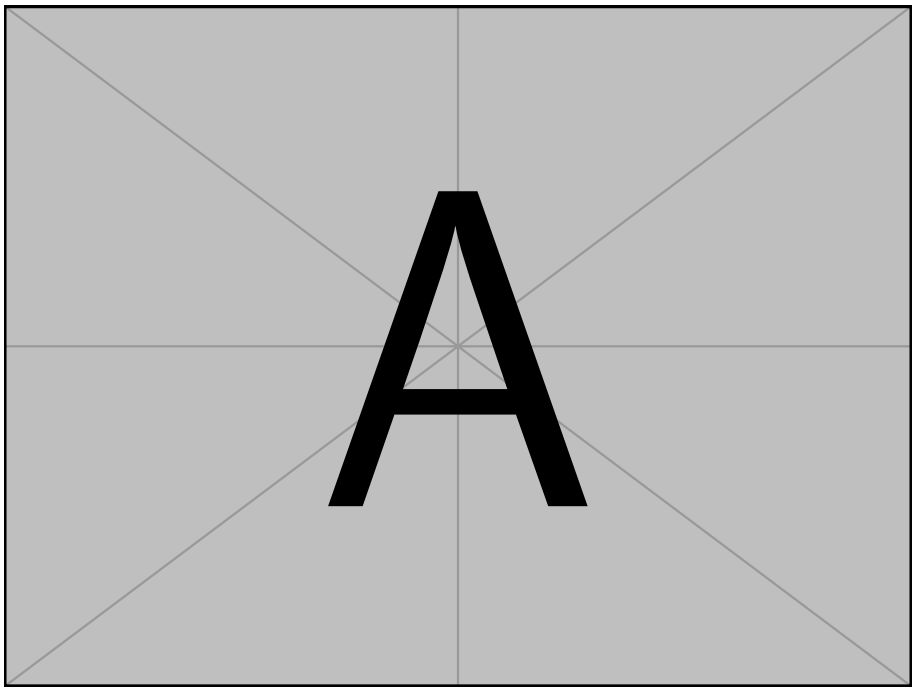
Surface-EMG Processing & Classification for Muscle Interfaces

University of Southeren Denmark

Supervisors:
Poramate Manoonpong (poma@mmmi.sdu.dk)
Xiaofeng Xiong (xizi@mmmi.sdu.dk)

Date:



1 Abstract

Hello, this is my abstract...

2 Acknowlegdements

Hello, here is some text without a meaning...

Glossary

abduction/adduction Movement away/towards the midline of the hand. 14, 15

Congential A disease or physical abnormality present from birth. 6

flexion/extension The bending movement of the finger. 14

phalange Bone in the finger. 14

sEMG surface-electromyography. 8

traumatic A disease or physical abnormality due to trauma. 6

Contents

3 Introduction

The human hand is one of the most important factors of the human identity. The hand allows a person to perform complex muscolatory combinations to interact with the surrounding world, express complex emotions during speech, and aid in defining a person's individuality and personalty [**???**]. The hands are controlled by a complex combination on precise muscles designed to perform gentle, precise control of the fingers. This allows a person to grasp objects in many different ways, perform complex tasks such as writing, playing musical instruments, or even constructing a house. The hand also acts like a sensory device allowing us to perform precise observations through feeling and touch. This allows a person to understand the environment without seeing it, the hand is able to sensor heat/cold, create complex understanding of geometries and texture through touch and manipulation.

Missing limbs, either Congential or traumatic amputation severely reduces a person's ability to interact with- / understand the world, express themselves and perform simple day-to-day tasks. In order to alleviate some of the drawbacks of missing a limb, amputees are often able to aquire a prosthetic replacement of their lost limb. The aquired prosthetic tries to imitate the movements of the lost limb, through muscle-activated interfaces, that is then used to control the movements of the prosthetic. In the case of hand prosthetics, the prosthetic allows the user to perform simple, day-to-day tasks, and is able to alleviate some of the stress caused to the non-amputated hand through overusage.

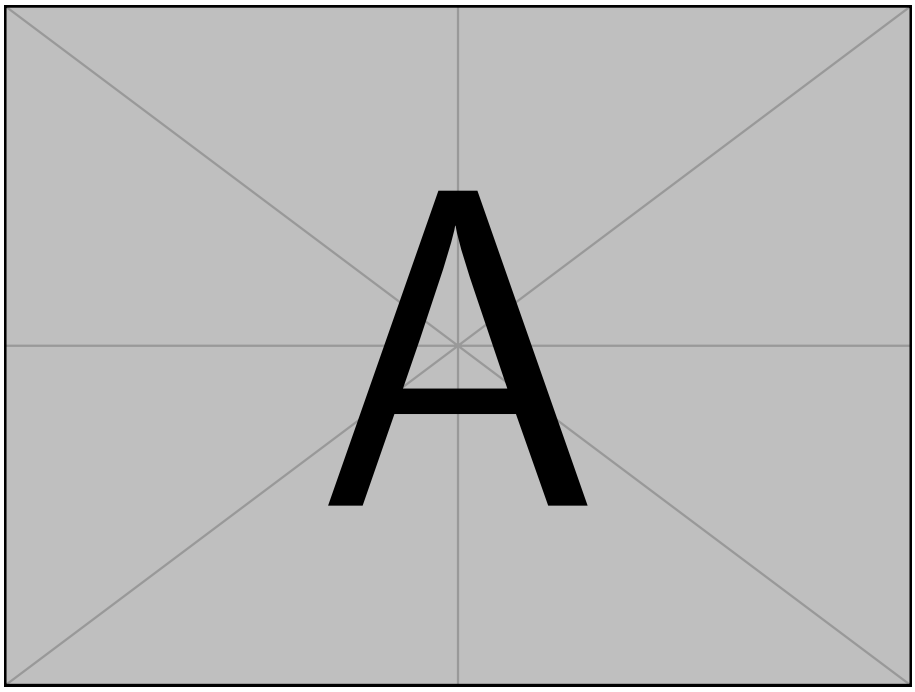


Figure 1: Example figure text

This thesis aims to summarize, and elaborate on current state-of-the-art research and products in the field of prostetics devices, and products the control of prosthetics and the existing limitations of these state-of-the-art products.

This thesis aims to contribute to the world of prosthetics control, by researching effective methods of collecting sensory data from the lower-/upper-arm, and by doing so, creating an state-of-the-art Artificial Intelligence (AI) based controller, that is able to imitate the intent and movements or a real hand. And by doing so, by improving sEMG controller design to increase functionality and the controllable DoF of the prosthetic, to provide a more true-to-life experience to the prosthetics user, and thus reduce the amount of patients that disregard prosthetics.

This thesis also aims to explore efficient methods of designing a network to identify lower-/upper-arm muscolatory intent, with the purpose of controlling a simulated prosthetics device, and by doing so, increase the controllable Degree-of-Freedom for the prostetics user.

4 Problem Specification

There is a large need for new technology that improves the effectiveness and ergonomics of human hand prosthetics.

Current state-of-the-art products on the market exhibits a severe reduction of controllable Degree of Freedom (Dof) compared to their biological counterparts. These products often rely on simple, grasp control based on 2 or more surface-electromyography (sEMG) interfaces, to classify "open/close" signals for the control scheme. The prosthetics user is then manually required to change grip control-scheme, creating very crude control dynamics that is very different from biological hand-control. A indebth explanation of "open/close" control can be seen in section **??** This is a great pitfall in the field of Research and Creation of prosthetics, as unsatisfactory function of prosthetics lead to amputees, that exhibit a great deal of stress douring the rehabilitation process.

This can cause the patient to repel the rehabilitation process and the prosthetic all-together. The repelling of the prostetic increase in the cases of the most severe cases of amputation, where the largest amount of control muscles are lost. These amputations are often located further up the limbs, where the loss of mobility and controllability are greatest. The amount of muscles leftover from amputation also dicates the type of prosthetic a patient is able to recieve. Patients of lower-arm amputation has less control over their prosthetic than patients of hand amputation, due to the loss of the muscles in the lower-arm. The loss of control increases as the amputation severity increases, and this is a problem in prosthetics design because it is impossible to create a standardized controller that suits most patient's needs.

State-of-the-art commercial prosthetics further decrease the controllable DoF in order to increase robustness of the control experience, this is further elaborated upon in **??**.

4.1 Motivation

The main goal of this thesis is to provide a meaningful contribution to the world of prosthetics design and control. In order to confine the workload done in this thesis, a set of development goals has been made:

1. Create a software-based, biology-inspired, anatomically realistic simulation of a humanoid lower-arm/hand that is able to imitate the movements of the humanoid limb.
2. Make the prosthetics simulation controllable from a widely-used robotics-software.
3. Design a sEMG muscle pre-prossesing pipeline for a prosthetics controller.
4. Design a state-of-the-art prosthetics controller based on AI, to control a simulated prosthetics device.
5. Create a custom dataset to train AI based controllers for prosthetics.

6. Test and Validate the created prosthetics controller against state-of-the-art methods.

5 Literature Review

As the capability of sensor technology increases the possebilities for biological sensoring becomes more important in state-of-the-art prosthetics development and research. There becomes a larger need for understanding and translating human-created sensor data into usable inputs for prosthetics and Human-Machine-Interfaces (HMI's). A lot of reserach has been done in this area, this research elaborates on different Machine learning or AI-based methods of understanding muscle-based sensor data. The pipeline for converting EMG sensor data to usable input data often contains a pre-processing step, where data is de-noisified, and cleaned of potential errors. The pre-processing step can also contain feature extraction such as ... The pre-processing step is then followed by a processing step, this step encompasses the use of a Machine-learning algorithm or a Neural Network, designed to either Classify a grip type, or Regress the angle of the joints. After Classification or Regression a post-processing step can be added where the actual kinematic data is created, and used as input to the prosthetic controller. Popular methods of processing EMG signals will be researched, and elaborated upon, with the aim of identifying robust, effective and implementable methodologies.

5.1 Introduction to Literature

Human Machine Interfaces (HMI), are control systems that enables humans to interact and control a mechanical or robotic system. As explained in the paper [1], researchers and prostecisists have been developing mechanical prosthetics for many years. One example of such devices would be the ankle-foot-orthoses, a support device strapped to the angle, used to relieably adjust the pressure applied by the body while walking, to help impared individuals with walking in a more natural way. [1] proposes that EMG signal based control research are a ongoing topic in rehabilitation and prosthetics. Generally, EMG is an experiment-based method of evaluating and recording electrical signals from muscles. Specifically, EMG signals eminate from the exiciability of muscle fibers through neural control, causing action potentials that cause depolarization and repolarization of the muscle membrane. These polaziration changes can be detected using EMG sensors, either through non-invasive or invasive techniques. Invasive EMG signal recording requires the use of a penetrating needle electrode to be placed in the muscle tissue, this method reduces the signal-to-noise ratio (SNR) but can be a cause of discomfort and infection. As an alternative, non invasive EMG sensors, placed at the surface of the skin are called sEMG sensors, they provide much less discomfort and propose no risk of infection to the amputee, at the cost of having an increased SNR. [1] explains that the muscle fiber membrane has a resting potential of -90 to $-90mV$ when resting. The paper also explains that the amplitude of sEMG signals have a voltage range from 0 to $10mV$, and a frequency range from 10 to $500Hz$.

5.2 Noise in EMG signals

The paper [1] poposes different noise types that contaminates EMG signals, these noises are defined as electrical signals that are not part of the decired EMG signal.

The different noise types found in EMG signals are

- Inherent Noise in Electronics Equipment,
- Ambient Noise,
- Motion Artifacts,
- Inherent Signal Instability,
- Electrocardiographic (ECG) Artifacts,
- & Cross Talking.

Noise in Electronics Equipment exists in all electronic devices, this noise has been proved to be reduced by using electrodes made of silver. **Motion Artifacts** affects EMG signals when the skin and electrodes move in relation to the movement of the underlying muscle. This can cause artifacts due to inconsistent displacement. **Inherent Signal Instability**, The amplitude of EMG signals are quasi-random. Frequency components less than 20 Hz are unstable and affected by firing rate of the motor units. This range is considered unwanted noise. Muscles change based on their active motor units, therefore the EMG signal changes too. **ECG Artifacts** is the electrical activity of the human heart is a huge interference component of EMG signals recorded from the Shoulder Girdle (Shoulder muscle groups). It is very hard to remove ECG artifacts from EMG signals, due to their relative charactaristics in the frequency spectrum! **Cross Talk** is undecired EMG signals from muscle groups not commonly monitored. It is a form of EMG leak from muscles not actively being recorded from.

5.3 sEMG Sensors for Prosthetics

The usage of sEMG sensors propose a lot of obstacles because of noise, but that does not stop sEMG sensors from being part of the state-of-the-art research in prosthetics. In the paper [2] proposes that the usage of sEMG sensors are of great importance in upper-limb classification for prosthetics devices. The paper uses sEMG sensors to classify reaching-to-grasping tasks using a Convolutional Neural Network (CNN) after pre-processing the signal with Principal Component Analysis (PCA) to reduce noise. The processing combination method of PCA-CNN proved to show higher accuracy than Machine Learning methods, such as Support Vector Machine (SVM) with an accuracy of $70.1 \pm 9.8\%$ based on 9 subjects. The paper proposes that the sEMG sensors are placed on the upper-body in combination with the upper-arm for grasping intention classification, specifically, the muscles *Pectoralis*, *Trapezius*, *Latissimus Dorsi* & the *Biceps/Triceps*, see section **??** for placements. In [2], the *Southhampton Hand Assessment Procedures (SHAP)* [3] was used to create a dataset. SHAP is designed for the assessment of musculoskeletal and neurological conditions, and can be used to test the effectiveness of prosthetics.

Another paper that proposes the usage of sEMG sensors for prosthetics is [4]. The paper proposes the usage of different sEMG devices, two of those are the wearable product “Myo Armband”, [5] a discontinued sEMG product consisting of 8 sensors

that can be placed below the elow joint, and the "Delsys Trigno" [6], a set of individual sEMG sensors that can be worn and record most muscle groups. [4] proposes the pre-processing of the sEMG data using a Notch filter of 50Hz. Furthermore, the target angles obtained as ground truth targets were reduced in dimensions through PCA, thus having the 6 dominant PC's be the targets. Then, using an "Inverse PCA algorithm", they compute the final control output for the prosthetic. In order to process the sEMG data, [4] proposes the use of a time window of $200ms$, with feature extraction for root mean square (RMS) & zero crossing (ZC). The extracted features were used as input to a nonlinear autoregressive exogeneous (NARX) network, that consists of a fully-connected multilayer-perceptron (MLP) network combined with a recurrent neural network (RNN).

5.4 Adaptive Grasping Methods

Most state-of-the-art methodologies consist of using sEMG data to predict grasp type classification or joint angle regression. The paper [7] proposes that this method becomes a burden for the HMI user, as the severity of the amputation increases and the loss of muscle recording arears become greater. [7] takes inspiration from evolutionary robotics, and propses the use of evolutionary computation to predict stable grasping methods based on touch sensor input. This is done by having a mapping between the touch sensor input of the fingers and the motorcontrol of the joints. [7] uses a simulation to train a RNN network, this RNN takes sEMG sensor data, Touch sensor data, distance to the object & object height into account. It is possible to compute distance to / height of object because grasping and training is done entirely in simulation, using a simulated target object, but that the method used would be realiseable for prosthetics. The paper concludes that alongside sEMG sensors, touch sensors could be used to appropiate joint motion could be predicted using contact states between hand and object.

The paper [8] proposes a passive solution to adaptiveness when grasping. Their method uses an underactuated, compliant linkage mechanism, where the joint rotation of the finger joints can be driven by a single motor. This allows the fingers to not rely on touch sensors, as the method in [7] does. [8] proposes the use of a sliding window with a size of $250ms$. The window is then processed using feature extraction of integral myoelectric value (iEMG), RMS, mean absolute value (MAV) & ZC with a threshhold to eliminate low signal fluctuation from noise. The extracted features are used for linear discriminanr analysis (LDA) to classify grasping intent. The paper proposes that LDA showed the highest accuracy out of different Machine Learning methods.

5.5 Grasping Intention from the Upper-arm

Another way of reducing the usage of lower-arm muscles when designing hand prosthetics would be to predict the intent of the user's actions based on upper-arm grasp prediction. The focuso of the paper [9] on recording and classification of upper-arm. The paper proposes a learning approach that decodes grasping intention during the reaching motion for upper-limb prosthetics. For pre-processing, a 30-$350Hz$ band-

pass filter is used on 12 muscles, 7 located in the upper-arm and 5 located in the lower-arm. These muscles are passed through a buttersworth filter with cut-off at $20Hz$. Furthermore, the elbow joint angle was measured using a goniometer. [9] proposes that a sliding window of $150ms$ should be used, and they use no dimensionality reduction method such as PCA. The paper tests 2 different machine-learning methods, LDA and Support Vector Machine (SVM), that utilize feature extraction of the average activation, waveform length and the number of slope changes for each window. Additionally, the paper tests an echo state network (ESN) using the given window of data with no feature extraction.

5.6 Alternatives to sEMG-based prosthetics

Research of prosthetics control interfaces expand into a multitude of areas. The paper [10] proposes the use of a brain-computer-interface (BCI) as an alternative to HMI. BCI is a type of technology that uses brain activity and the brain's neural information to control machine interfaces such as computers, assistive technology & prosthetics. BCI's have great benefit in areas where access to muscolatory information such as sEMG is impossible due to loss of muscles in the target recording area, or due to paralyzation where it becomes impossible for the patient to activate the targeted muscle groups. One dominant method of acheiving a BCI interface is electroencephalography (EEG). The purpose of [10] is to detect individual finger control using EEG sensors. EEG functions similarly to EMG but with the focus on recording brain activity instead of muscle activity. EEG is a non-invasive, portable and low-cost sensoring type, that provides a high temporal resolusion in comparison to other methods that detects brain activity, according to the paper [11]. [11] explains the great need for assisted rehabilitation devices and prosthetics, and that the need will increase in the future. Brain Computer Interfaces using EEG sensors needs to be further researched to increase overall performance of the system. [11] proposes that that the most used method of controlling a prosthesis or a rehabilitation device using EEG is to pre-process/filter the recorded data before segmenting it using a sliding window. Using the windows, feature extraction in both time & frequency domains are used as input to a feature reduction algorithm. These features are then subject to a classification network in order to transform the EEG data into motorcontrol for the BCI. It can be noted that the EEG sensor contains artifacts from other parts of the brain, such as eye movement, cardiac activity or contraction of the scalp muscles. Overall performance of using EEG for prosthetics control is low compared to more conventional methods such as EMG [11] . Furthermore, the setup for EEG recording is more complicated than EMG.

5.7 Summary of Literature

The state-of-the-art literature spans different methods of creating human machine interfaces. The main areas of creating interfaces are Muscle-/neuron-based recording and brain-based recording. Due to the large amount of cross-talk noise from brain-based sensoring, it is apparent that to increase overall controlability and robustness of a prosthetics device, the prosthetic is required to use EMG based sensoring for its

contro. By reviewing the state-of-the-art literature in sEMG-based hand prosthetics, it is apparent that if a Machine-Learning & Neural Network (except RNN), are to be used, we need to use a sliding window technique with a size of $150ms$ to $250ms$. It can be noticed that all methods use a pre-processing filter step on the raw EMG data, but that the choise of filter varies greatly. Amongst the most used filters are Buttersworth & lowpass filters with diffent frequency responces. 3 different methods of classification/regression of sEMG data are used: **Machine Learning**, **Window-Based Neural Networks**, **Recurrent Neural Networks**. algorithms such as LDA, SVM & ... As an alternative to ML, most literature proposes the use of **Neural Networks**, on the sliding window. Alternatively, methods not using a sliding window are recurrent networks, such as the echo state network or the NARX network. These Recurrent methods would also be applicable due to the continorous nature of the data.

6 Methodology

6.1 Anatomy

The hand is an anatomically-complex appendage designed to facilitate a large amount of control in different usage scenarios. The hand consists of 27 bones, 14 of these are called phalnages, and make up the 4 fingers and the thumb. These bones, alongside a complex set of $\sim$ 30 muscles are able to perform 24 Degree-of-Freedom (DoF) of rotational motion. The individual finger consists of 3 bones called phalange, arranged linearly from the palm of the hand. The 3 finger bones are called the proximal phalange, middle phalange and distal phalange. The joints between the phalanges have 1 DoF and are able to do flexion/extension movement, while the base of the finger have 2 DoF and are able to do flexion/extension & abduction/adduction movement. The anatomical muscle and joint structure of the hand/forearm can be seen in the figure 2.

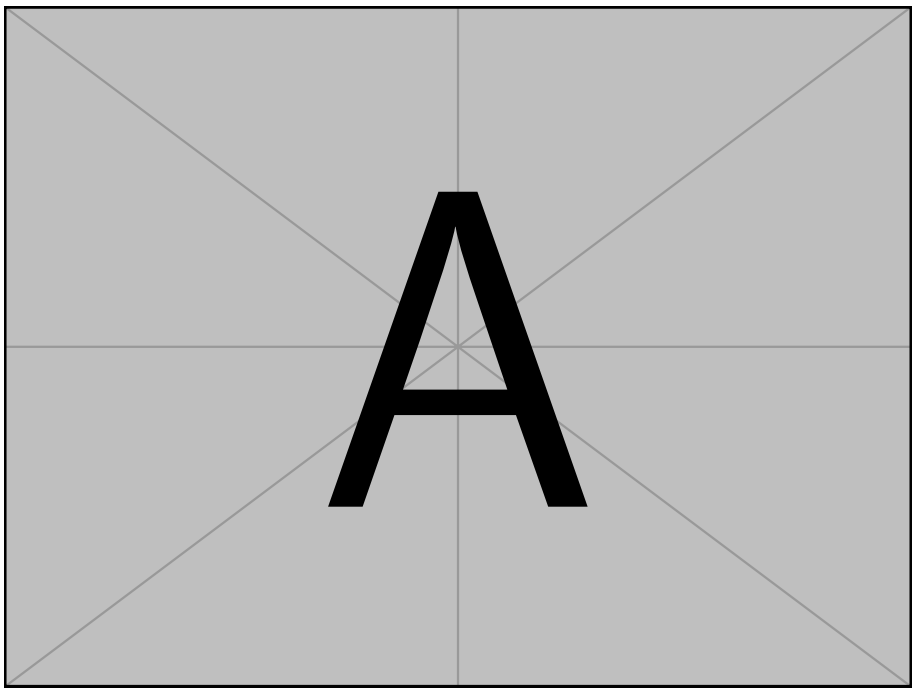


Figure 2: A rendering of the anatomical muscle and joint strucutre of the lower-arm. The complexity of the muscles controlling the hand can be seen. Source [**???**]

6.2 Brief of used Software

In order to design a simulated prosthetic device that facilitates the requirements stated in section 6.4, an effective simulation software needs to be chosen. The software needs to be controllable from an external source, such as ROS, and because of this, software like Gazebo [12] or CoppeliaSim [13] is ideal. Both simulation softwares allow for the creation of advanced dynamic-body simulations. CoppeliaSim [13] was chosen due to its intuitive development workflow.

In order to record the joint movements of the hand, while simultaneously record

the sEMG activity of the body, products that support hardware-synchonization needs to be used. Products designed to capture sEMG data are widely spread. As mentioned in the paper [4], the Myo armband [5] or the Delsys Trigno package [14] could be used. The Delsys Trigno [14] facilitates a hardware-synchonization recording mode out of the box, while the myo armband [5] does not. Because of availability, and synchronization capabilities it is ideal to use the Delsys Trigno as the sEMG recording device.

Lastly, a product designed to capture the joint movements of the hand needs to be chosen. A simple solution to hand tracking would be to use a software like MediaPipe [15], a python based landmark tracker that can track the joint angles of the hand using Machine Learning from a video feed. An alternative is mentioned in the paper [4], where a glove containing flex sensors could be used. This approach would provide relieable joint angle data without relying on a camera setup. Lastly, a capture glove could be fitted with 3D markers, detectable from a high-accuracy and high-framerate system such as the Motive motion capture setup [16] created by OptiTrack. It was chosen to use the Motive setup due to availablility and the additional benefit of having a hardware-synchnonization feature, that can be combined with the Delsys Trigno [14] setup.

6.3 Dataset Creation

In order to train a simulated hand prosthetic, a sofisticated dataset containing the measured relation between muscle activity and the finger placements is needed. In order to create a dataset of the hand, it is important to choose a set of day-to-day movements the dataset should contain. The paper [2] proposes the "Southampton Hand Assessment Procedure" (SHAP) [3] as its main method of dataset creation, an alternative would be the "Sollerman Hand Function Test" [17]. Both procedures are used in assessing the function and moveability of the human hand. By analyzing the most important gripping motions in both tests, it should be possible to denote a suitable set of grips that the dataset needs to contain. The sollerman grip types are ranked based on their usage percentage in activities of daily living. From this, the most used grip types according to sollerman are the *Pulp pinch*, *Lateral pinch*, *Five-Finger pinch & Diagonal Volar grip (Power grip)*, see [17] for further details. SHAP also proposes a set of grip types that are used in day-to-day tasks, these are *Spherical grip*, *Tripod pinch*, *Power grip & Lateral pinch*. In order to create a dataset mimics the movements of day-to-day tasks, the most important grips from [17] & [3] has been chosen. The grip types that needs to be part of the dataset and their usage descriptions can be seen in table 1. A set of consise grip types has been chosen as an alternative to a larger set of general grips. This is done as a basis of creating a specialized dataset that would be easier to train and work with as a proof of concept.

6.3.1 Existing datasets

In addition to creating a dataset for this thesis, it would be intresting to compare with an existing state-of-the-art dataset. The paper [18] proposes the use of their

Grip Type	Finger Usage Description
Pulp pinch	Between thumb, index and middle finger
Lateral pinch	Between thumb & side of index finger
Five-Finger pinch	Between thumb, and all four fingers
Power grip	Between thumb, and all four fingers with contact to palm

Table 1: The most used hand grips in day-to-day tasks based on *Southampton Hand Assessment Procedure* [3] & *Sollerman Hand Function Test* [17].

dataset [19]. The dataset contains a very large set of recordings, consisting of precise anatomical angles of the hand, with the assosicated muscle activity of the forearm. The dataset consists of 572 recordings from 22 subjects, in both reaching, gripping & releasing actions. Another dataset is explained in the paper [20], as an alternative of using finger joint angles as the ground truth data needed to train a network on sEMG data, the dataset uses a set of 16 gesture classes for a classification algorithm. The paper explains that the sEMG data in the dataset has been subject to a pre-processing step using a 10 to $500Hz$ bandpass buttersworth filter, and in order to remove powerline noise a $60Hz$ notch filter was used. The dataset was created on 43 participants.

6.3.2 sEMG Sensor Locations on the Body

To create a dataset that correlated muscle activity to the movements of the hand, it is important to choose a set of active muscles to record. $\sim$ 30 muscles are used in total to control the hand, but it would not be possible to record all of them simultaneously. 6 sEMG sensors were available in the Delsys Trigno [6] system, and these needed to be placed optimally in order to increase recording of the most important muscles. Target areas on the lower-arm for sEMG sensors are proposed by the paper [18]. Additionally, target areas of the upper-arm & upper-body were chosen based on the suggestions from the paper [9]. The chosen muscles to record with sEMG sensors can be seen in table 2.

Muscle	Main Functionality
(A) *Flexor Digitorum Superficialis*	Flex of the 4 fingers
(B) *Extensor Digitorum*	Extension of the fingers
(C) *Extensor Carpi Radialis Longus*	Wrist exstension & hand abduction
(D) *Triceps Brachii (long head)*	Extension of the arm
(E) *Biceps Brachii*	Flexion of the arm
(F) *Pectoralis Major / Frontal Deltoid*	Movement of the Arm

Table 2: The muscles targeted with the 6 available sEMG sensors, as recommended by the papers [18] & [9]. The targeted muscles contributes to a lot of the movements of hand & lower-arm, and are ideal for the dataset.

The target muscles were chosen as they contribute to a lot of the overall movements of the hand & lower-arm. The muscles in table 2 are numbered, and the locations of the individual sensor with the corresponding muscle can be seen in figure 3.

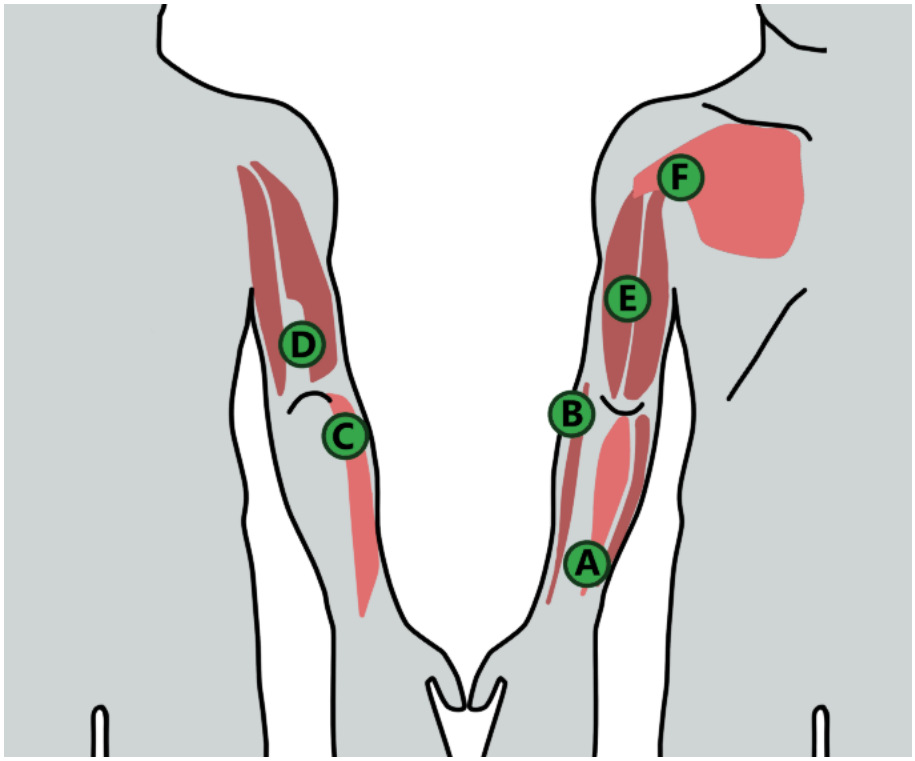


Figure 3: Muscle diagram, containing the target sEMG locations. The numbering corresponds to the muscle names in table 2. The left image shows the muscles on the rear of the arm, while the right shows the muscles on the front of the arm.

It was chosen to split the amount of sensors evenly along the lower-arm and the rest of the body in order to asses how important the different muscles are in prediction of the hand.

6.3.3 Motion Capture Glove

The motive capture system created by OptiTrack is a set of 8 high-quality cameras mounted to cover a target capture area. The capture system detects flouresent 3D markers in the given capture area, with the purpose of triangulating the markers and effectively calculate 3D poses for the markers in the scene down to an effective accuracy of $\pm 0.2mm$ [21]. In order to get precise recordings of the motion of the hand and fingers, flourecent 3D markers were placed on a glove. The pattern of the marker positions were closen in order to calculate the angles of the individual finger bones. The markers are used in sets of 3, this allows for the calculation of the triangle angles in 3D space. The positions of the 3D markers on the recorder glove can be seen in figure 4. An example of flexion with the capture glove can be seen in figure 5.

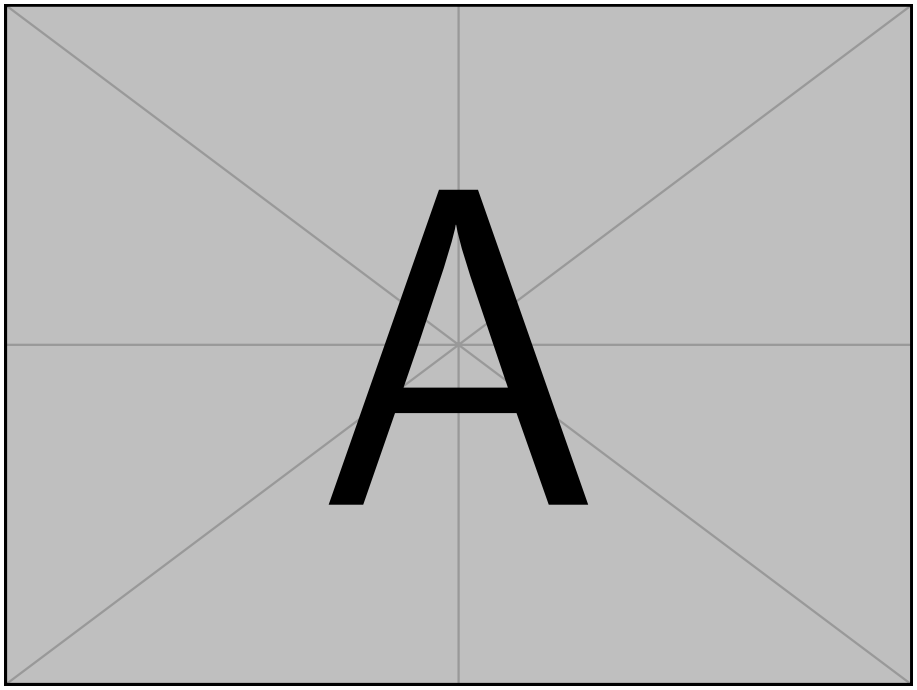


Figure 4: The positions of 3D markers on the capture glove, designed to be detectable using the Motive Capture software [21].

6.3.4 Problems with Motive Tacking Software & Capture Glove

During testing and the creation of the dataset, several problems with the capture glove design and the capture setup became apparent. These problem severely reduced the amount of time available to create and process a large dataset for this thesis. The Motive Motion Capture software [21] had a lot of difficulty with finding, recording and optimizing the 3D markers placed on the capture glove.

The Motive sofware used to generate the 3D poses is designed to detect 3D markers in the scene. Once the markers are found in the individual cameras, they are subject to an optimization algorithm that effectively tries to determine if the markers seen in 1 camera is the same as the markers in the other cameras. The first problem that occoured was specular reflection in the scene caused by shiny surfaces redirecting light in the target area. This creates *phantom* markers that can be seen in only a small set of frames. Another problem with the capture setup is the optimization algorithm. During recording of the dataset, it became apparent that it would not be possible to recort multiple fingers at the same time. 3D markers very close to each other seemed to be connecting, and only become a single marker in the tracking software. This problem specifically occoured when the markers at the finger tips of the glove came in contact with each other causing marker data to be lost. Furthermore, the Motive Capture software uses an auto-labeling system that is able to determine if the markers seen in one frame, are the same markers in the next frame. This system proved to only be semi-relieable, giving labeled marker sequences of less than ~ 200. During a sequence of 15 seconds of recording, a set of 7 markers in the scene became over 100 labeled marker tracks. And the

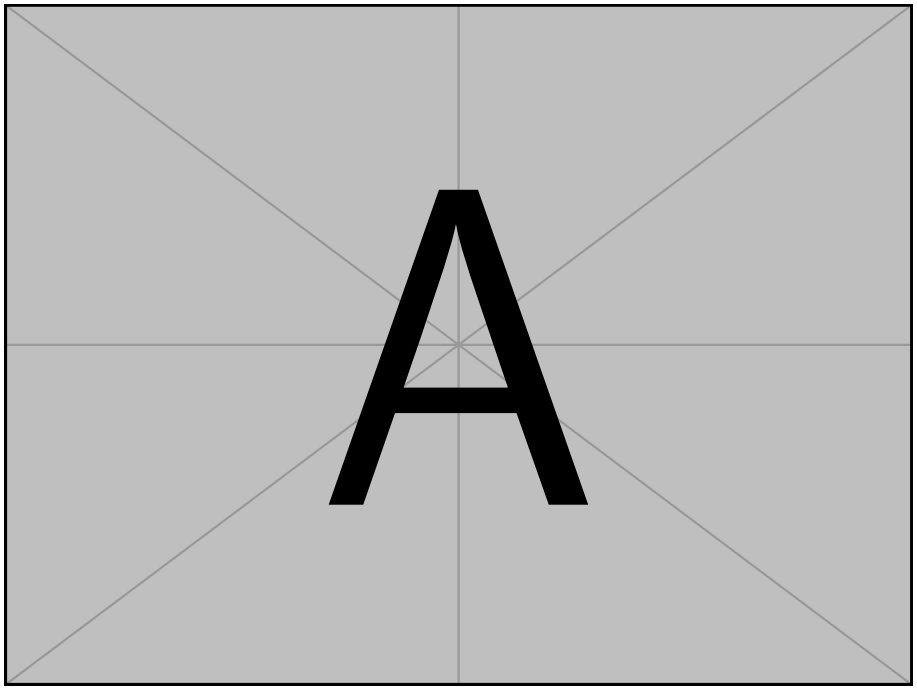


Figure 5: Flexion of the index finger and its transformation of the 3D markers on the capture glove.

problem increased as 30 seconds of fotage could create up to 1000 marker labels. The unreliable labeling of markers can be seen in appendix F.

6.3.5 Reduction of Tracking Software Problems

The large amount of problems observed using the Motive Motion Tracking software makes it impossible to use the tracking output as training data. The problems requires the user to manually combine labeled marker sections into larger labeled sections, in order to reduce changing labels on the tracked markers. It would be ideal if this functionality existed in the Motive software, but this was not the case. Because of a lack of functionality in Motive, software was created to replay the 3D markers in real-time, with their associated labels. Furthermore, software that was able to remove, merge and subsample markers was used to clean the labels. The software created to fix unreliable labeling of markers can be seen in appendix F.

6.4 Design of a Simulated Prosthetic Hand

In order to design a state-of-the-art simulated prosthetic hand, a number of anatomical design choises needs to be considered. This thesis tries to create the most anatomically-correct hand simulation available, this will hopefully have a number of positive effects on prosthetics research. By having access to an advanced simulation, it would in turn be able to test and visualize more advanced movement controllers that can facilitate more DoF than current commercial prosthetics. By creating an anatomically correct prosthetic hand simulation, it is hoped that prosthetics users

can have more advanced rehabilitation, and learn to have more natual control of their prosthetics. This would create a more natual usage experience, and decrease the percentage of users that reject the usage of their prosthetic alltogether. A set of requirements The simulated anatomically correct hand should be determined in order to create a state-of-the-art prosthetics simulation. The simulated prosthetic should:

1. Facilitate the same DoF as an anatomically correct hand.
2. Have porpotions that closely resemble that of an anatomically correct hand.
3. Be simulated and be controllable in a commonly used robotics software to increase accesability for researchers.

The resulting prosthetic hand simulation can be seen in appendix A.

6.4.1 Simulated Hand Articulation design

In order to translate the biology and anatomy of a real hand explained in 6.1 into an robotics simulation, we start by denoting the relative lengths of the wrist bones and phalnages by refrence, as can be seen in figure 6.

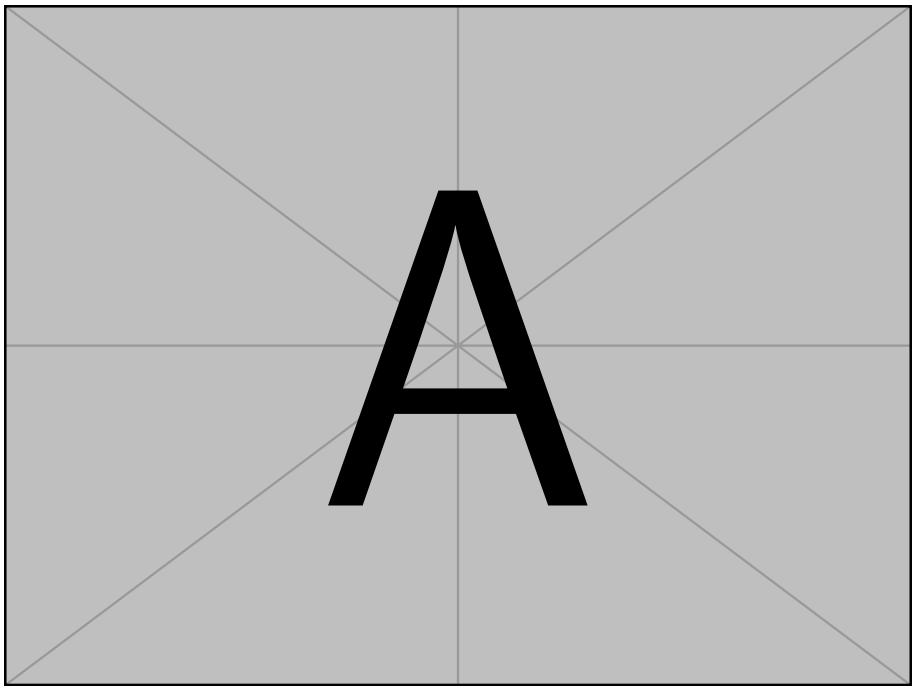


Figure 6: Example figure text

The porpotions of the refrence is used to denote the bone lengths for the model. The model is implemented in Coppeliasim **??**, The model is created in a hierachy, the bones are created with cylinders and the joints are created using 1 DoF Revolute joints. As specified in section 6.1, some joints of the human hand facilitates 2 DoF of rotation. This is needed in order for the wrist and finger base joints to be able to do abduction/adduction. To simulate this, two 1 DoF revolute joints were placed in series, thus allowing 2 DoF.

Based on the literature review in Section , it becomes apparent that there exists a standard pipeline for translating sEMG recordings into predicted motorcontrol for a prosthetic.

6.5 sEMG Data Processing

Raw sEMG data contains a lot of unwanted noise as explained in section 5.2. Some of the noise can be can be removed through signal processing using filters. State-of-the-Art papers use a lot of different methods to reduce noise in sEMG data. The paper [22] proposes the use of a $50Hz$ Notch filter, while papers [23] & [20] respectively chooses a $20Hz$ cutoff Buttersworth filter & a $10 - 500Hz$ band-pass Buttersworth filter, lastly the paper proposes a $60Hz$ notch filter in order to remove powerline noise. Due to me many different filtering types used in state-of-the-art, it was determined that the use of a filter depends on the specific sEMG data, and that the best filter for this thesis is to be determined through testing.

7 Datasets

7.1 Training Dataset

Due to the problems with the Motive motion capture software explained in section 6.3.4, it was determined that the main dataset for training the chosen methods would be based on dataset [19] from the paper [18]. The dataset contains sections of sEMG data and kinematic movements of the hand. These sections are labeled as either *reaching*, *manipulation* or *releasing*. Due to the dataset containing data for regression and classification, it is possible to explore multiple methods of HMI's. The Pre-processing step is not covered for the training dataset [19], as the dataset has been pre-processed using a 4th-order bandpass filter with a band of 25 to $500Hz$, then the data was subject to a 4th-order low-pass filter at $8Hz$. The main objective of these tests are to review and compare possible methods of regression & Classification of sEMG data in order to determine the movements of the hand.

7.2 Created Dataset

As explained in section **??**, creation of a custom dataset to train and test models on proved to be difficult. The dataset created using the Motive Capture Software in section **??** during the development of this thesis is not very large, and has no labeling for clasification. Because of this, the dataset is not suitable for training in any of the methods.

8 Machine-Learning Intent Classification

Due to the dataset containing labels associated with movements of the hand, it is possible to use the dataset to train a intent prediction classifier. As mentioned in the papers [**??**] & [**??**], it is ideal to feature extract windows of the raw sEMG data. The papers propose that the features *Zero Crossing* & *Root Mean Square* are ideal for intent classifiers. Furthermore, the paper [**??**] proposes that the Machine-Learning classifier LDA has the highest classification rate out of similar methods. Another paper [**??**] proposes the use of the Support-Vector-Machine ML method for intent classification. Both types of classification will be explored and compared.

8.1 Implementation

In order to classify intent from sEMG muscle data, the data needs to be converted to feature space before it can be used as input to a machine learning algorithm. For this, it was chosen based on state-of-the-art literature, that the feature extraction methods would be zero-crossing & RMS. The input channel for a single sEMG window was converted into a vector of zeros and ones for the zero-crossing features, where a one is set if the data crosses over the zero line. Furthermore, the RMS for a window was calculated, and inserted to the end of the input vector. This effectively converts the 6 sEMG channels of a window size of N into a feature vector of size $N + 1$. The feature-based inputs were extracted, and their movement labels were used as the target classification. The dataset of features and ground truths were then used as input for discriminant analysis, and SVM classifiers.

The purpose of **Discriminant Analysis** is to classify inputs into groups. Discriminant analysis is a classical multivariate method of Supervised Learning, that takes a input set with their ground-thuth classes and tries to maximize seperation between these different classes. The models assumes that data follows a gaussian distribution. For this ML method, two types of discriminant analysis were tested on the dataset, namely Linear Discriminant Analysis (LDA) & Quadratic Discriminant Analysis (QDA). The main difference between LDA and QDA is that LDA makes the assumption that the covariance matrices of the classes are the same. Equal covariance matrices will result in linear seperations between classes. QDA as an alternative does not assume this equal covariance, this results in a classifier that assumes quadratic seperations between classes. Just as Discriminant Analysis **Support Vector Machine** based methods are used to seperate inputs into groups based on classes. SVM is a robust supervised learning method that groups dataset classes. SVM tries to maximize the width between classes by converting the input to a higher dimensional feature vector, and using this vector, the SVM algorithm is able to efficiently calculate a non-linear classification. The feature space conversion is determined by the choise of SVM kernel in the algorithm. For this ML method, 3 types of SVM classifiers were tested, a SVM using a Radial Basis Function (RBF) kernel, a SVM using a Sigmoid Kernel and a NuSVM method with the RBF kernel and a parameter for the number of support vectors used.

8.2 Tests & Results

The methods are tested on 50 sets of data that was not part of the training set. The discriminant analysis methods can be seen in figure 7, with the average accuracy of the methods respectively being 0.61% & 0.55% over the 50 tests.

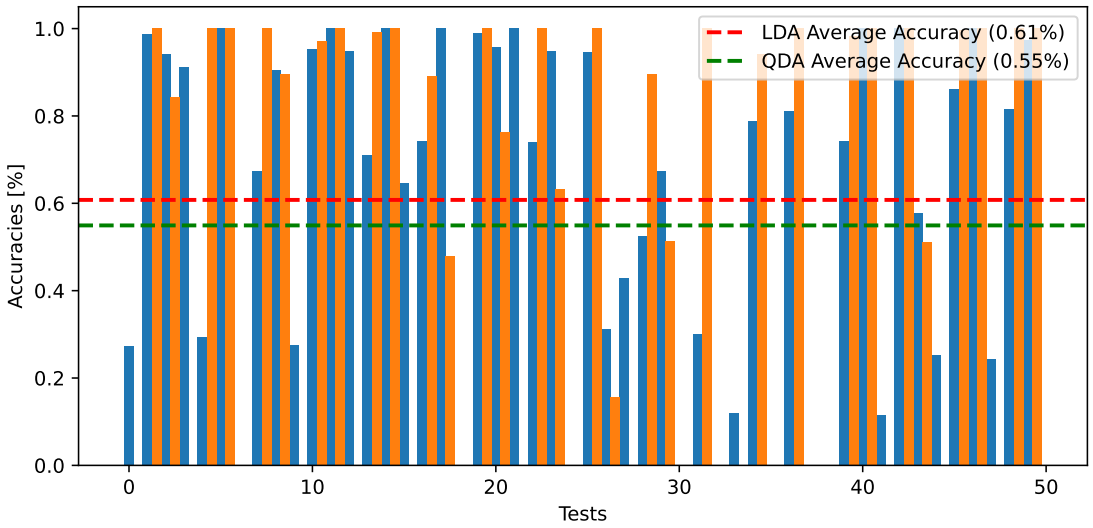


Figure 7: Comparison of Discriminant Analysis methods (LDA/QDA) over 50 tests.

The Support Vector Machine methods can be seen in figure 8 , with the average accuracy of the methods respectively being 0.54%, 0.56% & 0.59% over the 50 tests.

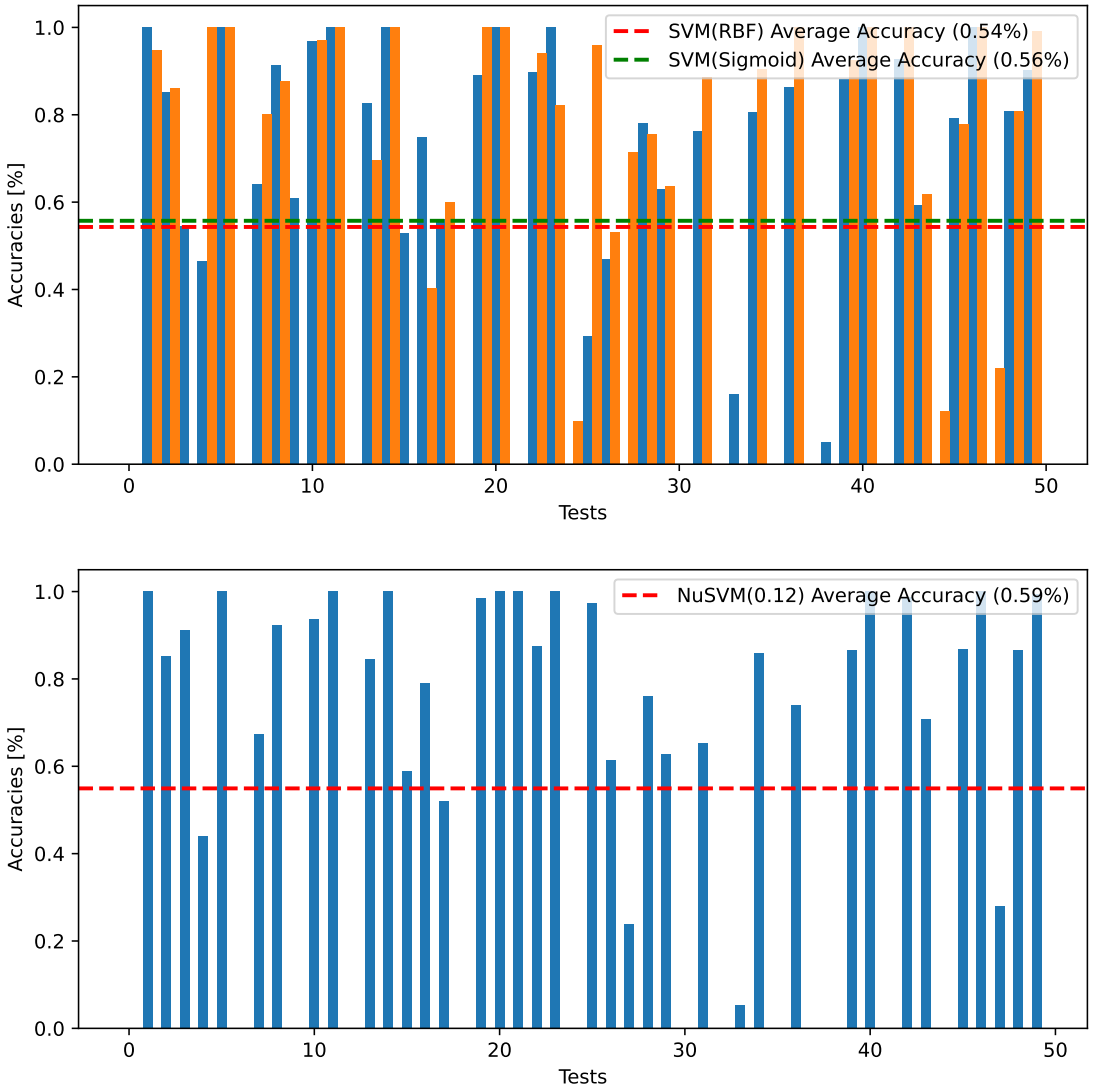


Figure 8: Comparison of Support Vector Machine Kernels & parameters over 50 tests.

8.3 Evaluation

The methods tested can be seen to have similar average accuracy of $\sim 0.6\%$ over 50 tests. Because of this, it is difficult to choose a superior method without further analysis. The choice of machine learning method depends on its overall classification rate over 50 tests, but it can also be important to rate the ML methods based on their Interquantile values. A full comparison of the machine learning methods can be seen in figure 9.

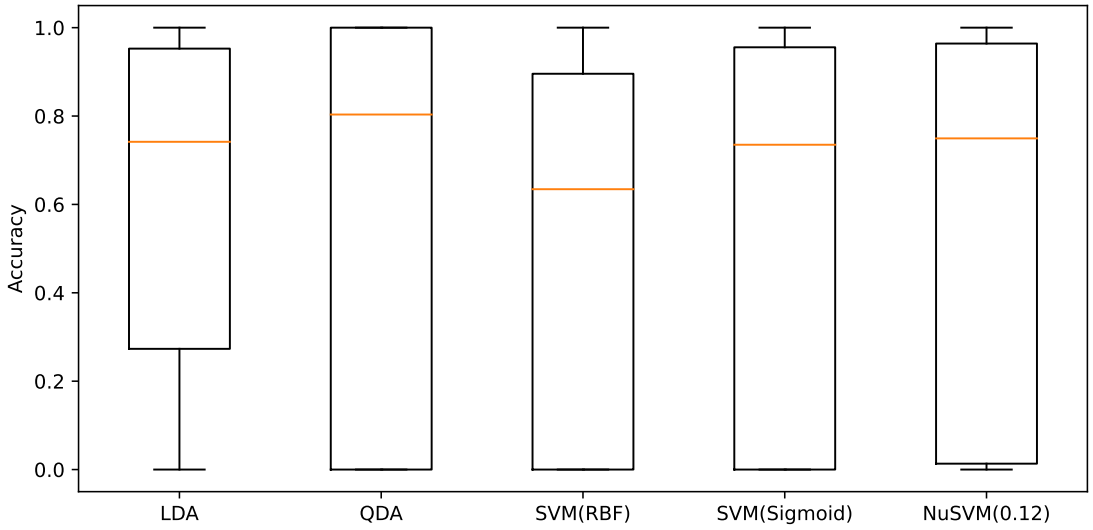


Figure 9: Boxplot comparison of chosen classification methods based on their 50 test accuracies.

It can be seen in the boxplot that most methods have a similar interquantile median, and fairly small wiskers. In order to get a clearer comparison of methods, the relevant values have been compared in table 3.

ML Method	avg-accuracy [%]	Interquantile Median	Interquantile Range
LDA	**0.61%**	0.741705	**0.679410**
QDA	0.55%	**0.803432**	1.000000
SVM (RBF)	0.54%	0.634426	0.895688
SVM (Sigmoid)	0.56%	0.735050	0.955580
NuSVM (0.12)	0.59%	0.749436	0.950715

Table 3: Comparison of relevant classification parameters for the chosen methods. The best performing parameters have been highlighted.

The relevant parameters in the table makes it possible to identify the most suitable method for intent classification. It can be seen that LDA has the overall best accuracy of all the methods of 61%, and the overall worst performing method is SVM (RBF). By denoting the median and the range of the Interquantile section of the boxplots, we can further elaborate on the spread of accuracies. LDA has the best average accuracy, and it also has the smallest interquantile range, indicating that LDA perform well across tests with a robustness to outlier tests. It can be seen that QDA performs best according to the interquantile median, but QDA is also the method that has the largest interquantile range indicating that QDA is not very robust to outlier tests. An alternative to the discriminant analysis methods, it can be seen that the NuSVM method with a number of support vectors denoted by 0.12. NuSVM has the sceond best average accuracy, the second best interquantile median & the third lowest interquantile range. From the table 3, it becomes apparent that

none of the methods have perfect performance, but that the LDA method is the best method for intent classification. This result is similar to the paper [**??**] that has the same conclusion. The resulting performance metrics of the ML methods are dependent on the chosen feature extraction techniques used. Zero crossing and MSE features can be used for the classifiers and provide insight into what parameters are important when working with sEMG data.

9 Neural Network Movement Regression

9.1 Windowed Convolusional Neural Network

As an alternative to intent classsification, the dataset could be used to train finger movement regression networks.

9.1.1 Implementation

The sEMG data used as input for regression networks is complex. The sequences of muscle activity needs to be understood by the proposed method and produced into a an output. A network type that is ideal to understand complex meaning in, 3-dimensionary inputs is a CNN network. A CNN consists of a set of Convolutional Layers that deforms inputs to abstract convolution features. The structure is considered sparcely connected due to the convolution layers are only recieving a subset of the previous convolution features. The feature conversion is ideal when a network needs to become robust and needs to recognize larger concepts in the input. This functionality is especially used when the input data takes the shape of a matrix. Once the input is convoluted to a chosen feature abstraction, it is given to a MLP strucure trained to convert the convolution features into a useable output. CNN networks have no buildin time-based functionality, they recieve a set of inputs and produce an output. It is possible however to represent time if the input is given in segments consisting of multiple timeframes. Because of this, the sEMG muscle activity from the dataset is segmented into windows of a desired size, the output angles that the network should do regression towards are then chosen to be the first angle after the window. A CNN network can be used in the field of sEMG processing due to the convolution layer's ability to learn abstract formations of the sEMG data. The network would be able to identify patterns in sEMG data and correlate them to an appropoate output. The CNN network structure seen in figure 10.

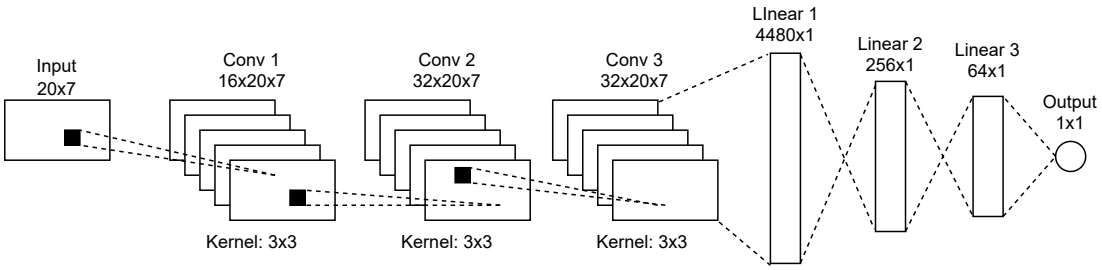


Figure 10: The chosen CNN network structure, consisting of 3 Convolution Layers and 3 Linear layers. Network input is a window of sEMG recordings, with the purpose of regression of a finger joint angle.

As visualized in the figure, the CNN network consists of 3 Convolution layers followed by 3 MLP layers. The chosen CNN network structure is chosen to be simple due to the small input window of 7 muscle sensors and a window size of 20 samples. For this reason, no pooling was used between the convolution layers. The Convolution layers extract features up to a size of 32 using a convolution kernel with a size of $[3, 3]$. The

convolution features are then flattened, to a size of 4480, before being subject to the MLP layers that convert the extracted features into a regression prediction.

The CNN network is trained to do regression of joint angles of the hand, because of this, the loss function is determined to be Mean Square Error (MSE) loss. The network is trained using an Adam optimizer with a learning rate of $1e^{-4}$. A subset of the dataset is split into a test and a validation set, and windows are extracted correlating input and output. The output is normalized, and the input/output sets are shuffeled before training takes place.

9.1.2 Tests & Results

In order to test the CNN network and assess its performance, the training MSE loss can be seen in figure 11.

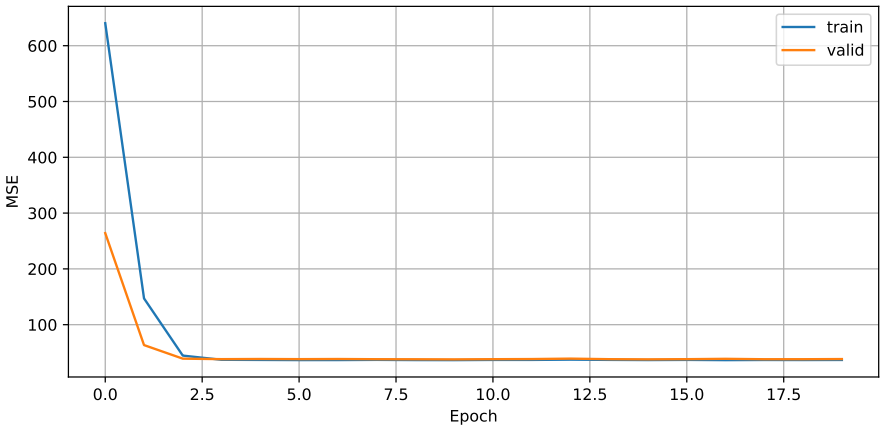


Figure 11: The Train and Validation MSE loss during training of the CNN network method.

The CNN network is trained with a batch size of 8 over 20 epocs, but as can be seen in the figure, the MSE loss settels at ~ 34 after just 3 epocs.

50 sets of regressions that have not been trained on are taken from the dataset for testing. The errors of the dataset are plotted, as can be seen in figure 12.

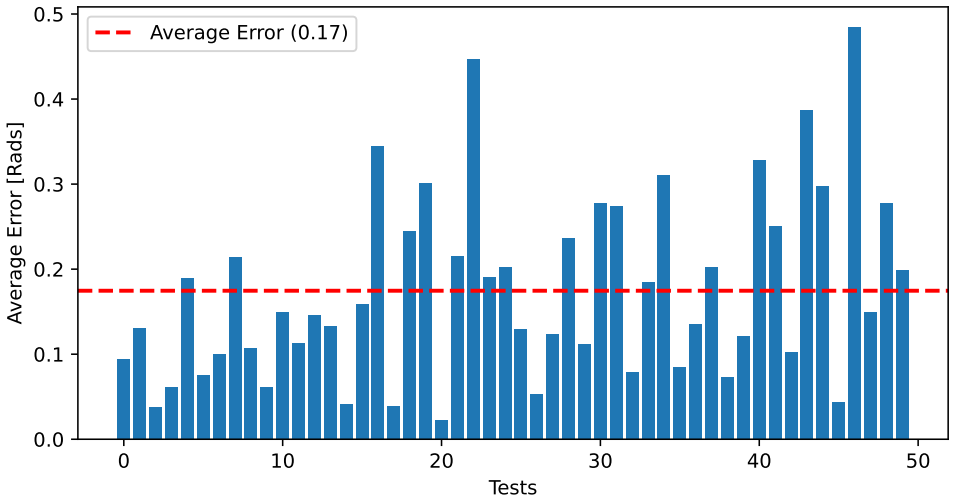


Figure 12: The errors of 50 regressions using the CNN network.

9.1.3 Evaluation

As can be seen in figure 11, the CNN network is able to be trained on the dataset with a validation set MSE loss of 0.34. The errors of the 50 test regressions in figure 12 show that a fairly large average regression error of 0.17 Radians exists in the test regressions, and that the average error is skewed upwards due to some outlier tests having large average errors up to 0.3 Radians.

9.2 Recurrent Neural Network Regression

Due to sEMG data being time-based, Recurrent network types are ideal for regression predictions. RNN networks are similar to MLP's with the main difference being that a RNN has hidden feedback memory state that is derived from earlier inputs. This functionality makes RNN networks ideal for concurrent, time-based systems where a sequencial input distribution needs to be converted to a output regression. Another use case of RNN-based regression systems are systems where responce time is critical. The RNN network does not use a window of data, it converts the input of the current time frame, and the feedback of the previous time frame into a prediction. Because of the hidden memory, a RNN network has a quicker response time than a window-based network. The RNN network structure designed to predict regression of finger movements from sEMG data can be seen in figure 13.

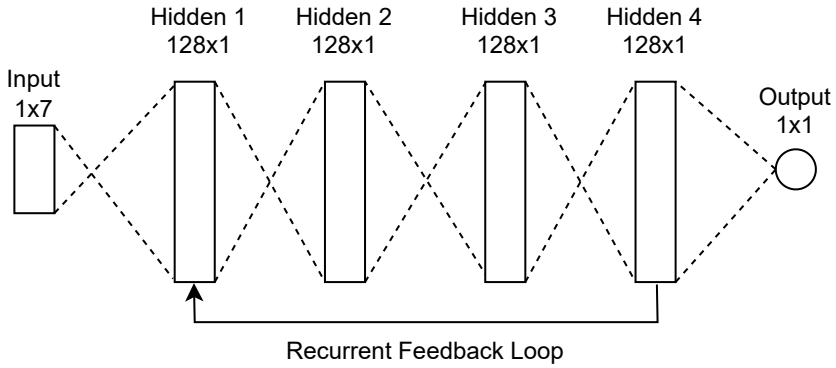


Figure 13: The chosen RNN network structure, consisting of a input layer, 4 hidden layers with a recurrent feedback loop, and a output layer.

9.2.1 Tests & Results

The RNN network is trained on the dataset [19], in a similar way as the CNN network, with the purpose of approximating finger angle regressions per time frame. The RNN network is trained with a batch size of 8, with the MSE loss function and the Adam optimizer with a learning rate of $1e^{-2}$. In order to test the network, a random set of 50 regressions that have not been trained on are taken from the dataset. The average errors of each regression test is plotted, as can be seen in figure 14.

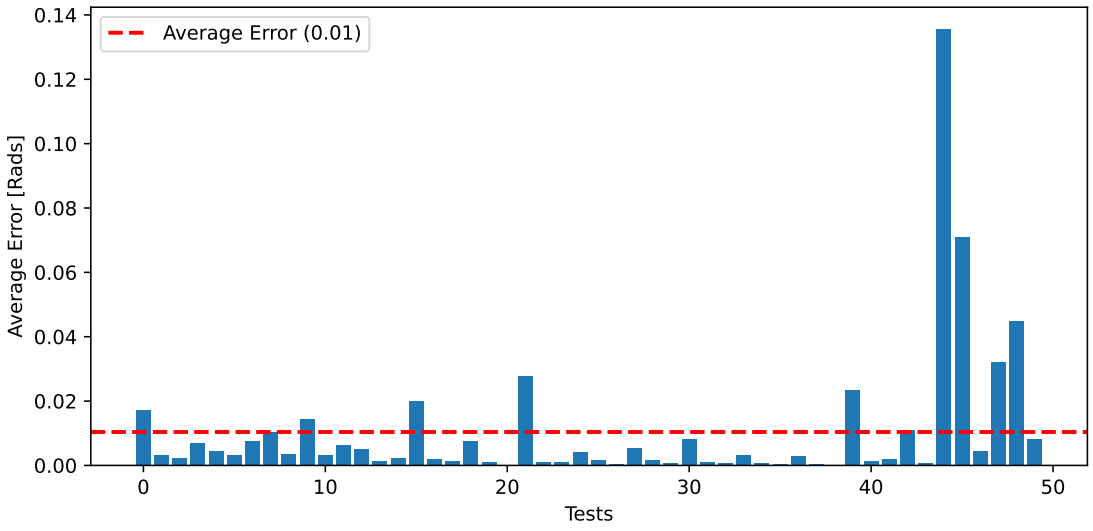


Figure 14: The errors of 50 regression tests using the RNN network.

9.3 Evaluation of Regression Methods

The errors of the 50 test regressions in figure 12 show that a fairly small error exists in the regressions, with a maximum regression error of ?? Radians.

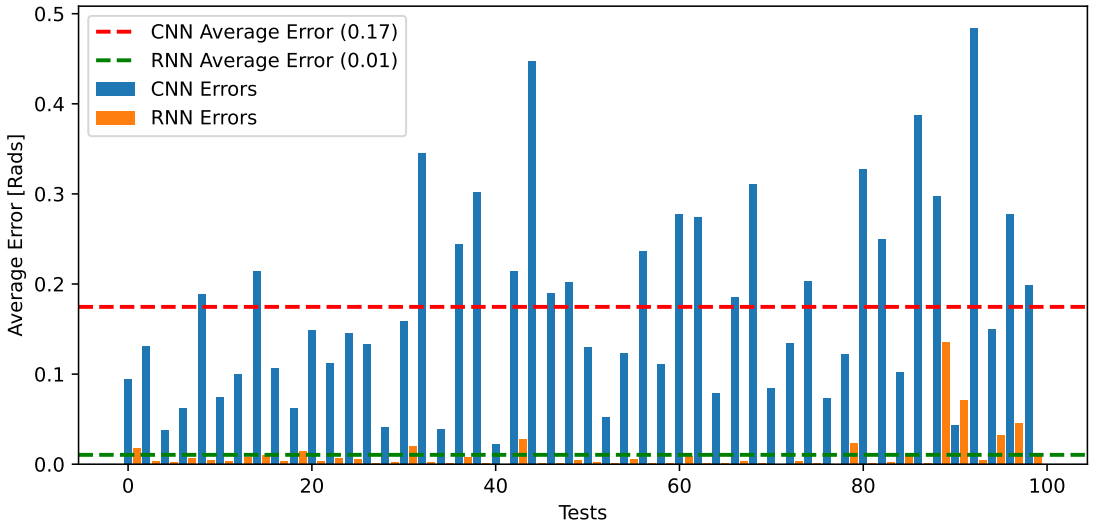


Figure 15: Comparison of Windowed CNN and RNN regression methods.

The figure clearly shows that there is a large difference in window-based and recurrent network types. The CNN shows a average test error of 0.17 Radians, while the RNN network has a average test error of 0.01 Radians. The reason for the large error in the CNN network test is unclear. One of the reasons the CNN network is unable to accurately learn regression could be because of the pre-processing done in the dataset. Alternatively, the network is unable to learn because of the window size. This would make sense since the RNN network with a really low error is able to learn to do regression, as it is not limited by a arbitrary window of input.

It becomes apparent that teaching a neural network to do regression of finger joints is a complicated task. Because of this, the tests propose that it is ideal to use a type of Recurrent Network for regression.

10 Simulated Prosthetic Hand

10.1 Useability of the Simulated Prosthetic Hand

Access to existing prosthetic hand simulations proved to be impossible to access and use, as explained in section **??**. they are either unavailable because their associated products have been discontinued or locked behind a pay-to-acces barrier where you have to pay to recieve a real prosthetic. Due to non-availability of prosthetic hand simulations, it was chosen that through this thesis, a open source prosthetic hand needed to be created for testing of the algorithms developed as part of this project. The creation of a prosthetic hand simulation can be seen in section 6.4. The prosthetic hand was designed to be as anatomically similar as a real hand as possible. This was done in order to accurately simulate the movement of a real hand, and because of it, allow users to test and validate control methods on a simulation before needing to construct a real prosthetic. The useability and moveability of the simulated prosthetic hand needs to be assessed. This is important because the useability of a prosthetic hand has a large impact on rehabilitation for the end-user.

10.2 Method

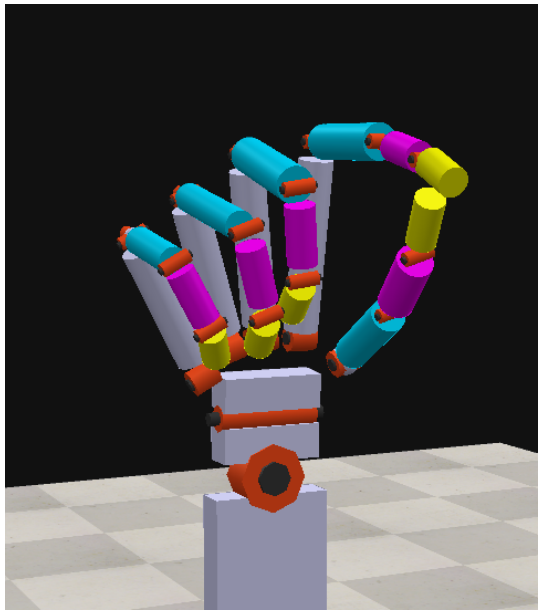
In order to test the useability and moveability of the simulated prosthetic hand, the hand was tasked with acheiving the end-poses of the most used day-to-day grips. These grips were explained in section 6.3 table 1, namely the *Pulp pinch*, *Lateral pinch*, *Five-Finger pinch* & *Diagonal Volar grip (Power grip)*. In order to pose the simulated hand, the interface created in ROS2 was used to send kinematic configurations to the hand simulation.

10.3 Test & Results

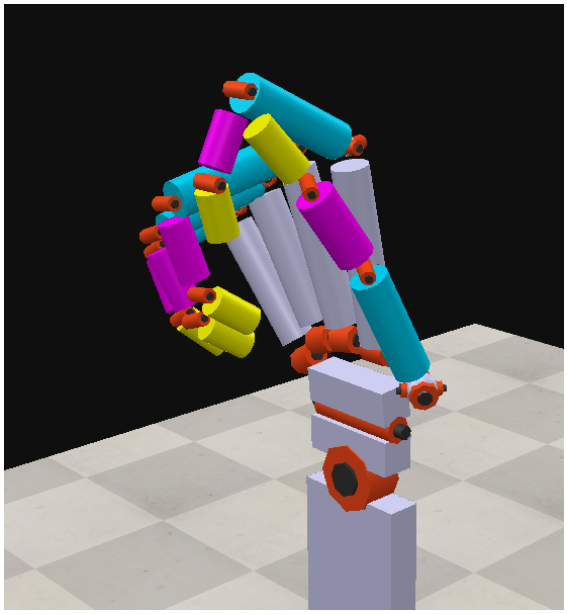
The hand was posed into all 4 day-to-day grip configurations for visual assessement. The resulting configurations of the hand can be seen in figure 16. The software used to pose the hand simulation can be seen in appendix **??**

10.4 Evaluation

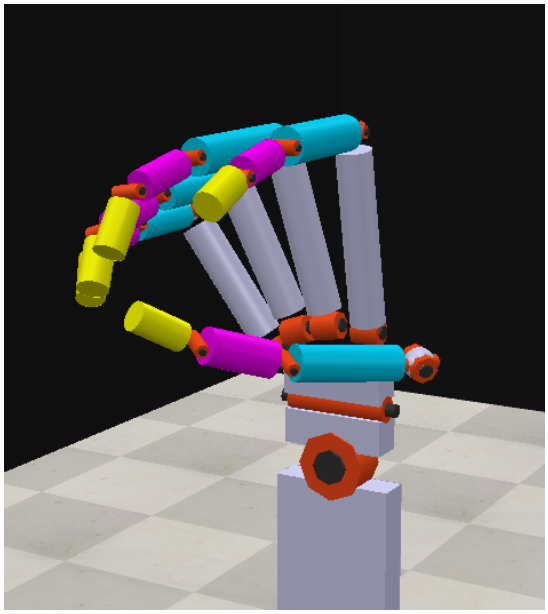
Based on a visual assessment of figure 16, it can be concluded that the prosthetic hand is able to be posed similarly to the real-life refrence. The ROS2 interface created for the simulation allows for easy manipulation of the joint rotations, and because of this, acheiving different poses becomes easy to do. The hand simulation is able to acheive a wide range of motion because of its design and its ability to mimic the anatomy of the real hand. From this it can be concluded that the prosthetic hand is porpotionally correct and thus also allows for accurate posing, applicable in different day-to-day scenarios.



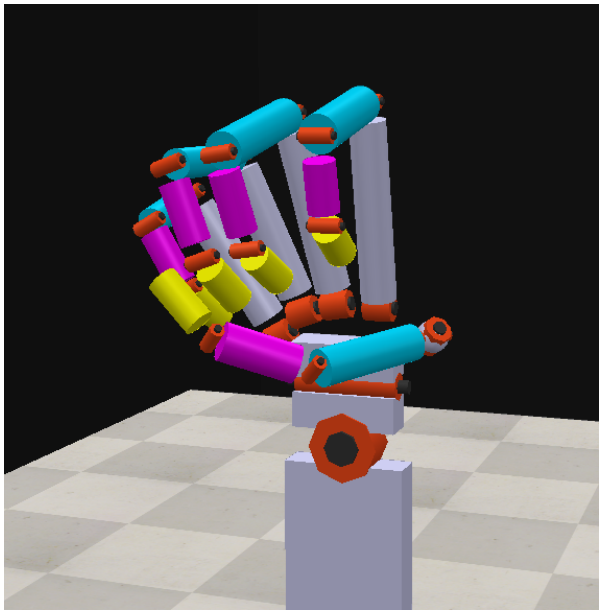
(a) Pulp pinch



(b) Lateral pinch



(c) Five-Finger pinch



(d) Diagonal Volar grip (Power grip)

Figure 16: The four most used day-to-day poses, tested in a hand prosthetic simulation.

10.5 Posing based on Network Output

As a result of testing different network types and their applicability for creating suitable motorcontrol output for a prosthetic hand, it would be ideal to test the network output on the prosthetic simulation. For full video see appendix ??.

11 Pre-Processing of sEMG data

12 Discussion

13 Conclusion

Hello, here is some text without a meaning...

13.1 Future Work

14 Bibliography

List of Figures

List of Tables

References

[1] M. Tech. "A Review paper on EMG Signal and its Classification Techniques". In: 2015.

[2] Keun-Tae Kim et al. "Upper-Limb Electromyogram Classification of Reaching-to-Grasping Tasks Based on Convolutional Neural Networks for Control of a Prosthetic Hand". In: *Frontiers in Neuroscience* 15 (2021). ISSN: 1662-453X. DOI: 10.3389/fnins.2021.733359. URL: https://www.frontiersin.org/articles/10.3389/fnins.2021.733359.

[3] *Southampton Hand Assessment Procedure (SHAP)*. http://www.shap.ecs.soton.ac.uk/.
Accessed: 2023-04-14.

[4] Zhaolong Gao et al. "A Multi-DoF Prosthetic Hand Finger Joint Controller for Wearable sEMG Sensors by Nonlinear Autoregressive Exogenous Model". In: *Sensors* 21.8 (2021). ISSN: 1424-8220. DOI: 10.3390/s21082576. URL: https://www.mdpi.com/1424-8220/21/8/2576.

[5] *Thalmic Labs Myo gesture control armband*. https://www.zdnet.com/article/thalmic-labs-shuts-down-myo-gesture-control-armband-project/.
Accessed: 2023-04-14.

[6] *Delsys Trigno sEMG Recording System*. https://delsys.com/trigno/.
Accessed: 2023-05-05.

[7] Yuki Kuroda et al. "Coevolution of Myoelectric Hand Control under the Tactile Interaction among Fingers and Objects". In: *Cyborg and Bionic Systems* 2022 (2022). DOI: 10.34133/2022/9861875. eprint: https://spj.science.org/doi/pdf/10.34133/2022/9861875. URL: https://spj.science.org/doi/abs/10.34133/2022/9861875.

[8] Yanchao Wang et al. "Design of an Effective Prosthetic Hand System for Adaptive Grasping with the Control of Myoelectric Pattern Recognition Approach". In: *Micromachines* 13.2 (2022). ISSN: 2072-666X. DOI: 10.3390/mi13020219. URL: https://www.mdpi.com/2072-666X/13/2/219.

[9] Iason Batzianoulis et al. "Decoding the grasping intention from electromyography during reaching motions". In: *Journal of NeuroEngineering and Rehabilitation* 15 (2018).

[10] Zhen Ma et al. "Decoding of Individual Finger Movement on One Hand Using Ultra high-density EEG". In: *2022 16th ICME International Conference on Complex Medical Engineering (CME)*. 2022, pp. 332–335. DOI: 10.1109/CME55444.2022.10063299.

[11] Maged S. AL-Quraishi et al. "EEG-Based Control for Upper and Lower Limb Exoskeletons and Prostheses: A Systematic Review". In: *Sensors* 18.10 (2018). ISSN: 1424-8220. DOI: 10.3390/s18103342. URL: https://www.mdpi.com/1424-8220/18/10/3342.

[12] *Gazebo - Gazeboism*. https://staging.gazebosim.org/home.
Accessed: 2023-04-28.

[13] *Coppeliasim - Coppeliarobotics*. https://www.coppeliarobotics.com/.
Accessed: 2023-03-21.

[14] *EMGworks - Delsys*. https://delsys.com/emgworks/.
Accessed: 2023-03-21.
[15] *MediaPipe landmark tracking - MediaPipe Python*. https://mediapipe.dev/.
Accessed: 2023-04-28.
[16] *Motive:Tracker - OptiTrack*. https://optitrack.com/.
Accessed: 2023-03-21.
[17] *Sollerman Hand Function Test*. https://www.physio-pedia.com/Sollerman_Hand_Function_Test.
Accessed: 2023-04-25.
[18] Nestor Jarque-Bou et al. "A calibrated database of kinematics and EMG of the forearm and hand during activities of daily living". In: *Scientific Data* 6 (Nov. 2019). DOI: 10.1038/s41597-019-0285-1.
[19] Nestor Jarque-Bou et al. *A calibrated database of kinematics and EMG of the forearm and hand during activities of daily living*. Version 1.0. Zenodo, July 2019. DOI: 10.5281/zenodo.3337890. URL: https://doi.org/10.5281/zenodo.3337890.
[20] Ashirbad Pradhan, Jiayuan He, and Ning Jiang. "Multi-day dataset of forearm and wrist electromyogram for hand gesture recognition and biometrics". In: *Scientific Data* 9 (Nov. 2022), p. 733. DOI: 10.1038/s41597-022-01836-y.
[21] *Motive Tracker Software - OptiTrack*. https://optitrack.com/software/motive/.
Accessed: 2023-03-21.
[22] Zhaolong Gao et al. "A Multi-DoF Prosthetic Hand Finger Joint Controller for Wearable sEMG Sensors by Nonlinear Autoregressive Exogenous Model". In: *Sensors* 21.8 (2021). ISSN: 1424-8220. DOI: 10.3390/s21082576. URL: https://www.mdpi.com/1424-8220/21/8/2576.
[23] Iason Batzianoulis et al. "Decoding the grasping intention from electromyography during reaching motions". In: *Journal of neuroengineering and rehabilitation* 15.1 (2018), pp. 1–13.

Appendices

See the attached Appendix zip for the appendix files. Alternatively, see `github???` for online refrence.

A Prosthetic Hand Simulation in CoppeliaSim

B Hand Simulation Poses

C ROS2 Controller for the Prosthetic Hand

D Modification & Replaying Scripts for Motive CSV Files

E Motion-Capture & sEMG Dataset

F Motive Marker labeling & Uncleaned marker set

G Pre-Processing Filter Graphs

H Network Creation/Training/Testing Code