg. [Accessed 28 01 2019].

[54] T. S. Korting, "How Support Vector Machines (SVM) algorithm works," Prezi, 20 08 2018. [Online]. Available: https://prezi.com/jdtqiauncqww/how-support-vector-machines-svm-algorithm-works/?utm_campaign=share&utm_medium=copy. [Accessed 28 01 2019].

[55] Augmented Startups, "Support Vector Machine (SVM) - Fun and Easy Machine Learning," Youtube, 15 08 2017. [Online]. Available: https://www.youtube.com/watch?v=Y6RRHw9uN9o. [Accessed 28 01 2019].

[56] A. Zhao, "Support Vector Machines: A Visual Explanation with Sample Python Code," Youtube, 21 08 2017. [Online]. Available: https://www.youtube.com/watch?v=N1vOgolbjSc. [Accessed 28

01 2019].

[57] A. Dertat, "Applied Deep Learning - Part 1: Artificial Neural Networks," Towards Data Science, 08 08 2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6. [Accessed 28 01 2019].

[58] M. Pound, "Neural Network that Changes Everything - Computerphile," Computerphile, 20 05 2016. [Online]. Available: https://www.youtube.com/watch?v=py5byOOHZM8. [Accessed 28 01 2019].

[59] Texas Instruments Incorporated, "MSP430 ultra-low-power MCUs – Products," Texas Instruments, 2019. [Online]. Available: http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/products.html. [Accessed 05 02 2019].

[60] E. Renard, "What are the strengths and weaknesses of Arduino and a typical microcontroller (coding, limitation, price, etc.)?," Quora, 19 01 2019. [Online]. Available: https://www.quora.com/What-are-the-strengths-and-weaknesses-of-Arduino-and-a-typical-microcontroller-coding-limitation-price-etc. [Accessed 05 02 2019].

[61] The Magpimagazine, "RASPBERRY PI 3: SPECS, BENCHMARKS & TESTING," The MagPi magazine, 2016. [Online]. Available: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/. [Accessed 10 02 2019].

[62] T. DiCola, "ADS1015 / ADS1115," Adafruit, 09 02 2016. [Online]. Available: https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115. [Accessed 10 02 2019].

[63] E. Upton, "Raspberry Pi 3 Model B+ on sale now at $35," Raspberry Pi Foundation, 14 03 2018. [Online]. Available: https://www.raspberrypi.org/blog/raspberry-pi-3-model-bplus-sale-now-35/. [Accessed 10 02 2019].

[64] F. Boylu Uz, "GPUs vs CPUs for deployment of deep learning models," Microsoft, 11 09 2018. [Online]. Available: https://azure.microsoft.com/en-us/blog/gpus-vs-cpus-for-deployment-of-deep-learning-models/. [Accessed 23 02 2019].

[65] Backyard Brains, "Experiment: Signal Classification," Backyard Brains, Inc, 2017. [Online]. Available: https://backyardbrains.com/experiments/RobotHand. [Accessed 27 02 2019].

[66] TIGA-MED Deutschland GmbH, "EKG-KLEBEELEKTRODE, NASSGEL, 30 STÜCK/ PACK," TIGA-MED Deutschland GmbH, unknown. [Online]. Available: https://www.tiga-med.de/Diagnostik-Geraete/EKG-Elektroden-Zubehoer/EKG-Klebeelektrode-Nassgel-30-Stueck-Pack. [Accessed 02 03 2019].

[67] Weiku, "Guohua Remote-Control Model Manufactory," Weiku, 2011. [Online]. Available: http://zsguohua.company.weiku.com/item/New-9g-Micro-Analog-Servo-12719571.html. [Accessed 01 05 2019].

[68] Sain Smart, "5-DOF Humanoid Robotic Arms & Hands," Sain Smart, 2019. [Online]. Available: https://www.sainsmart.com/products/diy-5dof-robot-humanoid-five-fingers-metal-manipulator-arm-left-hand-w-servos-for-robot-1. [Accessed 3 04 2019].

[69] InMoov, "Gallery V2," InMoov, unknown. [Online]. Available: http://inmoov.fr/gallery-v2/. [Accessed 02 05 2019].

[70] InMoov, "Gaël LANGEVIN – French InMoov designer," InMoov, unknown. [Online]. Available: http://inmoov.fr/. [Accessed 15 05 2019].

[71] MathWorks, "Arduino Support from Simulink," MathWorks, Inc, unknown. [Online]. Available: https://www.mathworks.com/hardware-support/arduino-simulink.html. [Accessed 20 04 2019].

[72] Ardunio, "Servo library," Arduino, unknown. [Online]. Available: https://www.arduino.cc/en/Reference/Servo. [Accessed 20 04 2019].

[73] MathWorks, "Classify Patterns with a Shallow Neural Network," Mathworks, Inc, unknown. [Online]. Available: https://www.mathworks.com/help/deeplearning/gs/classify-patterns-with-a-neural-network.html. [Accessed 12 05 2019].

[74] MathWorks, "Neuro-Fuzzy Designer," MathWorks, Inc, unknown. [Online]. Available: https://www.mathworks.com/help/fuzzy/neurofuzzydesigner-app.html. [Accessed 12 05 2019].

[75] MathWorks, "MATLAB App Designer," MathWorks, Inc, unknown. [Online]. Available: https://www.mathworks.com/products/matlab/app-designer.html. [Accessed 5 05 2019].

[76] Terawell, "Design Doll," Terawell, unknown. [Online]. Available: http://terawell.net/terawell/. [Accessed 10 05 2019].

[77] Noth Inc., "Myo Gesture Control Armband tech specs," North Inc., unknown. [Online]. Available: https://support.getmyo.com/hc/en-us/articles/202648103-Myo-Gesture-Control-Armband-tech-specs. [Accessed 13 05 2019].