

Filip Arthur Blaa fjell
Mathieu Steketee
Robert Asp rem
Andreas Blindheim

Multi-Articulated Myoelectric Prosthesis

Costumized for the Norwegian Market

Bachelor's thesis in Automation and Intelligent Systems
Supervisor: Øystein Bjelland
Co-supervisor: Aleksander Skrede
May 2024

Filip Arthur Blaa fjell
Mathieu Steketee
Robert Asp rem
Andreas Blindheim

Multi-Articulated Myoelectric Prosthesis

Costumized for the Norwegian Market

Bachelor's thesis in Automation and Intelligent Systems
Supervisor: Øystein Bjelland
Co-supervisor: Aleksander Skrede
May 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences



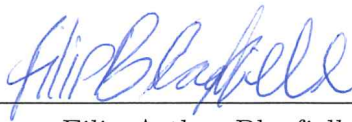
Preface

This thesis is the culmination of our teams pursuit of a Bachelors degree in Automation and Intelligent Systems at NTNU Ålesund, and encompasses the wide knowledge and possibilities gathered throughout our studies. The decision to explore this area was driven by its major humanitarian possibilities, the technological challenges it involved and the valuable experience it encompasses.

Our team would like to give a special thanks to our project supervisor and PhD candidate Øystein Bjelland for his continuous support and extremely insightful guidance throughout this development. Furthermore, we would like to thank Adam Leon Kleppe, Kai Erik Hoff and other academics involved with Ålesund Biomechatronics Lab for their their support. We would also like thank the Department of ICT and Natural Sciences at NTNU for supplying our team with the necessary resources to make this project successful.

The experiences gathered through the development of this thesis has been invaluable for our team. Not only have they improved our academic and theoretical knowledge, but also, in our opinion, furnished professionalism. We look forward to make further contributions in our future endeavours by utilizing our experiences from this journey.

Ålesund, 20/05/2024



Filip Arthur Blaa fjell



Robert Asp rem



Andreas Blindheim



Mathieu Stek etee

Abstract

This thesis presents the design, development and assessment of a multi-articulated myoelectric prosthesis customized for the Norwegian market. Prosthetics are essential for restoring function and improving the quality of life for individuals who have lost limbs, but today's advanced multi-articulated prostheses are prohibitively expensive for many who need them. This study includes a detailed literature review, theoretical background, methods of development and assessments of results. The methodology covers the approaches and solutions of all hardware and software components. By employing structured data collection, advanced signal processing and a one-dimensional convolutional neural network, a real-time pattern recognition software was developed, delivering an 84% accurate gesture classification within 350 ms of delay. Another framework was the successful approximation of the human hand joint DOFs, grip functionality and other factors of anthropomorphism. The mechanical and electrical aspects, such as tendon driven actuation and PCB design, proves effective in activities of daily living and anthropomorphic properties and cost. The results of this research demonstrates critical functionality of a low cost multi-articulated myoelectric prosthesis that can fill a gap in the Norwegian market.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Related Work	2
1.3	Problem Definition	2
1.4	Thesis structure	3
2	Theoretical Basis	4
2.1	Hand Anatomy	4
2.1.1	Robotic emulation	5
2.2	Surface Electromyography (sEMG)	6
2.2.1	Application in Prosthetics	6
2.3	Feature Extraction Techniques	6
2.3.1	Discrete Wavelet Transform (DWT)	6
2.3.2	Integrated EMG (iEMG)	7
2.3.3	Mean Absolute Value (MAV)	7
2.3.4	Root Mean Square (RMS)	7
2.4	Convolutional Neural Network (CNN)	8
2.4.1	1D CNN for sEMG Classification	8
3	Method: System Implementation	9
3.1	Mechanical Design	9
3.1.1	Actuation System Components	9
3.1.2	Tendon Drive Actuation System	10
3.1.3	Forearm Design	11
3.2	Electrical Design	12
3.2.1	PCB Design	12
3.2.2	MCU Selection	13
3.2.3	Battery-pack Development	14
3.3	Data Acquisition and Preparation	16
3.3.1	Muscle Sensor	16
3.3.2	Data Collection	16
3.3.3	Preparation and placement	17
3.4	Signal Processing and Software Design	19
3.4.1	Data Pre-processing	19
3.4.2	Feature Extraction	21
3.4.3	CNN Development	22
3.4.4	Model Validation	23
3.5	Microcontroller (MCU) Programming	24
3.5.1	STMicroelectronics (STM) Ecosystem	24

3.5.2	X-Cube AI Expansion Pack	24
3.5.3	Actuator control	25
4	Evaluation of Critical Functionality	26
4.1	Assessment of Anthropomorphic Appearance	27
4.1.1	Motivation	27
4.1.2	Method	27
4.1.3	Results	27
4.2	ADL Performance Evaluation	28
4.2.1	Motivation	28
4.2.2	Method	28
4.2.3	Results	29
4.3	Assessment of CNN Off-Line Performance	29
4.3.1	Motivation	29
4.3.2	Method	30
4.3.3	Results	30
4.4	Assessment of CNN Real-Time Performance	31
4.4.1	Motivation	31
4.4.2	Method	32
4.4.3	Results	33
4.5	Assessment of Computation Time	34
4.5.1	Motivation	34
4.5.2	Method	34
4.5.3	Results	35
4.6	Assessment of Accessibility	35
4.6.1	Motivation	35
4.6.2	Method	35
4.6.3	Result	36
5	Discussion	37
6	Conclusion	44
	References	45
	Appendix	49
A	Github Repository	49
B	Components lists	49
C	Morphology Chart	51
D	Wiring Schematics	52
E	Guideline	54

List of Figures

2.1	Visualization of human hand joints	5
2.2	Visualization of human hand degree of freedom (DOF)	5
2.3	Discrete Wavelet Transform (DWT) with 4 Levels	7
2.4	Example architecture of a stacked 1D CNN	8
3.1	Actuator with coil	10
3.2	Split view of folded finger	10
3.3	Motor placement	11
3.4	Complete forearm design split into two parts	12
3.5	2D overview of the bare PCB designed to fit the forearm	13
3.6	STM32F405RG MCU used in the prosthesis	14
3.7	Custom made battery pack assembly with three 18650 batteries	14
3.8	Battery Management System (BMS)	15
3.9	Finished battery pack assembly with three 18650 batteries	15
3.10	3D-printed enclosure housing three Sparkfun Myoware 2.0 sensors	16
3.11	Overview of the data collecting system	17
3.12	sEMG electrode placement on the forearm	17
3.13	Overview of the different gestures	18
3.14	EMG signal processing workflow	20
3.15	Feature extraction matrix	21
3.16	TensorFlow Lite conversion workflow	23
3.17	PID control loop.	25
4.1	Final prototype of prosthesis powered on in resting position	26
4.2	Gesture comparison with biological hand	27
4.3	Grasping objects from selected SHAP ADL tasks.	29
4.4	CNN model accuracy and loss	30
4.5	Confusion Matrix	31
4.6	Real-time gesture prediction system	32
4.7	Real-time Feedback System for Gesture Prediction	33
4.8	Real-time accuracy of the participants	34
4.9	Realtme testing off the CNN model on the STM32	35
5.1	Testing friction in fingers on a simple model	38
5.2	Capacitor soldered on PCB	40
5.3	Analysis of filtering techniques used in this thesis	42
1	Morphological chart	51
2	MCU wiring schematic	52
3	GPIO wiring schematic	52
4	Motors wiring schematic	53

5	Power wiring schematic	53
---	----------------------------------	----

List of Tables

3.1	Overview of the 6 different participants included in the dataset	19
3.2	Overivew of the 1D-CNN used in this thesis	22
4.1	Comparison of anthropomorphic properties.	28
4.2	Results in selection of SHAP ADL tests.	29
4.3	Price comparison of MMHPs	36
1	Components list with description and links	49
2	Bill of materials for printed circuit board	50

Acronyms

sEMG	Surface electromyography	ADC	Analog-to-digital converter
MMHP	Multi-articulated myoelectric prosthesis	LSTM	Long Short Term Memory
SMHP	Single-articulated myoelectric prosthesis	MCU	Micro Controller Unit
NTO	Norsk Teknisk Ortopedi	ML	Machine learning
CNN	Convolutional neural network	STM	STMicroelectronics
1D CNN	One-dimensional convolutional neural network	GPIO	General-purpose input/output
DWT	Discrete wavelet transform	IDE	Integrated development environment
NN	Neural network	SP	Setpoint
ADL	Activities of daily living	PV	Process variable
PWM	Pulse width modulation	BMS	Battery management system
PCB	Printed circuit board	Ag/AgCl	Silver/Silver chloride
SMD	Surface mounted device	SENIAM	Surface EMG for noninvasive assessment of muscles
DOF	Degree of freedom	ReLU	Rectified Linear Unit
CAD	Computer aided design	TF	Tensorflow
MP	Metacarpal joints	PID	Proportional Integral Derivative
PIP	Proximal interphalangeal joints	SHAP	Southampton Hand Assessment Procedure
DIP	Distal interphalangeal joints	AIT	Average inference time
CMC	Carpometacarpal joint	PLA	Polylactic acid
IP	Interphalangeal joint	RMS	Root mean square
iEMG	Integrated electromyography	I ² C	Inter-integrated circuit
MAV	Mean absolute value	PROS-TLX	Prosthesis Task Load Index
USART	Universal synchronous and asynchronous receiver-transmitter	NASA-TLX	NASA Task Load Index
		EMI	Electromagnetic interference

Chapter 1

Introduction

1.1 Background and Motivation

The field of prosthetic design has seen substantial advancements throughout the past decades, especially in myoelectric prostheses. These devices utilize surface electromyography (sEMG) from residual limbs to control functionality. By leveraging sEMG data acquisition, processing, and machine learning, the quality and user experience of prostheses have greatly improved. However, despite rapid progress and substantial scientific investment, several critical issues still limit their widespread market adoption.

Recent advancements in prosthetic technology have led to the development of multi-articulated myoelectric hand prostheses (MMHP), aimed at enhancing dexterity and functionality for users. However, studies, including those by Kerver et al. [1], reveal significant challenges in their practical application. Despite the potential benefits of multi-articulation, users often face difficulties with control mechanisms that are both time-consuming and cognitively demanding. These complexities do not necessarily lead to improved functionality or user satisfaction in everyday activities [1][2]. Moreover, the increased cognitive load and the need for constant attention to grasping and adjusting can lead to user frustration and decreased acceptance of the prosthesis. Furthermore, robustness is a commonly arising concern among users of MMHPs [1][2]. Patients may become more passive in their use of prostheses or even reject them due to concerns about damage or failure during crucial tasks. As a result, they might prefer conventional single-articulated myoelectric prostheses (SMPH) for their reliability over advanced functionality.

These issues align with the view of Norsk Teknisk Ortopedi (NTO), a Norwegian orthopedic workshop specialised in arm prostheses, and the only of its kind in Scandinavia [3]. NTO points out that SMPHs, specifically mentioning a 8-10kg grip force Otto Bock hand, offer a reliable and robust user experience, despite its limited functionality compared to MMHPs. Comparatively, while the MMHPs such as the iLimb hand¹, the BeBionic hand² and the Michelangelo hand³ may be more adaptive in some situations, detrimental factors such as grip force, movement speed, fragility, weight and noise are usually a concern. Additionally, since all prostheses are state subsidised in Norway, patients are demanded to wear a standards prosthesis for a year, attend a comprehensive needs assessment with a arm prosthetic team, and if the application is accepted, attend obligatory training with a rehabilitation team[3].

¹<https://www.ossur.com/en-us/prosthetics/arms/i-limb-quantum>

²https://www.ottobock.com/en-us/product/8E7*

³<https://www.ottobock.com/no-sc/product/8E500>

1.2 Related Work

Due to its major humanitarian implications, myoelectric prosthetics is a rapidly expanding area and a substantial amount of research has been conducted in the field. The development of advanced prosthetics encompasses an extensive variety of expertise, each serving as their own research areas. Robotic emulation of human anatomy, real-time prosthetic control and rehabilitation and therapeutics are all examples of such areas contributing to advancing prosthetics. The market of prosthetics is extensive and numerous products are available for patients. This ranges from customized devices for special applications, such as specific work environments, to advanced anthropomorphic devices aiming to fully emulate the human hand functionality.

A key aspect of myoelectric prosthetic development, as mentioned, is control strategies. Recently, several studies have shown promising results using a combination of feature extraction and deep learning methods to classify sEMG signals to specific hand gestures. One-dimensional convolutional neural networks (1D CNNs) in combination with, for instance wavelet transforms or Fourier transforms has proved quite effective in classification tasks involving physiological signals, such as sEMG. For instance, Zafar et al. [4] developed a 1D-CNN performing with 99.3% accuracy using the discrete wavelet transform (DWT) to extract features from sEMG signals as well as a hyperparameter optimization algorithm.

Another crucial aspect of myoelectric prosthetic development is robotic anthropomorphism. This entails emulating the human anatomy, physical likeness and function using electronics and mechanics. This has proven a challenging task, as the complexity of the human anatomy currently outperforms technological adaptations in speed, force and precision. Developers such as those at Touch Bionics and Ottobock have made strides with MMHP models like the i-Limb and BeBionic. However, even these state of the art devices still fall short in delivering the function that a biological hand does [5].

In conclusion, while several issues remain in MMHP development, the field also presents major opportunities to contribute and advance innovation and impact. This thesis takes advantage of the amount of conducted research and highlighted issues to further add insightful findings to the field.

1.3 Problem Definition

To what extent can a low-cost MMHP mimic the anthropomorphism of the human hand, reduce the cognitive demand of device control, and perform in daily support hand tasks to potentially be more accessible and increase acceptance rates?

This thesis aims to consider these issues and present the development and assessment of a prosthesis in the middle of the function/reliability spectrum by merging the benefits from MMHPs and SMHPs. This entails the following tasks:

- **Conduct thorough research, assessing methods of development, performance and the market status to identify the key aspects of development.**
- **Develop a robotic emulation of the human hand, mimicking the function of joints, tendons and muscles through CAD, 3D printing and PCB design.**
- **Create a robust neural network (NN) control system through sEMG data collection, data preprocessing, feature extraction and NN gesture classification.**
- **Evaluate the performance in activities of daily living (ADLs), assess the effi-**

ciency of the real-time gesture prediction software and appraise varying factors of the robotic anthropomorphism.

1.4 Thesis structure

This thesis is structured into six chapters with theoretical basis, methodology, results, discussion and conclusion remaining:

- **Chapter 2 - Theoretical basis:** This chapter includes important theoretical information applied in the methodology, laying a sound foundation for the rest of the thesis.
- **Chapter 3 - Method: System Implementation:** This chapter covers all methods of development and preliminary results of the thesis, giving insight into how the results were acquired.
- **Chapter 4 - Evaluation of Critical Functionality:** This chapter integrates a assessment of the final achievements, covering the methods of testing and presenting the final statistics.
- **Chapter 5 - Discussion:** This chapter integrates a analysis of the methodology and results chapters, such as potential improvements, the scope of the the thesis and the implications of the results.
- **Chapter 6 - Conclusion:** The conclusion simply summarizes the entire thesis and its implications.

Chapter 2

Theoretical Basis

2.1 Hand Anatomy

The human hand anatomy is a intricate composition of muscle, bones, tendons, joints and nerves, enabling humans to perform a multitude of complex tasks over 27 degrees of freedom (DOFs) [6]. The joints of the human hand is visualized in figure 2.1 and can be categorized followingly:

- **Metacarpal joints (MP):** Joints connecting the palm to the innermost finger part as well the innermost thumb part to the middle thumb part.
- **Proximal interphalangeal joints (PIP):** Joints connecting the innermost and middle parts of the fingers.
- **Distal interphalangeal joints (DIP):** Joints connecting the middle and outermost part of the finger.
- **Carpometacarpal joint (CMC):** Joint connecting the palm to the innermost thumb part.
- **Interphalangeal joint (IP):** Joint connecting the middle and outermost thumb part.

The degree of freedoms (DOF) direction each joint accounts for is visualized in figure 2.2 and can be categorized followingly.

- **Finger flexion-extension:** Bending and retracting the finger MP, PIP, DIP and joints downwards towards the palm..
- **Finger abduction-adduction:** Spreading and closing the finger MP joints.
- **Thumb opposition-retroposition:** Rotating the CMC joint inwards and outwards towards the palm.
- **Thumb radial abduction-adduction:** Moving the CMC joint towards the index finger and back in the plane of the palm.
- **Thumb palmar abduction-adduction:** Moving the CMC joint inwards and outwards perpendicular to the palm.

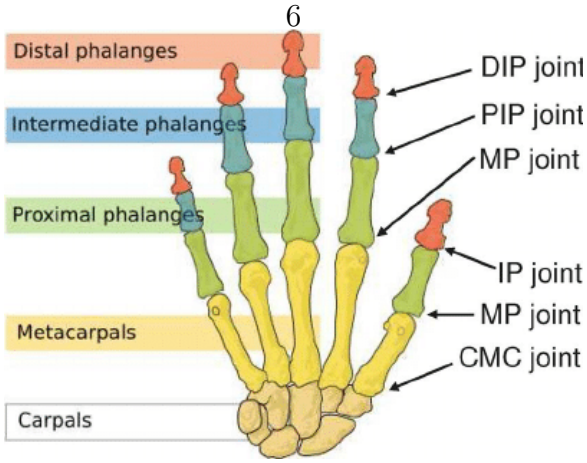


Figure 2.1: Visualization of human hand joints [7].

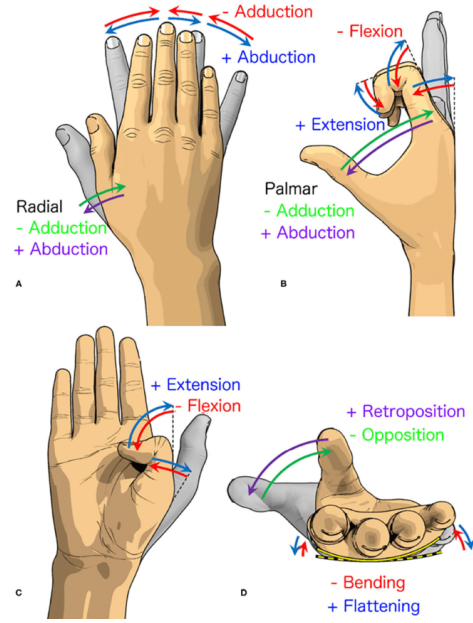


Figure 2.2: Visualization of human hand DOFs [8].

2.1.1 Robotic emulation

Designing a robotic emulation of the human hand includes several crucial principles. Structural emulation is an essential factor when designing a prosthetic device. This entails replicating the physical form of the human hand. Even though it may not be optimal for functionality, a one to one structural replica is crucial in prosthetics as it enhances the patients feeling of having a real hand. Furthermore, due to the complex anatomy of the hand, making an exact replica the human hand DOFs through a combination of mechanics and actuation is likely not efficient. Approximation the hand DOFs, usually happens through specific selection of joint articulation needed for the device application.

A solution to simplify the articulation of the hand in MMPH acutation is tendon drives. This simplifies the human hand function by utilizing one motor for each finger connected to a tendon running thorough the joints. While this disregards the ability of individual joint control, as humans have to a certain degree, it enables even joint articulation across the flexing of the fingers. This solution does however require some resistance in the joints to accommodate the extension of the fingers and apply even torque in the joints. This can be solved using torsion springs. To choose a suitable spring for the applied motor torque, the following relationship can be used:

$$\theta = \frac{64MDN_a}{Ed^4}$$

Where θ is the spring angular deflection (amount of rotation form equilibrium), M is the applied motor torque, D is the spring center diameter, d is the spring wire diameter, E is Young's modulus of elasticity of the spring material and N_a is the number of active turns (coils) [9].

2.2 Surface Electromyography (sEMG)

Surface electromyography (sEMG) is a technique used to capture the voltage produced by muscles [10]. This method is widely used in including clinical diagnostics, rehabilitation, and the control of assistive devices such as prostheses.

2.2.1 Application in Prosthetics

sEMG signals can be leveraged in the development of myoelectric prostheses [11]. These devices utilize the electrical signals generated by muscle contractions to control the movements of the prosthetic limbs. The use of sEMG in prosthetics allows for real-time user control, as the prosthetic movements are directly linked to the user's muscular signals.

The sEMG signals are processed to determine the intended motion or force output by the user. This is typically done through various signal processing techniques, including filtering, rectification, and normalization, to enhance the quality and interpretability of the signals. Advanced algorithms, such as pattern recognition, can then be employed to translate these processed signals into specific commands that control the movements of the prosthesis[4].

2.3 Feature Extraction Techniques

Extracting meaningful features from surface electromyography (sEMG) signals is crucial for effective prosthesis control. Feature extraction methods help transform raw sEMG data into useful information that can be used for classification and analysis [4].

2.3.1 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is a mathematical method used to decompose signals into several different frequency components, also known as wavelets [4]. DWT is widely adopted in various applications due to its ability to handle both the time and frequency information of a signal concurrently. This signal processing method enables the extraction of specific signal features, making it a highly efficient preprocessing step for pattern recognition algorithms. DWT on a signal $x(n)$ can be mathematically expressed as:

$$c_{j,k} = \langle x, \psi_{j,k} \rangle = \sum_n x(n)\psi_{j,k}(n) \quad (2.1)$$

where $c_{j,k}$ are the wavelet coefficients, $\psi_{j,k}(n)$ is the scaling function [4].

In figure 2.3 the diagram visually represents this process, showing the hierarchical breakdown of the input signal into approximation and detail coefficients at each level. The diagram effectively illustrates how the signal is progressively broken down into different frequency components across four levels of decomposition.

The DWT stands out as a valuable tool for analyzing sEMG signals due to its ability to decompose signals into different frequency components while retaining both time and frequency information. Compared to other feature extraction methods like the Fourier transform, DWT has demonstrated superior classification performance in detecting hand movements using EMG signals. This advantage arises from DWT's effectiveness in capturing transient features and local patterns that are often lost in Fourier transforms. In particular, using the Daubechies 10 (db10) wavelet has proven very efficient for capturing transient features and local patterns in sEMG data [4].

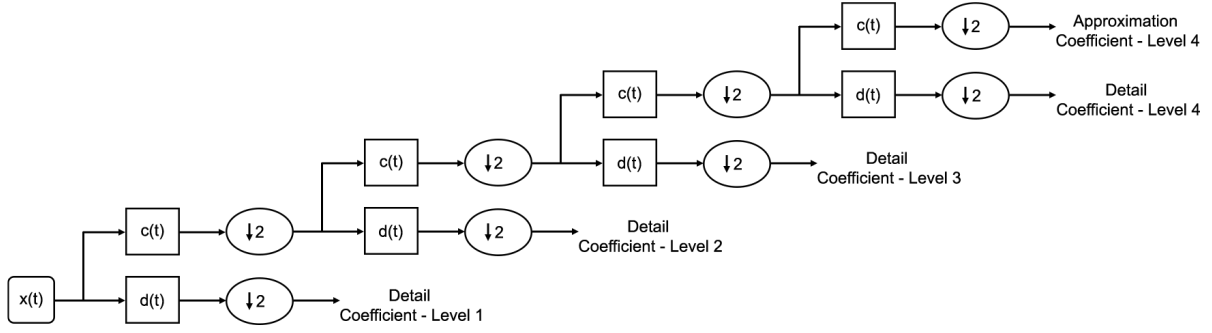


Figure 2.3: Discrete Wavelet Transform (DWT) with 4 Levels. This diagram illustrates the DWT process, decomposing an input signal $x(t)$ into its individual frequency components across four levels.

2.3.2 Integrated EMG (iEMG)

Integrated EMG (iEMG) is a feature extraction technique that involves summing the absolute values of the EMG signal over a specified time window [12]. This method provides a measure of the total electrical activity of the muscle over the window, reflecting the overall muscle effort. iEMG can be mathematically expressed as:

$$\text{iEMG} = \sum_{i=1}^N |x_i| \quad (2.2)$$

where x_i is the EMG signal value at the i -th sample and N is the total number of samples in the window.

2.3.3 Mean Absolute Value (MAV)

Mean Absolute Value (MAV) calculates the average of the absolute values of the EMG signal within a given window [12]. This feature provides a straightforward measure of the signal's intensity, offering a balance between computational simplicity and effective representation of muscle activity. MAV can be mathematically expressed as:

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2.3)$$

where x_i is the EMG signal value at the i -th sample and N is the total number of samples in the window.

2.3.4 Root Mean Square (RMS)

Root Mean Square (RMS) is a statistical measure of the magnitude of the EMG signal, calculated as the square root of the mean of the squares of the signal values [12]. RMS is widely used due to its ability to represent the power of the EMG signal, providing insights into the muscle contraction's strength and consistency. RMS can be mathematically expressed as:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |x_i|^2} \quad (2.4)$$

where x_i is the EMG signal value at the i -th sample and N is the total number of samples in the window.

2.4 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a specific subset of deep learning neural networks [13]. CNNs are particularly effective in tasks that use images, videos, time-series or other spatial and temporal input data. The convolutional layers are the core of the deep learning operation as they create feature maps from the input data. A convolutional layer consists of filters, strides and an activation function. The filters, also known as kernels, slide over the input with a given stride and perform an element-wise multiplication. Subsequently the product at each stride is summed and added together with a bias term and passed through the activation function to add non-linearity. Convolutional layers are usually stacked to create a deep learning algorithm, enabling it to capture different features based on the different parameter configurations [13].

2.4.1 1D CNN for sEMG Classification

When using CNNs for classifying sEMG signals as a specific body movement, a one-dimensional CNN (1D CNN) is usually adopted [4]. This refers to the input shape of the CNN, and since sEMG data is temporal, the input shape is one-dimensional. To efficiently capture the different features in sEMG data, such as muscle contraction intensity or specific contraction patterns, a variety of convolutional layers are usually stacked. An example architecture of a 1D CNN is visualized in figure 2.4.

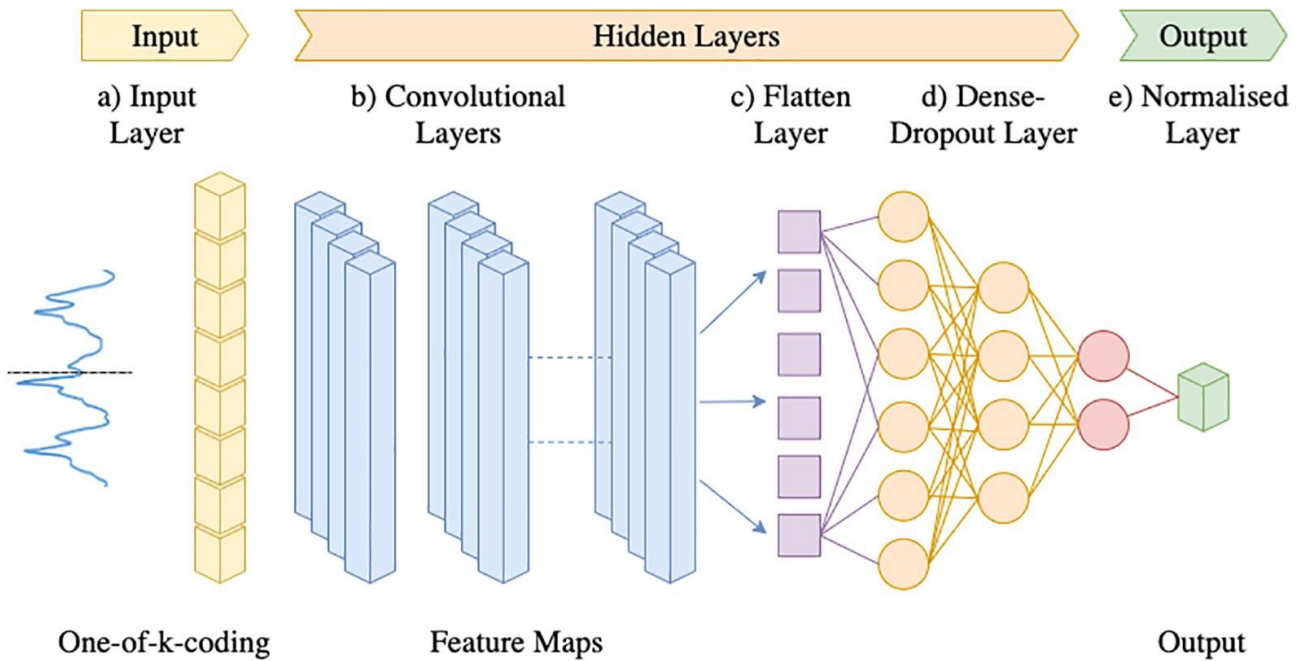


Figure 2.4: Example architecture of a stacked 1D CNN [14].

Chapter 3

Method: System Implementation

This chapter encompasses a detailed explanation of the various approaches and methods utilized in this thesis. This entails step by step explanations of component and design choices as well as software development. All aspects of hardware and software is discussed and aims to give insight into how the theoretical concepts are translated into practical development.

3.1 Mechanical Design

The mechanical design of the prosthesis is a approximated emulation of the human hand and was designed using a low-cost approach. Several simplifications of the human hand were necessary to increase the feasibility and lower the cost of the project. This includes a reduction of Degrees of freedom (DOF), removal of certain joints and usage of low-cost materials. A major goal of the mechanical design was to mimic the structural anthropomorphism of the human hand by fitting all electronics within the device, such as the actuation system, battery assembly, Printed circuit board (PCB), the myoelectric sensors and wiring. Another crucial goal was to integrate the necessary and most important human hand DOFs, enabling it to perform a variety of gestures.

3.1.1 Actuation System Components

To control the fingers and thumb independently and enable multi-articulation, a tendon driven actuation system was chosen. A six motor configuration was chosen to control the most essential DOFs in the human hand and mitigate overcomplication. This includes one motor for each flexion-extension of the fingers and thumb, as well as one extra motor for the opposition-reposition DOF enabled by the thumb carpometacarpal (CMC) joint. To ensure an anthropomorphic appearance and strength, the actuators need to be placed inside the device as well as deliver approximately 11.57 mNm at rated load.

Moreover, the actuators had to be implemented in a upright position to not make the hand to wide. Therefore, to align the shaft rotation with the finger movement direction, a 90°shaft angle was required. DC motors are usually quite small in size and still deliver a comparatively high torque. However, compared to, for instance, servo and stepper motors, they require some sort of feedback sensors to control accurately. After a technology analysis, the team discovered the GW1812-N20-E, a 68 rpm geared DC motor with a 90°shaft angle. With its small volume of (size), the motors would fit nicely into the palm of the prosthesis.

However, to control the DC motors accurately, some sort of feedback sensors also had to be

implemented. While the DC motors have a integrated encoder position feedback, it requires a static set-point for each gesture. To ensure a dynamic set-point, current sensors was ultimately chosen due to the lack of mechanical parts, its proportionality to motor torque, small size and low cost. Moreover, since the micro controller unit (MCU) does not deliver the necessary voltage and current to run the actuators, motor drivers had to be implemented. Here, the TB6612 driver was chosen for its low cost, applicability and, especially, its dual H-bridge configuration. The H-bridge configuration enables a pulse-width modulation (PWM) amplification without affecting the duty cycle and enables directional motor control. Furthermore, the dual H-bridge entails that each driver can control two motors each, reducing the amount of components and complexity of the electrical design.

3.1.2 Tendon Drive Actuation System

Several steps were taken to develop the tendon drive actuation system. Firstly, the team had to ensure the spring properties of the torsion springs inside the MP and PIP joints were suitable. The angle of the joints at maximum flexion (angle between the palm and finger) is 75° for the MP joint and 69° for the PIP joint. By using the formula for angular deflection in torsion springs discussed in section 2.1.1, the needed spring parameters to actuate the joints to maximum flexion was found.

As seen in figure 3.2, the tendon runs through a channel in the finger and is connected inside the outermost part using print in place technique. This methodology involves pausing the 3D print mid-process to enable the insertion of components, ensuring secure attachment of the tendon within the printed structure. A major part of the design was to choice to have a static DIP joint in the fingers. This was mainly done to reduce the complexity of the design and to increase flexion strength of the fingers by having one less spring to drive. Moreover, the DIP joint probably does not have much function when fully extended and is usually in a semi-articulated position when resting as well as when grasping. Another crucial aspect of the actuation system was to design and fasten a pulley on the motor shaft. The tendon is attached to this pulley and reeled in during flexion.

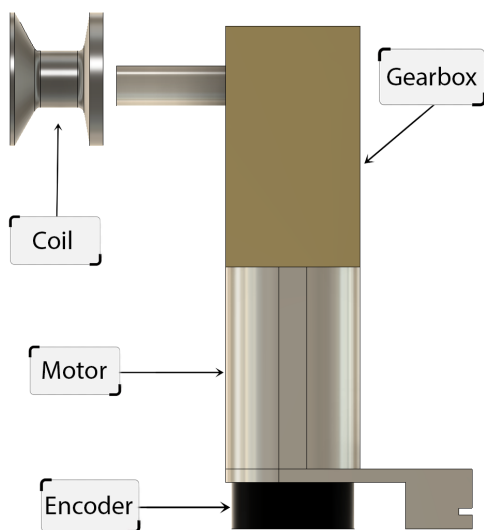


Figure 3.1: Actuator with coil



Figure 3.2: Split view of folded finger

The thumb actuation differs from the fingers as an extra motor is used to increase functionality. The opposition-retroposition DOF enabled by the CMC joint of the thumb is crucial in hand functionality and is therefore included in this design [15]. However, to articulate this joint in addition to the flexion-extension of the MP and IP joints, an additional actuator was implemented. This actuation does not rely on the tendon drive system implemented for flexion-extension movement, but utilizes a ball bearing instead. As seen in figure 3.3b, the thumb assembly is connected to the palm with a motor shaft and a ball bearing, enabling articulation of the whole system. The motor facilitating the flexion-extension movement of the thumb is located within the thumb design. The palm was designed as two parts to accommodate simple assembly, where one has placeholders for all the motors and channels for the wiring. The finger and thumb assembly on the palm as well as motor placements can be seen in figure 3.3a.

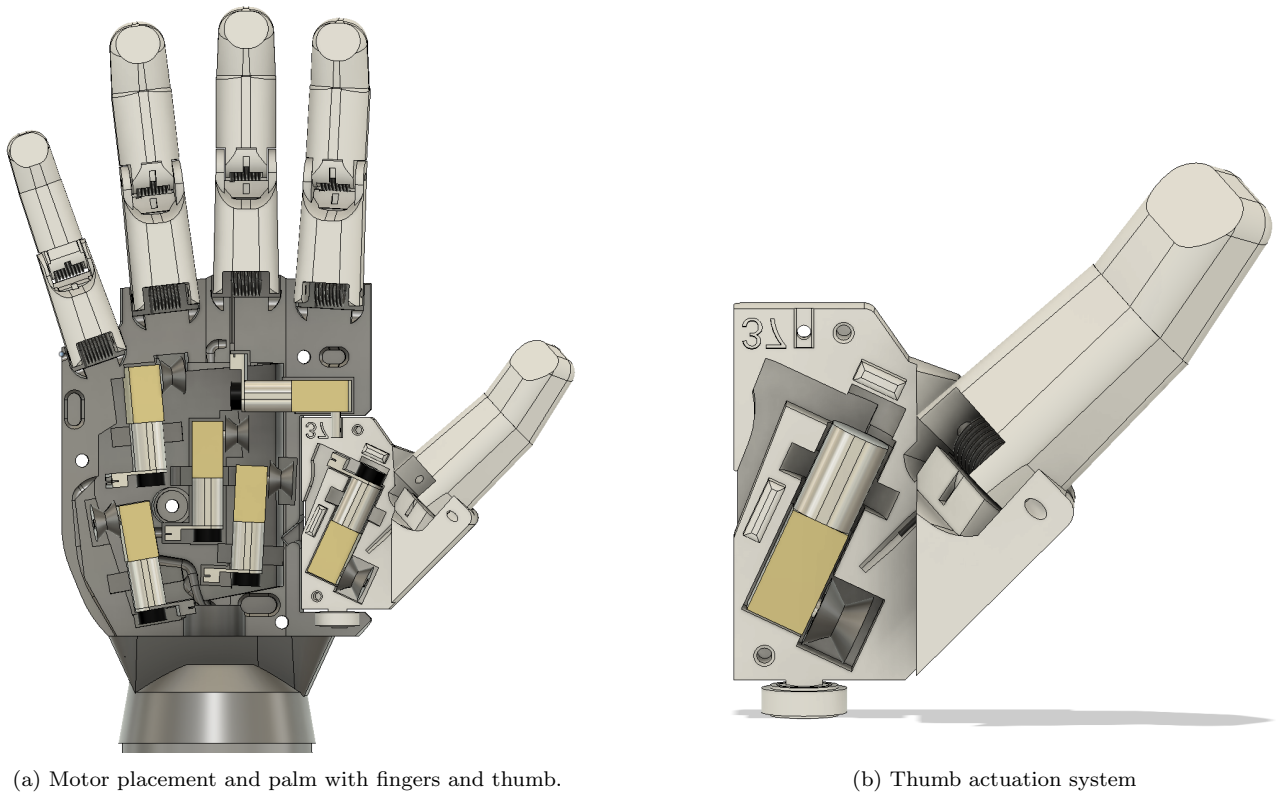
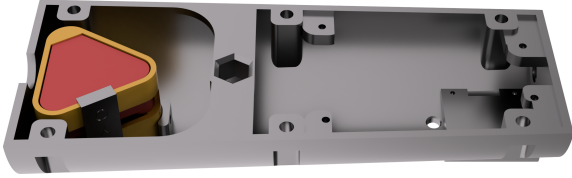


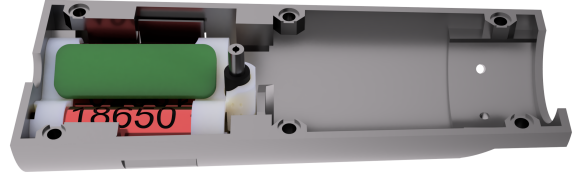
Figure 3.3: Motor placement

3.1.3 Forearm Design

The forearm design is developed to encapsulate all necessary components for controlling the prosthesis, apart from those housed within the hand itself. This includes a printed circuit board (PCB), sEMG sensors for detecting muscle signals, and an OLED display, which provides users with real-time status updates on gestures, battery level and motor torque. Additionally, the design incorporates a through-hole for a tactile button and a dedicated slot for securely mounting the custom battery pack. The forearm is not cylindrical but rather conical to better match the shape of the human forearm, reflecting on the goal to achieve an anthropomorphic design. The assembly is split at the middle into two parts, allowing for the proper mounting of components inside the forearm before final assembly and enclosure of the entire prosthesis.



(a) Forearm upper side. This houses sEMG sensors, Printed Circuit Board, OLED display and through-hole for button.



(b) Forearm lower side. This houses custom made battery pack.

Figure 3.4: Complete forearm design split into two parts. The forearm houses all the necessary electrical components apart from the actuators which are located in the palm of the hand.

3.2 Electrical Design

The electrical system of the prosthesis is designed to ensure high performance and reliability. It encompasses several critical components including the Printed Circuit Board (PCB) design, Micro Controller Unit (MCU) selection, electrical schematics, and battery-pack development. Every component is designed to function together seamlessly, delivering power, control, and functionality to the prosthesis.

3.2.1 PCB Design

The PCB design is a critical aspect of the prosthesis’s electrical system. It was designed to accommodate the intricate network of components while maximizing efficiency and minimizing space consumption. A 4-layer-stack-up was chosen to support the 98 components required to control the prosthesis. Surface mounted devices (SMD) was favoured over through-hole components due to their smaller footprint, enabling high-density and efficient use of limited space.

The layout was designed to minimize interference, and includes a low-power OLED display for showing the running status of the prosthesis, such as active gestures and battery status. The components have a wide variety of voltages, with the VCC being 3.3 V for the majority of the components. The current-sensing Hall effect sensors and the RGB chip operate on 5 V, while the motor drivers are served close to 13 V and are limited by PWM with a duty cycle of 92 %, which is explained in section 3.2.3. Additionally, the PCB is fitted with a tactile button for multi-purpose control. Furthermore, to facilitate easy programming, the PCB includes a USB-C port connected to a CH340C serial-to-UART chip, which works in addition to the ST-Link V2 programming tool.

The output voltage V_{OUT} of the LM2577-ADJ boost converter was calculated using the following formula:

$$V_{\text{OUT}} = 1.23 \text{ V} \left(1 + \frac{R1}{R2} \right)$$

where $R1 = 19\,000 \Omega$ and $R2 = 2000 \Omega$, making the theoretical output

$$V_{\text{OUT}} = 1.23 \text{ V} \left(1 + \frac{19\,000 \Omega}{2000 \Omega} \right) \approx \underline{12.92 \text{ V}}[16]$$

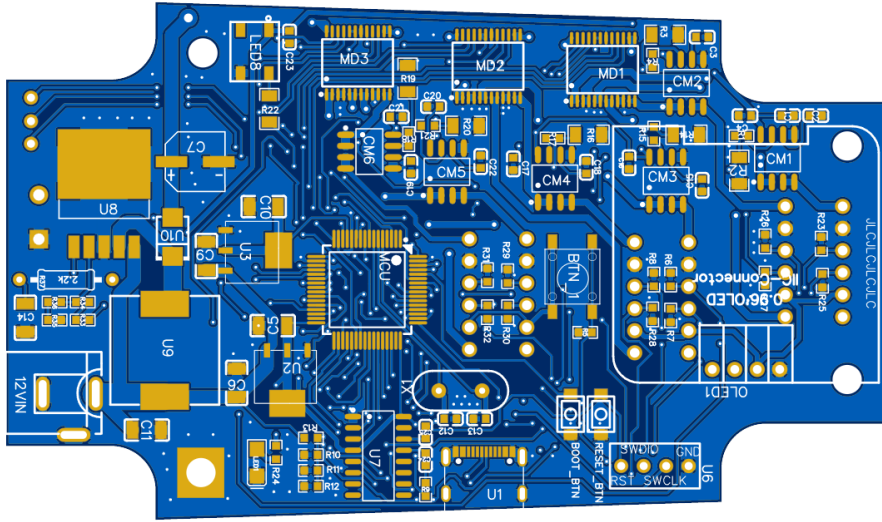


Figure 3.5: 2D overview of the bare PCB designed to fit the forearm. It features 95 of 98 components mounted on the front, primarily using SMD components.

The PCB was designed using EasyEDA ¹, a software that facilitated several critical steps, including schematic design, component placement, and thorough design rule checks (DRCs). A four-layer PCB was chosen to accommodate all the necessary traces and components, especially given the small distance between soldering pads on the selected STM microcontroller, which only is 0.127 mm. The additional layers were essential to ensure proper routing of signals and power distribution.

The outermost layers (first and fourth) were designated for current-bearing traces, such as those for the motors. These layers were chosen because of their capability to conducting higher currents than the buried layers. In addition to high-current traces, the first and fourth layers, as well as the second layer, were also used for the signal traces. The third layer was dedicated to be a ground layer. This ground layer was crucial, as it provided a reliable ground connection for the remote components through the use of vias.

The top and bottom layers were shielded with ground planes to enhance signal integrity and reduce electromagnetic interference (EMI) [17]. By strategically organizing the layers and utilizing ground planes, the PCB design ensured robust performance and reliability while meeting the small space requirements of the components.

See appendix D for the complete schematic drawings, and appendix B for chosen components.

3.2.2 MCU Selection

The MCU is crucial component for the system, as it is tasked to take input from sEMG sensors, process the data, run the neural network (NN) model inference and control the actuators. Several steps were taken to determine the choice of the MCU, including size, clock speed, memory and number of general-purpose input/output (GPIO). A sufficient MCU clock speed and memory is essential when implementing resource consuming software like real-time machine learning (ML) inference. Lastly, a vast amount of GPIO was vital to control the multitude of sensors and actuators.

¹<https://easyeda.com/>

After reviewing several MCU's, such as the ESP32 and the ATMEGA, a STM32 was chosen, specifically the STM32F405RG. This 32-bit MCU, built on the ARM M4-Cortex core, has a strong power to size ratio and is widely implemented in embedded systems requiring more intensive mathematical processing. Additionally, it includes 43 GPIOs which supports the vast amount of components in the circuit. The integrated development environment (IDE) for the STM32 also includes the X-Cube expansion pack, which facilitates convenient ML deployment.

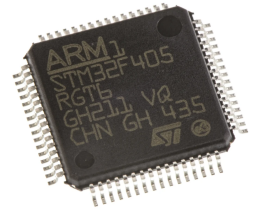


Figure 3.6: STM32F405RG

3.2.3 Battery-pack Development

For the battery pack, 18650 lithium-ion cells with a capacity of 2500mAh were chosen. These cells are known for their high capacity and ease of use in pack building, particularly when it comes to welding. Li-ion batteries are known for their impressive performance characteristics, including high specific energy, long life, and the ability to operate over a wide temperature range [18].

The motors in the project require a voltage of around 12V. Considering the voltage range of each 18650 cell (3V - 4.2V), three cells in series provide a voltage ranging from 9V - 12.6V. This configuration meets the requirements for the motors.

To ensure the direct current motors operate efficiently without being affected by the state of charge of the battery, a fixed output voltage is necessary. A buck-boost converter could be used for fixing the voltage at 12V. However, due to its cost and complexity, a boost converter was selected instead. The boost converter raises the voltage to a consistent 13V. Although this is within the acceptable input for the motor drivers, it exceeds the maximum allowed voltage for the motors. To address this, the Puls Width Modulation (PWM) duty cycle is limited to 92%, ensuring the motors receive a maximum voltage of 12V, calculated as $\frac{12V}{13V} \approx 92\%$. This solution balances the need for stable voltage while considering cost and design complexity.

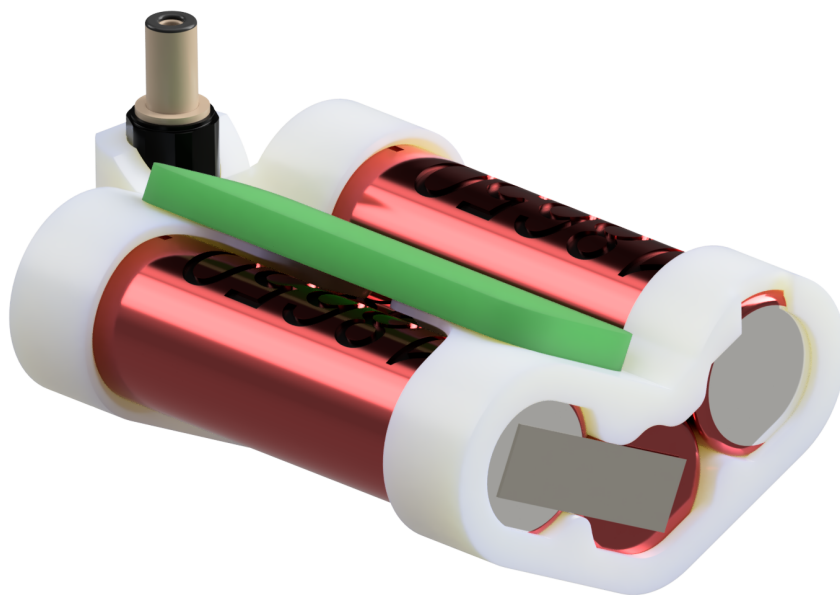


Figure 3.7: Custom made battery pack assembly with three 18650 batteries in series. Designed to fit in the forearm.

To determine the current draw of the entire PCB at the lowest possible battery voltage (9 V), the stall current of six motors, each at 230 mA is calculated. The total current for the motors is:

$$6 \times 230 \text{ mA} = 1380 \text{ mA}$$

Including the other components, which draw an estimated 200 mA from a linear voltage regulator to achieve 3.3 V, the total current is:

$$1380 \text{ mA} + 200 \text{ mA} = 1.58 \text{ A}$$

The pure nickel strips have dimensions of $0.2 \times 8 \text{ mm}$, likely capable of handling sufficient current for this application.

It is crucial to avoid excessive heating of the batteries during assembly, as prolonged exposure to high temperatures can degrade their performance and safety [18]. For this reason, spot welding is preferred over soldering. Spot welding minimizes the heat exposed to the battery cells compared to soldering, which involves directly heating the terminals and applying tin. The welder was set to 50 J to effectively weld the 0.2 mm thick pure nickel strip [19]. This method ensures a secure and reliable connection without compromising the integrity of the battery cells.

The Battery Management System (BMS) plays a crucial role in ensure safe operation of the battery pack. It monitors the state of charge for each battery series, while providing protection against overcharging, over-discharging and short circuits. For this project, a 3s BMS was selected to match the 3 in series, 1 in parallel (3s1p) configuration of the battery pack. This ensures that each cell in series is balanced and operates withing safe voltage and current limits, thereby enchaining the longevity and reliability of the battery pack.

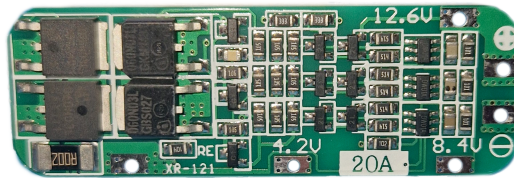


Figure 3.8: Battery Management System (BMS)

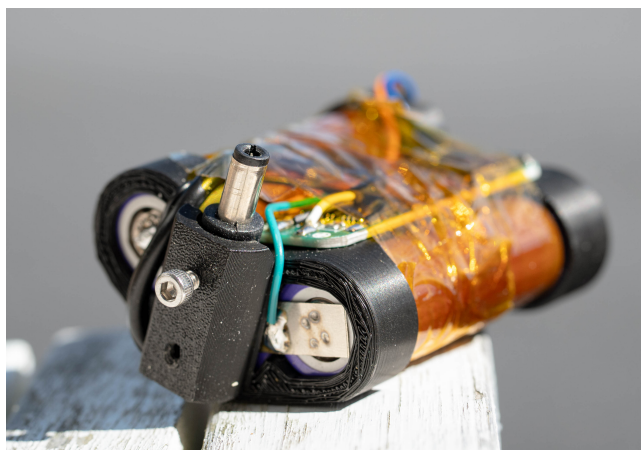


Figure 3.9: Finished battery pack assembly with three 18650 batteries in series, welded with pure nickel strips and soldered to a Battery management system for safe charging and discharging.

3.3 Data Acquisition and Preparation

3.3.1 Muscle Sensor

The data for this project was acquired using surface electromyography (sEMG), a non-invasive method for detecting electrical signals generated by muscles [20]. The team utilized Sparkfun Myoware 2.0 sensors ² to interpret the signals from the electrodes. These affordable sensors, available since 2015, are widely used in various research studies [21], making them ideal for this project. The Myoware sensors feature an instrumentation amplifier with a gain of 200x, low power consumption, and a high signal-to-noise ratio (SNR). Additionally, they include a bandpass filter with a low-pass cutoff frequency of 20Hz and a high-pass cutoff frequency of 498Hz [21].

For the project setup, three Myoware sensors were fitted together inside a 3D-printed enclosure, as illustrated in figure 3.10. This enclosure ensured stability and allowed the entire assembly to be easily transferred from the test setup, as explained in section 3.3.2, to the forearm of the prosthesis as illustrated in figure 3.4a.

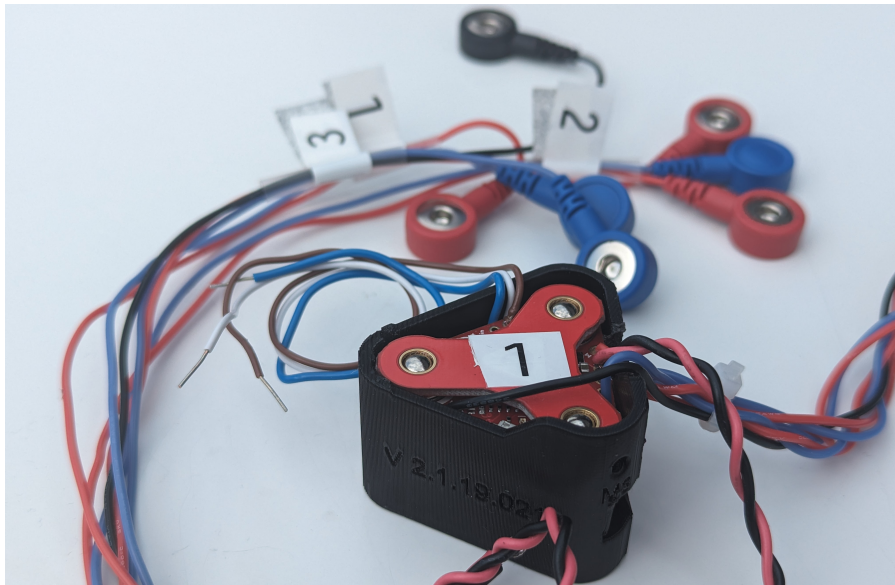


Figure 3.10: 3D-printed enclosure housing three Sparkfun Myoware 2.0 sensors, including necessary cables, used for detecting sEMG signals. This setup ensures stability and allows easy transfer between the prosthesis forearm and test setup.

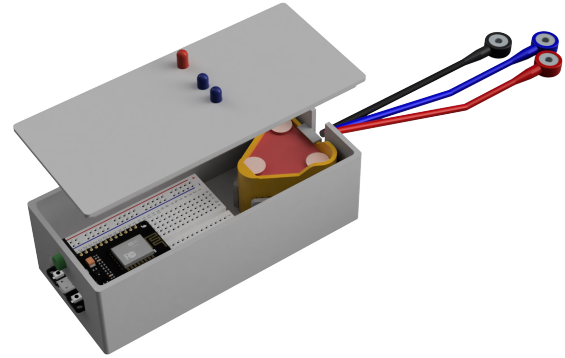
3.3.2 Data Collection

For collecting the data it was used three sensor pairs with a common ground strategically placed on the forearm in combination with an ESP32, running at a resolution of 12bit and a sampling rate with approximately 1938Hz. The program running the collection procedure toggles between two modes based on a 5 second countdown for each. In the gesture/activity mode, the participants are being informed to relax after a period of 4 seconds so the dataset may include a labelled data where the gesture goes from active to rest position. The system, as depicted in figure 3.11, connects to a PC by utilizing a USB cable where the data captured are stored in separate files, each suitably labelled for clarity for each participant, gesture and position.

²<https://myoware.com/products/muscle-sensor/>



(a) Data collecting done on a participant



(b) CAD design for the data collecting encapsulation

Figure 3.11: Overview of the data collecting system. The box contains all the necessary hardware for collecting data from participants without needing anything else than a laptop and USB cable

3.3.3 Preparation and placement

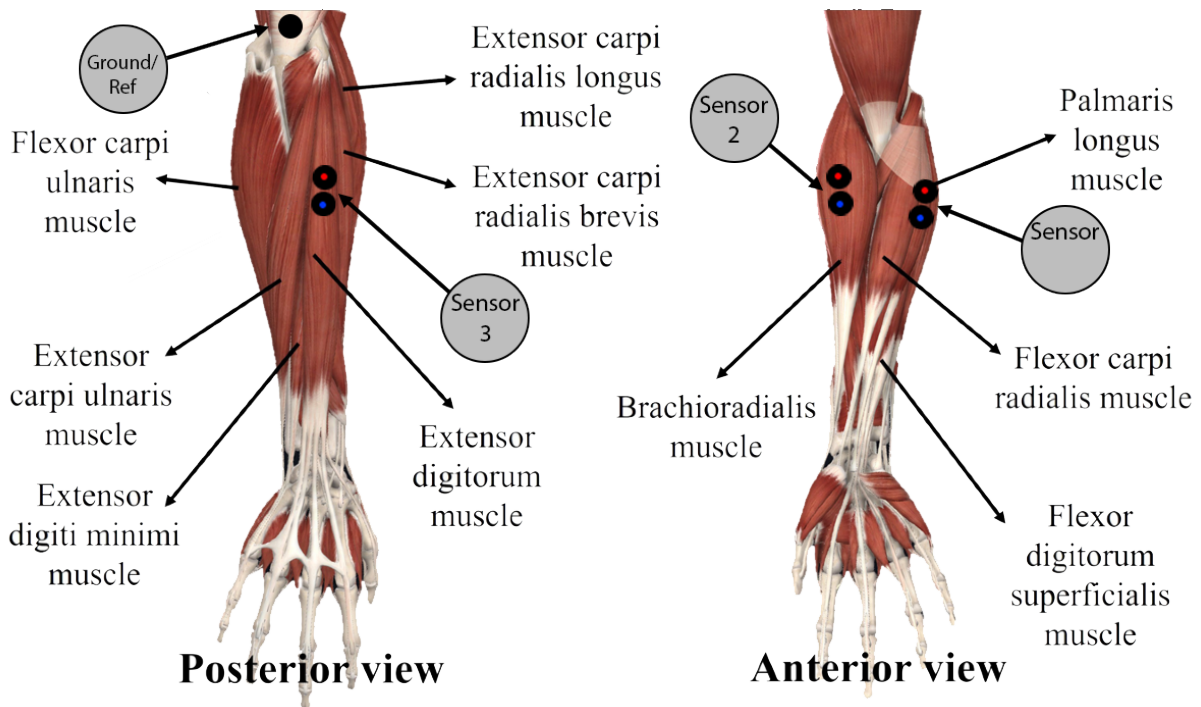


Figure 3.12: sEMG placement: The rough placement of these electrodes was first defined by following the guidelines from the surface EMG for non-invasive assessment of muscles project [22], in combination with earlier studies [23]. Then, the placement was fine-tuned by following an empirical approach for finding the optimal placement within that area. - Image: Modified from [23]

As illustrated in figure 3.12, the three sensors are placed respectively on the palmaris longus muscle, brachioradialis muscle, and extensor digitorum muscle. In addition, the reference/-

ground electrode are placed on the elbow. Given a total of seven disposable Ag/AgCl electrodes (diameter: 24mm)³, in a bipolar configuration, used for each participant. The rough placement of these electrodes was first defined by following the guidelines from the surface EMG for non-invasive assessment of muscles (SENIAM) project [22], in combination with earlier studies [23]. Then, the placement was fine-tuned by following an empirical approach for finding the optimal placement within that area. Before applying the different sensor it is crucial to ensure good contact between the electrodes and the skin. Here the skin preparation guide from SENIAM [22] was followed. Skin surface was first cleaned with razor to remove any hair, thereafter cleaned with either alcohol or non-perfumed soap and dried before the electrodes were placed. After connecting the sEMG sensors to the electrodes on the skin, the setup was firmly secured with sports tape, as depicted in figure 3.11, for minimizing any movements of the sensors or cables that could disturb the measurement with low frequency noise [20].

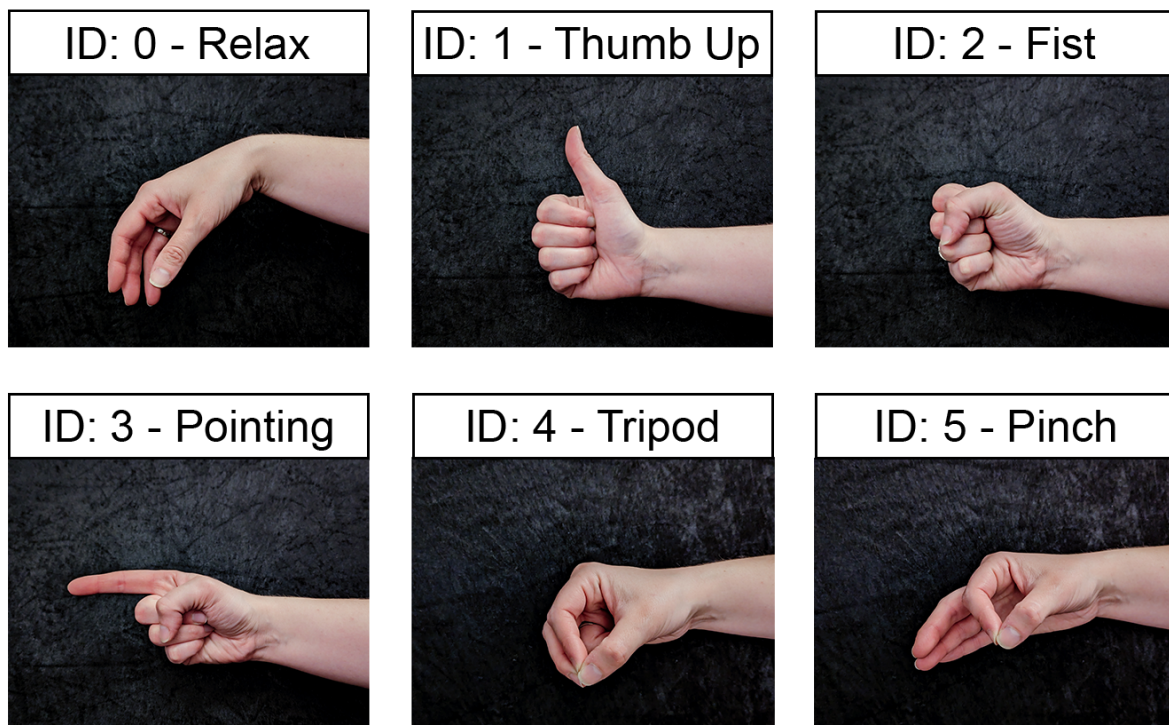


Figure 3.13: Overview of the different gestures with corresponding label used in the project. The unique ID helps the supervised machine learning to classify each gesture

The choice of the gestures is based on different research done in the field [24], [4], as well as the restriction of the prototype hand when it comes to mechanical design. With this approach it was concluded to include a total of five gestures in addition to the resting position. The gestures, as depicted in figure 3.13, are labeled with each own unique ID in the the dataset to help classify each gesture in the machine learning model. For the collection, each gesture is repeated 20 times in three different position (in total 60 repetition for each gesture pr subject). For later identification of the data, each position are given an unique position (POS) as described in the list below:

- Standing, hand hanging straight down - POS: 1
- Standing, forearm parallel to the floor - POS: 2
- Sitting, forearm resting on a table - POS: 3

³<https://cdn.sparkfun.com/datasheets/Sensors/Biometric/H124SG.pdf>

The data was collected by six healthy participants, all right-handed, as listed in the table 3.1 below. Each participant was guided through a step-by-step guideline, as described closer in appendix E, and was given time to practice on the different gestures and methods before the actually data collecting took place.

ID	Gender	Age
01	M	39
02	M	24
03	M	45
04	F	37
05	M	24
06	M	25

Table 3.1: Overview of the 6 different participants included in the dataset, with corresponding unique ID.

3.4 Signal Processing and Software Design

3.4.1 Data Pre-processing

When it comes to EMG signal processing and machine learning for hand gesture recognition, it is crucial to segment the EMG signal into windows for further analysis. The selection of an appropriate windowing technique is essential to ensure accurate feature extraction and classification. Overlapping and non-overlapping windowing techniques are commonly used, with overlapping windows providing a balance between classification accuracy and delay in decision-making [25]. After the EMG onset is identified, a window is chosen to capture the specific EMG data for the gesture. This window serves as a segment of the signal where various signal processing and feature extraction methods are applied to extract descriptive features for subsequent machine learning analysis. The choice of window length is critical in real-time applications to balance classification accuracy and system responsiveness. Larger window lengths generally lead to higher classification accuracy, but may introduce delays in the classifier’s decision-making process [25].

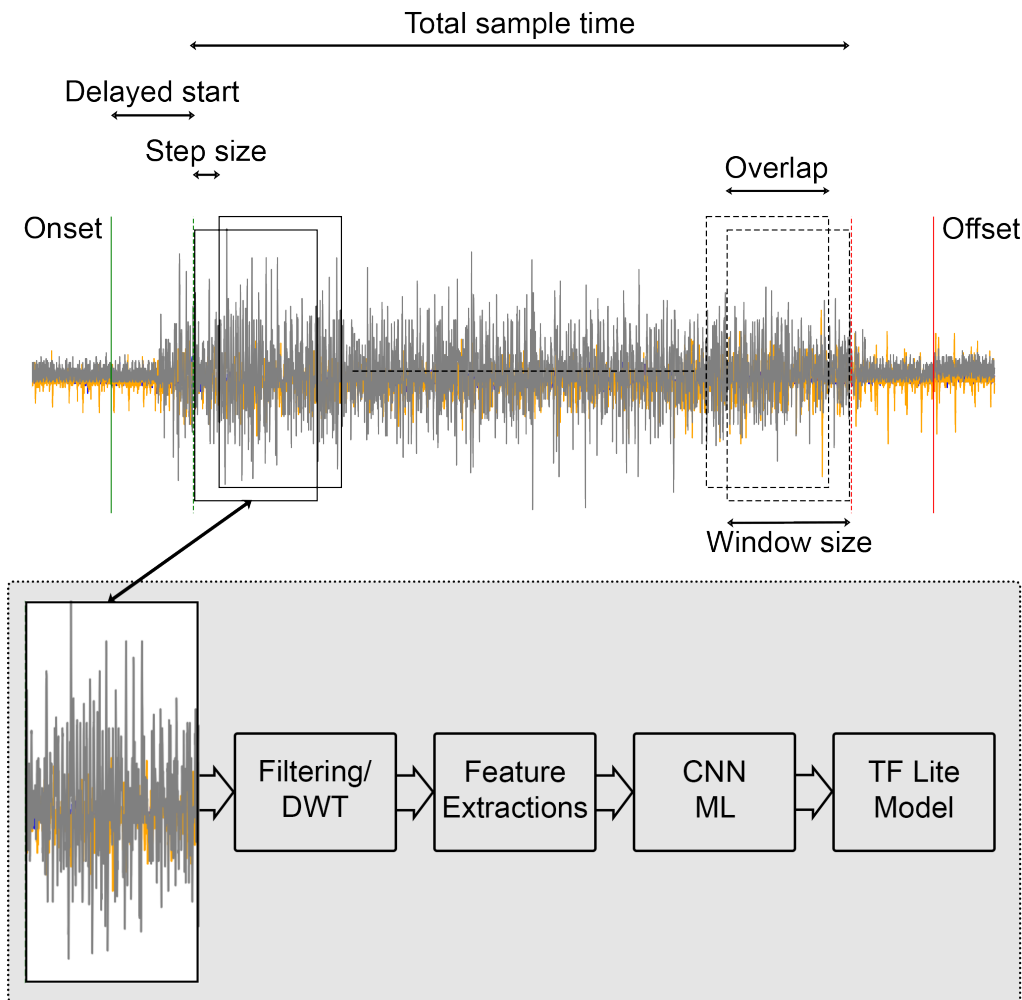


Figure 3.14: EMG signal processing workflow. The signal is segmented into 300-millisecond windows with a 50-millisecond step size over a 4-second total sample time, ensuring no critical information is lost and sufficient data is generated for the machine learning model.

Figure 3.14 illustrates the EMG signal processing workflow employed in this thesis. After a 500ms delayed start following the onset detection, which accounts for human reaction time of about 235ms [26] and allows for some extra time to capture the most relevant gesture signals, the signal is segmented into 300-millisecond windows with a step size of 50 milliseconds, totaling 4 seconds of sample time. This delayed start ensures that no critical information is lost at the window boundaries, generating sufficient data for the machine learning model.

Ensuring high-quality data acquisition is fundamental in EMG signal analysis, making proper filtering an essential step. This phase crucially includes the use of notch and bandpass filters to enhance signal clarity and reliability of the underlying muscle activity [4].

Specifically, a notch filter set at 50 Hz is deployed to eliminate power line noise. This filter standardizes the baseline across different recording windows, ensuring consistent data quality. Furthermore, to capture only the relevant frequencies, a bandpass filter is utilized. This filter isolates the crucial portions of the EMG signal pertinent to muscle activity analysis by excluding lower frequencies that may contain slow movement artifacts and higher frequencies prone to high-frequency noise.

In this context, a 4th order Butterworth bandpass filter, spanning a frequency range of 20

to 350 Hz, was implemented. This decision was guided by empirical data testing and supported by existing research in the field [4], [27].

The Discrete Wavelet Transform (DWT) is employed to extract key features from the sEMG signals for hand gesture recognition, adopting the approach from earlier research [4] using db10 wavelet functions and 4 levels of decomposition. By applying DWT to a 300-millisecond window, the signal is analyzed at multiple scales, providing a multi-resolution representation of the gesture. This method captures both time and frequency domain information, allowing for further feature extractions across different frequency sub-bands. The DWT coefficients highlight the energy patterns unique to each gesture, with lower frequency bands offering insights into overall trends, and higher frequency bands capturing detailed, transient activities. This multi-resolution approach enhances the feature space for subsequent classification tasks, making it possible to distinguish subtle differences in hand gestures effectively.

3.4.2 Feature Extraction

Feature extraction was implemented utilizing three key metrics: integrated EMG (iEMG), mean absolute value (MAV), and root mean square (RMS) [12]. Initially, iEMG was applied to the filtered raw EMG signal, which empirically gave robust results. Subsequently, MAV was calculated using the normalized and filtered signal to ensure consistent amplitude representation across the data set.

Further extraction was performed on the normalized signal processed through DWT across all four decomposition levels, including the last approximation coefficient. This process was thoroughly applied to iEMG, MAV, and RMS metrics, enabling a detailed multi-scale analysis of the EMG signals.

In total, as depicted in figure 3.15, each sensor generated 17 distinct features, resulting in a final feature extraction output shape of 17 features across 3 sensors for the machine learning model.

Features extracted from window: 20000 of 108000
Labeled gesture for window is: 4

	Sensor1	Sensor2	Sensor3
0	671.7888869128268	3830.3795640099966	4924.56432906083
1	0.7678847015363592	0.7594335554748345	0.7751539152949191
2	109.27512546055476	70.83930710964088	70.94882293558697
3	2.706102719934918	1.691344194192832	1.659987121118123
4	2.0236134344547176	1.3118390205489052	1.3138670913997588
5	82.98165886861807	129.34819750983127	109.35254408315551
6	2.0331774019977362	3.006847247533634	2.575274976648646
7	1.5366973864558902	2.395336990922801	2.025047112651028
8	105.8003935912765	113.80161787549815	86.7664617474167
9	1.5287014521054951	1.616935154514185	1.2285351273503204
10	1.1887684673177135	1.278669863769642	0.9749040645777157
11	99.86608063939377	55.293970959983135	47.846099667131995
12	0.8282265870117937	0.4332196450323334	0.40109828711117673
13	0.628088557480464	0.3477608236476927	0.30091886583101884
14	13.67474500103845	5.359512807847353	3.9911650857303274
15	0.07118973710643035	0.023988091214822684	0.017144916020978774
16	0.045582483336794835	0.01786504269282451	0.01330388361910109

Figure 3.15: Feature extraction matrix: Features extracted from window 20000 of 108000. The table displays 17 distinct features generated by each of the three sensors used in the thesis. These features include integrated EMG (iEMG), mean absolute value (MAV), and root mean square (RMS), processed through both the RAW signal and the Discrete Wavelet Transform (DWT) across all four decomposition levels. The labeled gesture for this window is 4, which correspond to the tripod grasp.

3.4.3 CNN Development

To obtain a successful gesture classification, a one-dimensional convolutional neural network (1D-CNN) model was chosen for its performance in sequential data handling and feature extraction performance [4]. TensorFlow and Keras, popular open source ML development libraries, were utilized to increase the coding efficiency through a variety of pre-built functions, debugging tools and model test functionalities [28]. The model was iteratively developed, constantly adding layers and functionality. To do this a combination of studies, educated guesses and, mostly, trial and error was used. By reducing the amount of data, batch sizes and other factors affecting the processing time, this method was quite effective in optimizing the model.

The CNN architecture consists of 1 input layer, three 1D convolutional layers, 1 flattening layer, 1 dense layer and 1 dense output layer. Table 3.2 showcases the CNN operations and their respective configurations.

- **Input layer:** The input layer is the first order of operation in the CNN model architecture and processes the 17 features per sensor.
- **Convolutional layers:** The convolutional layers utilize 64 and 128 filters and capture different features, such as sudden changes in the signal. Rectified Linear Unit (ReLU) is chosen as the convolutional activation. To alleviate overfitting, L2 regularization was also implemented.
- **Flattening layer:** After the convolution, a flattening layer was added to convert the data into a single vector. This was necessary as the fully connected dense layer requires a single dimension data shape.
- **Dense layer:** The first dense layer is configured with 128 units and a ReLU activation.
- **Output layer:** The dense output layer is defined with a softmax activation and 6 units, reflecting the 6 different classifications at hand.

Layer	Configuration Details
Input layer	Input shape (17,3)
Conv1D layer 1	Kernel size: 2, Filters: 64, Activation: ReLU, Padding: same, L2 regularization: 0.015
BatchNormalization	
MaxPooling1D	Pool size: 2, Strides: 2, Padding: same
Conv1D layer 2	Kernel size: 2, Filters: 128, Activation: ReLU, Padding: same, L2 regularization: 0.015
BatchNormalization	
MaxPooling1D	Pool size: 2, Strides: 2, Padding: same
Conv1D layer 3	Kernel size: 2, Filters: 128, Activation: ReLU, Padding: same, L2 regularization: 0.015
BatchNormalization	
MaxPooling1D	Pool size: 2, Strides: 2, Padding: same
Flatten layer	
Dense layer	Units: 128, Activation: ReLU
Dropout	Rate: 0.4
Dense output layer	Units: 6, Activation: softmax

Table 3.2: CNN layers, functions and configurations for the 1D-CNN used in this thesis.

The compilation and of the model includes several steps to optimize the process. Firstly, the dataset was split into 60% training, 20% validation and 20% test data. The test data was used to asses the model performance in a TensorFlow lite format, which will be discussed later. The compilation function of the model was configured with the Adam Optimizer and Sparse Categorical Crossentropy was utilized as the loss function.

In the learning phase of the model, a batch size of 128 was found to be ideal to balance the computational efficiency and model memory usage. To mitigate overfitting and time-waste during learning, a early stopping callback of was implemented to the process. This makes the number of epochs dynamical and terminates the training if the validation loss does not improve over 10 epochs. Furthermore, a model checkpoint callback was also integrated in the training process, simply always saving the best model performance during training. Lastly, a learning rate reduction callback was implemented. This reduces the learning rate as the validation loss does not improve.

As the CNN was supposed to be deployed on a MCU for real-time classification, a essential process was the conversion of the model to a TensorFlow Lite (TFLite) format. This allows for model compression and low resource consumption, ideal for the embedded environment. Converting the model involves several key steps to ensure the model does not loose efficiency when being compressed. By utilizing the TFLite default optimization and quantization configurations, the model was saved in a TFLite format as shown in figure 3.16.

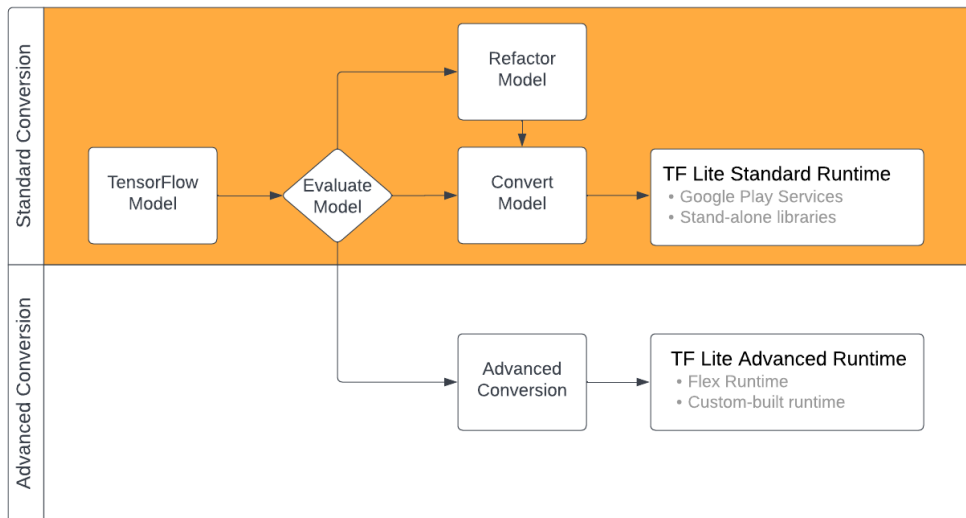


Figure 3.16: TFLite conversion workflow [29]

3.4.4 Model Validation

Model validation for 1D CNNs applied to EMG data involves assessing the model’s performance using metrics like confusion matrix, model accuracy, and loss as well as evaluation metrics like F1 score, trainable parameters, and average inference time [30]. The confusion matrix helps in understanding the model’s predictive capabilities by showing the true positive, true negative, false positive, and false negative predictions [31]. Additionally, model accuracy indicates the proportion of correct predictions made by the model, while model loss quantifies how well the model is performing during training, with lower values indicating better performance.

The F1 score provides a comprehensive measure by giving the weighted average of precision and recall. The closer its value is to 1, the better the model's ability to classify accurately. Trainable parameters refer to any parameter in the model that is learned during the training process. The number of trainable parameters directly impacts the training duration, with more parameters generally requiring longer to learn. This aspect is critical for understanding the scalability and efficiency of the model in practice [30].

Average inference time, the duration it takes for the model to produce classification results for some test input data, is termed average inference time. For real-time applications, it is crucial that the average inference time remains low, ensuring that the model can deliver prompt results suitable for on-the-fly decision-making [30]. This measure is a key factor in the deployment of models in real-world scenarios, where delays can be detrimental to the application's effectiveness.

3.5 Microcontroller (MCU) Programming

3.5.1 STMicroelectronics (STM) Ecosystem

The STM32 ecosystem is a comprehensive suite of tools and software designed to facilitate the development and programming of STM32 microcontrollers (MCUs). Central to this ecosystem is STM32Cube (Cube), an extensive set of software components, utilities, and middleware that simplify the development process. STM32CubeMX (CubeMX) is a graphical configuration tool that allows developers to initialize peripherals, configure clock settings, and generate initialization code tailored to specific STM32 MCUs. This tool streamlines the setup process, ensuring that developers can quickly get started with their projects.

Complementing CubeMX is STM32CubeIDE (CubeIDE), an integrated development environment that combines project management, code editing, compiling, and debugging capabilities. CubeIDE supports a range of programming languages, including C and C++, and integrates seamlessly with other Cube software components.

Given the choice of the MCU as described in section 3.2.2, the STM32 ecosystem is an ideal fit for our project. The comprehensive tools and software provided by Cube, including CubeMX for configuration and CubeIDE for development, streamline the setup and programming processes. The ecosystem's extensive middleware libraries and example projects further enhance our ability to implement advanced functionalities efficiently. Additionally, the inclusion of the X-CUBE-AI expansion pack facilitates the seamless deployment of machine learning models, aligning perfectly with our need for real-time neural network inference.

3.5.2 X-Cube AI Expansion Pack

Building on the strengths of the STM32 ecosystem, X-CUBE-AI ⁴ is a great expansion package for CubeMX, enabling the integration of AI algorithms into STM32 MCUs. This package allows for the deployment of pre-trained neural network models, making it ideal for applications like anomaly detection, image recognition, and predictive maintenance.

In conjunction with TFLite, X-CUBE-AI converts TensorFlow models into a format executable on STM32 MCUs. The workflow includes training a neural network with TensorFlow, converting it to TFLite, and using X-CUBE-AI to generate optimized C code. This ensures efficient execution of neural networks within the resource constraints of an STM32 MCU.

⁴<https://stm32ai.st.com/stm32-cube-ai/>

X-CUBE-AI also provides tools for model analysis and validation directly on the MCU, allowing for testing the model speed and performance in real-time for further analysis.

3.5.3 Actuator control

To control the motors accurately a Proportional Integral Derivative (PID) controller can be used. This controller functions by dynamically changing the motor control input signal based on a error computed from a setpoint and the feedback. The control signal is computed using the PID formula [32]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.1)$$

Where $u(t)$ is the control signal, $e(t)$ is the error between the full flexion position and current position given by the encoders, K_p is the proportional gain, K_i is the integral gain and K_d is the derivative gain [32]. The closed control loop is visualized in Figure 3.17.

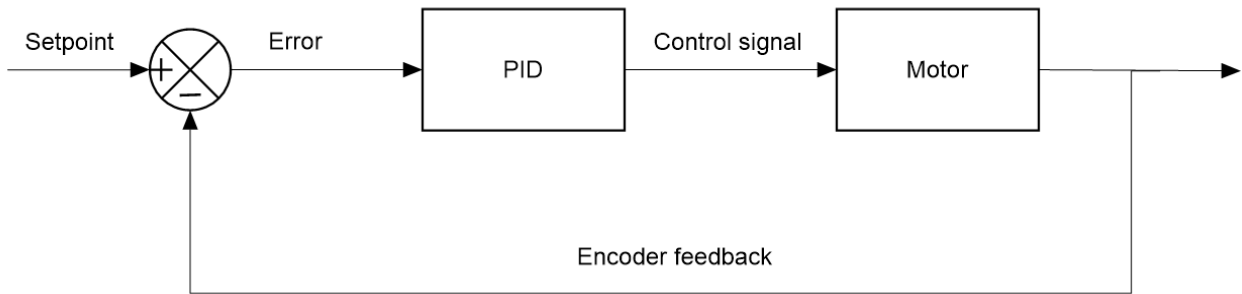


Figure 3.17: PID control loop.

To enable dynamical grasping a current threshold can be implemented using the current sensors. This way, the PID control can be terminated and the motors can be stalled at the present PWM value when the current threshold is exceeded. When a resting position signal is received the PID activates again and regulates towards the given setpoint. However, due to time constrains, PCB short-circuiting and general MCU issues, the control system was not implemented on the MCU.

Chapter 4

Evaluation of Critical Functionality

This chapter presents the main findings of the thesis and the culmination of our approach. By creating a robust testing framework and by presenting detailed statistics, the results aims to showcase the potential the prosthesis and add valuable information to the research field. This includes a detailed evaluation of the function quality, real-time software performance, anthropomorphic properties and accessibility. The final prototype utilized for evaluation can be viewed in figure 4.1.



Figure 4.1: Final prototype of prosthesis powered on in resting position

4.1 Assessment of Anthropomorphic Appearance

4.1.1 Motivation

Research has proven that an anthropomorphic design in myoelectric prostheses can increase users' acceptance rates [33]. Structure, color, weight, and general appearance all contribute to the feeling of having a real hand, and is therefore important and evaluated.

4.1.2 Method

Each gesture of the prosthesis is visually compared to a biological hand to assess the likeness. Furthermore, in table 4.1, a detailed comparison of the dimensions, weight, and DOFs of the hands is showcased. The hand used in this test is the hand of the team member from which the prosthesis has been designed. Therefore, the dimensions of the prosthesis can not be compared to average hand sizes. The dimension is simply measured as the length from the tip of the middle finger to the bottom of the forearm, and the width is measured from the widest point of the thumb to the widest point of the little finger. The weight is determined as a percentage of body weight stated in research conducted by Ramachandran et al. [34].

4.1.3 Results

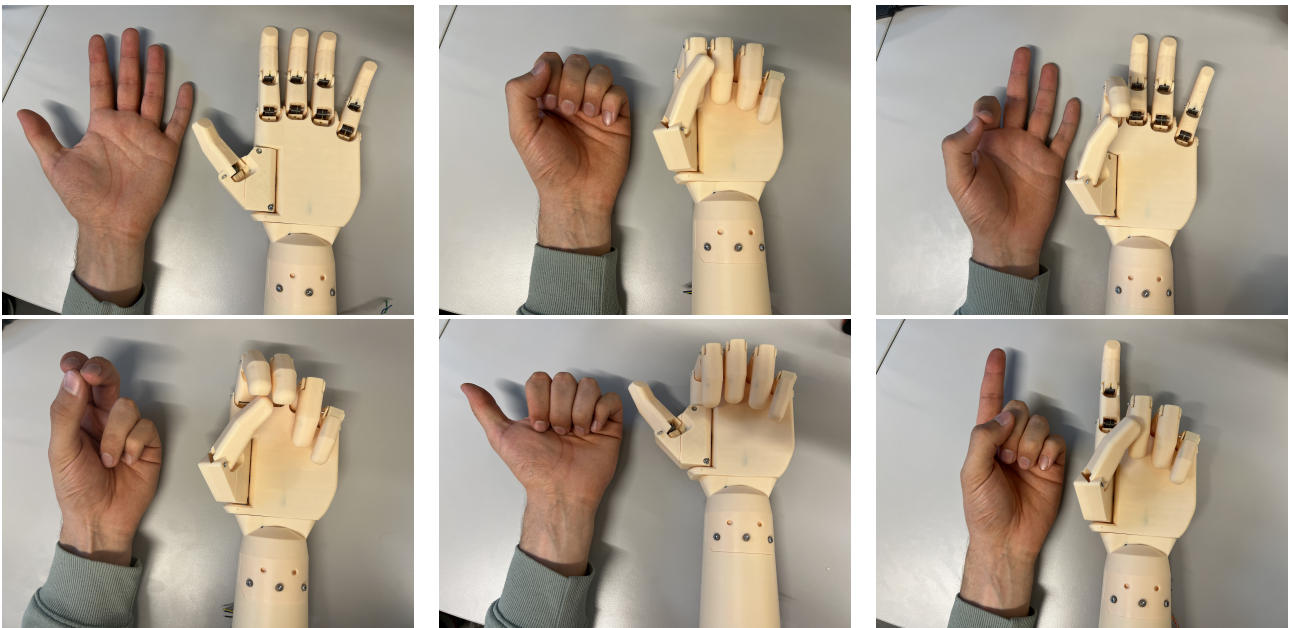


Figure 4.2: Gesture comparison with biological hand

As seen in figure 4.2, the gestures in the MMHP prove quite anthropomorphic when compared to the biological hand. Even though individual joint control is not implemented and the DOFs have been significantly reduced, the mechanical design and tendon drive actuation system provides human-like characteristics. Moreover, in table 4.1 the comparison of number of joints, number of DOFs, weight, and dimensions are made. While the weight may seem to be low in the prosthesis, it is important to note that the muscles contributing to the high weight in biological hands is used to carry the weight itself. When wearing a prosthesis, the residual limb has to carry the entire weight itself. The discrepancy seen in the number of joints and DOFs are mainly due to the disregarding of the finger DIP joints, not including wrist articulation and not including individual joint control.

	Biological hand	Prosthesis
Weight	2132g [34]	795g
Number of DOFs	27 [35]	6
Number of joints	18 [36]	10
Dimensions	420mm x 178mm	423.9mm x 179.4mm

Table 4.1: Comparison of anthropomorphic properties.

4.2 ADL Performance Evaluation

4.2.1 Motivation

Testing Activities of Daily Living (ADLs) are essential when evaluating the functional performance of a prosthesis [37]. The Southampton Hand Assessment Procedure (SHAP), is a testing framework created to evaluate the performance of prosthetic hands and its users in ADLs [38]. This method was chosen as the function assessment in this thesis due to it being widely adopted by researchers as well as it being well-validated and thoroughly documented [39]. However, this method requires a specific kit and data, which has not been obtained for this test due to pricing and availability reasons. Therefore, a modified version of this procedure will be done.

4.2.2 Method

The evaluation method is two parted. The first procedure is an evaluation of the specific grasping functions of the MMHP. This entails gripping and moving abstract objects reflecting specific grip patterns. However, due to time constraints and PCB issues, this will not be evaluated in this instance. The second procedure is originally a test of 14 different ADLs using a variety of objects and equipment:

- **Pick up coins:** Drag coins across table and into a jar.
- **Button board:** Undo buttons of shirt.
- **Simulated food cutting:** Use knife to cut plasticine into two pieces.
- **Page Turning:** Simulated page turning by picking up cards, flipping. them and placing them down again.
- **Jar lid:** Open jar lid using both hands.
- **Glass jug pouring:** Pour 100ml water from glass jug.
- **Carton Pouring:** Fill carton with 200ml of water.
- **Lifting a heavy object:** Lift full jar of water over the carton and place down on table.
- **Lifting light object:** Lift empty can over carton and place down on table.
- **Lifting a tray:** Lift tray and place on edge of the table.
- **Rotate key:** Turn key in keyhole.
- **Open/Close zip:** Open and then close a zipper.
- **Rotate a screw:** Pick up screwdriver and rotate a screw.
- **Open door handle:** Open up a door using door handle.

In this evaluation, the three grasping functions of the MMHP will be tested five times each on a chosen SHAP ADL task reflecting the type of grasp. As the part of the test, the other task were neglected mainly due to PCB issues. The pinch grasp will be tested using the open/close zip ADL, the fist grasp was tested using the lift heavy object task and the tripod grasp was tested using the rotate key test. The pinch test is determined successful if the grasp can pull the zipper of a sweater down and up again without losing the grip. The fist grasp is determined as successful if it can hold the heavy object, in this instance, a half full 0.5L bottle, for 10 seconds without dropping it. Lastly, the tripod grasp task is evaluated as a success if it can hold the key and rotate it to open the lock. Additionally, a medical glove was cut and threaded on the fingers and thumb of the prosthesis to increase the friction and grip function.

4.2.3 Results

As seen in table 4.2, the results are quite promising. The fist grasping performs well with a 100% success rate in its respective task. However, the MMHP struggled the most in the rotation of a key using the tripod grip. The rotation of the key in the keyhole was too hard in comparison to the grip strength unless if the grasp of the key itself was optimal. Lastly, the opening and closing of the zipper was quite effective, with it grip only failing in one instance. The grasping of the different ADL related objects can be seen in figure 4.3.

	Open/close zipper	Lift flask	Rotate key
Trial 1	Success	Success	Success
Trial 2	Success	Success	Fail
Trial 3	Fail	Success	Success
Trial 4	Success	Success	Success
Trial 5	Success	Success	Fail

Table 4.2: Results in selection of SHAP ADL tests.

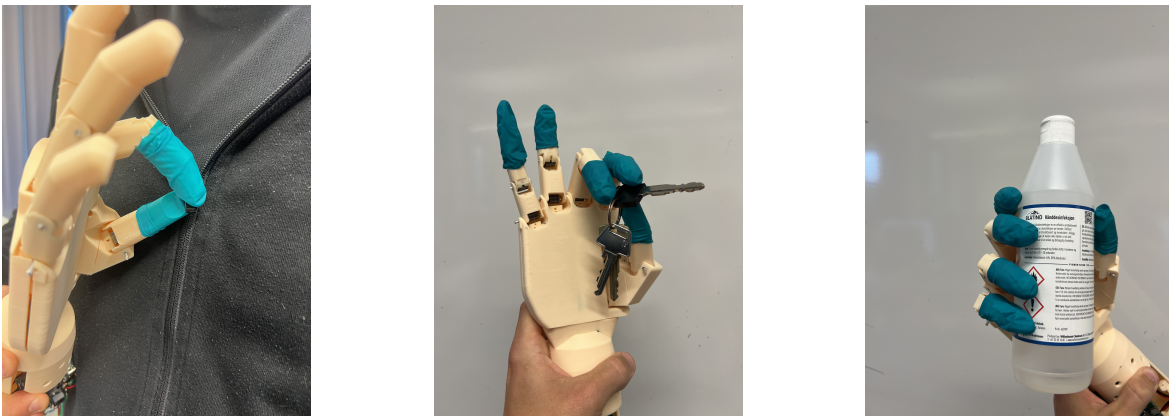


Figure 4.3: Grasping objects from selected SHAP ADL tasks.

4.3 Assessment of CNN Off-Line Performance

4.3.1 Motivation

In order to ensure the robustness and reliability of the CNN model, it is essential to evaluate its performance in an off-line environment before deploying it in real-time applications. This step allows for thorough testing and validation of the model’s capabilities, ensuring that it meets the necessary criteria for accuracy, efficiency, and effectiveness.

4.3.2 Method

After the execution of the data preparation steps outlined in section 3.4.1, the machine learning model was fed a substantial total of 108,000 data windows. These were evenly distributed among five distinct gestures, thereby establishing a robust and comprehensive dataset optimized for rigorous training and thorough evaluation [40].

4.3.3 Results

In the evaluation of the CNN model, several key metrics were examined to determine the efficacy and feasibility of the model for real-time applications. Firstly the F1 score achieved was 0.86, indicating a high level of precision and recall in the model's classification ability; this score suggests that the model is robust in distinguishing between different classes with a good balance between sensitivity and specificity. Regarding the model's complexity, it contains 100,550 trainable parameters, which is relatively moderate for deep learning models in EMG applications [30]. This parameter count suggests a balance between model complexity and computational efficiency, ideal for models requiring real-time processing. Most notably, the average inference time (AIT) was recorded at 0.124 milliseconds per sample. This exceptionally low inference time is critical for real-time applications, where quick and accurate responses are essential. The combination of these metrics demonstrates that the model not only performs well in terms of accuracy but is also efficient, making it suitable for deployment in environments where rapid processing of EMG data is required.

Secondly the model's performance during training and validation is illustrated in figure 4.4. As shown, the accuracy steadily increased across epochs, indicating effective learning and generalization capabilities. Both the training and validation accuracy's converge at around 86%, suggesting a well-balanced model that avoids over-fitting. Similarly, the loss plots reveal a significant decrease in both training and validation loss, stabilizing at low values which reflects the model's increasing reliability in making precise predictions

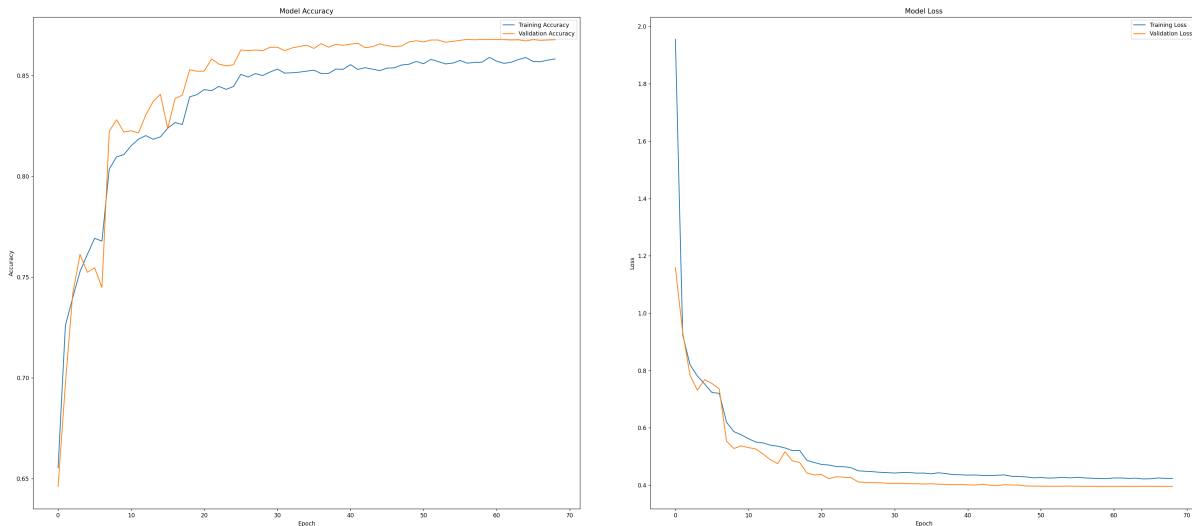


Figure 4.4: Model accuracy and loss plots illustrating the CNN model's accuracy steadily increased across epochs, converging at around 86%. This indicates effective learning and a well-balanced model that avoids over-fitting. The loss plots show a significant decrease in both training and validation loss, stabilizing at low values, reflecting the model's reliability in making precise predictions.

And at last to further assess and benchmark the TFLite model, the processed and quantized TFLite file was analyzed using Python's sklearn library to generate a confusion matrix. The

confusion matrix, a vital tool in machine learning, helps to visually summarize the performance of a classification model, providing insight into the accuracy of predictions against the true labels [31]. This rigorous evaluation yielded an overall accuracy of 85.85%, as detailed in figure 4.5. Notably, each gesture achieved over 80% accuracy, with the pinch grasp emerging as the most dominant gesture, excluding the resting gesture. This robust performance establishes an excellent baseline prior to deployment in a real-time application, showcasing the model’s potential efficacy and reliability.

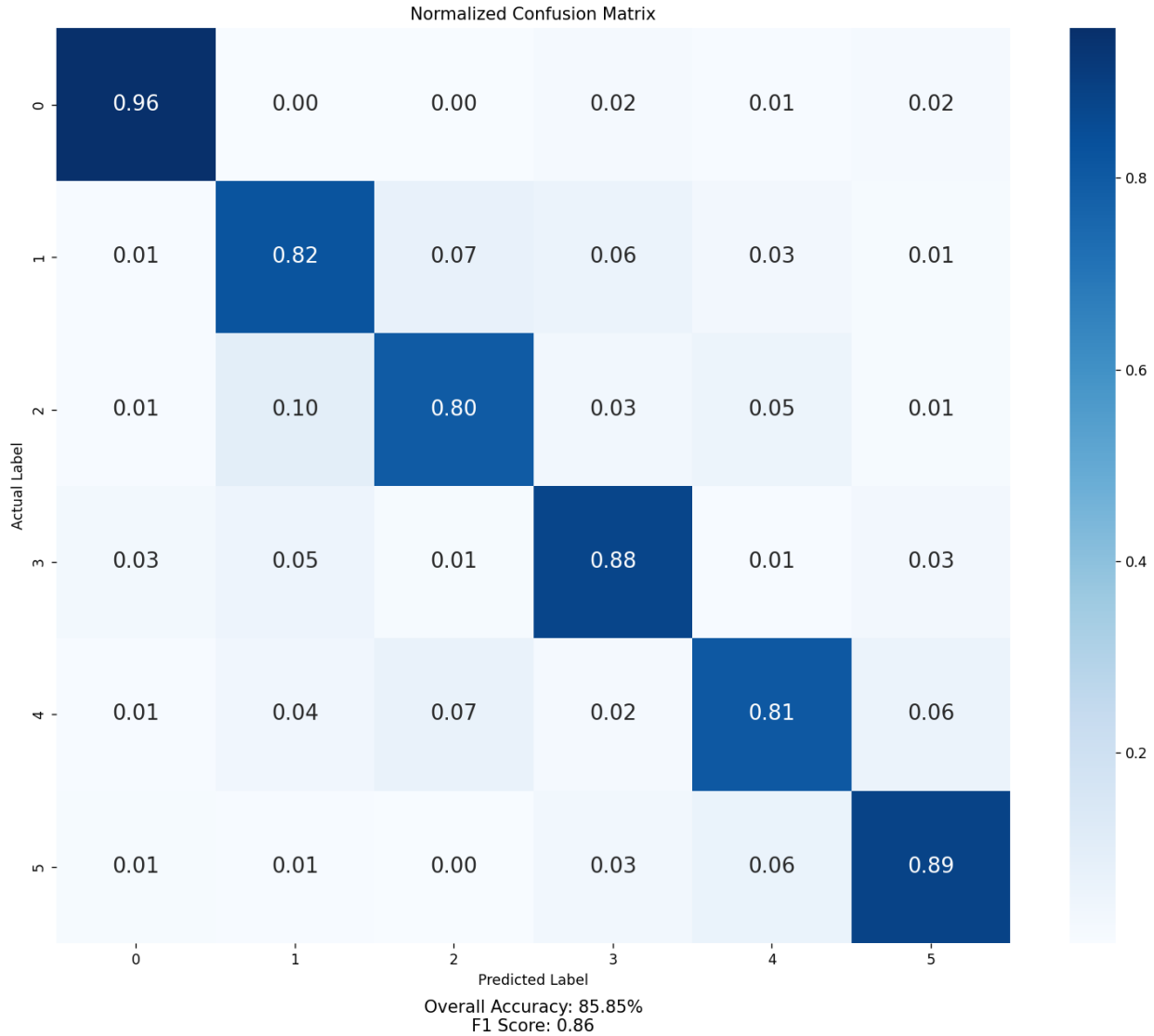


Figure 4.5: Confusion Matrix evaluation yielded an overall accuracy of 85.85%, with each gesture achieving over 80% accuracy. The pinch grasp emerged as the most dominant gesture, excluding the resting gesture. This robust performance establishes an excellent baseline for real-time application deployment, showcasing the model’s potential efficacy and reliability.

4.4 Assessment of CNN Real-Time Performance

4.4.1 Motivation

Evaluating the CNN model in a real-time environment is crucial to ensure its practical applicability and reliability. This step verifies that the model can handle live data streams efficiently, maintaining accuracy and performance under real-world conditions.

4.4.2 Method

For testing and verifying the model in a real-time environment, the system previously used for data collection in section 3.3.2 was adapted to continuously stream data via serial communication to a local PC, maintaining the same sampling frequency of 1938Hz for each sensor. The testing was conducted with Python 3.11 on a Lenovo laptop equipped with Windows 11 64-bit, 32GB RAM, and an Intel i7-1260P processor running at 3.4GHz. The filtering, preprocessing, and feature extraction processes were identical to those used during data collection, including the same window size. To mimic real-life usage of the prosthetic hand, the TFLite model developed during the training phase was imported using the TensorFlow Python module.

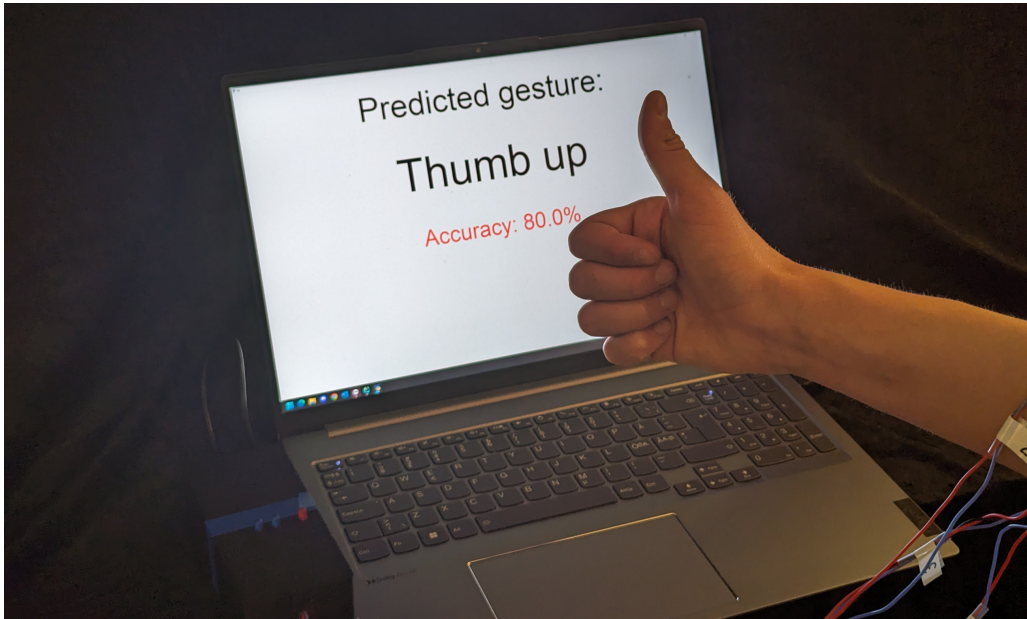


Figure 4.6: Real-time gesture prediction system displaying the predicted gesture and its corresponding confidence level, effectively documenting the accuracy of the real-time CNN model.

A structured test involving four participants was conducted to evaluate the efficacy of the real-time gesture prediction system. Participants 1 and 2, unfamiliar with the machine learning model and methodology, represented novel data inputs, while Participants 3 and 4 had their data included in the initial training dataset. Participants were instructed on how to perform five different gestures and how to grasp an object using both pinch and tripod techniques. As depicted in Figure 4.6, participants received real-time feedback on a screen, displaying both the gesture predicted by the model and the associated accuracy. The system evaluates the reliability of these predictions by using a prediction buffer, which stores a sequence of recent predictions. To calculate the confidence level of the most frequently predicted gesture within this buffer, the system determines how often this gesture appears relative to the total number of predictions in the buffer. This frequency is then expressed both as a percentage and as text corresponding to the gesture. As illustrated in figure 4.7 if the buffer has 10 slots and the same gesture is predicted 8 times out of those 10, the confidence level for that gesture would be 80%. By continuously updating this buffer with new predictions and calculating the confidence level, the system provides an ongoing assessment of prediction stability. This method enhances the clarity of real-time results by smoothing out transient mispredictions, ensuring that the displayed gesture and accuracy are representative of the user's actual movements.

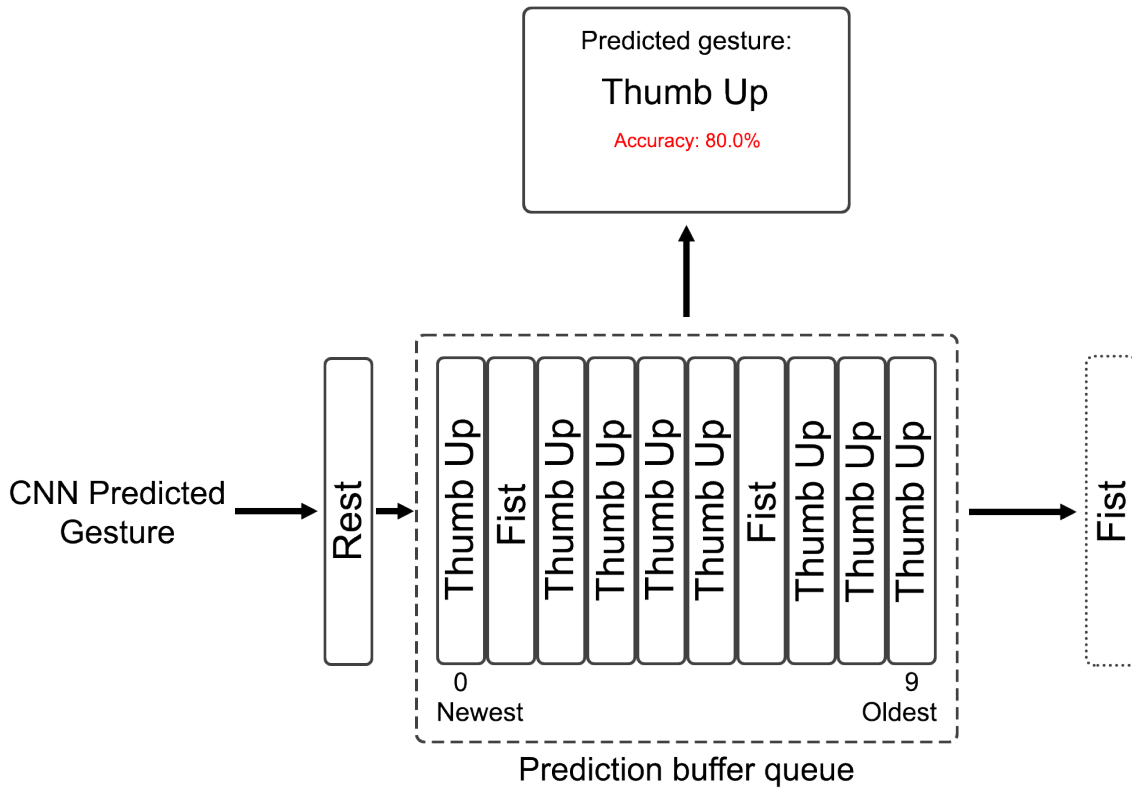


Figure 4.7: Real-time Feedback System for Gesture Prediction: Participants received immediate feedback on a screen, displaying the predicted gesture and its accuracy. The system uses a buffer of the 10 most recent predictions to measure reliability. The confidence level is calculated by finding the frequency of the most common gesture in the buffer and expressing it as a percentage. For example, if a gesture appears 8 times in the buffer, its confidence level is 80%. This method smooths out transient errors and provides a clear, stable real-time assessment.

4.4.3 Results

As illustrated in figure 4.8, the results for the real-time model are quite promising. The model demonstrates enhanced performance metrics when processing data from Participants 3 and 4, achieving an overall accuracy of 84%, which is indicative of the model's capability to effectively recognize and process familiar patterns. In contrast, the accuracy for Participants 1 and 2, who provided unseen data, is approximately 78%. The differential in gesture performance is also reflected in the confusion matrix in figure 4.5, where the trend aligns with these findings, thereby lending additional credibility to the model's robustness and its predictive accuracy.

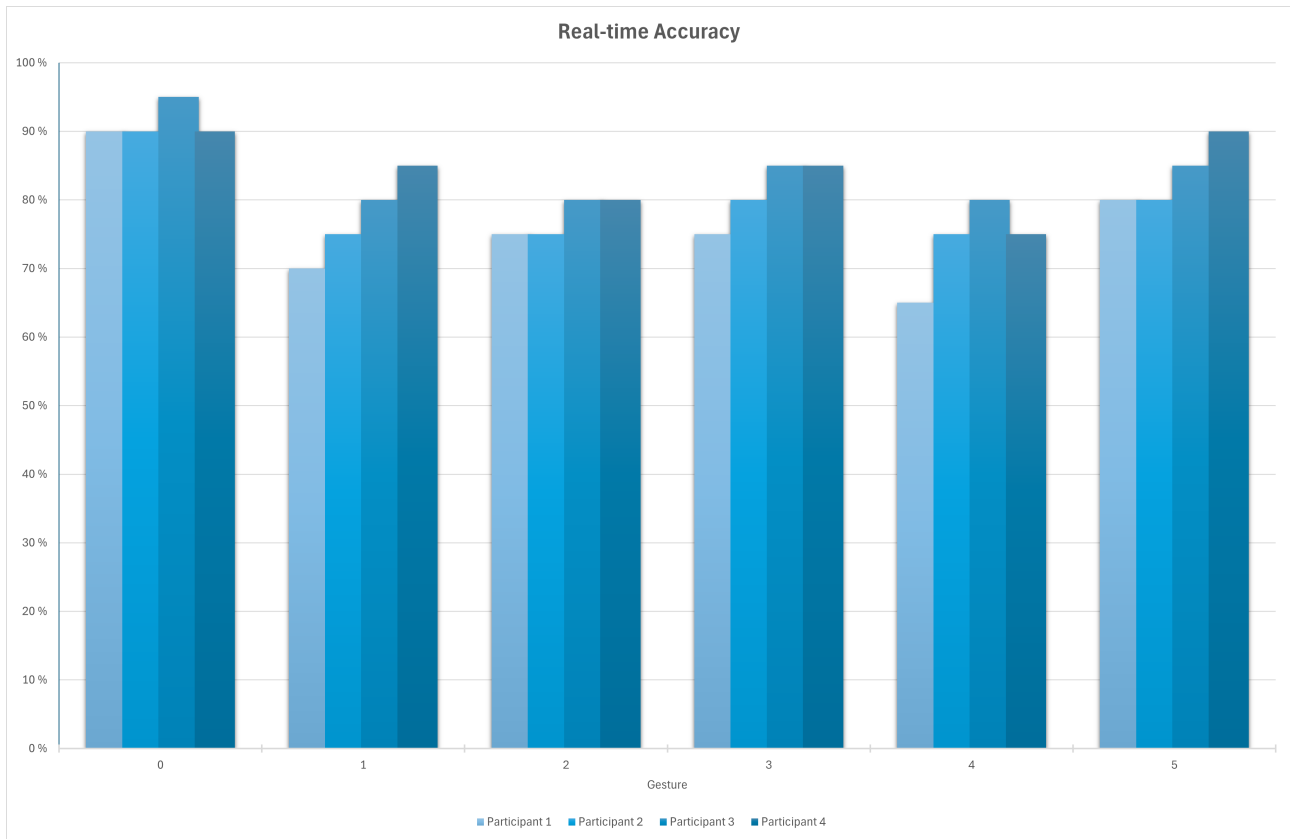


Figure 4.8: The real-time model shows promising results, with an overall accuracy of 84% for Participants 3 and 4, indicating effective recognition of familiar patterns. In contrast, Participants 1 and 2, who provided unseen data, achieved an accuracy of approximately 78%. This differential is supported by the confusion matrix, reinforcing the model’s robustness and predictive accuracy.

4.5 Assessment of Computation Time

4.5.1 Motivation

Evaluating the computation time for both inference and feature extraction of each window segment is crucial for assessing the real-life performance of the model. This analysis helps identify potential bottlenecks in the software that could delay gesture recognition. By pinpointing these areas, we can optimize the system to ensure swift and accurate responses, which is essential for effective real-time applications.

4.5.2 Method

The STM ecosystem, as outlined in Section 3.5, offers robust capabilities for pre-implementation testing of models. These tests can generate valuable statistics on the model’s memory usage on the microcontroller unit (MCU), provide insights for optimization, and measure the model’s inference time. The test was conducted manually 10 times using a pre-extracted window directly from a signal, where feature extraction was already completed and the output gesture (ID: 2) was known.

Additionally, tests are conducted to determine the time required for feature extraction on a window segment. These tests are performed locally on a PC using Python, as described in section 3.3.2, which has its own set of advantages and limitations that will be discussed later.

4.5.3 Results

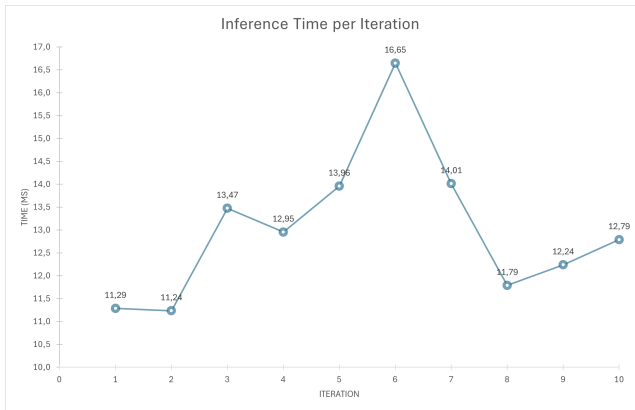
Figure 4.9a shows the distribution of inference times across different nodes in the model when run directly on the STM32. The average inference time across all tests was approximately 13.04 ms, with the slowest being 16.65 ms as shown in figure 4.9b. Additionally, the model achieved an accuracy of 90%, demonstrating its optimization for real-time application.

When examining the time required for various filtering and feature extraction processes, the average time was approximately 2.1 ms. This highlights the importance of having a simplified feature extraction process to maintain a quick response time for the system [41], [42].

```

ST.AI Profiling results v1.2 - "cnntest"
-----
nb sample(s)      : 1
duration          : 16.646 ms by sample (16.646/16.646/0.000)
macc              : 380256
cycles/MACC       : 4.73
CPU cycles        : [1,797,758]
used stack/heap   : not monitored/0 bytes
-----
Inference time per node
-----
c_id  m_id  type          dur (ms)  %   cumul  CPU cycles  name
-----
0     1     Conv2D (0x103) 0.895    5.4%  5.4% [ 96,626 ] ai_node_0
1     1     NL (0x107)     0.053    0.3%  5.7% [ 5,676 ]  ai_node_1
2     3     BN (0x102)     0.116    0.7%  6.4% [ 12,536 ] ai_node_2
3     6     Pool (0x10b)   0.070    0.4%  6.8% [ 7,539 ]  ai_node_3
4     9     Conv2D (0x103) 6.170    37.1% 43.9% [ 666,375 ] ai_node_4
5     9     NL (0x107)     0.055    0.3%  44.2% [ 5,963 ]  ai_node_5
6     11    BN (0x102)     0.123    0.7%  44.9% [ 13,312 ] ai_node_6
7     14    Pool (0x10b)   0.071    0.4%  45.4% [ 7,638 ]  ai_node_7
8     17    Conv2D (0x103) 6.440    38.7% 84.1% [ 695,486 ] ai_node_8
9     17    NL (0x107)     0.032    0.2%  84.2% [ 3,403 ]  ai_node_9
10    19    BN (0x102)     0.070    0.4%  84.7% [ 7,590 ]  ai_node_10
11    22    Pool (0x10b)   0.041    0.2%  84.9% [ 4,408 ]  ai_node_11
12    24    Dense (0x104) 2.451    14.7% 99.6% [ 264,737 ] ai_node_12
13    24    NL (0x107)     0.008    0.0%  99.7% [ 844 ]    ai_node_13
14    25    Dense (0x104) 0.043    0.3%  99.9% [ 4,621 ]  ai_node_14
15    26    NL (0x107)     0.009    0.1%  100.0% [ 1,004 ] ai_node_15
-----
total                16.646                [ 1,797,758 ]

```



(a) Distribution of inference times across different nodes in the model when run directly on the STM32 (b) Inference time for each iteration of the test runned directly on the STM32

Figure 4.9: Realtime testing off the CNN model on the STM32

4.6 Assessment of Accessibility

4.6.1 Motivation

Accessibility includes both affordability and the ease of access to technology and to support services. The adoption of affordable components, such as 3D-printed materials and commercially available microcontrollers, significantly reduces production costs. As a result, the cost of a MMHP is brought within the financial reach of more individuals. In Norway, as mentioned, prosthetics are state subsidised and only available trough a long application process. However, it is likely safe to assume that the high cost of the available MMHPs is a major reason for this. Therefore, a comparison of the price of these devices to the MMHP developed in thesis is a good indication of increased accessibility.

4.6.2 Method

In Norway, the Össur iLimb, the Otto Bock BeBionic and the Otto Bock Michelangelo hands are available. The pricing of such products are rarely publicly available and difficult to determine as third party sources report varying numbers. However, some published articles has stated the cost of the devices.

4.6.3 Result

It is important to note that this prosthesis does not include the cost of development, manufacturing or a profit margin. The cost of the commercially available prostheses encompasses such factors which highly affects their pricing. However, as seen in table 4.3, the large difference still indicate that a 3D printed prosthesis with low-cost components will likely have a drastically lower price, even with the addition of manufacturing cost, development cost and profit margins. Moreover the possibilities of making all software and hardware publicly available can be explored. By making detailed guides the project could be open source, which then again drastically increases the factor of accessibility.

Prosthesis	Cost
BeBionic	\$11 000 [43]
iLimb	\$18 000 [44]
Michelangelo	\$25 000 [44]
This prosthesis	\$400

Table 4.3: Price comparison of MMHPs

Chapter 5

Discussion

When reflecting and discussing this thesis, its findings and implications, it is essential analyze the problem statement: *To what extent can a low-cost MMHP mimic the anthropomorphism of the human hand, reduce the cognitive demand of device control, and perform in daily support hand tasks to potentially be more accessible and increase acceptance rates?*. The team has developed in accordance to this and provided results with corresponding to to this statement. This chapter includes discussion about these future directions, analysis and possible mitigation of encountered issues, and general reflection on the methods employed throughout the development.

Unfinished actuation system

Our team decided to employ a tendon drive actuation system to drive the articulation of the prosthesis. However, several issues arose in this process resulting in it being unfinished. Firstly, late in the development state, the current sensors on our custom printed circuit board (PCB) short-circuited and significantly impacted the project results. This happened during a manual voltage measurement on the PCB where the operator accidentally touched the wrong pin, sending too much voltage through the low-voltage part of the circuit. While the encoders still where operational and each motor was controllable, the sensory feedback needed to accurately control dynamic actuation was not.

While this problem created several issues, it also provided useful insight to robust PCB design. To address this issue in future development, several correctional steps will be implement. Firstly, adding more protective measures to prevent short-circuiting when troubleshooting the PCB is necessary. The use of surface mounted devices (SMD) greatly reduce the required footprint size on the PCB, as the components can be fitted in a smaller package. However this comes at the cost of more difficulties while soldering and later troubleshooting when the system does not behave the way that was initially planned. One example to make the troubleshooting easier, could be breaking out certain useful pins for easier measuring further apart from higher voltage pins, such as with the motor drivers for the Pulse width modulation (PWM) control signal. Additionally, applying conformal coating to the PCB, particularly in areas with higher voltage levels, can provide extra protection against short circuits and environmental factors. This coating insulates the components and exposed copper, preventing accidental shorts during probing and ensuring reliable operation in various conditions.

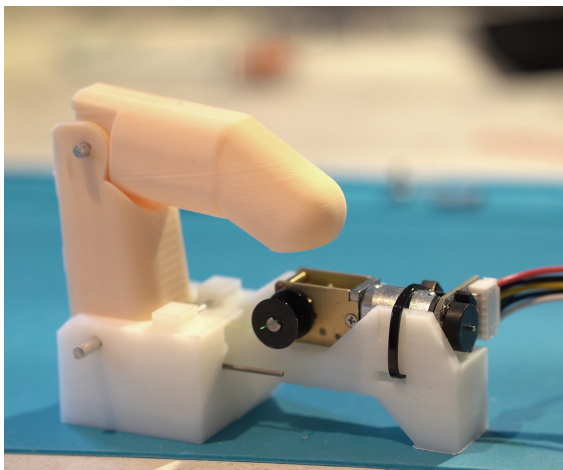
Moreover, implementing a more comprehensive testing and pre-assembly protocol is essential when designing more advanced PCBs. This especially includes using a simulation tool to test

voltages and currents and their affect on all circuit components. This way, our team could more easily identity errors, areas where short-circuits might appear and a extremely more efficient troubleshooting methods if unexpected incidents happens in later stages.

Another seemingly small issue that resulted in the unfinished actuation system was the team receiving the wrong DC motors. According to the supplier the DC motors would deliver 0.0034N/m (35 gf.cm) at 6 V. This torque was supposed to be at rated load and 68 rpm. However, when the team received the motors and tested them the specifications did not match with the motor. The motors did not deliver enough torque while running on 6 V at 31. Additionally, reading the encoder values also gave odd results that did not match the data sheet of the motor.

The motors were initially expected to run at 6 V, and the voltage regulator was planned to be a buck converter to step down the variable battery pack voltage (ranging from 9 V to 12.6 V) to a consistent 6 V for the motors. However, the motors received were not up to specification and required 12 V to operate effectively. Consequently, a buck-boost converter was considered to set the output voltage at a fixed 12 V, as the battery voltage would be both below and above this desired voltage. Ultimately, a simpler solution was chosen, as detailed in section 3.2.3, involving the use of a boost converter to increase the voltage to 13 V and then limiting it in software to 12 V.

During testing, it was observed that the motors did not move the fingers to the desired position when mounted in the hand. It was suspected that the friction where the lines were being dragged was too high. To investigate, a simple model was made to hold the fingers and move it with minimal friction without mechanical moving parts like for instance ball bearings. This was done by running the wire over a slippery aluminium rod up to the finger. Two versions of the finger were tested in this process: The original one, implemented in the hand assembly (Figure 5.1a), and one where the line did not drag inside the first joint (Figure 5.1b). As shown in Figure 5.1, the original finger bends more on the second joint, and the modified finger bends more on the first joint. The total travel distance is about the same for both versions, concluding that the friction problem we are seeing with the assembled prosthesis likely is from the actuator inside the palm up to the first joint of the fingers. A potential solution could include repositioning the actuators for a straighter, less function-prone route - which would necessarily be in the forearm, given the tight space in the palm of the hand. Another potential solution could be manufacturing the prosthesis with materials that have lower friction than polylactic acid (PLA) plastic, or using stronger actuators.



(a) Original finger used in hand assembly



(b) Modified finger without canals

Figure 5.1: Testing friction in fingers on a simple model

An additional functionality that could be explored is using the hall-effect current sensors to estimate the force applied to object from the prosthesis when engaging in a grip. This could be achieved by calibrating the current sensors to correlate the current-draw with the force applied by the actuators. During this calibration, known forces would be applied to the actuator, and the corresponding current-draw would be measured to create a calibration curve which later could be used for estimating the force being applied during grip activities. This information could then dynamically adjust the grip strength, ensuring a secure but gentle hold on object, thereby enhancing the prosthesis's functionality and improving the user experience.

MCU difficulties

During the project, one of the major challenges the team faced was the micro controller unit (MCU). Once the MCU was mounted on the PCB, the first problem occurred: establishing a connection to the MCU with an ST-Link USB did not seem to work. After some troubleshooting and researching the solution was use the ST-Link V2, which is the original development tool for debugging and programming. However, when trying to connect to the MCU for the second time, no connection was established. Luckily, after some troubleshooting and support from the ST Community, the problem was discovered. The problem was that the development tool needed a 3.3V supply as the target voltage. Once the target voltage was sorted out, a connection was achieved.

Another problem that occurred was the uploading the bootloader to the MCU. The bootloader is the first program that runs once the MCU is powered on and is responsible to set up the main software. The choice of the bootloader depends on the features that are deem important, in this case the universal synchronous and asynchronous receiver-transmitter (USART) and inter-integrated circuit (I²C) communications are one of them. There are many bootloaders available online, but since the PCB and all the components are custom-made for this project, the bootloader also needs to be custom. Once all the pinout and configuration was completed, in the STM32CubeMX software, the code was generated. This code included .hex and .bin files which were utilized as the bootloader for the MCU.

The USART communication was another challenge for the team. The initial plan was to use an USB-C port via a serial chip to program the MCU. This however showed to be more difficult then intended. With all the components soldered in place on the PCB, there was no feedback from the USB-C port. A lot of troubleshooting was tried like: measuring the right voltage on the serial chip, checking connections between the MCU and the chip, different types of USB-C cables, downloading different drivers and testing with different computers. None of these attempts unfortunately seem to solve the issue. The team ended up not using the USART communication and rather use the ST-Link development tool to program the MCU.

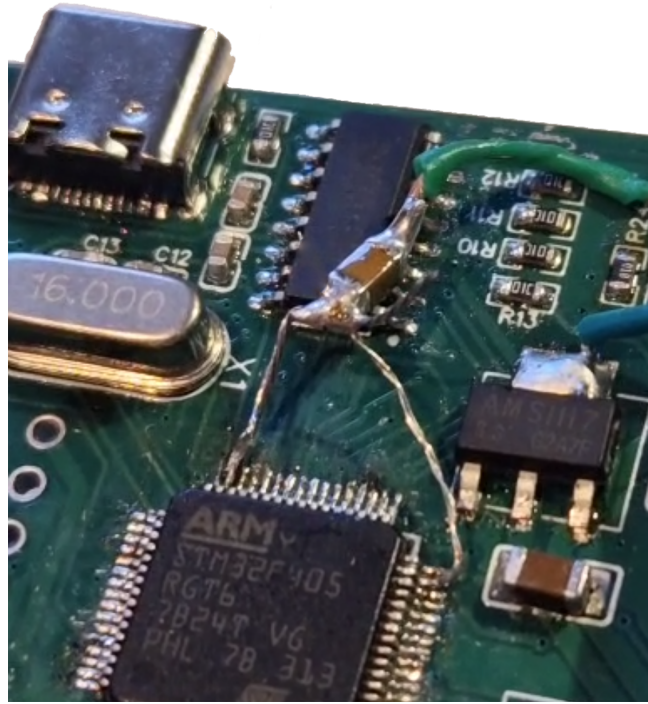


Figure 5.2: 4.7 μF capacitor soldered on PCB

Finally, during testing of the general-purpose input/output (GPIO) pins, a discovery was made that they were always high. This problem was tried by fixing in the software with several attempts, but no success. Various measurements were also tried, before asking the ST community for some guidance. The community came with a suggestion to add a 4.7 μF capacitor between the two "VCAP" pins and ground. Once the capacitor was soldered onto the MCU, as depicted in figure 5.2, the GPIO pins functioned normally.

Due to these issues, not all programming of the MCU was completed. With several components failing and many hours spent on adjustments and troubleshooting of the PCB design, not all the code has been written. Within Appendix A, the outline of the code for the STM32 can be found in the Github repository under the STM32Code folder.

Indefinite cognitive load evaluation

Assessing the cognitive demand of controlling the prosthesis is essential in this thesis as it is a large part of the problem definition. However, this would require an assemblance of prosthesis users doing extensive testing, which is quite unfeasible due to the both sensitivity around the subject and the time constrains at hand. Our team concluded that by substantially reducing the amount of gestures in the prosthesis the cognitive load would most likely be reduced. While this is a statement is based on research, a definite assessment including metrics and testing is crucial in further development. Our team is determined to launch a extensive user test to properly evaluate this metric. The Prosthesis Task Load Index (PROS-TLX) is a adaptation of the well known NASA Task Load Index (NASA-TLX) testing framework, developed specifically to asses factors such as cognitive, psychical and emotional demands of prosthesis users [45]. Employing this test and comparing outcomes to results from other studies will substantially increase the validity of this study and emphasize the potential of a decreased cognitive load.

Standardized SHAP Evaluation

In future work, a major prioritization is to conduct the SHAP with the original kit and instructions to acquire a standardized evaluation of the performance of the MMHP in ADLs. The

results can then be compared to case studies that have evaluated the state of the art prostheses as well as other studies whom have conducted the procedure on their own developments. This way, the prominence of our findings will be elevated as well as we will have important data to increase the functionality of the MMHP in ADLs.

Dataset

At the beginning of the project, we focused on building a dataset using the publicly available datasets from Ninapro [24]. However, this posed some challenges as it was unclear if, and in which way, the data had been pre-processed, and in addition they were sampled at different frequencies and resolutions. To address these issues the group thereby decided to collect own data, inspired by an earlier research article [4]. The dataset in the article uses a 300ms window per on-set detection, resulting in a small dataset containing 600 different samples. Given our goal for the prosthesis to perform well universally on various users, both known and unknown to the model, we aimed to create a much larger dataset. Understanding the importance of a large dataset for universal prosthesis performance, efforts were made to expand the dataset to encompass a wide range of examples. This enables the inclusion of a more diverse range of examples in the training process, helping the CNN to learn more robust and generalizable features. This broader dataset captures a wider range of variations and patterns in the data, enhancing performance on unseen data and reducing overfitting risks. The more extensive data points prevent the model from merely memorizing training examples, fostering the learning of meaningful features applicable across different inputs [31]. Based on this understanding, we decided to collect data with the users in various positions to vary the supporting muscles used during different exercises and the extent of muscle usage. Additionally, we aimed to extract as much data as possible from each exercise to provide the model with deeper insights into how an exercise's execution can vary over time. Based on the results from both the offline and real-time performance of the model the foundation of the dataset seems well placed. To improve the performance in a better way we must look closer to the parameter tuning as well as the quality of data.

Quality of data

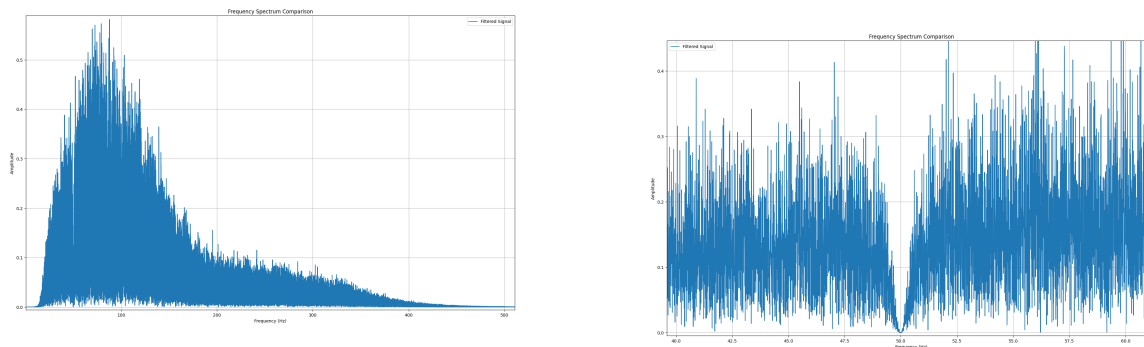
Factors such as noise and interference can significantly impact the integrity of EMG signals, potentially leading to their misinterpretation. It is essential, therefore, to verify that EMG signals are of high quality before proceeding with any further analysis. Ensuring signal purity is crucial for accurate and reliable EMG data interpretation [46]. Initially, the STM32 microcontroller chosen for the prosthesis was also intended to be used for data collection to ensure the signal was as close as possible to the final model. Due to delays, a decision was made to use an ESP32 the team had available to avoid further project delays. Like the STM32, the ESP32 features a 12-bit analog-to-digital converter (ADC), which in theory should provide adequate signal quality. However, it was later discovered that the ADC in the ESP32 is of very poor quality, introducing a lot of noise and poor signal-to-noise ratio (SNR) ¹. Since the project had progressed significantly by this point, this error was accepted, and the focus shifted to creating the best possible starting point for the data. Additionally, there were some issues with the timing needed to sample the signal accurately at 2kHz; this was also resolved after data collection had begun, thus locking the sampling rate at 1938Hz to avoid redoing data collections from the participants.

The Myoware 2.0 muscle sensors, as described in section 3.3.1, come with an integrated bandpass filter to remove unwanted high- and low-frequency noise. However, after testing the signal, the team decided to implement a more advanced software Butterworth bandpass filter to further

¹<https://forum.arduino.cc/t/fixing-the-non-linear-adc-of-an-esp32/699190>

reduce noise and irrelevant signals. Based on successful results from previous research [4], the team initially set the bandpass filter to 20-300Hz. Upon closer analysis, some signal data near 300Hz was observed during certain gestures. To ensure no critical data was lost, the low-pass cutoff was adjusted to 350Hz. Additionally, a 4th order filter was chosen to minimize disturbances near the cutoff frequencies. The decision to use a 4th order Butterworth filter was based on its widespread use in research [47] and the positive results it provided during test, as illustrated in figure 5.3a.

For the 50Hz notch filter, minimal noise was experienced during data collection. This could be attributed to the system running on DC power during sampling and the team’s efforts to eliminate potential sources of interference, such as cables and lights. As illustrated in figure 5.3b, the notch filter effectively reduces the signal in a narrow band around 50Hz, indicating that the system is functioning as expected. In further work, it would be beneficial to investigate the impact of not using a notch filter, considering based on the empirical testing and plots provided that most of the main energy of the EMG signal is in the area of 20-150Hz [48].



(a) Frequency response of the 4th order Butterworth filter, showing the adjustment of the low-pass cutoff to 350Hz to ensure no critical data is lost and to minimize disturbances near cutoff frequencies (b) Effectiveness of the 50Hz notch filter in reducing signal noise. The narrow band reduction around 50Hz indicates proper system function.

Figure 5.3: Analysis of filtering techniques used in this thesis

CNN Parameter optimization

The parameter tuning for the model will reflect the size of the trainable parameters which in return will affect both the accuracy of the classification as well as the real-time interference [30]. The team based the approach on previous research and subsequently evaluated the model based on this. Then, empirical testing was used where we employed model accuracy and confusion matrix as tools to assess how well the different parameters performed. This was an extensive process since each test had to be conducted individually and with slight adjustments of each parameter before combining the different ones again to see which were optimal. Though this method was thorough, employing advanced techniques such as Bayesian optimization [49], which has been demonstrated to effectively tune hyperparameters in various studies, could further enhance the efficiency and efficacy of the model tuning process. Another promising method is the Henry Gas Solubility-based Stacked CNN (HGS-SCNN) model [4], which provides a novel approach to parameter optimization, building upon the foundational research that underpins this thesis.

Machine Learning Algorithms

The work in this thesis is based around CNN as this have shown great results in a different research around EMG signal. CNNs are particularly effective at extracting spatial features from

EMG data, making them popular in applications like prosthetic control and gesture recognition. However, even though CNNs are highly effective, alternative hybrid models like CNN + Long Short Term Memory (LSTM) could also be considered for this task. Since LSTMs can capture temporal dependencies in sequential data, potentially improving performance by leveraging both spatial and temporal features. Integrating LSTM with CNN can enhance the model's ability to handle the temporal nature of EMG signals, resulting in better overall performance for complex tasks [50].

Offline and Real-Time Performance

To achieve the presented results, the team underwent significant trial and error. Various methods for feature extraction, window size, step size, and the impact of different captured positions, as described in section 3.4.1, were tested. It was found that including all positions resulted in a slightly lower F1 score in the final model but seemed to work better for real-time testing. Using only position 2 (standing, with forearm parallel to the ground) in the CNN model achieved an F1 score of 0.9 with an overall accuracy of 89.65%, with three gestures scoring over 90%. However, testing with real-time data did not yield equally good results and varied between tests.

This highlighted the importance of systematic testing to verify measurements against real-world conditions. In hindsight, the team could have been more precise in their approach, but much of the methodology developed alongside the model.

Regarding the real-time validation, a more systematic and methodical approach could have been beneficial. The current results were somewhat coarse due to the manual test routine and the small number of participants. Nevertheless, the results provided an indication of the system's response and accuracy, especially when dealing with unknown data. This underscored the importance of generalization in model training and testing to ensure robust performance in real-world applications.

The real-time test for feature extraction on a single window was performed locally on a PC using Python. While the local PC was much faster than the STM32 MCU used for this project, resulting in less credible computation time measurements, Python is slower than the C/C++ languages intended for the STM32 [51]. Although the code and testing on the STM32 were not completed, the tests on the PC indicated that a real-time scenario was achievable, providing valuable feedback.

Sensory Feedback

Human sensations are essential in the hand functionality and has a major affect on how we operate [52]. Feedback from sensations such as temperature and touch enables us to avoid harm and optimize grip. These feelings can be emulated in massively increase anthropomorphism in prosthetics through haptic feedback. Haptic feedback functions by using specific and advanced vibration patterns as sensation input. By employing components such as pressure sensors and temperature sensors and transforming the signals into into specific vibrations, the user can learn to perceive these as sensations. Including this in prosthetics can substantially decrease the cognitive demand, increase function quality and, especially, improve social aspects [52].

Chapter 6

Conclusion

In conclusion, this thesis presents and demonstrates the potential of a low-cost, anthropomorphic, multi-articulated myoelectric prostheses and its possibilities in the Norwegian market. The research conducted in thesis was focused on addressing the critical need for more affordable as well as functional multi-articulated prostheses. Despite the challenges remaining in prototype, the methods employed shows promising results in the development and possible commercialization of the prosthesis. The anthropomorphic design approach and tendon driven actuation system provides the prosthesis with essential human properties such as appearance, low weight, structure and performance in activities of daily living. Moreover, by leveraging data collection, advanced signal processing and a convolutional neural network, a real-time sEMG signal to hand gesture classification software with 84% accuracy was achieved. Furthermore, the usage of low cost and commercially available components and 3D printing contributes to increase the accessibility of the device. Since this thesis has addresses current issues in the myoelectric prosthetic research field and provided valuable insight, our team hope that can it inspire and improve the important technology in future development.

References

- [1] Nienke Kerver et al. “The multi-grip and standard myoelectric hand prosthesis compared: does the multi-grip hand live up to its promise?” In: *Journal of NeuroEngineering and Rehabilitation* 20.22 (2023), pp. 1–18. DOI: [10.1186/s12984-023-01131-w](https://doi.org/10.1186/s12984-023-01131-w).
- [2] Yinlai Jiang et al. “Development and evaluation of simplified EMG prosthetic hands”. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)* (2014), pp. 1368–1373. DOI: [10.1109/ROBIO.2014.7090524](https://doi.org/10.1109/ROBIO.2014.7090524).
- [3] OrtoNor. *Products*. Accessed on: 2024-04-25. 2024. URL: <https://www.ortonor.no>.
- [4] Muhammad Hamza Zafar, Even Falkenberg Langås, and Filippo Sanfilippo. “Empowering human-robot interaction using sEMG sensor: Hybrid deep learning model for accurate hand gesture recognition”. In: *Results in engineering* 20 (2023), p. 101639.
- [5] Joseph T. Belter et al. “Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review”. In: *Journal of Rehabilitation Research and Development* 50.5 (2013), pp. 599–618. DOI: [10.1682/JRRD.2011.10.0188](https://doi.org/10.1682/JRRD.2011.10.0188). URL: <https://www.rehab.research.va.gov/jour/2013/505/pdf/page599.pdf>.
- [6] George El-Khoury and Michael D. McCool. “Scalable Distribution of Quality”. In: *University of Toronto* (2003). URL: <https://www.dgp.toronto.edu/~gelkoura/noback/scapaper03.pdf>.
- [7] Zhi-Hong Mao et al. “Information Capacity of the Thumb and the Index Finger in Communication”. In: *IEEE Transactions on Biomedical Engineering* 56.5 (2009), pp. 1535–1543. DOI: [10.1109/TBME.2008.2011817](https://doi.org/10.1109/TBME.2008.2011817).
- [8] John-John Cabibihan et al. “Suitability of the Openly Accessible 3D Printed Prosthetic Hands for War-Wounded Children”. In: *Frontiers in Robotics and AI* 7 (2020), p. 594196. DOI: [10.3389/frobt.2020.594196](https://doi.org/10.3389/frobt.2020.594196).
- [9] Davide Borchia. *Torsion Spring Calculator*. <https://www.omnicalculator.com/physics/torsional-spring>. Accessed: 2024-05-16. 2024.
- [10] Lara McManus, Giuseppe De Vito, and Madeleine M. Lowery. “Analysis and Biophysics of Surface EMG for Physiotherapists and Kinesiologists: Toward a Common Language With Rehabilitation Engineers”. In: *Frontiers in Neurology* 11 (2020), p. 576729. DOI: [10.3389/fneur.2020.576729](https://doi.org/10.3389/fneur.2020.576729). URL: <https://www.frontiersin.org/articles/10.3389/fneur.2020.576729/full>.
- [11] Drishti Yadav and Karan Veer. “Recent trends and challenges of surface electromyography in prosthetic applications”. In: *Biomedical Engineering Letters* 13 (2023), pp. 353–373. DOI: [10.1007/s13534-023-00281-z](https://doi.org/10.1007/s13534-023-00281-z). URL: <https://doi.org/10.1007/s13534-023-00281-z>.
- [12] Christopher Spiewak. “A comprehensive study on EMG feature extraction and classifiers”. In: *Open Access Journal of Biomedical Engineering and Biosciences* 1.1 (2018).

- [13] Rikiya Yamashita et al. “Convolutional neural networks: an overview and application in radiology”. In: *Insights into Imaging* 9.4 (2018), pp. 611–629. DOI: [10.1007/s13244-018-0639-9](https://doi.org/10.1007/s13244-018-0639-9). URL: <https://doi.org/10.1007/s13244-018-0639-9>.
- [14] Khaled Ferkous et al. “A novel learning approach for short-term photovoltaic power forecasting - A review and case studies”. In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 108502. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2024.108502>. URL: <https://www.sciencedirect.com/science/article/pii/S0952197624006602>.
- [15] Visakha K. Nanayakkara et al. “The Role of Morphology of the Thumb in Anthropomorphic Grasping: A Review”. In: *Frontiers in Mechanical Engineering* 3 (2017), p. 5. DOI: [10.3389/fmech.2017.00005](https://doi.org/10.3389/fmech.2017.00005). URL: <https://www.frontiersin.org/articles/10.3389/fmech.2017.00005/full>.
- [16] Texas Instruments. *LM1577/LM2577 Simple Switcher Step-Up Voltage Regulator*. <http://www.ti.com/lit/ds/symlink/lm2577.pdf>. Accessed: 2024-05-16. 1999.
- [17] Altium. “PCB Ground Plane Best Practices in Your Multilayer Stackup”. In: (2024). Accessed: 2024-05-16. URL: <https://resources.altium.com/p/pcb-ground-plane-best-practices-your-multilayer-stackup>.
- [18] F. C. Krause et al. “Performance of commercial li-ion cells for future nasa missions and aerospace applications”. In: *Journal of the Electrochemical Society* 168.4 (2021), p. 040504. DOI: [10.1149/1945-7111/abf05f](https://doi.org/10.1149/1945-7111/abf05f).
- [19] Brisbane Maker Space. *kWeld Spot Welder*. Accessed: 2024-05-14. n.d. URL: <https://wiki.brisbanemaker.space/tools/electronics/kweld>.
- [20] Lara McManus, Giuseppe De Vito, and Madeleine M. Lowery. “Analysis and Biophysics of Surface EMG for Physiotherapists and Kinesiologists: Toward a Common Language With Rehabilitation Engineers”. In: *Frontiers in neurology* 11 (2020), p. 576729. ISSN: 1664-2295.
- [21] José Manuel López Villagómez et al. “Hand movement classification by time domain feature extraction in EMG signals”. In: *2023 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Vol. 7. 2023, pp. 1–6. DOI: [10.1109/ROPEC58757.2023.10409406](https://doi.org/10.1109/ROPEC58757.2023.10409406).
- [22] Hermens et al. *Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles*. URL: <https://www.seniam.org/> (visited on Apr. 4, 2024).
- [23] Manuela Gomez-Correa et al. “Forearm sEMG data from young healthy humans during the execution of hand movements”. In: *Scientific data* 10 (2023), p. 310. ISSN: 2052-4463.
- [24] Manfredo Atzori et al. *Building the NinaPro Database: a Resource for the Biorobotics Community*. eng. n.d.
- [25] Mojisola Grace Asogbon et al. “Appropriate Feature Set and Window Parameters Selection for Efficient Motion Intent Characterization towards Intelligently Smart EMG-PR System”. In: *Symmetry (Basel)* 12.10 (2020), p. 1710. ISSN: 2073-8994.
- [26] David L. Woods et al. “Factors influencing the latency of simple reaction time”. In: *Frontiers in Human Neuroscience* 9 (2015). ISSN: 1662-5161. DOI: [10.3389/fnhum.2015.00131](https://doi.org/10.3389/fnhum.2015.00131). URL: <https://www.frontiersin.org/articles/10.3389/fnhum.2015.00131>.
- [27] R. Merletti and G.L. Cerone. “Tutorial. Surface EMG detection, conditioning and pre-processing: Best practices”. In: *Journal of electromyography and kinesiology* 54 (2020), p. 102440. ISSN: 1050-6411.

- [28] Ekaba Bisong. “TensorFlow 2.0 and Keras”. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019, pp. 347–399. ISBN: 978-1-4842-4470-8. DOI: [10.1007/978-1-4842-4470-8_30](https://doi.org/10.1007/978-1-4842-4470-8_30). URL: https://doi.org/10.1007/978-1-4842-4470-8_30.
- [29] TensorFlow Lite. *TensorFlow Lite Model Conversion*. <https://www.tensorflow.org/lite/models/convert>. Accessed: 2024-05-20.
- [30] Krishnapriya S, Jaya Prakash Sahoo, and Samit Ari. “Surface Electromyography based Hand Gesture Signal Classification using 1D CNN”. In: *2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*. 2023, pp. 1–6. DOI: [10.1109/ISACC56298.2023.10083548](https://doi.org/10.1109/ISACC56298.2023.10083548).
- [31] Sami Briouza et al. “A Convolutional Neural Network-Based Architecture for EMG Signal Classification”. In: *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*. 2021, pp. 107–112. DOI: [10.1109/ICDABI53623.2021.9655876](https://doi.org/10.1109/ICDABI53623.2021.9655876).
- [32] Robert A. Paz. “The Design of the PID Controller”. In: *Klipsch School of Electrical and Computer Engineering, New Mexico State University* (2001). Available at https://www.researchgate.net/publication/237528809_The_Design_of_the_PID_Controller.
- [33] S. Fani et al. “Assessment of myoelectric controller performance and kinematic behavior of a novel soft synergy-inspired robotic hand for prosthetic applications”. In: *Frontiers in Neurobotics* 10 (2016). DOI: [10.3389/fnbot.2016.00011](https://doi.org/10.3389/fnbot.2016.00011).
- [34] Hari Krishnan Ramachandran et al. “Estimation of Mass Moment of Inertia of Human Body, when Bending Forward, for the Design of a Self-Transfer Robotic Facility”. In: *Journal of Engineering Science and Technology* 11.2 (2016), pp. 166–176.
- [35] Yang Gao et al. “Research on the Usability of Hand Motor Function Training based on VR System”. In: Oct. 2021, pp. 354–358. DOI: [10.1109/ISMAR-Adjunct54149.2021.00080](https://doi.org/10.1109/ISMAR-Adjunct54149.2021.00080).
- [36] Jovana J. Belic and A. Aldo Faisal. “Decoding of human hand actions to handle missing limbs in neuroprosthetics”. In: *Frontiers in Computational Neuroscience* 9 (2015), p. 27. DOI: [10.3389/fncom.2015.00027](https://doi.org/10.3389/fncom.2015.00027). URL: <http://journal.frontiersin.org/article/10.3389/fncom.2015.00027>.
- [37] Patricia Capsi-Morales et al. “Functional assessment of current upper limb prostheses: An integrated clinical and technological perspective”. In: *PLOS ONE* 18.8 (2023), pp. 1–23. DOI: [10.1371/journal.pone.0289978](https://doi.org/10.1371/journal.pone.0289978). URL: <https://doi.org/10.1371/journal.pone.0289978>.
- [38] Skyler A Dalley, Daniel A Bennett, and Michael Goldfarb. “Functional Assessment of the Vanderbilt Multigrasp Myoelectric Hand: A Continuing Case Study”. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE*. 2014, pp. 6195–6198. DOI: [10.1109/EMBC.2014.6945044](https://doi.org/10.1109/EMBC.2014.6945044).
- [39] Johannes G.M. Burgerhof et al. “The Southampton Hand Assessment Procedure revisited: A transparent linear scoring system, applied to data of experienced prosthetic users”. In: *Journal of Hand Therapy* 30.1 (2017), pp. 49–57. ISSN: 0894-1130. DOI: <https://doi.org/10.1016/j.jht.2016.05.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0894113016300618>.
- [40] Angkoon Phinyomark and Erik Scheme. “EMG Pattern Recognition in the Era of Big Data and Deep Learning”. In: *Big Data and Cognitive Computing* 2.3 (2018). ISSN: 2504-2289. DOI: [10.3390/bdcc2030021](https://doi.org/10.3390/bdcc2030021). URL: <https://www.mdpi.com/2504-2289/2/3/21>.
- [41] Nawadita Parajuli et al. “Real-Time EMG Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation”. In: *Sensors (Basel, Switzerland)* 19.20 (2019), p. 4596. ISSN: 1424-8220. DOI: [10.3390/s19204596](https://doi.org/10.3390/s19204596). URL: <https://www.mdpi.com/1424-8220/19/20/4596>.

- [42] L. Nieuwoudt and C. Fisher. “Investigation of Real-Time Control of Finger Movements Utilising Surface EMG Signals”. In: *IEEE sensors journal* 23.18 (2023), pp. 1–1. ISSN: 1530-437X.
- [43] T. Chen et al. “Design of 3d-printed cable driven humanoid hand based on bidirectional elastomeric passive transmission”. In: (2020). DOI: [10.21203/rs.3.rs-60462/v1](https://doi.org/10.21203/rs.3.rs-60462/v1).
- [44] K. M. Leong et al. “Hand gesture design using bionic prosthetic hand”. In: *Journal of Human Centered Technology* 2.1 (2023), pp. 43–50. DOI: [10.11113/humentech.v2n1.41](https://doi.org/10.11113/humentech.v2n1.41).
- [45] J. V. V. Parr et al. “A tool for measuring mental workload during prosthesis use: The Prosthesis Task Load Index (PROS-TLX)”. In: *Journal of Prosthesis and Orthotics* 35.1 (2023), pp. 1–15.
- [46] Emma Farago et al. “A Review of Techniques for Surface Electromyography Signal Quality Analysis”. In: *IEEE Reviews in Biomedical Engineering* 16 (2023), pp. 472–486. DOI: [10.1109/RBME.2022.3164797](https://doi.org/10.1109/RBME.2022.3164797).
- [47] Zhuo Zheng et al. “A Review of EMG-, FMG-, and EIT-Based Biosensors and Relevant Human–Machine Interactivities and Biomedical Applications”. In: *Biosensors* 12.7 (2022). ISSN: 2079-6374. DOI: [10.3390/bios12070516](https://doi.org/10.3390/bios12070516). URL: <https://www.mdpi.com/2079-6374/12/7/516>.
- [48] Qian Xiao, F Wu, and F Wang. “Complex System Based Automation Technology for Rehabilitation System Research”. In: *BIO web of conferences*. 59 (2023), p. 03006. ISSN: 2273-1709.
- [49] Hassan Ashraf et al. “Optimizing the performance of convolutional neural network for enhanced gesture recognition using sEMG”. In: *Scientific reports* 14.1 (2024), p. 2020. ISSN: 2045-2322.
- [50] Dezhen Xiong et al. “Deep Learning for EMG-based Human-Machine Interaction: A Review”. In: *IEEE/CAA Journal of Automatica Sinica* 8.JAS-2020-0829 (2021), p. 512. ISSN: 2329-9266. DOI: [10.1109/JAS.2021.1003865](https://doi.org/10.1109/JAS.2021.1003865). URL: <https://www.ieee-jas.net/en/article/doi/10.1109/JAS.2021.1003865>.
- [51] Mathieu Fourment and Michael R Gillings. “A comparison of common programming languages used in bioinformatics”. eng. In: *BMC bioinformatics* 9.1 (2008), pp. 82–82. ISSN: 1471-2105.
- [52] Charles H. Moore et al. “Grasping Embodiment: Haptic Feedback for Artificial Limbs”. In: *Frontiers in Neurorobotics* 15 (2021), p. 662397. DOI: [10.3389/fnbot.2021.662397](https://doi.org/10.3389/fnbot.2021.662397). URL: <https://www.frontiersin.org/articles/10.3389/fnbot.2021.662397/full>.

Appendix

A Github Repository

<https://github.com/aspr3m/Myoelectric-Hand-Prostheses>

B Components lists

Components

Component	Details	Quantity	Link
DC Motor	GW1812-N20-E	6	Aliexpress
Muscle Sensor	MyoWare 2.0	3	Sparkfun
Sensor cable	Myoware 2.0	3	Sparkfun
Cable shield	Myoware 2.0	3	Sparkfun
Disposable Surface EMG	Myoware 2.0	10	Sparkfun
USB Isolator	Adafruit isolator	1	Adafruit
Battery	18650 2500 mA h	3	Batterionline
Battery management system	3 cells in series	1	Aliexpress
Torsion springs	9 mm, 9 laps, 1 mm	9	Aliexpress

Table 1: Components list with description and links

Bill of materials for PCB

Designator	Footprint	Value	Quantity
MCU	LQFP-64	STM32F405RGT6V	1
OLED1	0.96OLED_4P	0.96 OLED IIC	1
BTN_1	BUTTON 4P 6X6	Button SMD	1
U2,U3	AMS1117-3.3V	AMS1117-3.3V	2
U4	AMS1117-3.3V	AMS1117-5V	1
U8	TO-263-5	LM2577S-ADJ	1
U9	IND-SMD	100 μ H	1
U10	SMB	10 μ F	4
MD1,MD2,MD3	SSOP-24	TB6612FNG,C,8,EL	3
CM1,CM2,CM3,CM4,CM5,CM6	SOP-8	ACS712ELCTR-05B	6
X1	OSC-TH	16 MHz	1
MYO_PWR	TERMINAL-5MM-2	CONN_025MM	1
MYO_SIG	3 PIN HEADER	3 pin	1
M3,M4,M1,M2,M5,M6	CONN-TH_XH6-SWD	XH6-SWD	6
LED8	WS2812B	WS2812B5050	1
R6-R8, R23-R26, R27-R32, R33-R46	R0603	10 k Ω	26
R2,R3,R14,R16,R19,R20	R1206	18 k Ω	6
R22	R1206	220 Ω	1
R35,R36	R0603	1 k Ω	2
C11	C1206	100 nF	1
C14	C1206	330 nF	1
C23	C0603	220 nF	1

Table 2: Bill of materials for printed circuit board

C Morphology Chart

Function	Solution				
Programming SW	Clion	Arduino IDE	STM32 CubeMX / STM32 CubeIDE	Matlab	
Programming language	Python	C/C++		Matlab	
ML model	TensorFlow	PyTorch	Jax		
Onset/Offset algorithm	Threshold based	Correlation Coefficient	ML	Manual/Visual	
Training data	Ninapro	Own data	Other students/external		
EMG sensor	DIY	MyoWare	uMyo	Delsys Trigno	Myo Gesture Control Armband
Flexion finger	Fishing line	4-bar link	Pulley with bearing and shaft		
Extension finger	Fishing line	Torsion springs	4-bar link	Pulley with bearing and shaft	
Torsion spring	3mm(3 laps), (strongest)	6mm(6 laps), (middle strength)	9mm(9 laps), (weakest and easiest to bend)		
Motor type	Geared DC Motor	Servomotor	Stepper motor	Linear Motor	
Microcontroller	ESP32	STM32	ATMEGA		
Battery	18650	21700			
Print filament	PLA	PETG	Carbon filled PETG	ABS	TPU
Finger joint axle	Aluminium shaft	Plastic filament	Stainless steel shaft	Bearing with shaft	
Thumb motor placement	2 motors (one inside the handpalm and one on the back of the hand)	2 motors (both inside the palm, one for flexion/extension and one for retroposition/opposition)	2 motors (one placed high in the handpalm for retroposition/opposition and one in sideways in the handpalm for flexion/extension)	2 motors (one placed low in the handpalm for retroposition/opposition and one in the forarm for flexion/extension)	2 motors (one placed high in the handpalm for retroposition/opposition and one in the forarm for flexion/extension)
Thumb retroposition/opposition	Motor, shaft and bearing	SMA actuation units			
Battery charging	12 VDC socket	USB-C	USB-Micro		
Programming port	USB-C	USB-Micro			
Grip strength	Silicon	Raw PLA / PETG / ABS	Rubber glove		

Figure 1: A morphological chart is a tool that visually maps out a product’s required functionality and explores various ways and combinations to achieve it. It lists possible solutions for each product function, offering a structured approach to evaluate alternative combinations. This facilitates early planning of the product’s architecture by considering new combinations of sub-solutions. The solutions highlighted in green is the chosen ones in this thesis.

D Wiring Schematics

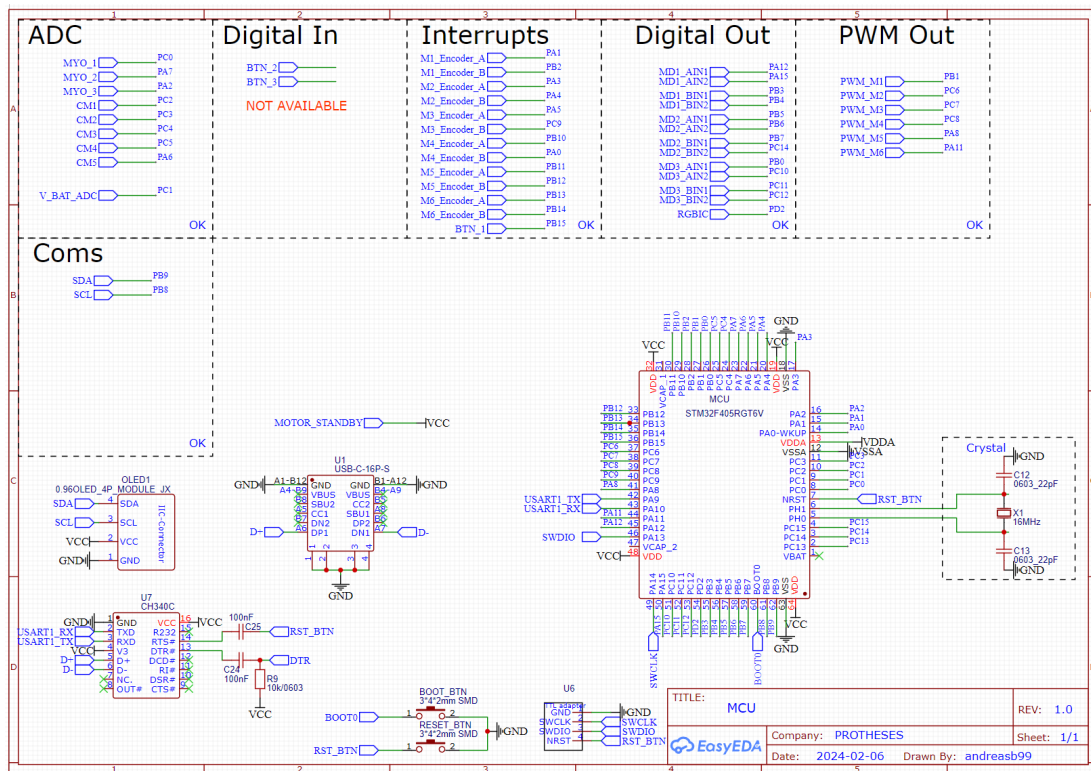


Figure 2: Schematic showing the wiring of the MCU, USB-C port for communications, input/output list, and communication protocols. The diagram includes the additional components for the MCU, such as crystal, Reset&Boot Buttons and the OLED display.

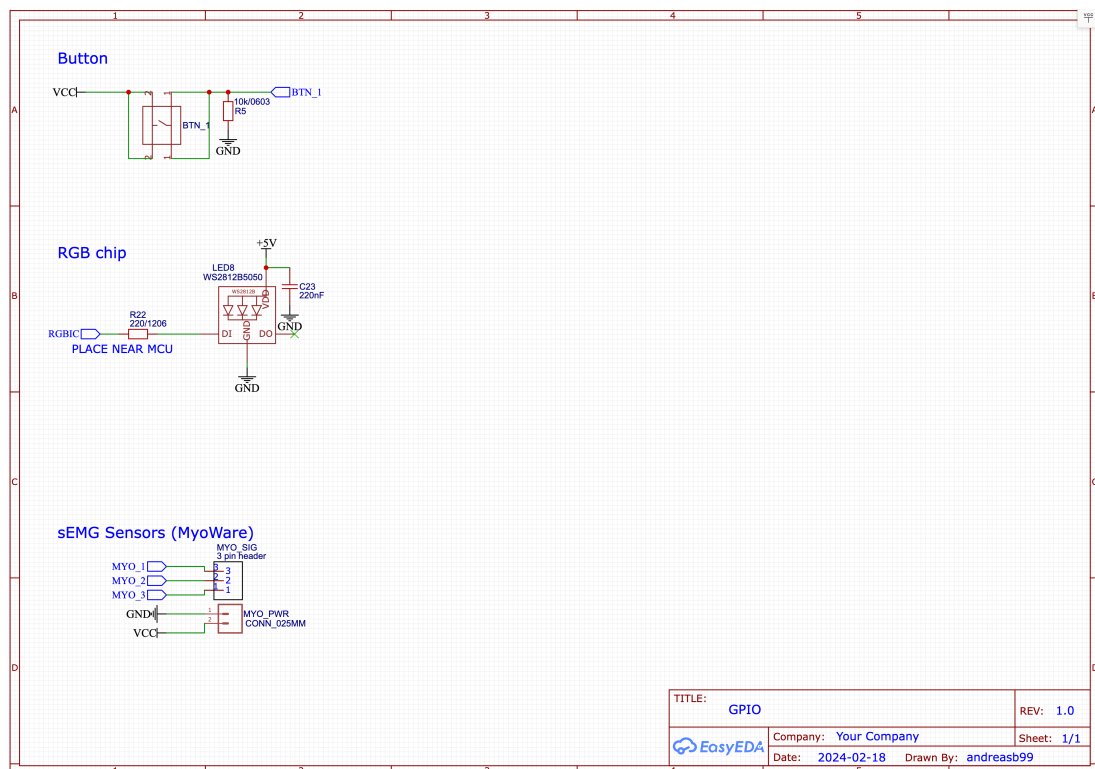


Figure 3: Schematic showing wiring of RGB chip (WS2812b 5050), tactile button for mode selections on the prosthesis, as well as the connection for the sEMG sensors from MyoWare.

E Guideline

Step-by-Step Guideline for sEMG Data Collection

Step 1: Participant Preparation and Sensor Attachment

- Explain the data collection process, including gestures, positions, and sequence.
- Prepare the forearm by shaving and cleaning the skin.
- Securely attach sensors to predetermined locations on the forearm, ensuring optimal skin-sensor contact. Fasten all electrodes with tape, allowing the subject to move freely.
- Test the sensor signals to ensure proper function and contact.

Step 2: Demonstration and Practice/Training

- Demonstrate each gesture (thumb, fist, point, tripod, pinch) and position (arm relaxed, forearm horizontal, hand on table).
- Allow the participant to practice until they are comfortable performing the gestures and positions without guidance.

Step 3: Data Collection Setup

- Prepare recording equipment and ensure it is ready to capture and store data.
 - **Important: Remove the AC adapter from the laptop!**
- Position the participant for the first data collection stance (arm relaxed by the side).
- Perform a final check to ensure sensors and recording devices are functioning correctly.

Step 4: Data Collection Process

For each gesture (thumb, fist, point, tripod, pinch) and each position (standing with arm relaxed, standing with forearm horizontal, sitting with hand on table):

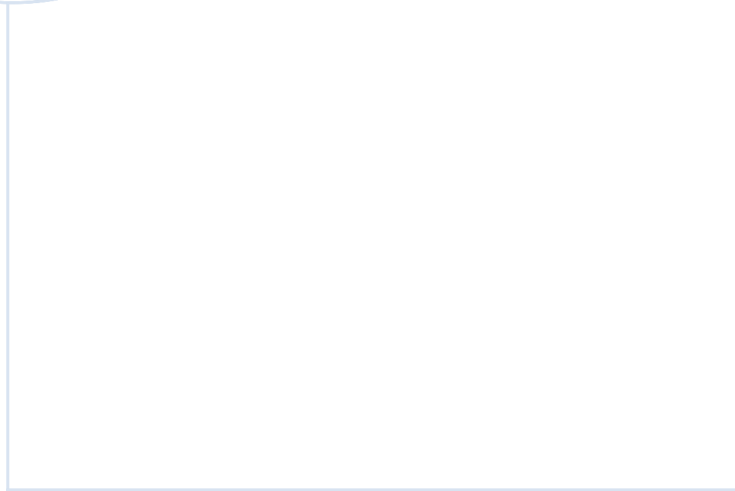
- Prepare the participant for the first gesture, starting with the resting position.
- Start recording, labeling the file according to the procedure in Step 6.
- Follow the LED timing: Maintain gesture for 4-5 seconds, followed by a 5-second rest:
 - First LED on: Participant prepares mentally.
 - Both LEDs on: Gesture activation.
 - Second LED off: Gesture release.
 - Both LEDs off: Resting period.
- Repeat the cycle 20 times.
- Ensure a short break between each gesture to minimize fatigue.

Step 5: Switching Position

- After completing all gestures for one position, move to the next position following the same procedure.
- Allow the participant to rest between switches to prevent fatigue.

Step 6: Data Labeling and Storage

- Label collected data immediately after each session with identifiable information (participant ID, gesture ID, position ID, trial number), e.g., 01-1-1_1.csv.
- Verify the quality of the data.
- Store the data in the dedicated folder in Git.



 **NTNU**

Norwegian University of
Science and Technology