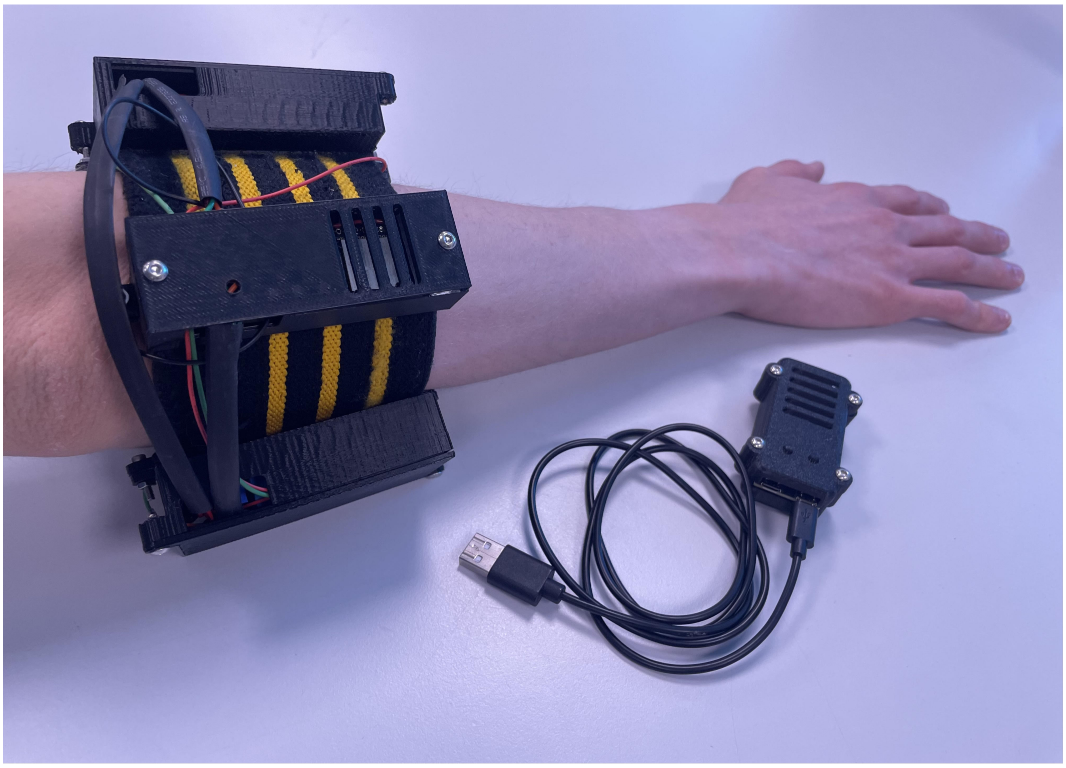


Design and Development of a Wireless Surface EMG Band for Real-Time Muscle Signal Monitoring

Bachelor's thesis in Electrical Engineering

Adam Söderberg, Darian Belenos, Emil Jägstedt,
Hampus Björnberg, Kristofer Ericsson, and Moa Carlsson

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Bachelor's thesis 2026

Design and Development of a Wireless Surface EMG Band for Real-Time Muscle Signal Monitoring

Adam Söderberg
Darian Belenos
Emil Jägstedt
Hampus Björnberg
Kristofer Ericsson
Moa Carlsson

CHALMERS

Department of Electrical Engineering
Division of Signal processing and Biomedical Engineering
EENX16-VT26-48
Chalmers University of Technology
Gothenburg, Sweden 2026

Design and Development of a Wireless Surface EMG Band for Real-Time Muscle Signal Monitoring

Adam Söderberg
Darian Belenos
Emil Jägstedt
Hampus Björnberg
Kristofer Ericsson
Moa Carlsson

Supervisor: Mehdi Shirzadi, Department of Electrical Engineering
Examiner: Silvia Muceli, Department of Electrical Engineering

Bachelor's Thesis 2026
Department of Electrical Engineering
Division of Signal processing and Biomedical Engineering
EENX16-VT26-48
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The wearable sEMG armband with the receiver node placed next to it

Template created by Magnus Gustaver, 2020
Typeset in LaTeX
Gothenburg, Sweden 2026

Abstract

Surface electromyography (sEMG) is a non-invasive method to monitor muscle activity. Due to its versatility and suitability for wearable systems, this technology can be applied in a wide range of fields, including rehabilitation, prosthetics, and human-machine interaction. This bachelor's thesis aims to design and develop a wireless sEMG armband for real-time monitoring of muscle activity, including movement classification of three key movements. The main aspects are hardware construction, wireless transmission, signal processing, and a graphical user interface (GUI).

The development of each part involved an iterative work process. A multi-channel bipolar method was used for the sEMG recordings. The electrodes were made of stainless steel with dimensions $15 \times 8 \times 0.5$ mm. Three measurement channels were used, with the signal from each amplified and filtered using an instrumentation amplifier, an RC band-pass filter, and an operational amplifier. The signals are then sampled and wirelessly transmitted via ESP-NOW to a computer. Each sample is processed using custom root mean square (RMS) and continuous wavelet transform (CWT) functions and displayed in a GUI. Using the extracted features, three movements are classified using linear discriminant analysis (LDA): rest, wrist extension, and flexion.

Each part of the system was tested and verified individually. The electrodes exhibited characteristics similar to those of commercially available Ag/AgCl wet electrodes. However, the amplification and cutoff frequencies differed from the calculated and desired values. The lower cutoff frequency varied between 13 and 15 Hz, and the upper between 299 and 350 Hz across the three channels. The latency for the wireless transmission is on average 20.8 ms, with 0% packet loss. Both the RMS and CWT functions were deemed effective and consistent with the theoretical expectations. The movement classification resulted in an accuracy of 93.9 %.

Despite isolated tests indicating that each part works, the system as a whole lacks adaptability and robustness. For instance, the movement classification heavily relies on manual calibration. To make the band viable for practical use, several improvements are required. However, it is concluded that the band achieves its objective of recording and monitoring muscle activity in real time and classifying key movements, while serving as a proof-of-concept for a wireless sEMG armband.

Keywords: sEMG, gesture recognition, human-machine interface, wearable device, signal processing, microcontroller, wireless communication, instrumentation amplifier, electrodes, continuous wavelet transform

Acknowledgements

Firstly, we would like to thank our supervisor, **Mehdi Shirzadi**, for his guidance. His ability to swiftly provide great advice and help has been invaluable to us throughout the project.

Mehdi Shirzadi, Gothenburg

We also want to thank **Rikard Karlsson**, lab manager at CASE labs. Especially for providing the solution to use conductive epoxy for the electrodes and helping with the application process.

Rikard Karlsson, Gothenburg

Lastly, we want to thank **Peter Bäckgren**, lab technician at FUSE, for using the water jet cutter to cut out the electrodes.

Peter Bäckgren, Gothenburg

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

ADC	Analog to Digital Converter
API	Application Programming Interface
CPU	Central Processing Unit
CWT	Continous Wavelet Transform
DFT	Discrete Fourier Transform
DSSS	Direct Sequence Spread Spectrum
ECG	Electrocardiography
EMG	Electromyography
FDM	Fused Deposition Modeling
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum
GUI	Graphical User Interface
HP	High pass
IDE	Integrated Development Environment
LDA	Linear Discriminant Analysis
LP	Low Pass
MAC	Media Access Control
MVC	Maximum Voluntary Contraction
PCB	Printed Circuit Board
PLA	Polyactic Acid
RMS	Root Mean Square
sEMG	Surface Electromyography
SHF	Super High Frequency
SIMD	Single Instruction, Multiple Data
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
UHF	Ultra High Frequency
WLAN	Wireless Local Area Network

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	A specific sample in the analysis window
τ	Integration variable (time-shift)

Sets

$A \in A_1, A_2, A_3$	Set of amplifiers
$C \in C_1, C_2$	Set of capacitors
$E \in E_0, E_1, E_2, E_{12}, E_{34}, E_{56}$	Set of electrodes or electrode pairs
$G \in G_1, G_2$	Set of gain values
$R \in R_1, R_2, R_{G1}, R_{G2}$	Set of resistors

Parameters

a	Scale factor
f_s	Sampling frequency
C	Capacitance
Δf	Frequency resolution
Δt	Temporal resolution
G_{tot}	Total amplification of the circuit
j	Imaginary number
m	Number of data points in the wavelet
n	Number of data points in the signal

N	Number of samples in the analysis window
R	Resistance
T_{tot}	Total sampling time variable

Variables

f	Frequency variable
k	Signal length
R_G	Resistance of potentiometer
t	Time variable
t_1	Latency variable
t_2	Latency variable not including sampling time
U_1	Input voltage variable
U_2	Output voltage variable
ω	Angular frequency variable

Functions

$\mathcal{F}(x)$	Fourier transform function
$h(t)$	Impulse response
$\boldsymbol{H}(j\omega)$	Transfer function
$\mathcal{O}(N)$	Computational complexity
$\boldsymbol{\Psi}(t)$	Morlet wavelet function
$x(t)$	Input signal voltage function
$\boldsymbol{X}(t, a)$	Continuous wavelet transform function

Contents

List of Figures

List of Tables

1

Introduction

Electromyography (EMG) is a technique that detects electrical activity produced during muscle contraction [1]. In contrast to intramuscular EMG, surface electromyography (sEMG) is a non-invasive method in which measurements are taken on the skin's surface without needles or other invasive techniques. Electrodes are placed directly on the skin over muscles to measure the electrical potential. The data received can then provide valuable insights into muscle activation, fatigue, and recovery [1]. Due to these traits, sEMG technology has recently been used for gesture recognition in wearable devices.

Gesture recognition can be achieved through several methods, which often require cumbersome setups that contradict the practical, easy-to-use nature of wearables. An example is vision-based gesture recognition, which requires a camera setup that records the gesturing body from all required angles. Because sEMG devices perform this recording using electrodes directly on the skin, they do not require a visual overview of the gesturing body, unlike vision-based gesture recognition. This allows for a much more compact system that can be neatly contained in a small wearable device, such as an armband.

When designing for wearability, wires are a constant problem; they might get stuck on objects or be in the way of clothes, making it uncomfortable for the user. Therefore, wearables are most often wireless. This introduces two hurdles to overcome: wireless transmission and real-time performance. Real-time transmission means transmitting data between devices without significant delay. As instantaneous feedback is required for many applications, real-time monitoring of muscular performance is crucial for system efficacy [2]. This requires both a reliable and efficient way to transmit and monitor the data so it is easily understandable to the user.

These requirements pose distinctive engineering challenges, as the system must filter out and amplify weak sEMG signals, transmit them wirelessly, and make them readily available to the user, all within a compact wearable device. This bachelor's thesis seeks to explore how these challenges can be overcome and to build a functioning device that addresses them.

1.1 Objective

The project aims to design and build a wearable sEMG armband. The armband should be able to wirelessly transmit recorded sEMG signals from the forearm muscles to a computer. Furthermore, it seeks to create a user interface that enables real-time monitoring of recorded signals through signal processing.

The project has been divided into six specific objectives:

1. Record the sEMG signals generated by the forearm muscles using electrodes.
2. Design and build a circuit that amplifies and filters the recorded sEMG signal.
3. Convert analog signals to digital and then transmit them to a secondary device wirelessly in real-time.
4. Build a wearable armband that can contain all the electronics required to perform the first three objectives.
5. Distinguish relevant sEMG signals from noise using signal processing, as well as differentiate and display those signals in a graphical user interface (GUI).
6. Classify wrist extension, flexion, and rest based on extracted signal features.

1.2 Scope and Delimitations

The objectives presented thus far are fundamental parts of the project and are expected to be completed. While further improvements and applications are possible for the band, the project is limited to the objectives presented, as they are more relevant to the sEMG band's core function. The primary reason behind the delimitations is time constraints. The following text presents these delimitations.

The project is limited to traditional signal processing methods and therefore does not utilize neural networks, which runs counter to the recent trend of using such tools. Due to the time required to train these models, this project does not include them.

Additionally, the method of wireless transmission is decided by a study of literature. Thus, only the chosen method is tested, primarily due to the time required to set up multiple wireless communication systems. Due to budget and time constraints, the electrode material is chosen through a similar process: decisions are based on the literature, followed by practical verification of the selected material.

Further, only disposable alkaline batteries are used. This is due to the risk of rechargeable lithium battery fires and the administrative time required to address the issue.

Finally, the project's focus is functionality. Aspects such as ergonomics and aesthetics are secondary to performance and usability. The band serves as a technical proof of concept rather than a finished product. Costs, manufacturing methods, and emissions are therefore not considered either.

1.3 Previous Work and Applications

There are currently two major areas related to wearable sEMG devices. The first one is consumer electronics. These devices are most often brought up in the context of human-machine interfaces. An example is the Myo-armband, developed by Thalmic Labs [3]. The band can recognize five basic hand and wrist movements, allowing the user to perform simple control of devices such as computers and phones. More recent alternatives, such as the Meta Neural-Band, can recognize finer gestures and aim to replace both the mouse and the keyboard [4]. This partly ties into ergonomics. Research has shown that prolonged use of the standard keyboard and mouse causes muscle strain, which over time can lead to musculoskeletal disorders [5]. These issues could be avoided by using sEMG wearables. Studies using ergonomic keyboards show reduced muscular activity compared to standard keyboards, while efficiency is comparable to that of standard keyboards after a few weeks of training [6]. Despite these applications for sEMG wearables, the technology has not yet achieved widespread commercial success.

The second large area of application is medical devices. In contrast to consumer electronics, medical applications often consider whether to use wired or wireless communication, dry or wet electrodes, or intramuscular EMG instead of sEMG [7]. Although intramuscular EMG can provide more precise measurements, it limits the user's ability to move as needles are inserted into the muscles. The invasive nature of intramuscular EMG also limits its versatility, as it requires the involvement of medical professionals [7]. It is partly for these reasons that sports scientists often prefer using sEMG when conducting research [8]. Wired sEMG allows for greater mobility than intramuscular EMG. However, it is still somewhat limited by the wires. For wearable user devices, dry electrodes are preferred over wet electrodes for their reusability and improved user convenience [9]. The user is also not required to know the exact electrode placement, only a general location, as they are integrated into a band.

A prominent application of sEMG within the medical field is prosthetics. The company Smartarm Robotics Inc. [10] was founded in 2018 and developed the smartARM sEMG-controlled prosthetic arm. For early prototypes, the Myo-armband was used to control the prosthetic [11][12]. However, in later versions of the smartARM [10], the electrodes are attached directly to the prosthetic, a method common to competitors such as the OpenBionics HeroArm [13]. This indicates that an sEMG armband can serve as a prototyping tool in prosthetic research, but not for use in an actual product.

2

Theory

This chapter serves as a foundation of information for the project to build upon. It introduces fundamental concepts, relevant technologies, and key principles necessary to understand the system's design and implementation. Furthermore, it seeks to explain and support the decision-making in the planning, building, and testing phases of the project.

2.1 Surface Electromyography

When muscle contractions are initiated, the brain sends signals that travel through the nervous system along motor neurons to the muscle fibers [14]. A motor neuron, together with the muscle fibers it controls, makes up one motor unit. When the brain activates a motor unit, an electrical signal called an action potential travels along the motor neurons. Once it reaches the muscle fibers, it triggers another action potential in the muscle fiber at the neuromuscular junction [14]. This signal spreads along the membrane of the muscle fibers, causing calcium ions to be released inside the muscle cell. This enables the two proteins actin and myosin to interact, with myosin pulling on the actin filaments, causing them to slide past each other and the muscle to contract. The electrical activity of the muscle fibers also induces small voltage changes on the skin surface [14]. When several motor units are activated simultaneously, their signals combine to form a larger, measurable signal.

These signals can be detected and measured using a technique called EMG [15]. A non-invasive method for recording EMG signals is sEMG, which uses electrodes placed on the skin's surface [16]. The basic function of an sEMG system is to acquire electrical signals from muscles, amplify them, filter out unwanted noise, and finally present the processed signal.

One commonly used measurement configuration is bipolar sEMG, in which three electrodes are used to measure a single signal [16]. Two of these, denoted as E_1 and E_2, are positioned directly over the measured muscle or group of muscles. When both electrodes are positioned on the same muscle, they must be located on the same side of the innervation zone [17]. The innervation zone is the region where motor neurons connect to muscle fibers, from which the signal propagates in both directions along the muscle. Placing the electrodes on the same side of this zone helps prevent signal cancellation [17]. The electrodes E_1 and E_2 are usually separated by an inter-electrode distance of approximately 20 mm [18]. The third electrode, denoted E_0, serves as a reference electrode to reduce noise [16]. It is therefore often placed in an area with low muscle activity, such as near the elbow or knee.

2.1.1 sEMG Electrodes

Electrodes are commonly classified into two main categories: wet and dry [19]. Wet electrodes require a conductive gel between the electrode and the skin and are typically made of Ag/AgCl. Due to their low skin-electrode impedance and high signal quality, they are widely used in clinical applications. Dry electrodes, in contrast, do not require conductive gel, which provides greater flexibility and ease of use. However, they exhibit higher skin-electrode impedance, which reduces signal quality. Common materials used for dry electrodes include silver, platinum, and stainless steel [19].

Electrode geometry is another important design consideration. Electrodes are typically shaped as either rectangles or circles, with no clear performance advantage for either geometry [18]. The choice is often driven by practical considerations or the desired surface area.

Electrode size can vary significantly, with proposed surface areas ranging from 5 mm^2 to 800 mm^2 [18]. Increasing the electrode size reduces the skin-electrode impedance [18]. When the electrode dimension is increased perpendicular to the muscle fibers, the effective pickup area increases. In contrast, increasing the length along the muscle fiber direction introduces a spatial integration effect that attenuates high-frequency components [18]. Consequently, electrode size must be carefully balanced to achieve low impedance while avoiding excessive pickup area and spatial integration.

2.1.2 Amplification and Filtering

The electrical signals measured in sEMG are of very low amplitude, typically on the order of millivolts or microvolts [16]. They are also affected by significant noise originating from various sources. One major source is power line interference, which occurs at a frequency centered around 50 Hz [20]. Another source is motion artifacts arising from changes in skin-electrode impedance as the electrode moves across the skin [20]. Movement of the cables can also contribute to such artifacts. Motion artifacts generally contain frequency components up to 50 Hz [20]. Baseline noise is another contributor, referring to unwanted signals present when the muscle is relaxed [20]. It is commonly modeled as white Gaussian noise. This type of noise reduces the signal-to-noise ratio (SNR), making it more difficult to distinguish low-amplitude sEMG signals [20]. Additional sources of interference include electrocardiography (ECG) signals originating from cardiac activity and crosstalk from neighboring muscles [20].

Since E_1 and E_2 are placed in proximity, the noise is similar. However, muscle signals are more local and will differ across electrodes [16]. Thus, a differential amplifier configuration can be used to subtract the signals, removing the common noise while preserving the sEMG signal.

To further reduce the signal's noise, filtering is required. This can be achieved both analogically and digitally, and usually, both are used to some extent [16]. Analog filters are constructed using frequency-dependent components [21]. There are many types of filters with different functions. Low-pass (LP) filters dampen high frequencies while allowing low frequencies to pass through. High-pass (HP) filters have the opposite function. Band-pass filters allow frequencies within a certain range, called the pass band, to pass and dampen all frequencies outside of this range [21][22]. Band-stop filters do the opposite, damping frequencies outside the so-called stop band. A notch filter is a band-stop filter with a very thin stop band.

In the ideal case, filters would completely exclude all frequencies outside the pass band [22]. In reality, it would be impossible to construct such a filter. Instead, there is a frequency band around the cutoff frequency where the signal gets partially damped. To describe the behavior of non-ideal filters, transfer functions are used [22]. Usually denoted as $\boldsymbol{H}(j\omega)$, the transfer function describes the relationship between the output and input voltage to the filter. The absolute value of the transfer function $|\boldsymbol{H}(j\omega)|$ describes the magnitude of the output signal in comparison to the input signal and thus describes the damping of the filter at a certain frequency [22]. A simple and common type of analog filter is the RC filter, consisting of resistors and capacitors [22]. The behavior of an RC filter can be described using a transfer function, such as the following function presented in [21] for a first-order RC LP filter

$$\boldsymbol{H}(j\omega) = \frac{\boldsymbol{U_2}}{\boldsymbol{U_1}} = \frac{1}{1 + j\omega RC} \tag{2.1}$$

where $\boldsymbol{U_1}$ is the input voltage, $\boldsymbol{U_2}$ is the output voltage, R and C are the resistance and capacitance of the components, and $\omega = 2\pi f$, with f being the frequency of the signal. This shows that the signal decreases gradually rather than abruptly. The cutoff frequency is then defined as the frequency at which the signal has decreased by 3 dB from its peak value [23]. When multiple RC filters are used in series, buffer amplifiers can be placed between the RC stages to prevent loading effects [23]. The filter then becomes an active filter.

For sEMG applications, it is common to choose an HP filter with a cutoff frequency between $5 - 30$ Hz and an LP filter with a cutoff frequency between $400 - 500$ Hz, although it can go as low as 350 Hz without losing much sEMG signals due to their low amplitude above that frequency [20]. While sEMG signals also occur outside this range, their amplitudes are much lower and difficult to distinguish from noise.

EMG signals are inherently analog and therefore continuous in nature [16]. Since computers operate in the digital domain, the EMG signal must be converted from analog to digital before display and processing. This conversion is performed by an analog-to-digital converter (ADC). The resolution of the resulting digital signal is dependent on the sampling frequency. According to the Nyquist sampling theorem, the sampling frequency must be at least twice the highest frequency component of the analog signal to avoid aliasing-induced distortion.

2.2 Wireless transmission

Many portable devices used today rely on some form of wireless connection. Laptops and phones, for example, almost always include both Wi-Fi and Bluetooth antennas. Some of the more widely available transmission methods, such as Wi-Fi, Bluetooth, and Peer-to-Peer networks, are presented in this section.

2.2.1 Wi-Fi

A wireless local area network (WLAN) operates using radio waves via a central access point, such as a router [24]. When discussing WLAns, it usually refers to the IEEE 802.11 standard, more commonly known as Wi-Fi. All connected devices, also known as stations, continuously perform handshakes with the central access point whenever data is transmitted [24]. This ensures that neither side is overwhelmed by data, but it might cause small delays.

Wi-Fi 5 and 6 mainly use the super-high-frequency (SHF) band and the lower ultra-high-frequency (UHF) band [24]. As many devices also use this frequency band, certain methods are required to avoid the signal being drowned out by noise. The Wi-Fi standard for these waves uses two main techniques, frequency-hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS), to avoid being disrupted by noise. FHSS splits the frequency band into several channels, and the signal then rapidly switches between them [24]. So, if there is noise on a specific channel, it is dialed down to the average across all channels. DSSS, on the other hand, is a method that makes it easier for the receiving station to translate the waves into data [24]. In short, it works by transmitting its data multiple times simultaneously, so if some of it is blocked or drowned out by noise, several versions of the same data remain intact.

2.2.2 Peer-to-Peer

Peer-to-peer networks can be constructed using WLAN instruments and share many features with Wi-Fi [24]. The main difference between this and WLAN is that, instead of several stations sharing a single network, peer-to-peer connections create a temporary, interconnected network among all participating stations. Some peer-to-peer features are also particularly well-suited for direct transmission between stations [24]. There is no need for a central unit to hold the data before passing it on to the next station. Any station can leave and rejoin the network as long as at least one participating station remains connected. The handshake process in peer-to-peer communication is also more efficient, reducing transmission delays [24].

One such peer-to-peer network is the ESP-NOW protocol [25]. This protocol is developed by Espressif Systems for the ESP32 microcontrollers. It is marketed as a quick, long-distance, ultra-low power alternative to their Wi-Fi and Bluetooth protocols. The bit rate, meaning the amount of bits transmitted at a time, defaults to the maximum bit rate of 250 Mbps [25]. The maximum data length, meaning the largest number of bytes per transmission, is 1470 bytes for the 2.0 version and

250 bytes for the 1.0 version. The ESP-NOW protocol has been demonstrated to achieve up to 300 meters of range, although speed and transmission success rate begin to decline after 75-150 meters, depending on terrain and interference [26]. It also features a long-range mode that has demonstrated clear transmission over 900 meters, with a drop in performance at 400-600 meters.

2.2.3 Bluetooth

Bluetooth uses the UHF band, but it prioritizes power management over range and efficiency [24]. Even though Bluetooth resembles WLAN in many ways, it is not a local area network. Rather, Bluetooth is closer to a peer-to-peer network but is much less flexible, still requiring a master node to connect to the network [24]. Instead, Bluetooth focuses on ease of use and implementation cost, making it a popular choice for portable devices such as headphones and smartwatches [24]. Therefore, there are often many preconfigured profiles that devices can use to recognize and connect to a unit via Bluetooth.

2.3 Digital Signal Processing

To effectively interpret signals, digital signal processing is required to extract representative features from them. In this section, two such methods are presented.

2.3.1 Root Mean Square

The root mean square (RMS) value is presented in [27] as

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i|^2} \tag{2.2}$$

where N represents the number of samples in the analysis window, and x_i denotes the signal voltage at sample i. First, the signal is squared, then averaged over a set number of samples, and finally, the square root of the average is calculated. It is a widely used operation when studying sEMG signals, as it quantifies the signal and reflects the electrical activity of motor units [28]. RMS therefore provides a measure of the intensity of muscle activation, where higher RMS values generally correspond to increased force level contraction [27]. In addition to estimating muscle activation, RMS is frequently used to assess muscle fatigue [29]. During constant-force fatiguing tasks, the RMS value typically increases to maintain the required force [29]. However, during sustained Maximum Voluntary Contraction (MVC), the RMS value may gradually decrease over time since the muscle is unable to maintain the initial force level. Consequently, muscle fatigue can influence the RMS value differently depending on the type of contraction.

In addition to muscle fatigue, RMS is sensitive to other peripheral factors such as muscle fiber properties, subcutaneous layer properties, and recording setup [28]. The measured RMS value during MVC can be used to express all future measurements as a percentage of that value to remove the individual physiological properties [28]. Conversely, this might be difficult to implement, as subjects may not reach true maximal contraction, due to discomfort or fear of pain.

2.3.2 Continuous Wavelet Transform

The frequency in an EMG signal can indicate muscle type and force level [30]. It decreases as the force level increases, and certain frequency bands can be used to determine the muscle type. The continuous wavelet transform (CWT) is a method that can be used to extract frequency information from this signal [31][32]. Compared to the short-time Fourier transform (STFT), the wavelet transform provides a higher time and frequency resolution trade-off, giving it the best possible resolution when extracting information from EMG signals [33]. This is important because temporal and frequency resolution cannot be infinite, which would be the ideal case [34]. The relation is described by

$$\Delta f \cdot \Delta t \geq \frac{1}{4\pi} \tag{2.3}$$

where Δf is the frequency resolution, Δt is the time resolution. The CWT of a signal is obtained by convolving the signal with a wavelet [33]. The wavelet consists of an oscillating signal and, in the case of a mother wavelet called the Morlet wavelet $\psi(t)$, a complex sinusoidal multiplied by a window function [31]. A benefit of the complex Morlet wavelet compared to a real-valued wavelet is that it produces a constant peak when applied to a fixed frequency instead of oscillations [35]. The window function for the Morlet wavelet is a Gaussian curve [31]. This function is described by

$$\boldsymbol{\psi}(t) = \pi^{-\frac{1}{4}} e^{j\omega_0 t - \frac{t^2}{2}} \tag{2.4}$$

where ω_0 is the central frequency, and t is the time. The mother wavelet is then rescaled using the scale factor a, which is proportional to the frequency being analyzed, and convolved with the signal x to create the wavelet transform X shown in

$$\boldsymbol{X}(t,a) = \int_{-\infty}^{\infty} \frac{\psi\left(\frac{t-\tau}{a}\right)}{\sqrt{a}} \cdot x(t)\, d\tau \tag{2.5}$$

where τ is the time shift [31]. The convolution is an expensive operation with a computational complexity $\mathcal{O}(n^2)$ [36]. This can be reduced using many methods, but one of the more commonly used methods to do this in the frequency domain [36]. This can be done using the fast Fourier transform (FFT) algorithm, which has a complexity $\mathcal{O}(n \cdot log(n))$. This relationship is described by

$$x \circledast h = \mathcal{F}^{-1}(\mathcal{F}(x) \cdot \mathcal{F}(h)) \tag{2.6}$$

where $\mathcal{F}$ is the Fourier transform [36]. Note that the Fourier method computes the circular convolution, which is not the same as the linear convolution needed to compute the wavelet transform [36]. This can be bypassed by zero-padding the wavelet and signal to at least $k = m + n - 1$, where m is the number of data points in the wavelet and n is the number of data points in the signal. Part of the circular convolution will then contain the linear convolution, which can be extracted [36]. There is a phenomenon arising when using a finite signal called the cone of influence [35]. It means that the wavelet coefficients near the edge become inaccurate because there is not enough data in the convolution with the wavelet. This zone becomes larger at lower frequencies due to the wider window.

The FFT is a fast algorithm to compute the discrete Fourier transform (DFT) at a lower complexity [37]. It does this by using properties of the complex root unity. The FFT works by recursively dividing the input into even and odd coefficients, computing DFTs on each subset, and combining the partial results using butterfly operations [37]. The inverse FFT can then similarly be calculated using the same method but with the inverse DFT [37].

2.4 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a fundamental technique for classifying biosignals and is widely used in machine learning and statistics [38]. By maximizing the posterior probability, under the assumption that the data are normally distributed and that the covariance matrices are equal, it determines linear combinations of features that best separate two or more classes. Consequently, LDA is widely used to classify motions when analyzing sEMG signals [39]. The selection of sEMG features is a critical factor in classification performance; recent methods have proposed including both time and frequency features to increase accuracy [39].

3

Methodology

In this chapter, the system's general design is described, with subsequent sections exploring each step in greater detail. Additionally, the testing and verification process is described in detail. Prior to this, relevant ethical considerations are mentioned.

The system is a multi-channel sEMG-recording armband that wirelessly transmits signals to a computer for processing and display. Figure 3.1 illustrates the simplified system architecture.

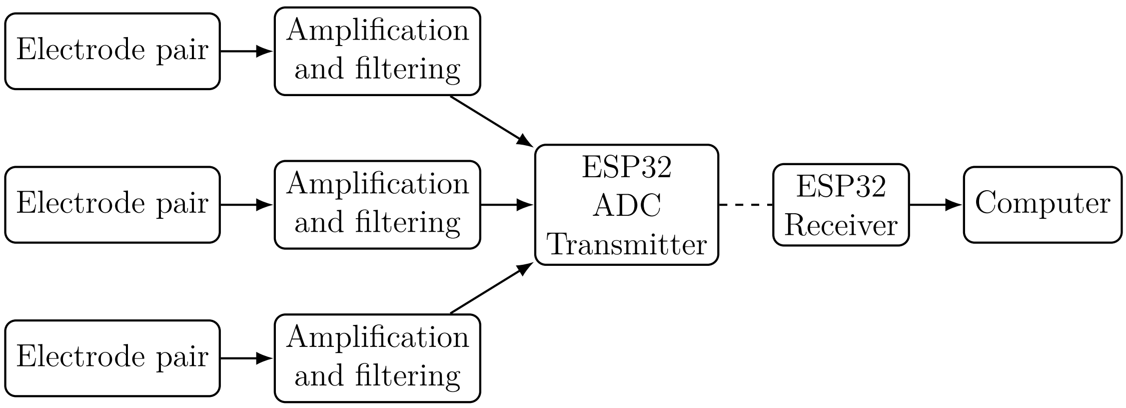


Figure 3.1: System architecture of the sEMG armband. The dotted line symbolizes wireless communication.

The system includes three channels. Each channel features a set of electrodes and an amplification and filtering circuit. A bipolar method is used to record the sEMG signals. Thus, each set consists of two electrodes, denoted E_1 and E_2. Additionally, all channels share one reference electrode, E_0. The amplification and filtering circuit amplifies the signal difference between E_1 and E_2 and filters out signals outside the desired frequency range.

An ESP32 microcontroller receives the amplified and filtered signal, converts it from analog to digital, and wirelessly transmits it to another ESP32 connected to a computer. Subsequently, the data received from the microcontroller is buffered, processed, and visualized in a GUI using various signal-processing methods. A classifier is applied to distinguish wrist movements.

Every program necessary for testing and operating the sEMG band is available in a public GitHub linked in A Appendix 1.

3.1 Ethics

The band measures bodily signals, which raises several ethical concerns. One area particularly relevant to this is privacy and data security. According to the EU regulation 2016/679 [40], any information that is related to an identifiable person is personal data and must, therefore, be in accordance with the GDPR. This includes the sEMG signals recorded by the band, which highlights the importance of data security. A clear vulnerability here is the wireless connection, which is susceptible to interception. The storage of personal data must also comply with the GDPR. These issues are exacerbated by the sensitive nature of biometric data. During this project, only the prototyping stage is treated. The risk of interception was therefore considered low. During the testing, all user data recorded is anonymized and given with consent and knowledge of how it will be used. Further, the extent of the presented data is limited, and identifying the test subject from this small dataset should be impossible. All data not presented is deleted after the relevant results have been documented.

An application of sEMG is in medicine, where it is used in post-stroke and neuromuscular rehabilitation [1]. For this application, it is crucial that the band's results are reliable. Unreliable readings could prolong rehabilitation time or even worsen the patient's condition. The band could also be used for prosthetics, where unreliable readings could cause prosthetics to make unintended movements, putting the user or others in danger. It is therefore important to once again note that the sEMG band developed in this project is a prototype and should under no circumstances be used in real-world applications, especially not for clinical use. If, at a later stage, the band is developed into a finished product, extensive testing would be required to ensure consistent performance under varying conditions.

To ensure the band's function, real sEMG data are needed. To acquire this data, human testing is conducted, which raises ethical dilemmas. To ensure that testing is conducted ethically, appropriate measures and considerations are in place. Primarily, sEMG is non-invasive. This means it monitors bodily functions with minimal impact. The circuit has a maximum voltage of 9 V, and no voltage will be applied to the electrodes, minimizing the risk of electrical injury. All components with applied voltage are enclosed in a casing to protect the user. Prior to testing, the circuit is controlled to ensure proper functionality. Thus, the risk of injuries related to the circuit is minimal. Further, the test subjects are well informed about the testing procedure, and all tests are done with their consent. The test subjects can also opt out of the testing at any time.

The development of sEMG technology also poses broader societal implications. These are not directly related to the sEMG band created in this project, but rather the implications that research on the subject could bring. Firstly, enabling the recording of biometric data raises several concerns about the ownership, access, and protection of user data. The data could be used maliciously to obtain sensitive user information, such as medical conditions or physical activity. This could be used to monitor the user or by insurance firms for dynamic pricing. Additionally, sEMG system performance can vary widely from user to user, depending on muscle mass, body fat, and skin properties, which may lead to discrimination. However, sEMG technologies can also be used to improve rehabilitation, enable muscle-controlled prosthetics, and facilitate human-machine interaction. These are all important fields that can improve human conditions, underscoring the value of the research, although ethical concerns remain important.

3.2 Electrode Design and Manufacturing

This section describes the design and manufacturing process for the electrodes. It was decided that the project would use a multi-channel bipolar sEMG method. Bipolar method refers to a configuration in which two electrodes are used to produce a single EMG signal [16]. The idea is that these electrodes have similar noise but different sEMG signals, so if the signals are subtracted, much of the noise cancels out while the sEMG signals remain. The term multi-channel refers to a band with several pairs of measuring electrodes that independently record EMG signals from different parts of the forearm. This is vital for distinguishing different wrist and hand movements. The band in this project has three channels, meaning that it features a total of three electrode pairs. In addition to these, bipolar measurements also require a reference electrode [16]. This electrode was placed on the elbow because there is low muscle activity in that area, resulting in fewer disturbances and a cleaner signal [18]. Since all three channels use the same reference electrode, the band features seven electrodes in total.

The design of sEMG electrodes requires careful consideration of several parameters that directly affect signal acquisition, noise sensitivity, and overall measurement quality [18]. One such parameter is the electrode material. To achieve the best signal, the material should provide low skin-electrode impedance and stable contact between the skin and the electrode [18]. This makes disposable Ag/AgCl wet electrodes an excellent choice for short-term EMG measurements [19]. They include a gel surface that improves electrode-skin contact and are secured to the skin with an adhesive that minimizes motion artifacts while ensuring good contact. Due to these advantages, this kind of electrode was used in various early tests. However, the gel and adhesive on wet electrodes dry out quickly after use, making them unsuitable for a reusable armband. The project was therefore limited to using dry electrodes for the final armband. Dry electrodes can be manufactured from many different materials, but the choice ultimately fell on stainless steel. It is mechanically strong, even when thin, and is resistant to corrosion [41]. While some materials have lower contact impedance, stainless steel offers sufficiently low impedance at an affordable price and has suitable properties for an armband. The steel used was AISI 304, chosen for its availability.

There were plenty of factors to take into account when deciding the geometry of the electrodes. It was decided early that the electrodes should be rectangular. This decision was made for practical reasons; there are no clear electrical advantages or disadvantages associated with the shape [18]. Using rectangular electrodes makes them slightly easier to manufacture and maximizes the electrode area in relation to the inter-electrode distance. When determining electrode size, there must be a balance between electrode impedance and pickup area [18]. Since the project uses dry electrodes with relatively high impedance, large electrodes are desirable. Due to the imprecise nature of an armband in combination with the high muscle density of the forearm, it might not be a downside to have a large pickup area. The measurements from each channel are thus not from a single muscle but rather from a general area of the forearm. To avoid spatial integration effects, the length in the direction of the muscle fibers was limited to 8 mm [18]. With these factors in mind, the electrode size was set to $15 \times 8 \times 0.5$ mm, with a thickness of 0.5 mm determined by the stainless steel sheet used. The inter-electrode distance was chosen to be 20 mm.

The stainless steel was purchased in a large sheet with the dimensions $500 \times 250 \times 0.5$ mm. Early prototypes were cut out using metal scissors. For the final electrodes, a water jet cutter was used. This allowed for high-precision cutting, making all electrodes nearly identical. The process of using the water jet included creating a CAD model of the electrodes.

To transmit the signal from the electrodes to the amplification circuit, a wire must be attached to the electrodes. Because of the oxide layer on the surface of stainless steel, conventional soldering methods could not be used without the use of aggressive soldering fluxes [42]. Due to safety concerns, 8330D silver conductive epoxy adhesive was opted for instead [43]. The stripped ends of the wires were carefully placed on top of the cut-out electrodes. Following this, the mixed two-part conductive epoxy was applied to the contact surface. The conductive epoxy was then left to cure for six hours. Lastly, a thin layer of liquid super glue was applied over the cured epoxy for extra protection. The finished electrode, with the silver epoxy connection, is shown in Fig. 3.2.

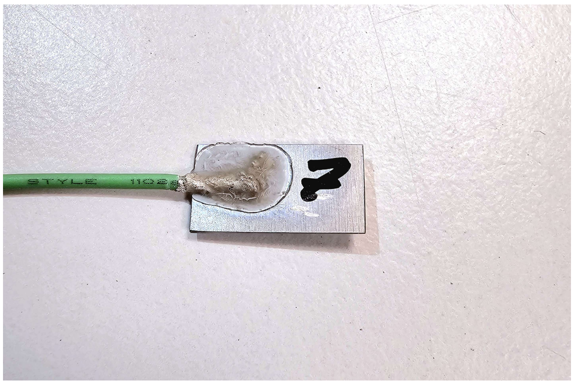


Figure 3.2: Picture of the electrode, with the silver epoxy connection.

3.3 Amplification and Analog Filtering

The recorded signals have low amplitude and are heavily noisy [16]. Amplification and filtering, therefore, play a vital role in improving the recorded signal. The first step is to amplify the signal from the electrodes. Since the bipolar method is used in this project, two electrodes are used to record sEMG signals. These two signals need to be amplified and subtracted from each other [16]. A component that achieves both these tasks is the instrumentation amplifier INA121 [44]. It first amplifies both signals independently using two separate OP-amplifiers, and then subtracts them using a third OP-amplifier, which amplifies the difference between the two signals. The function of the INA121 is illustrated in Fig. 3.3.

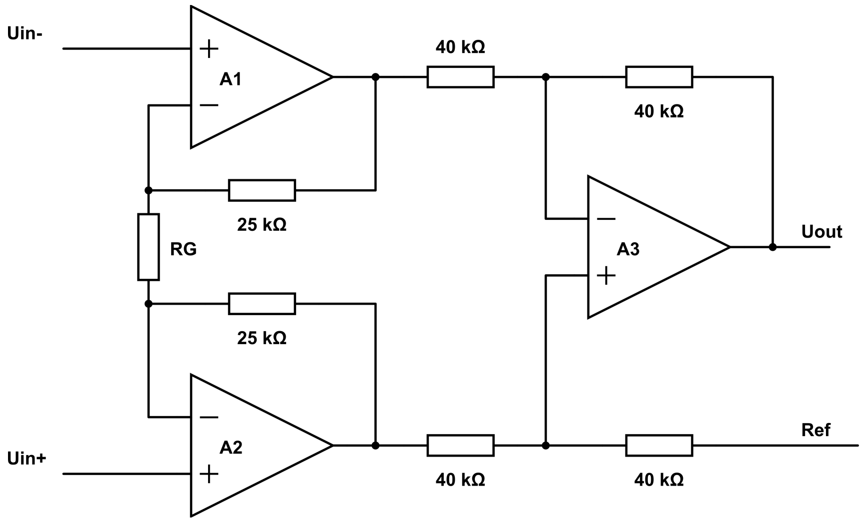


Figure 3.3: Simplified circuit diagram of INA121 [44].

Figure 3.3 shows that the input signals are first separately amplified by the A_1 and A_2 amplifiers [44]. The gain can be modified by changing the R_G resistance. The A_1 signal is then subtracted from the A_2 signal by the A_3 amplifier to produce the output signal. Since the reference input only affects the positive input of A_3 as shown in Fig. 3.3, its voltage is unaffected by the subtraction and passed on to the output signal. It can therefore be used to introduce a DC component to the signal. The gain of the INA121 is presented in [44] as

$$G = 1 + \frac{50\ k\Omega}{R_G} \tag{3.1}$$

and can thus be adjusted by changing the R_G resistance. For the armband, a potentiometer with a range of $0 - 10$ kΩ is used as the resistance. This, in theory, allows the range to be adjusted to a minimum of 6 times and a maximum of 10 000 times, which is the maximum gain of the INA121 [44]. Figure 3.4 illustrates how the potentiometer is included in the circuit, as well as the other connections to the INA121.

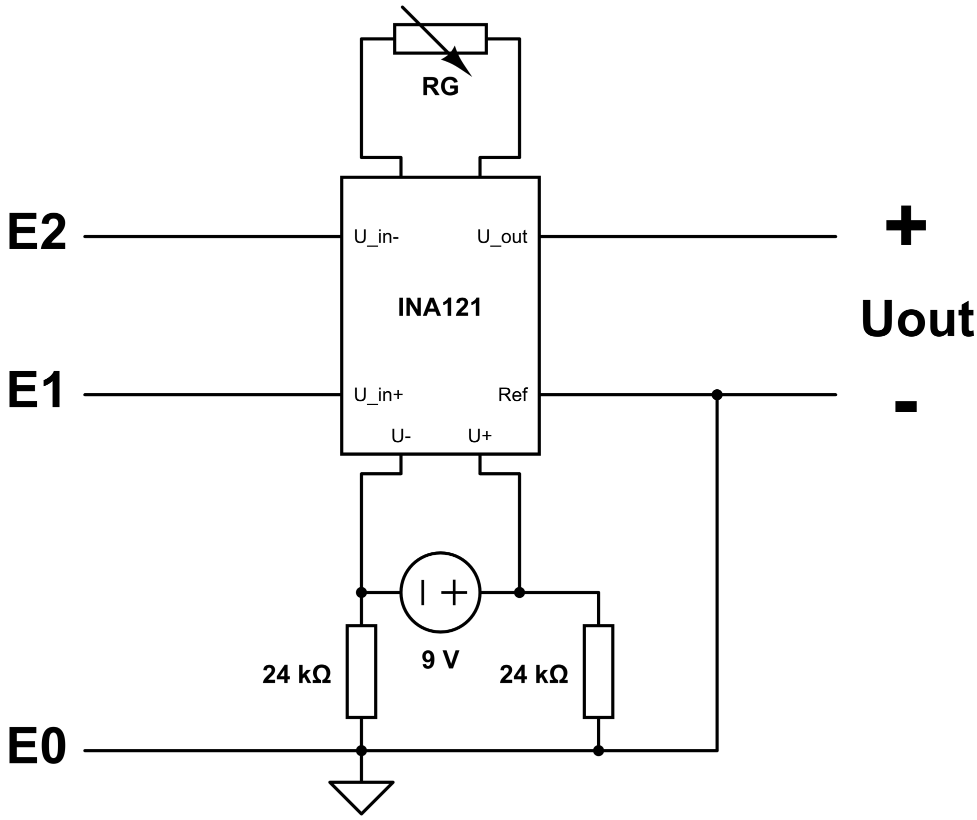


Figure 3.4: Circuit diagram for the first amplification step using INA121 instrumentation amplifier.

As illustrated in Fig. 3.4, the U_{in}^{+} port is connected to the E_1 electrode, and the U_{in}^{-} port is connected to the E_2 electrode. A 9 V battery powers the amplifier. The two 24 kΩ resistors act as a voltage divider. Therefore, the amplifier is given ±4.5 V in relation to the reference electrode E_0. This ensures that both positive and negative sEMG signals in comparison to ground are included in the output.

After the first amplification step, the signal is filtered. Since sEMG signals are low-amplitude, placing the filter before the INA121 might attenuate the signal too much before the first amplification stage. On the other hand, placing the filter after the last amplification step would cause unwanted noise to be unnecessarily amplified. Therefore, placing the filter between the amplification steps was determined to be the superior choice.

It was decided that the filter would be constructed as an RC filter. While other filters might have better characteristics, RC filters are easy to design and relatively compact while offering sufficiently good performance. Further, it was decided that the filter should be second-order to provide a sharper roll-off. Figure 3.5 illustrates the filter design.

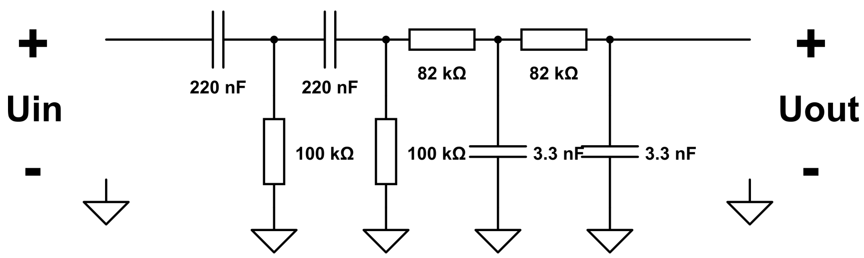


Figure 3.5: Circuit diagram of the second-order HP and LP filters.

As shown in Fig. 3.5, the filter begins with a second-order HP filter, followed by a second-order LP filter. The transfer function of the combined filter was calculated to decide the component values. With the assumption that two second-order RC filters in series can be described with the product of their separate transfer functions, their combined transfer function is

$$\boldsymbol{H}(j\omega) = \left(\frac{j\omega R_1 C_1}{1 + j\omega R_1 C_1}\right)^2 \cdot \left(\frac{1}{1 + j\omega R_2 C_2}\right)^2 \tag{3.2}$$

which gives the amplitude

$$|\boldsymbol{H}(j\omega)| = \frac{(\omega R_1 C_1)^2}{1 + (\omega R_1 C_1)^2} \cdot \frac{1}{1 + (\omega R_2 C_2)^2} \tag{3.3}$$

where $\omega = 2\pi f$, f is the frequency, R_1 and C_1 are the component values for the HP-filter, and R_2 and C_2 are the component values for the LP-filter. While the assumption made might be rather rough, 3.3 provides a sufficiently accurate description of the filter's function to determine the component values, since it is not that delicate a matter.

The main aspect to consider when deciding the component values for the filter is the cutoff frequency. It can be calculated graphically by plotting the transfer function 3.3 and finding the points where the amplitude has decreased -3 dB or $1/\sqrt{2}$ from the maximum value. According to [20], the HP filter should have a cutoff frequency of $10-30$ Hz, and the LP filter should have a cutoff frequency of $400-500$ Hz, or slightly below. To avoid aliasing, the LP filter should also have a cutoff frequency at most half the ESP32's sampling frequency, according to the Nyquist sampling theorem. The sampling frequency is 2 kHz. During early testing of the configuration, much high-frequency noise was detected. It was therefore decided that the cutoff frequency of the LP filter should be in the lower end of the recommended range. During the same testing, it was also discovered that a filter with too low impedance caused problems such as significant signal attenuation and shifted cutoff frequencies, most likely due to a high load on the INA121. Taking this into consideration, components with the marked values presented in Table 3.1 were chosen.

Table 3.1: Filter component values.

Component	Value
R_1	100 $k\Omega$
R_2	82 $k\Omega$
C_1	220 nF
C_2	3.3 nF

Using these values, the transfer functions of the filter were calculated according to 3.2. The amplitude of the transfer function was plotted and is shown in Fig. 3.6.

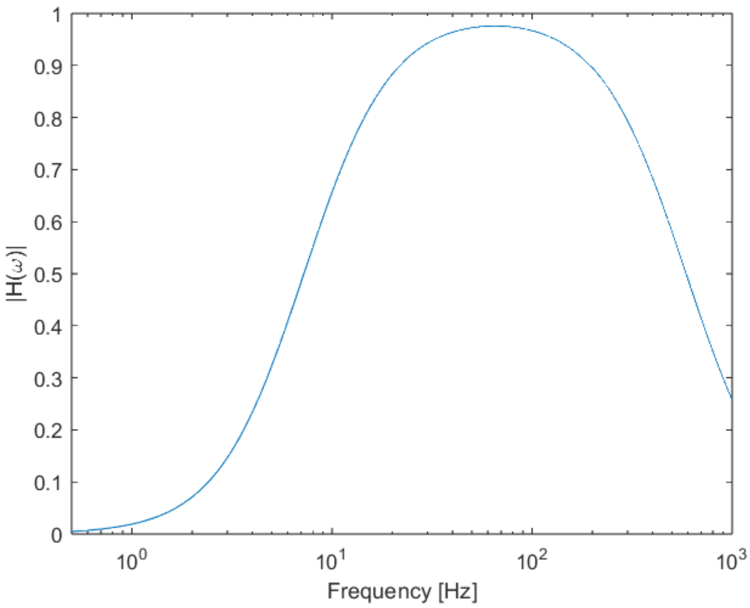


Figure 3.6: Theoretical amplitude of the transfer function for the filter in semilogarithmic scale.

The cutoff frequencies were calculated using a graphical method, in which the points where the graph had decreased by $1/\sqrt{2}$ from its maximum value were found. This gave an upper cutoff frequency of about 394.0 Hz and a lower cutoff frequency of about 10.8 Hz.

After filtering, the signal goes through a final amplification step. For this, the MCP6021 operational amplifier is used. This amplifier was chosen mainly for its low noise and rail-to-rail capability, meaning it can output signals across nearly the entire supply voltage range [45]. It is also designed to operate at low voltages and frequencies. The circuit used for this is illustrated in Fig. 3.7.

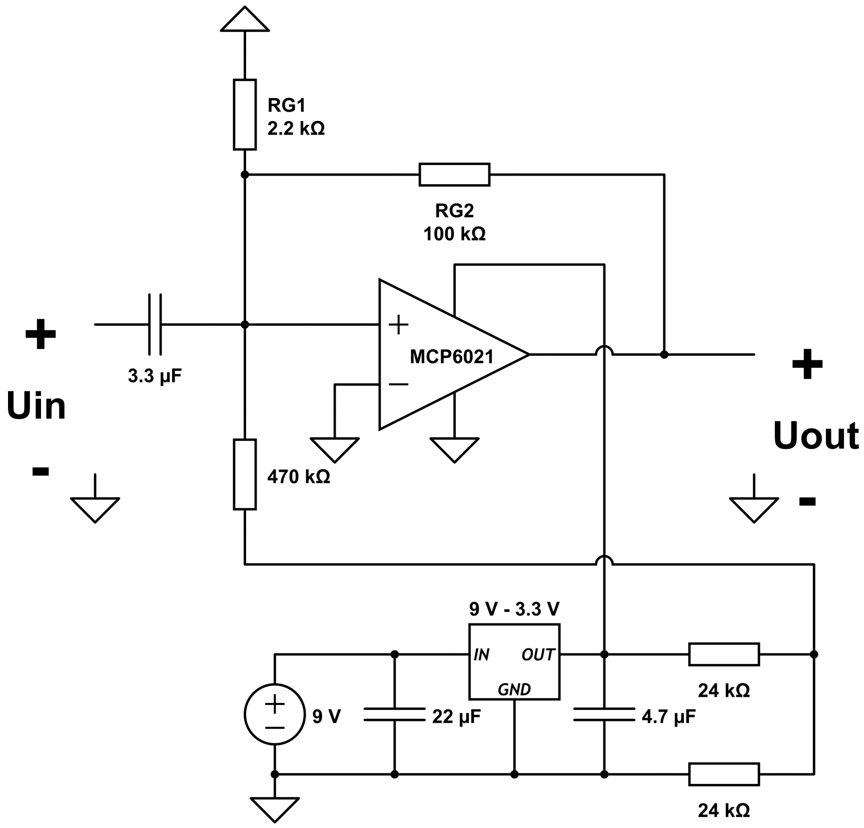


Figure 3.7: Circuit for the final amplification using MCP6021.

As shown in Fig. 3.7, the circuit begins with a 3.3 μF series capacitor. The purpose is to decouple this step from the rest of the circuit, so that only the signal can pass through. Since the circuit is a non-inverting amplifier, the gain is decided by the R_{G1} and R_{G2} resistors. Reference [21] suggests that the gain of ideal non-inverting amplifiers can be calculated as

$$G_2 = 1 + \frac{R_{G2}}{R_{G1}} = 1 + \frac{100k}{2.2k} \approx 46.5 \tag{3.4}$$

which is the theoretical amplification of the circuit. This means that the total theoretical amplification of the circuit, if the voltage drop across the filter is neglected, is

$$G_{tot} \approx G_1 \cdot G_2 = 46.5 G_1 \tag{3.5}$$

where G_1 is the amplification by the INA121, which can vary between $6 - 10\ 000$ times. In theory, this allows the gain to be adjusted between 279 and 465 000 times, although the upper range may not be accurate in practice. Since sEMG signals commonly occur in the range of hundreds of microvolts to tens of millivolts, and the desire is to amplify the signal to around 1 V, the expected amplification needed is somewhere around a couple of hundred to a couple of thousand times, which means that it should be within the possible range with this configuration [16].

The MCP6021 is supplied with 3.3 V on the positive and 0 V on the negative supply pin. This is to ensure it does not output signals exceeding 3.3 V, the maximum safe input level for the ESP32 [46]. To achieve this voltage, a voltage regulator is used. The capacitors on either side of the voltage regulator are used for stability. The MCP6021 is also supplied with 1.65 V on the reference pin. This adds a DC offset to the input signal, ensuring that both the positive and negative parts are amplified. Note that this was left out of the circuit diagram for illustrative reasons. The entire circuit diagram is presented in Fig. 3.8.

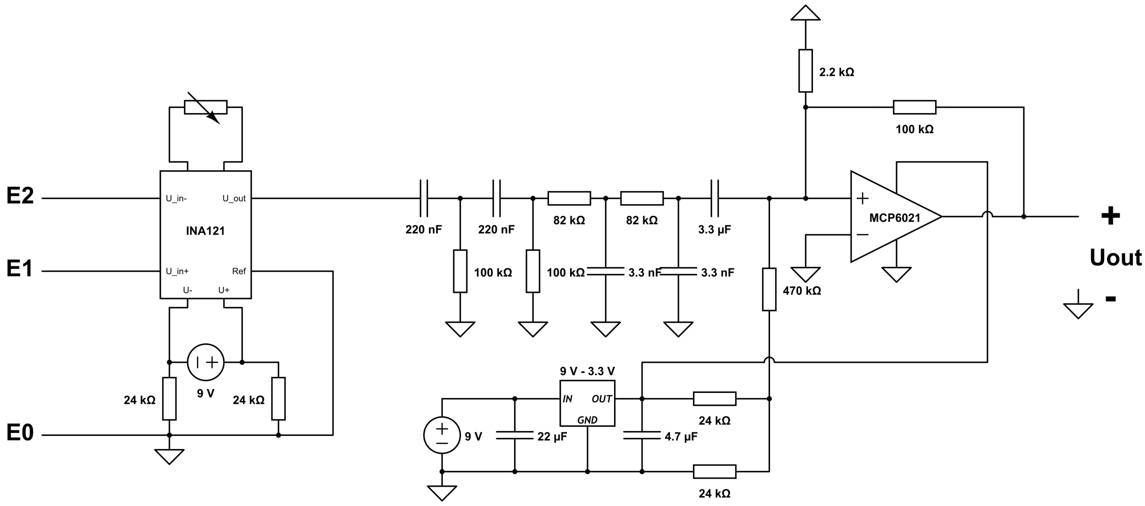


Figure 3.8: Circuit diagram for the complete amplification and filtering circuit.

The final armband features three of these circuits, one for each electrode pair. The circuits were realized first on a breadboard and then soldered onto prototype boards. They are all powered by the same two 9 V batteries and have their outputs connected to separate analog pins on the ESP32, which is also powered by the 9 V batteries through a simple circuit featuring a 3.3 V LD1117 voltage regulator [47]. The function of the voltage regulator is to convert the 9 V input into a stable 3.3 V output by dissipating excess power as heat [48].

3.4 Data Acquisition and Wireless Communication

The next step for the measured sEMG signal is to pass through an ADC and then be transmitted to the computer. This part of the project is the key to a smooth, reliable wireless connection, and requires carefully selected components and methods. Wireless communication can easily become a bottleneck for the project if not optimized for low data loss and quick responsiveness.

The ESP32-S3-DevKitC microcontroller is the component of choice. The board has several essential components for the project and is equipped with a Bluetooth and WiFi antenna operating in the 2.4 GHz band, a dual-core processor running at 240 MHz, and two 12-bit ADCs [46]. Furthermore, the microcontroller can be powered by 5 V or 3.3 V, which is important in this project because the band is completely wireless and battery-powered.

There are several options for wireless communication, including WiFi, Bluetooth, or ESP-NOW [46][25]. Traditional Wi-Fi communication was considered less suitable because it requires network configuration and connection management during operation, when ESP32 devices must simultaneously access another network. This can be solved using either Bluetooth or ESP-NOW, since they can operate without occupying the secondary device's Wi-Fi module. ESP-NOW was favored over Wi-Fi because this method of one-way communication does not use traditional Wi-Fi connection establishment procedures between the transmitter and receiver [25]. ESP-NOW can instead be used to send data directly to the receiver ESP32 device using its unique address, known as the media access control (MAC) address [25]. This communication method can be referred to as connectionless Wi-Fi communication, since it does not use the traditional setup and communication model, even though it uses the same hardware. For ESP-NOW communication in this application of building a wireless sEMG band, two ESP32 devices are required for this system to work. One connected to the electrodes in the band, and one to the computer. An advantage to this method is that no Wi-Fi or Bluetooth pairing is required on the computer, since the ESP32 manages the wireless communication and forwards the data to the computer via USB. This means the band has its own wireless system rather than relying on the computer's wireless transmitter and receiver, which can vary in performance between users. This makes ESP-NOW a more reliable choice over Bluetooth since the band can be its own system, independent of what it is connected to.

For programming the two microcontrollers, the Arduino IDE was used. The Arduino IDE is easy to use and set up, and has a large library ecosystem. The downside of the Arduino IDE is its entry-level-to-intermediate user interface design and layout. The Arduino IDE is not designed for professional users, as it lacks a developed code editor and advanced debugging tools. This makes the integrated development environment (IDE) less appealing for professional use and more suited for prototyping. Although Arduino is not the most advanced IDE, its capabilities are well within the requirements for this project. The IDE supports libraries for ESP-NOW wireless communication and for consistent ADC sampling using the ESP32 microcontroller's integrated hardware timer. These are two important requirements for this project: wireless communication and a consistent sampling timer.

The firmware was implemented using the Arduino framework for the ESP32 platform. The firmware code was written using the ESP-NOW protocol for wireless peer-to-peer communication [25]. A hardware timer was configured to trigger periodically at a fixed sampling rate. This ensured deterministic timing for the analog signal acquisition, independent of other tasks the ESP32 handles, such as communication. The timer configuration was based on the Arduino-ESP32 timer application programming interface (API) [49].

The code for the ESP32 microcontroller on the sEMG band is interrupt-driven using the hardware timer [49]. The measurements are taken with a frequency of 2 kHz, which means that the timer in the code starts the ADC sampling every 0.5 ms. The samples from the three channels are stored in a buffer that can hold 32 values from each channel. Each value is a 12-bit integer, but takes 16 bits when stored. A full buffer with 96 values or 192 bytes is transmitted to the receiver ESP32. The analog sampling is assigned to run on a different core than the ESP-NOW data transmission to the receiver ESP32. The ESP32 receiver waits for data to arrive, and the program runs only when data is available. The data is then stored in a queue that can hold up to 10 packages of data, 1 920 bytes. After a package enters the queue, the task of sending the data to the computer starts. In this stage, the data is labeled with a specific word to sync the computer with the data, so the channels do not get mixed if any data is lost in a package. The sync word is placed at the beginning of every packet and consists of the hexadecimal number 8080_{16}, which is 32 896 in decimal. This combination is used because it cannot occur in the data, even if one byte is lost. The task that receives and sends data to the computer via USB is programmed to run on either of the two possible cores.

3.5 Digital Signal Processing

With the electrodes, hardware, and wireless communication in place, the focus shifts to processing the recorded sEMG signals. While both MATLAB and Python were considered at the beginning of the project, Python was favored over MATLAB for the GUI due to its versatility and aesthetic GUI design options, which enable a more user-friendly interface.

The two signal processing methods used to process the recorded sEMG signals are RMS and CWT. Due to the heavy computational demands of the CWT, it was implemented in C++ and made available in Python via a .dll library. MATLAB was also used to record and store sEMG data in .mat files, and as a working tool to facilitate CWT analysis.

To ensure real-time performance and prevent data loss, multithreading was implemented. Since Python cannot run tasks simultaneously on different central processing unit (CPU) cores due to the global interpreter lock, multithreading improves efficiency by running tasks concurrently [50]. In this case, the background thread handles data acquisition from the serial port. Because it is an I/O-bound task, the program automatically switches threads while waiting for the next data packet to arrive. This allows the CPU-bound main thread to manage processing and the GUI. The Python script begins by initializing the GUI layout and performing the necessary data preparations. Then the background thread waits to receive the data transmitted from the ESP32 device via the USB port. When the sync word is found, the data is unpacked and stored in separate buffers depending on the electrode channel. In the main thread, each sample is scaled to voltage, and the offset added during amplification is filtered out. Instead of setting a fixed offset, it is estimated for each sample using exponential smoothing

$$s_t = \alpha x_i + (1 - \alpha) x_{i-1} \tag{3.6}$$

where s_t is the filtered sample, α the smoothing factor, x_i the current sample, and x_{i-1} the previous sample. The smoothing factor is $\alpha = 0.99$. The initial offset value was set to 1.65 V because it is in the middle of the ADC's sampling range. After removing the offset, each sample is processed and visualized.

3.5.1 Root Mean Square

RMS was chosen as one of the signal processing methods because it is commonly used to assess muscle activation in sEMG signals and involves basic mathematical operations. After scaling and offset removal, each recorded sample is passed to a separate function. It stores the sample in a buffer, updates the RMS sum, and calculates the latest RMS value. These values are stored separately for each electrode channel.

To achieve a good trade-off between a quick response to changes and a smooth curve, the number of samples in the analysis window was, after manual tuning, set to $N = 512$. Consequently, before the first RMS value is calculated, the data buffers for each channel have to be filled with 512 samples. When full, only the last value in each data buffer is updated, saving time and data power.

3.5.2 Continuous Wavelet Transform

The CWT was implemented in C++ to reduce its execution time, given its high computational cost. A high computational cost was deemed a valid trade-off because the sEMG data is processed on a computer that should have sufficient computational power. The CWT is a better choice than STFT for producing a frequency-time plot, which is a part of the GUI. According to the Nyquist sampling theorem, the CWT requires a sampling frequency at least twice the highest frequency of interest. Frequencies up to 500 Hz contain significant EMG signals [20], but because the anti-aliasing filters are analog, a frequency of 2 000 Hz was chosen. This ensures that the aliased signal is significantly attenuated before it folds back into the frequencies of interest.

The Morlet wavelet was chosen for its simple implementation and complex-valued nature, which suppresses oscillations in the graph for constant signals. The wavelet is divided into even and odd parts to enable the compiler to use single instruction, multiple data (SIMD) instructions, thereby speeding up the transform's execution time [51]. The wavelet was sampled in the interval

$$n \in \mathbb{Z} \cap \left[-\frac{N}{2}, \frac{N}{2} - 1\right] \tag{3.7}$$

where N is the length of the data array and n is the sample index. The FFT-based convolution in the CWT limits the size of the data array to a power of two. The time t, in seconds, is the sample number divided by the sampling frequency, as described in

$$t = \frac{n}{f_s} \tag{3.8}$$

Where f_s is the sampling frequency. The wavelet used is based on the scaled Morlet wavelet, but it has been modified. The first modification is that the scale factor is chosen as $a = \frac{1}{2\pi f}$. This is because the oscillation frequency should match the frequency being analyzed. The second change is that the scale factor in front of the exponential term is set equal to the frequency only. This scaling through testing made the transform's value proportional to the signal's energy across all frequencies. There is also no constant factor of $\pi^{-0.25}$, because the actual value of the wavelet transform is not important; only the ratio matters. Computational power can thereby be conserved. The even wavelet is calculated and zero-padded to double the length of the data array shown in

$$\psi_{even}[n, f] = \begin{cases} f \cdot \cos\left(\frac{2\pi f n}{f_s}\right) e^{\frac{-0.5 f^2 n^2}{f_s^2}}, & -\frac{N}{2} \leq n < \frac{N}{2} \\ 0, & \frac{N}{2} \leq n \leq \frac{3N}{2} \end{cases} \tag{3.9}$$

where f is the frequency being analyzed and $\psi_{even}[n, f]$ is the discretized, zero-padded real wavelet. Similarly, the odd wavelet is calculated using

$$\psi_{odd}[n, f] = \begin{cases} f \cdot \sin\left(\frac{2\pi f n}{f_s}\right) e^{\frac{-0.5 f^2 n^2}{f_s^2}}, & -\frac{N}{2} \leq n < \frac{N}{2} \\ 0, & \frac{N}{2} < n \leq \frac{3N}{2} \end{cases} \tag{3.10}$$

where $\psi_{odd}[n, f]$ is the discretized, zero-padded imaginary wavelet. Note that the square brackets denote that the function is discrete. The convolution is then performed via multiplication of the FFTs of the odd and even wavelets separately with the data $x[n]$, which is zero-padded to the length $2N$. The linear convolution can then be extracted in the interval shown in

$$X[n, f] = \mathcal{F}^{-1}\Big(\mathcal{F}(x[n]) \cdot \mathcal{F}(\psi[n, f]\Big), \quad n \in \mathbb{Z} \cap \left[\frac{N}{2}, \frac{3N}{2}\right] \tag{3.11}$$

where $X[n, f]$ is either the odd or even wavelet transform of the signal, and $\psi[n, f]$ is either the odd or even wavelet, respectively. These transforms are then combined to calculate the amplitude of the full transform seen in

$$|X[n, f]| = \sqrt{X_{odd}^2[n, f] + X_{even}^2[n, f]} \tag{3.12}$$

where $|X[n, f]|$ is the amplitude of the wavelet transform.

3.6 Movement classification

For movement classification, an LDA model from the scikit-learn library is used [52]. The classification process is divided into two phases: a training phase and a real-time execution phase. During the training phase, the data is recorded in MATLAB and then imported into Python to build the model. To classify wrist extension, flexion, and rest, three data sequences are required. One where the subject varies between rest and wrist extension, one where the subject varies between rest and wrist flexion, and one where the subject is resting. Each training sequence lasts 10 seconds, corresponding to 40 000 samples distributed across three channels. This includes the sync words from the ESP32 device, which are removed before processing.

To establish the training model, two datasets are required. The first dataset is a feature matrix, containing the RMS and CWT values from the recorded training data. Instead of considering only one frequency in the feature matrix, a broader frequency range is accounted for by calculating the mean of the CWT values over $50 - 450$ Hz. This range was chosen because motion artifacts generally contain frequency components up to 50 Hz, and the sEMG signal amplitude drops off notably at frequencies above $400-500$ Hz [20]. The second dataset is a target array indicating wrist extension, wrist flexion, or rest at each time step. The target labels in the target array are generated by analyzing resting data to determine the maximum RMS and CWT values, which are used as thresholds. Each sample in the training data is then compared against these thresholds to classify the signal as rest, wrist extension, or flexion.

The position of the electrodes of the band is strategically aligned with some of the muscle groups used for these specific movements [53]. Specifically, Channel 1 targets the Extensor Carpi Radialis, Channel 2 corresponds to the Extensor Carpi Ulnaris, and Channel 3 monitors the Flexor Carpi Radialis.

In the real-time code, every 50th sample is processed, and the RMS and CWT values are extracted and sent to the classification function. The classified movement is then displayed in the GUI.

3.7 Graphical User Interface

Python has many options for GUI libraries. The aim was to create a simple, user-friendly GUI that displays the processed data in real time. Consequently, PyQt-Graph was the most suitable option for this system due to its real-time capabilities and ease of implementation. To provide an overview of the processed data, it is visualized in various plots, as shown in Fig. 3.9 and 3.10.

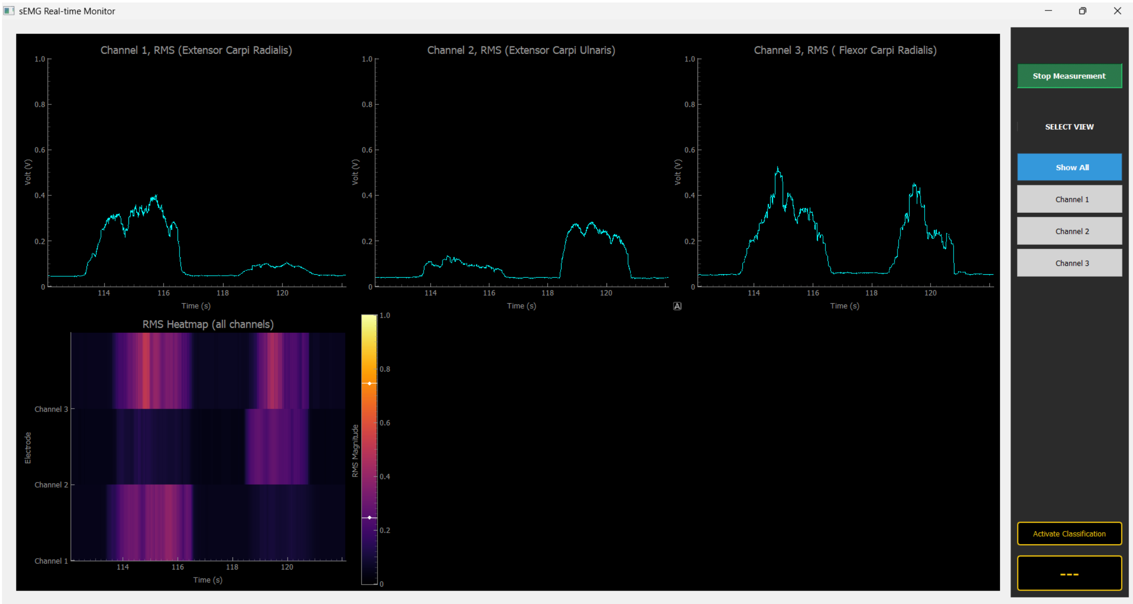


Figure 3.9: An overview of the RMS plots and heatmap for each channel in the GUI. The figure displays two contractions, resulting in two peaks in the RMS curve and increased color intensity in the heatmap.

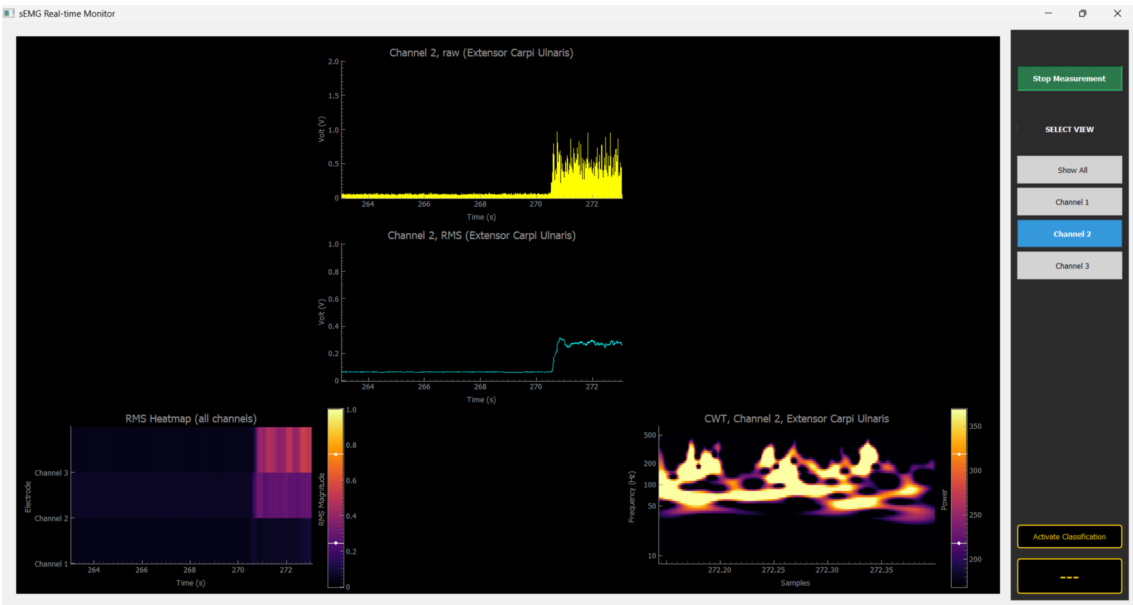


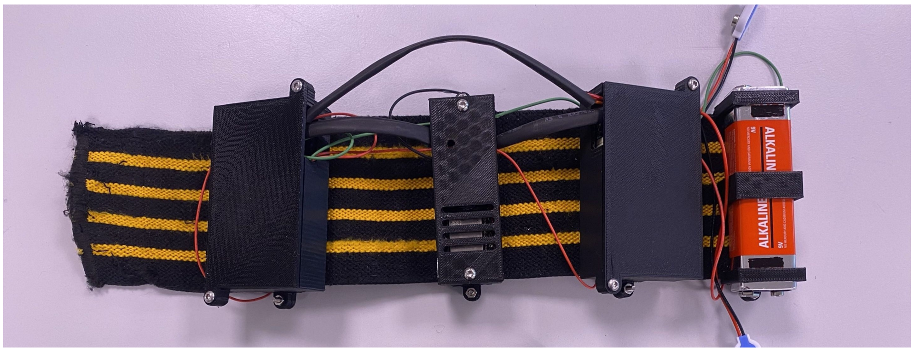
Figure 3.10: The plots for the filtered sEMG signal, RMS curve, heatmap, and CWT scalogram for Channel 2. The figure displays a contraction, resulting in a peak in the RMS curve and increased color intensity in the heatmap and scalogram.

For each channel, the raw sEMG data is displayed as a graph. The CWT is represented as a scalogram, with RMS values shown as a curve and in a combined heatmap of all channels simultaneously. With push buttons, it is possible to change views between the channels and have an overview of all RMS plots at once. The movement classification is also enabled through a push button, and the classified movement is displayed as a string.

Due to the heavy CWT calculations, only one CWT scalogram is shown at a time to prevent lagging. Both the scalogram and the heatmap have interactive color bars, allowing the user to perform real-time intensity scaling by manually adjusting the color scale's minimum and maximum values. Subsequently, the user can optimize the contrast to enhance weak muscle activations or prevent color saturation during high-intensity contractions.

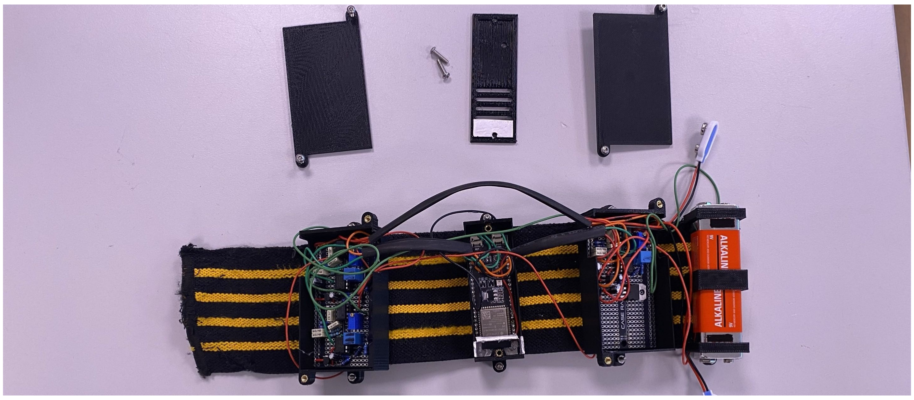
3.8 Design and Manufacturing of sEMG Band

The sEMG band was designed to be easily modifiable to support the project's iterative work process and the rapid prototyping of different layouts and components. The band is made of fabric with elastic rubber strings inside, making it stretchy. There are four modules around the band. Each module has two parts, one facing the arm and one facing out from the arm. The part facing the arm has electrodes mounted to it, and the other side holds the circuits or batteries as shown in Fig. 3.11.



(a) Outwards-facing side of the sEMG band, rolled out.

(b) Inwards-facing side of the sEMG band, rolled out.



(c) Outwards-facing side of the sEMG band with open circuits, rolled out

Figure 3.11: Inside and outside of the sEMG band with and without exposed circuits, rolled out.

Starting from the left side in Fig. 3.11, the first module has two amplification and filtering circuits and one electrode pair. The second module has the ESP32 device and the reference electrode. The third module has one amplification and filtering circuit, a voltage regulator circuit, and one electrode pair. The fourth module has two 9 V batteries and an electrode pair. All modules are connected by wire, since each circuit needs power and must communicate with the ESP32 device. The modules are designed in the CAD program Autodesk Inventor and 3D-printed in polyactic acid (PLA) using a fused deposition modeling (FDM) printer.

The purpose of the outwards-facing casings is to protect and hold the circuits and battery. Each casing has a lid fastened using M3 bolts and heat-set inserts. For the module housing the ESP32 device, a dedicated compartment for the antenna was designed. It has five of its six surfaces covered with aluminum foil to prevent transmitted power from interfering with the circuit.

The purpose of the inward-facing part of the module is to hold the electrodes in place. To ensure a stable skin-electrode contact, the electrodes are placed on an elevated surface at the desired inter-electrode distance. Two slots in the plastic ensure correct placement and raise the electrodes slightly over the plastic surface. The electrodes are attached with superglue and further secured with hot glue. Furthermore, integrated cable channels are built into the plastic piece to protect cables. To connect the modules, the under and over pieces are fastened with M2.5 bolts and nuts.

For the receiver device, a protective casing was designed using the same methods and materials as those used for the band's modules. This was done to prevent exposure of the ESP32 device, shielding it from accidental contact and potential damage. The assembled band, including all modules, is shown in Fig. 3.12.

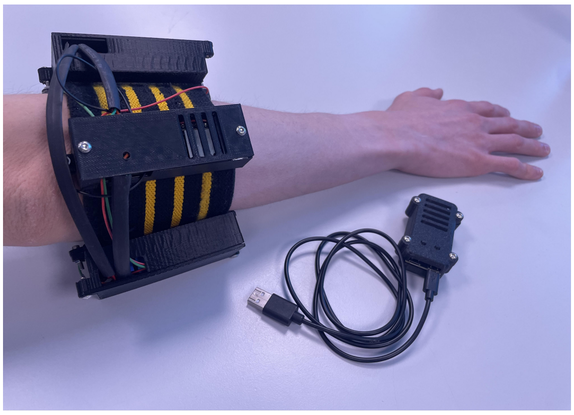


Figure 3.12: Subject wearing sEMG band with the receiver device next to it.

In the figure, the band is worn, with the ESP32 module on the elbow in the middle of the picture, and the modules at the top and bottom are amplification, filtering, and power-supply circuits. The battery module is not visible and is placed on the opposite side of the ESP32 device.

3.9 Testing and Verification

This section provides details on how the sEMG band was tested to verify its functionality and performance and to assess whether its purpose and goals were fulfilled. To verify the sEMG band's functionality, each of its subsystems was tested. The electrodes were compared with Ag/AgCl electrodes, and their impedance was measured. The circuit's filtering characteristics and amplification range were also tested. For wireless communication, both the latency and packet loss were analyzed. The function of the digital signal processing was verified by testing the CWT and RMS individually. Finally, a movement classification test was performed.

3.9.1 Testing Electrodes

To test and verify that the electrodes worked, they were compared to wet Ag/AgCl electrodes NM 3351 OFI by Ceracarta. One of the electrode pairs, as well as the ground electrode, was soldered to one of the amplification circuits. The circuit was then connected to the ESP32 device, which was connected to a computer via a USB cable. The area on the forearm where the electrodes were to be placed was marked. This was done to ensure that testing for all electrode pairs was as similar as possible. The first electrode pair and the ground electrode were then attached to the arm at the marked spots using sports tape.

The test was carried out by performing an MVC and recording the maximum amplitude. A program was made in the Arduino IDE to record the signal for five seconds. During this time, one MVC was performed. The maximum recorded value was then printed in the serial monitor. The values were presented as 12-bit integers, but they were converted to volts, with 0 corresponding to 0 V and 4095 corresponding to 3.3 V. This measurement was performed 10 times with 10 seconds of rest in between, and the average was calculated. The test was repeated once for every electrode pair. The same amplification and filtering circuit, with the same gain, was used for every electrode pair, once more to make every test as similar as possible and reduce sources of error. Lastly, the same test was performed using the Ag/AgCl electrodes. The constructed electrodes could then be compared with the Ag/AgCl electrodes to assess their performance.

In addition, the electrodes' impedance was measured. This was done by conducting 4-wire measurements using a Keysight 34465A multimeter. This was done to verify the electrode's connection to the wire, since a good connection would result in a low resistance.

3.9.2 Testing of Amplification and Filtering Circuit

Testing the amplification and filtering circuit consisted of two tests. The first test analyzed the circuit's frequency behavior. The primary purpose of this test was to assess the filter's performance and to determine whether the theoretical cutoff frequencies hold in practice. To perform this test, the NI Elvis II platform was used. It was connected to a computer using a USB cable. The amplification and filtering circuit was then connected to the platform using solid-core wires. A BNC cable with a probe in one end was connected to the platform. The probe was connected to the circuit's output. A frequency sweep was then performed from 1 Hz to 5 000 Hz, with the signal amplitude set to 10 mV. The output signal was plotted in a Bode plot across the frequency range, showing the filter's characteristics.

The second test analyzed the range of the circuits' gain. This was done by measuring the input voltage to the circuit and the output voltage from the circuit. Two Keysight 34465A multimeters were used, one measuring input voltage and one measuring output voltage. The input signal was generated by a function generator as a sine wave with a frequency of 100 Hz and an amplitude of 1 V peak-to-peak. This signal was passed through a voltage divider, constructed from a 1 MΩ and a 1 kΩ resistor, to reduce the amplitude before the circuit's input. This was done to avoid saturating the output signal. However, because the circuit's gain is high, the signal remains saturated at maximum gain. The voltage after the voltage divider, and the output voltage were measured to calculate the gain. It was calculated as

$$G = \frac{U_{out}}{U_{in}} \tag{3.13}$$

where U_{in} is the input voltage, and U_{out} is the output voltage. This was done three times. The first is with the potentiometer set to the lowest value, giving the circuit its highest gain. Then the potentiometer was set to the lowest value where the output signal was not saturated, and the input and output were measured. An oscilloscope was connected at the output to identify when the signal was saturated. Lastly, the potentiometer was set to its largest resistance, giving the lowest gain, and the input and output voltages were measured. Both tests were run three times, once for each circuit.

3.9.3 Testing of Wireless Communication

The performance of a wireless communication system is primarily determined by two factors: latency and data loss over time. The purpose of the latency test was to measure and evaluate the time required for the measured muscle signal to reach the computer. Specifically, the latency time includes the sampling time, the ESP-NOW transmission time, the USB data transfer from the microcontroller to the computer, and the time for the program to read it. The measured time does not include any signal-processing delays. This is due to the signal processing time varying with hardware and software implementations across different use cases.

Measured latency can be interpreted in relation to human perception of real-time feedback. In a test described in [54], feedback time as low as 45 ms on average between haptic and visual perception was detected by participants. Furthermore, the optimal controller delay time for prehensors was reported to be between $100-125$ ms [55]. Prehensors are the terminal devices of a prosthetic arm, responsible for grasping and interacting with objects.

To make this test possible, the transmitter code was modified, while the receiver code remained the same as for the sEMG band. The modification enabled the transmitter to send a batch of measurements on demand. One batch consists of 96 measurements, distributed evenly across three channels. Each measured value requires two bytes. Additionally, one full batch also requires two alignment bytes, for a total of 194 bytes.

A Python test program enabled the transmitter ESP32 to start data collection and send when a batch is full. The time is tracked by the test program from the moment the transmitter is notified to start its work process. The receiver then sends the data to the USB buffer for the test program to read. The test program unpacks a full data buffer from the receiver, which stops the time tracking. The latency is measured 100 times and averaged to obtain a reliable result; the highest and lowest values are also stored to analyze deviations from the average. During testing, the band and the receiver were 0.5 meters apart. This specific distance was chosen due to cable length limitations between the two ESP32 devices, since both devices need a wired connection to the computer. When using this method of measuring time, the time of sampling the data is included, as well as unwanted USB communication time. However, the USB communication time is assumed to be low and therefore neglected in the analysis of the test results. The sampling time can be calculated from the sampling frequency and the number of data points transmitted. The data is gathered with a frequency of 2 kHz, which means that the time between samples is 500 μs. This means the total time it takes to sample one full buffer is

$$T_{tot} = \frac{1}{f_s} \cdot n = \frac{1}{2000} \cdot 32 = 0.016s = 16ms \tag{3.14}$$

where T_{tot} is the total time, f_s is the sample frequency, and n is the number of samples per channel for one buffer. This means that every 500 μs, a sample from all three channels is taken.

To further verify the wireless system's performance, a packet-loss test was performed. The test was designed to determine whether any data packets are lost during wireless data transfer between the two ESP32 devices. Packet loss happens when a packet of data fails to reach its destination. This can be due to the environment or errors in the device. To verify that all data packets reach the receiver device, a test was conducted to determine packet loss during transmission.

For the test, each data packet was assigned a sequential identification number by the transmitting device before transmission. The receiving device then checked each packet's identification number to determine whether any packets had been lost in transit. The packet loss was calculated, and for every tenth packet, the packet loss was displayed in the Arduino IDE serial monitor.

The test was conducted indoors, with a clear line of sight between the devices, and the transmission was allowed to run for five minutes. Two tests were conducted at different distances. The first test was conducted with the ESP32 devices 0.5 m apart, both connected to the same computer, whereas the second test was performed at 7 m apart, with the devices connected to two separate computers. These distances were estimated as upper and lower limits for general use.

3.9.4 Digital Signal Processing Verification

To ensure that the RMS algorithm is correct, it was applied to a sine wave. The RMS value of a sine wave with peak amplitude U_{peak} is given by

$$U_{RMS} = \frac{U_{peak}}{\sqrt{2}}. \tag{3.15}$$

The sine wave was given the properties stated in Table 3.2, where the expected RMS value is also calculated.

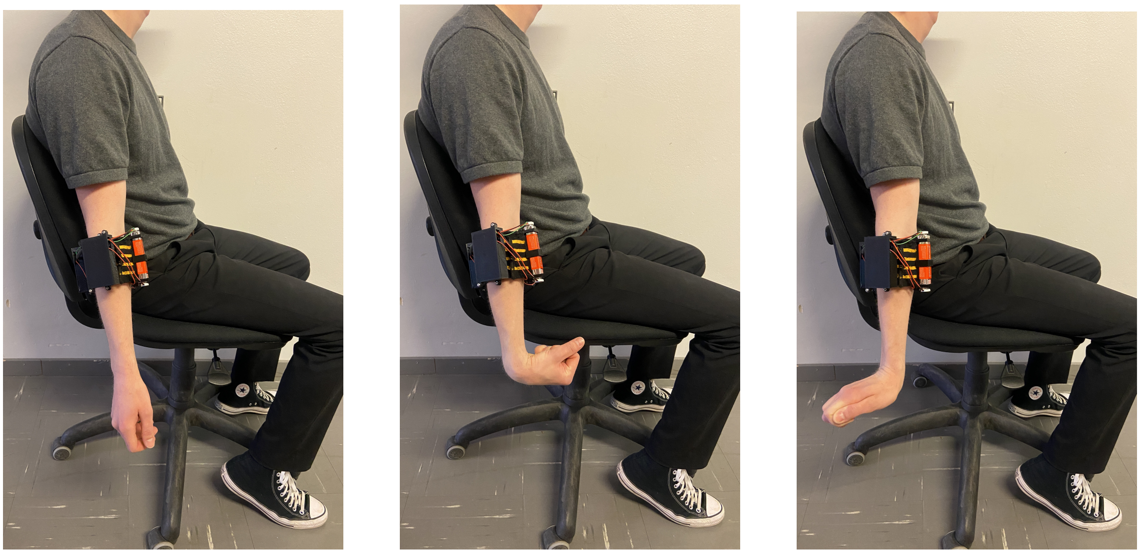
Table 3.2: Properties of the generated sine wave.

Sampling frequency	2000 Hz
Frequency	50 Hz
Amplitude	1 V
Offset	1.65 V
Expected RMS value	$\frac{1}{\sqrt{(2)}} = 0.707$

The wavelet transform code was tested by comparing it with MATLAB's built-in CWT function. The data used were synthetic and comprised 512 samples at a sampling frequency of 2 000 Hz. It included four fixed-frequency sine waves starting at time 0, with frequencies of 20, 50, 100, and 300 Hz, as well as a linear chirp function, where the frequency of the sine wave starts at 1.1 Hz and increases by 1.1 Hz per sample. All signals had the same amplitude of 1 V. In addition, uniform random noise between 0 and 0.1 V was added using MATLAB's rand() function. This was done to evaluate how the C++ CWT handles noise.

3.9.5 Movement Classification Verification

The movement classification test was performed on a single subject, beginning with the recording of training data in MATLAB. The subject was a young adult male. To account for session-specific factors, training data were recorded immediately before testing. For the duration of the recording, the subject sat in an office chair approximately 1 meter from the receiving microcontroller, with their arm relaxed at their side, as displayed in Fig. 3.13.



(a) Rest. **(b)** Wrist flexion. **(c)** Wrist extension.

Figure 3.13: The figure displays the subject's posture and the movements performed during the recording of training data and testing.

The subject is resting in Fig. 3.13a, performing wrist flexion in Fig. 3.13b, and wrist extension in Fig. 3.13c. For the training sequences including active movements, the subject performed three contractions per sequence, aiming to sustain each contraction for one to two seconds, while resting during the remaining time.

This data was subsequently used to set the thresholds and train the LDA model in Python. To ensure a balanced distribution of classes between rest and muscle activation in the target array, the thresholds were manually calibrated with static multipliers.

Once the model was trained, the subject maintained the same position as during the training phase and performed a total of 16 active movements. The testing was conducted in two stages, with the second one following immediately after the first to ensure consistent testing conditions. First, a sequence of a total of 6 movements, alternating between wrist extension and flexion, with rest between each movement. Second, a sequence of a total of 10 movements, five flexions and five extensions, performed in a non-sequential order, with each movement separated by rest. All sequences were screen-recorded in the GUI to facilitate analysis, which was performed after testing. Subsequently, each performed movement and its corresponding classification were documented in a confusion matrix. The confusion matrix was populated through an analysis of the screen recordings of the GUI. To ensure accuracy, the displayed real-time classification was cross-referenced with the corresponding RMS curves, which provided a visual representation of the contractions. By monitoring both the predicted class and RMS curves, it was possible to verify the intended movement against the classification.

4

Results

This chapter presents the results of the tests described in the section 3.9 Testing and Verification. The findings are presented as measured or calculated values and are visualized using tables and graphs. Key outcomes and observations from the testing process are also described.

4.1 Electrodes

The measured impedance for the electrodes and their connections to the wires is presented below in Table 4.1. Observe that the measurement uncertainty presented in the table comes from shifting reading values, which were greater than the uncertainty of the measurement instrument.

Table 4.1: Test of electrode and connection impedance.

Electrode	Impedance
E_1	$0.71 \pm 0.01\ \Omega$
E_2	$1.29 \pm 0.01\ \Omega$
E_3	$1.00 \pm 0.01\ \Omega$
E_4	$0.82 \pm 0.01\ \Omega$
E_5	$0.98 \pm 0.01\ \Omega$
E_6	$1.12 \pm 0.01\ \Omega$
E_0	$0.42 \pm 0.01\ \Omega$

Lower impedance indicates a better electrical connection between the two media. The results show that the electrodes have low impedance. This leads to the EMG signal being less attenuated by the electrode construction itself.

Table 4.2 presents the results of the maximum recorded amplitude for the electrodes when performing MVC. E_{12} represents the electrode pair of electrodes 1 and 2. Similarly, E_{34} refers to electrodes 3 and 4, and E_{56} refers to electrodes 5 and 6. Ag/AgCl are the wet electrodes used as a benchmark in the test.

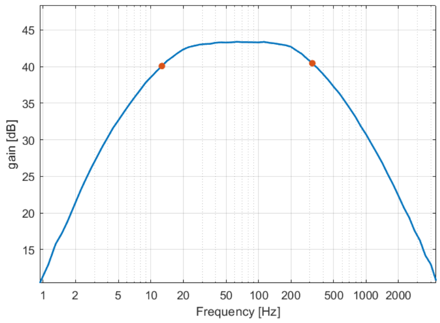
Table 4.2: Test of the electrodes' maximum amplitude for MVC measured in volts.

Peak	E_{12} [V]	E_{34} [V]	E_{56} [V]	Ag/AgCl [V]
1	1.82	2.00	1.85	2.02
2	1.89	1.91	1.81	1.91
3	1.87	1.87	1.77	1.91
4	1.89	1.98	1.82	1.86
5	1.87	1.75	1.90	1.84
6	1.92	1.93	1.88	2.08
7	1.75	1.83	1.88	1.97
8	2.13	1.81	1.90	2.45
9	1.77	1.86	1.83	1.83
10	1.75	1.81	1.75	2.09
Mean	1.87	1.88	1.84	2.00

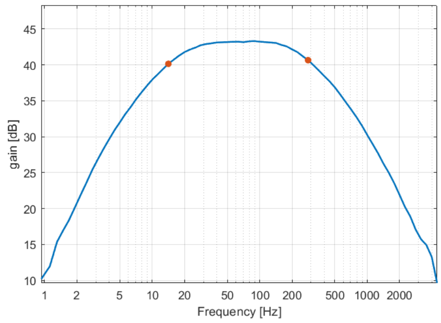
It is evident that the three electrode pairs, E_{12}, E_{34}, and E_{56}, give a similar amplitude as shown in Table 4.2. Meanwhile, the medical-grade Ag/AgCl electrodes yield a higher amplitude than the stainless steel electrodes.

4.2 Amplification and Filtering Circuit

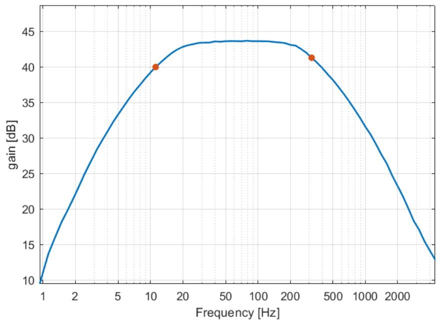
Figure 4.1 shows the Bode plots for the three filters between the frequencies 0 and 5 000 Hz. The main attribute is the cutoff frequencies, marked as red dots on the graphs. The y-axis is the gain in dB scale, and the x-axis is the frequency in logarithmic scale.



(a) Filter circuit 1.



(b) Filter circuit 2.



(c) Filter circuit 3.

Figure 4.1: Graphs showing the characteristics of the three filtering circuits in a Bode plot. The gain in dB is on the y-axis, and the frequencies, in a logarithmic scale, are on the x-axis. The red dots signify the cutoff frequencies.

As shown in Fig. 4.1a, circuit 1 has cutoff frequencies of around 13 Hz and 320 Hz. Circuit 2 has cutoff frequencies of around 15 Hz and 299 Hz, which can be seen in Fig. 4.1b. Lastly, circuit 3 has cutoff frequencies of around 13 Hz and 350 Hz, as shown in Fig. 4.1c. This means that the relative differences between the upper and lower cutoff frequencies are similar. The higher cutoff frequencies differ by 51 Hz between the highest and lowest, while the lower cutoff frequencies differ by 2 Hz.

Table 4.3 shows the maximum gain of the three amplification circuits. When the circuit was set to maximum gain, the output was saturated. Therefore, the gain calculated in this test is not the maximum gain that could be achieved with a lower amplitude signal. Observe that the measurement uncertainty presented in the tables comes from shifting reading values, which were greater than the uncertainty of the measurement instrument.

Table 4.3: Test of maximum gain. The RMS value was used for the voltages U_{in} and U_{out}.

	U_{in} [mV]	U_{out} [mV]	G
Circuit 1	0.53 ± 0.04	$1\,500 \pm 30$	$2\,380 \pm 221$
Circuit 2	0.48 ± 0.03	$1\,520 \pm 40$	$3\,170 \pm 215$
Circuit 3	0.53 ± 0.04	$1\,520 \pm 40$	$2\,870 \pm 229$

Table 4.4 shows the maximum gain for which the circuits did not saturate.

Table 4.4: Test of maximum gain without saturation. The RMS value was used for the voltages U_{in} and U_{out}.

	U_{in} [mV]	U_{out} [mV]	G
Circuit 1	0.53 ± 0.03	870 ± 60	$1\,640 \pm 146$
Circuit 2	0.47 ± 0.03	880 ± 70	$1\,870 \pm 191$
Circuit 3	0.52 ± 0.04	780 ± 70	$1\,500 \pm 177$

The last test was measured with the lowest possible gain, as shown in Table 4.5.

Table 4.5: Test of minimum gain. The RMS value was used for the voltages U_{in} and U_{out}.

	U_{in} [mV]	U_{out} [mV]	G
Circuit 1	0.53 ± 0.04	28 ± 2	53 ± 5
Circuit 2	0.48 ± 0.03	28 ± 4	58 ± 10
Circuit 3	0.53 ± 0.04	30 ± 4	58 ± 9

In both Tables 4.3 and 4.5, circuit 2 has the highest gain, circuit 3 the second highest, and circuit 1 the lowest gain. In Table 4.4, circuit 2 also has the highest gain, but circuit 1 has the second highest, and circuit 3 has the lowest.

4.3 Wireless Communication

The measured delay, from sampling on the ESP32 device to the computer, is presented below. Since it is possible to estimate the time required to gather all data points for a single full packet, two types of measurements are presented in Table 4.6. The total time measured is t_1. This includes sending the start instruction, sampling, wireless communication, sending the full packet to the computer, and the program reading it. The sampling time is calculated in section 3.9.3, and t_2 is defined as t_1 minus the sampling time, 16 ms.

Table 4.6: Latency for wireless transmission.

Type of data	t_1 [ms]	t_2 [ms]
Average	20.8	4.8
Minimum	19.3	3.3
Maximum	22.9	6.9

The table includes the average time, as well as the highest and lowest measured values, from a dataset of 100 measurements. The highest and lowest values show the deviation and the spread of the measured values around the average time.

The packet loss test yielded the values shown in Table 4.7. Two tests were made at different distances, 0.5 and 7 meters. Both tests were performed for 5 minutes.

Table 4.7: Packet loss at 0.5 m and 7 m of the wireless system.

Distance	0.5 m	7 m
Packet Loss	0.00%	0.00%

No data packets were lost during transmission in either test.

4.4 Digital Signal Processing

Figure 4.2 displays the result from applying the RMS algorithm to a generated sine wave. The left plot presents the input sine wave with the offset filtered out. The right plot presents this sine wave with the applied RMS algorithm.

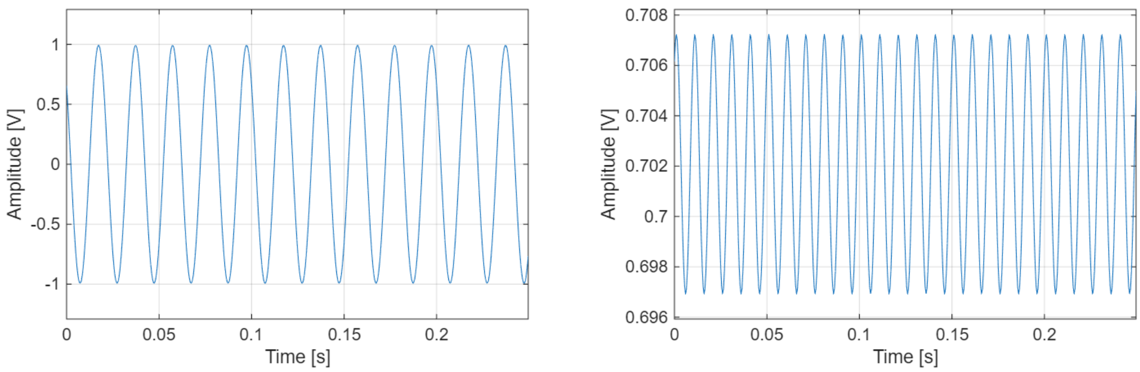


Figure 4.2: The left plot presents the input sine wave with the offset filtered out. The right plot presents this sine wave with the applied RMS algorithm.

The RMS value oscillated between 0.707 V and 0.697 V with a mean of 0.702 V. The offset has been filtered out, causing the unprocessed sine wave to center around 0.

The results from the comparison of MATLAB's CWT and the C++ CWT are shown in Fig. 4.3.

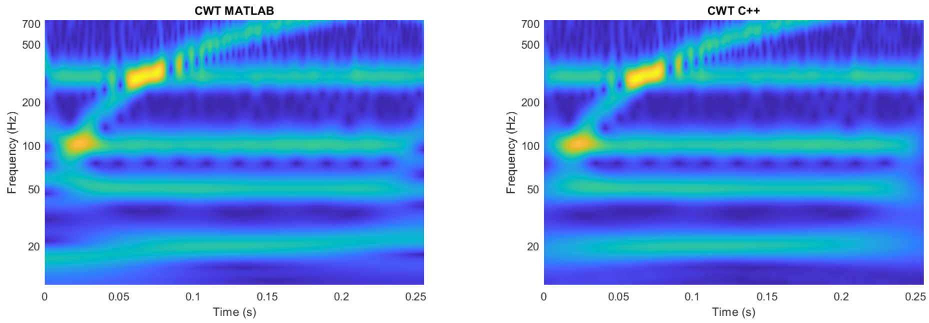


Figure 4.3: MATLAB's built-in CWT function to the left and the CWT made in this project to the right. The transformed data consists of 512 sample points from 4 fixed-frequency sine waves, a linear chirp, and some uniform noise.

The two methods produced visually similar results, which indicates that the C++ CWT works correctly. Figure 4.3 shows that the MATLAB method has frequency distortions toward the edge of the transform. The C++ method exhibits greater frequency-resolution loss at the edges.

4.5 Movement Classification

The results from the movement classification are presented in the confusion matrix in Table 4.8. The left-most column presents the performed movements, and the top row presents the classified movements.

Table 4.8: Result from movement classification.

	Classified Rest	Classified Extension	Classified Flexion	Total Performed
Performed Rest	17	0	0	17
Performed Extension	2	6	0	8
Performed Flexion	0	0	8	8
Total Classifications	19	6	8	33

Six out of eight wrist extensions were correctly classified as wrist extensions. Two out of them were misclassified as rest. All performed wrist flexions and rest were correctly classified. As the matrix indicates, 31 out of 33 movements were correctly classified. The accuracy can be calculated according to

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of classifications}} = \frac{31}{33} = 93.9\% \tag{4.1}$$

which results in a 93.9% accuracy.

5 Discussion

This chapter provides an analysis and interpretation of the results presented in the previous chapter, 4 Results. The findings are discussed in relation to the project's objective and relevant theory. In addition, the implications, limitations, and potential sources of error are examined. Finally, possible improvements outside the scope of the project are discussed.

5.1 Electrodes

The impedance between the electrode plate and the end of the connected wire is low, as shown in Table 4.1, which indicates a good electrical connection. Low impedance is important, as high impedance can attenuate the signal and introduce additional noise, potentially lowering the SNR. However, it is important to note that these impedances are only between the electrode plate and the connected wire, not between the skin and the electrode. The latter has a higher impact on the signal quality and is influenced by factors such as electrode material, contact pressure, and local skin properties [19]. It is typically lower with wet electrodes, such as the Ag/AgCl electrodes used as a benchmark in the tests.

The three electrode pairs show similar mean MVC amplitudes of 1.87 V, 1.88 V, and 1.84 V, respectively. This indicates a good consistency and repeatability between the different electrode pairs. While the average of the ten measurements shows similar values for each of the three electrode pairs, the individual MVC trials exhibit more varied values, as shown in Table 4.2. A contribution to this variation is physiological factors such as the muscle being contracted with different force or muscle fatigue. Additionally, minor shifts in the electrodes' position relative to the muscle fiber may have contributed to differences between tests.

Compared with the three electrode pairs, the Ag/AgCl electrodes yielded a higher mean amplitude of 2.00 V. One of the main reasons for this is the gel on the Ag/AgCl electrodes, which reduces the impedance between the skin and the electrode. Lower skin-to-electrode impedance yields a higher signal quality.

Another factor that affects the result in this test is the size of the electrodes. The Ag/AgCl electrodes have a smaller surface area than the constructed electrodes. This gives the Ag/AgCl electrodes a disadvantage in terms of skin-to-electrode impedance [18]. However, this is balanced by wet electrodes generally having lower skin-to-electrode impedance. No test was performed to test the skin-to-electrode impedance. Therefore, the exact impedances of either electrode are unknown. The smaller size of the Ag/AgCl electrodes offers an advantage in providing a more precise pickup area and lower spatial integration [18].

The two conducted tests suggest that the manufactured electrodes perform well in terms of signal amplitude. However, it is important to note that these tests did not account for effects such as motion artifacts or SNR. While it does provide some information on the electrodes' performance, a more thorough test would be required to determine whether they perform as intended.

5.2 Amplification and Filtering Circuit

The amplification and filtering circuit has the important task of turning the weak, noisy sEMG signal from the electrodes into a clearer, more distinct signal. It turned out to be difficult to quantify the circuit's performance since it is heavily dependent on both the electrodes' performance and the software interpreting the signal. Therefore, the two tests served to isolate the circuit from the rest of the band's components.

The first test was specifically designed to evaluate the filter's performance. The results show that the cutoff frequencies differed considerably between the circuits. The upper cutoff frequency is $299-350$ Hz, while the lower cutoff frequency is $13-15$ Hz. This most likely has to do with the non-ideal nature of analog components, where the component value can differ from the marked one. Differing impedances will result in differing cutoff frequencies. Soldering can also affect filter performance, with a poor solder connection slightly increasing the component's impedance. What is clear, however, is that all the circuits have narrower band gaps than in the ideal case. This is most likely due to the different RC steps loading each other. This effect is not accounted for in theoretical calculations but occurs when multiple RC steps are placed in series without a buffer between them [23]. The risk associated with a too-narrow band gap is that parts of the sEMG signal get filtered out as well. With this, valuable information that could help classify the movement is attenuated.

When assessing the accuracy of the filter test, it is important to keep in mind its purpose: to ensure that the filter has cutoff frequencies in the desired regions. High accuracy was therefore not a priority. While the NI Elvis II platform might not be the most accurate, its results are sufficient for this purpose.

The purpose of the second test was to obtain a rough estimate of the circuit's maximum and minimum gain. This was done to ensure that all circuits could be adjusted to amplify sEMG signals of varying amplitudes. As previously stated, EMG signals typically range from hundreds of microvolts to tens of millivolts [16]. Those signals should be amplified to lie around ± 1.65 V and not exceed the $0-3.3$ V range. The test shows that the circuits have a minimum amplification of $53-58$ times and a maximum amplification of $2\ 380-3\ 170$ times, indicating that they should be capable of amplifying EMG signals to the desired amplitude. However, the amplification range differs considerably from the theoretical range of $279-465\ 000$

times.

The differing upper limit primarily reflects a flaw in the test. To test the amplification, a signal was generated to mimic an EMG signal. It was set to an amplitude of about 1 mV and a frequency of 100 Hz. This meant that at high amplification, the signal would saturate the MCP6021, which was supplied with a $0 - 3.3$ V supply. The true maximum amplification could therefore not be measured. There are two reasons why this method is used despite its flaws. The first is that the circuit should be tested with signals it is most likely to encounter in its intended use case. The second reason is that generating a small enough signal to test the true maximum amplification would be very difficult with the available resources. The signal would have to be in the microvolt range, and in that region, the background noise will be larger than the signal.

The signal's small amplitude also explains the high measurement uncertainty. To improve this, the test would need a better signal generator for small amplitudes and a measurement environment with much lower noise. However, for the same reasons as in the filter test, high accuracy is not a priority. The test ensures that the circuit can deliver the desired amplification, rather than indicating the absolute range of amplification.

Another contributor to the lower-than-expected amplification is the filter. In the calculations, it is assumed that the filter causes no attenuation [23]. However, because there is no buffer between the RC steps, the steps are affected by each other's impedance. Each step of the filter becomes parallel to the others, causing a greater voltage drop across the series components than usual. The drop-off can be negated by increasing the amplification in the first step, but to make the circuit behave more like the calculations, hardware changes to the filter would be needed.

During testing, it was discovered that the battery charge level can affect the results. When new 9 V batteries were used, the noise was noticeably higher compared to slightly used batteries. This is most likely due to thermal throttling in the voltage regulator supplying power to the ESP32 device. Hundreds of milliamperes can be drawn by the ESP32 device [46], and the voltage regulator lowers the voltage by dissipating excess power as heat [48]. Higher input voltage results in greater power dissipation, potentially leading to overheating and explaining the increased noise.

Another issue present was that the noise floor could change when objects came close to a channel's circuit. A possible reason for this is that the ground has high impedance relative to the body. This problem might be solved by using a bigger ground electrode. The amplification and filtering tests do not reflect this problem because the circuit is connected to the electrical ground, but the problem becomes apparent when the circuit is integrated into the complete system.

Despite these shortcomings, the amplification and filtering circuit appears to provide signals of sufficient quality for further processing under favorable conditions. The isolated tests indicate that the circuit is capable of providing both amplification within the intended range and filtering with a passband relevant for sEMG applications. While these tests were performed using generated signals rather than sEMG data, the movement classification test suggests that the circuit also functions adequately when integrated into the complete system. In the test, 31 out of 33 movements were correctly classified. However, this result alone cannot be used to validate the performance of the amplification and filtering circuit, since the classification accuracy depends on the entire processing chain. Further, a classification algorithm could potentially adapt to systematic imperfections in the circuit if they are consistent, thereby correctly classifying movements despite a highly flawed circuit. Nevertheless, when considered alongside the isolated circuit tests, the results suggest that the circuit performs sufficiently well to identify and classify distinct movements, although further improvements are still necessary.

To improve the circuit's performance and bring it closer to its theoretical limits, several key improvements could be made. One of them is to use active filters instead of the currently used series RC-steps. This would both bring the cutoff frequencies closer to the calculated values and ensure that there is no unexpected voltage drop across the filter. Other changes could also be made to the filter, such as increasing its order. A higher-order filter has a steeper cutoff, reducing the attenuation of desired signals and increasing the attenuation of unwanted signals. Another important change would be to realize the circuit on a printed circuit board (PCB) rather than on a prototype board. A PCB is more reliable and would minimize the differences between the circuits. It would also have the added benefits of being easier to mass-produce and potentially having a smaller footprint, making the entire band more compact. In addition, to further reduce the footprint, surface-mounted components could be used.

5.3 Wireless Communication

The results of the delay test are shown in Table 4.6. The measured average total latency t_1 was 20.8 ms, of which approximately 16 ms was attributable to sampling time. This leaves an estimated communication and transfer delay t_2 of 4.8 ms. These results indicate that the majority of the measured latency originates from the sampling process rather than from wireless transmission, USB communication, and computer delays. This reasoning relies on the assumption that the sampling time is ideal, which may not hold in practice.

The minimum and maximum values of t_1 deviate from the average by 7.2% and 10.0 % respectively. This indicates that the system exhibits stable, consistent latency under the test conditions. For the calculation of the percent deviation from the average value, the time t_1 is used, since estimating the sample time to be exactly 16 ms is unnecessary. If the receiver ESP32 device becomes overloaded, data handling can slow down. Inside the receiver, a ten-slot queue is used to handle longer delays and prevent data loss. In the worst-case scenario, while still avoiding data loss, data delay will result in a time delay of approximately ten times the measured average. During testing, this has never occurred as a problem, but it is a possibility if the ESP32 device or the computer malfunctions.

From an application perspective, a total latency of approximately 20 ms is acceptable for this real-time monitoring of muscle activation use case [54]. For time-sensitive applications, such as precise real-time control systems, this delay can fit within the optimal usable level for prosthetic prehensors [55]. Note that the measured 20 ms delay is not the entire system's delay. This means that the latency might need to be reduced further when implemented in a real-time control or monitoring system. The latency comparisons are therefore not very accurate, since the communication time is compared against the full system's optimal delay.

Several limitations in the measurement method should be noted. The test includes the time required to send a start command from the computer, which introduces additional delay not present when the system is running continuously. These delays occur during USB communication with the ESP32 transmitter device. They are included in the times t_1 and t_2 in Table 4.6 because they are difficult to measure independently. Furthermore, the delay is expected to be shorter when running continuously, as both cores on the ESP32 are used simultaneously. The ESP32 can then handle both data sampling and transmission, or receive data and send it to the computer in parallel. The fixed distance of 0.5 m between the two devices at which the measurements were taken limits the ability to generalize the result further. Increased distance or environmental interference could impact the wireless transmission time.

In addition, the sampling time is dependent on the chosen sampling frequency and buffer size. A higher sampling rate or smaller buffer size could reduce latency, but may introduce trade-offs such as increased data loss.

Another consideration is the method for tracking the time between sending a start command and receiving a full data packet. This was done in Python; the script includes a function that checks whether all expected data has been received by the computer. To avoid overloading the CPU, a 1 ms delay is used between checks. This means that if the data arrives exactly when a new sleep function starts, the actual and measured times will differ by 1 ms. For more accurate measurements, it might be preferable to reduce the sleep function. Time tracking is done using perf.counter() function. This means that time tracking relies on the function's accuracy, which should be sufficiently low.

The results for the two packet loss tests showed no packet losses in either test, as shown in Table 4.7. This indicates that the wireless communication is stable in open spaces at short distances. Tests at longer ranges or in varied terrain pose a higher risk of packet loss. As mentioned in section 3.9.3, none of those situations have been considered for the use case of this project.

To improve the verification process of the transmission, the packet loss could have been accompanied by a data integrity test. This would have provided an oversight into how well the data in the packets retains its values during transmission.

Future tests could increase the range to that of a football field or an athletics track to verify whether the band is viable for application on a larger scale in sports. The band's performance in less open environments, such as a forest or a crowded city, could also be further examined to enable a wider range of applications.

5.4 Digital Signal Processing

The RMS algorithm was used to extract the sEMG signal intensity corresponding to muscle activation. By generating a sine wave with known properties and applying the RMS algorithm to it, the aim was to verify the algorithm. The result from this analysis is displayed in Fig. 4.2. The oscillations of the RMS value partly occurred because the window length did not correspond to an exact integer number of signal periods, resulting in window ripple. Since the offset estimation works like an HP filter, the RMS value is slightly lower than expected. Hence, a larger estimated offset results in a higher attenuation of the RMS value. Despite these minor fluctuations, the RMS algorithm remains consistent with the theoretical expectations, confirming that it works as intended.

The two implementations of the CWT are very similar. However, the C++ implementation exhibits fewer frequency distortions toward the cone of influence but loses more frequency resolution toward the edge. Reducing distortions is beneficial in this case, since distorted frequency data can provide false information about muscle contraction. Some improvements to the CWT could include introducing functions to extract data points near the cone of influence. The wavelet transform could also be implemented as a shader or with multithreading to significantly improve runtime performance.

5.5 Movement Classification

Although the movement classification achieved high accuracy during testing, this performance was heavily dependent on manually calibrating the RMS and CWT thresholds to account for varying physical and environmental factors. It was noted that a shift in arm position from a supported table during the training phase to an unsupported posture during testing had a significant impact on the classification, making the thresholds ineffective. Furthermore, the battery charge level of the sEMG band proved to be a critical factor: a higher charge level increased the noise floor, causing the algorithm to classify noise as muscle activation based on the manually set thresholds. However, the increased offset for the RMS values affected only specific channels, potentially due to slight differences in their circuitry. In addition to hardware imbalances, physical factors such as electrode placement, skin-to-electrode contact, or crosstalk from adjacent muscles can lead to inconsistent baseline offsets for the RMS values. These factors further emphasize that the model lacks adaptability, as it is tailored to the specific conditions of the recordings.

To address the dependencies in the movement classification, one important factor is the hardware. The amplification and filtering circuit currently suffers from highly variable results due to varying environmental factors. To improve the repeatability of movement classification, the received signal must be more consistent. Another improvement would be to streamline the training process for the LDA model. Instead of recording the training data separately, integrating that process into the GUI would make it easier to account for the physical and environmental factors during that session for that specific subject. Also, the features used in the model, the RMS values, and the mean of CWT values between 50 and 450 Hz, could be further assessed. For instance, the frequency band used to calculate the mean CWT value could be divided into several narrower bands or into a single narrower band. This would capture more detailed data during contractions. Instead of averaging data from the entire window, using the latest data point within the cone of influence would be more accurate. Likewise, the model could benefit from a time-domain feature that is more robust to noise.

The proposed improvements concern both signal acquisition and processing, and the classification system relies on both the software and hardware developed during this project. Hence, it is challenging to isolate the primary source of these limitations with complete certainty. Using a commercial sEMG device in combination with the current classifier could help identify whether the main issues are rooted in the software or hardware. However, these factors do not change the classification model itself. It is important to keep in mind that using an existing LDA library entails certain mathematical assumptions that may affect classification. While creating the LDA classifier from scratch would offer complete customization, more critical issues were identified in the classification process. To avoid recalibrating the model for each session, a neural network-based classification model is an option. It can recognize patterns in data and would be adaptable across multiple subjects. However, that would require a more advanced setup and a substantial amount of training data to generate a robust trained model.

5.6 Further Improvements

This section introduces and discusses potential further improvements for the band that are outside the scope of this project. One such improvement is to the band's physical design. In its current form, the band is neither very practical nor very comfortable to wear. While this is suitable for a prototype, a finished product would benefit from a more compact and ergonomic design. Several approaches can improve this aspect. Currently, the amplification and filtering circuit constitutes the largest part of the overall system volume. As previously discussed, replacing the current implementation with a PCB and surface-mounted components would enable a more compact design. In addition, the present design requires multiple supply voltage levels for different parts of the circuit, necessitating several separate power supply circuits. By selecting components that operate at 3.3 V, a single supply rail could be used instead. Integrating all three amplification and filtering circuits onto a single PCB would further allow them to share the same supply circuitry, thereby reducing the overall size. The ESP32 chip could also be integrated directly onto the PCB, eliminating the need for a separate development board. This would enable the entire system to be implemented on a single PCB, eliminating the need for multiple modules and resulting in a lighter, more compact band.

A reduced circuit size would also facilitate improvements in the band's comfort and ergonomics. To further enhance comfort, the PLA casing could be replaced with a softer, rubber-like material. This would additionally increase surface friction, potentially reducing the required strap tension.

The circuit's power consumption has not been tested. However, observations during the tests indicate that the circuit consumes relatively high power. One source of this might be the voltage regulators that dissipate excess energy as heat. To determine which components consume the most power and which solutions could be implemented, further testing is required. Additionally, switching to a more energy-dense battery, such as a rechargeable lithium-ion battery, could reduce the band's size or increase battery time. The band would also be more practical, allowing the user to recharge the batteries rather than replace them.

Another improvement that would bring the system closer to a finished product is developing a more application-specific GUI. While the current implementation is functional, its visual design is limited in terms of aesthetics and user experience. In a finalized product, the GUI should be adapted to the intended application. For medical use, emphasis would likely be placed on functionality and ease of use. A simplified interface similar to the current version may therefore be appropriate. In human-machine interaction applications, a minimal interface consisting primarily of a settings menu for customizing control parameters could be sufficient. For some use cases, such as prosthetics, a dedicated GUI may not be required at all.

The ESP-NOW protocol supports several ESP32 devices connecting to the same network. That feature enables a telemetry system to monitor multiple sEMG armbands. This would allow for the monitoring of several limbs or people simultaneously. For example, a rehabilitation clinic could simultaneously track all its patients' muscle activity. Another improvement that could be made to the band is adding multiple channels. An increased number of channels allows a larger portion of the forearm muscles to be monitored and more muscle activation combinations to be detected. The additional information about muscle activation can enable the classification of more movements and higher classification accuracy. This could make the band more suitable for use in applications such as advanced prosthetic control.

6

Conclusion

This bachelor's thesis aims to design and develop a wireless sEMG armband for real-time monitoring of muscle activity, including movement classification. Although the objective is considered achieved, the results highlight several limitations in the system.

The isolated testing indicated that each part of the system performed in accordance with the specific objectives. The electrodes, along with the amplification and filtering circuit, were able to record and process the sEMG signals. The wireless transmission was implemented with no noticeable latency or packet loss. The signal processing methods yielded reliable results, and the GUI displayed the data in real time. Finally, the movement classification achieved high accuracy in a controlled environment. However, after integrating all components, it was found that the entire system lacked robustness and adaptability. This was most apparent during the movement classification. Since the entire system was developed during this project, these issues are most likely due to dependencies between the hardware and software.

Potential sources of error could be mitigated by changing some features of the band. One such change is to increase the size of the ground electrode to lower the skin-to-electrode impedance, thereby reducing noise in the measured data. The circuit could also be made more power-efficient, as the voltage regulator dissipates excess energy as heat, possibly adding noise. During the project, Ag/AgCl electrodes were used as a benchmark to verify the performance of the stainless steel electrodes. Likewise, commercially available devices or software could have been used to identify additional problems involving the synergy among parts. For instance, it could be used for the electrodes and the amplification and filtering circuit, or digital signal processing and movement classification. This would facilitate the identification of integration-related issues in the individual components.

The method and supporting theories presented in this thesis can be used in the future to develop similar sEMG bands, while the discussion serves as a set of technical guidelines for further optimizations. While the band does not introduce new concepts or applications, it remains a proof-of-concept model for a wireless sEMG armband. Ultimately, the project demonstrates that the primary objectives are achievable within the established delimitations, validating the feasibility of the system.

Bibliography

[1] M. Al-Ayyad, H. A. Owida, R. De Fazio, B. Al-Naami, and P. Visconti, "Electromyography monitoring systems in rehabilitation: A review of clinical applications, wearable devices and signal acquisition methodologies," *Electronics (Switzerland)*, vol. 12, no. 7, pp. 1–35, Mar. 2023. DOI: 10.3390/electronics12071520.

[2] V. Tiwari et al., "From muscle potentials to computer interfaces: A multiscale review of electromyography for clinical and computational applications," *Sensing and Bio-Sensing Research*, vol. 51, pp. 2–13, Jan. 2026. DOI: 10.1016/j.sbsr.2026.100969.

[3] *Lesson one: Getting started with myo*, Myo Gesture Control Armband Instruction Manual, Thalmic Labs Inc. Accessed: May 11, 2026. [Online]. Available: https://images-na.ssl-images-amazon.com/images/I/A1Imj9kUBiL.pdf.

[4] K. Landram, *Cmu, meta seek to make computer-based tasks accessible with wristband technology*, Carnegie Mellon University - College of Engineering, Jul. 9, 2024. Accessed: May 11, 2026. [Online]. Available: https://engineering.cmu.edu/news-events/news/2024/07/09-wearable-sensing-tech.html.

[5] C. Gaudez and F. Cail, "Effects of mouse slant and desktop position on muscular and postural stresses, subject preference and performance in women aged 18–40 years," *Ergonomics*, vol. 59, no. 11, pp. 1473–1486, Apr. 2016. DOI: 10.1080/00140139.2016.1148783.

[6] G. P. van Galen, H. Liesker, and A. de Haan, "Effects of a vertical keyboard design on typing performance, user comfort and muscle tension," *Applied Ergonomics*, vol. 38, no. 1, pp. 99–107, Jan. 2007. DOI: 10.1016/j.apergo.2005.09.005.

[7] I. Karacan and K. Türker, "A comparison of electromyography techniques: Surface versus intramuscular recording," *European Journal of Applied Physiology*, vol. 125, pp. 7–23, Oct. 2024. DOI: 10.1007/s00421-024-05640-x.

[8] M. A. C. García and T. M. Vieira, "Surface electromyography: Why, when and how to use it," *Revista Portuguesa De Pneumologia*, vol. 4, pp. 17–28, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:70661880.

[9] E. Romero Avila, E. Junker, and C. Disselhorst-Klug, "Introduction of a semg sensor system for autonomous use by inexperienced users," *Sensors (Switzerland)*, vol. 20, no. 24, pp. 1–17, Dec. 2020. DOI: 10.3390/s20247348.

[10] smartARM, *Prosthetic arms powered by ai vision*, 2026. Accessed: May 4, 2026. [Online]. Available: `https://www.smartarm.ca/`.

[11] C. Yarkoni, *Meet your 2018 imagine cup champions – smartarm of canada!* Official Microsoft blog, Jul. 25, 2018. Accessed: May 4, 2026. [Online]. Available: `https://blogs.microsoft.com/blog/2018/07/25/meet-your-2018-imagine-cup-champions-smartarm-of-canada/`.

[12] S. Mick et al., "Smart arm: A customizable and versatile robotic arm prosthesis platform for cybathlon and research," *Journal of NeuroEngineering and Rehabilitation*, vol. 21, no. 1, Dec. 2024. DOI: `10.1186/s12984-024-01423-9`.

[13] Open Bionics, *Open bionics - turning disabilities into superpowers with bionic hands*, 2026. Accessed: May 4, 2026. [Online]. Available: `https://openbionics.com/`.

[14] J. D. B. John D. Enderle Susan M. Blanchard, *Introduction to Biomedical Engineering*, 2nd ed. Burlington, MA, USA: Academic Press, 2005.

[15] W. F. Brown, *Electromyography*, in AccessScience, New York, NY, USA: McGraw-Hill. [Online]. Accessed: Feb. 2, 2026, Jan. 2020. DOI: `10.1036/1097-8542.223500`.

[16] M. Ferrante, *Comprehensive Electromyography: With Clinical Correlations and Case Studies*. Cambridge, UK: Cambridge University Press, 2018.

[17] H. Piitulainen, T. Rantalainen, V. Linnamo, P. Komi, and J. Avela, "Innervation zone shift at different levels of isometric contraction in the biceps brachii muscle," *Journal of Electromyography and Kinesiology*, vol. 19, no. 4, pp. 667–675, Aug. 2009. DOI: `10.1016/j.jelekin.2008.02.007`.

[18] H. J. Hermens, B. Freriks, C. Disselhorst-Klug, and G. Rau, "Development of recommendations for semg sensors and sensor placement procedures," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 361–374, Oct. 2000. DOI: `10.1016/S1050-6411(00)00027-4`.

[19] A. Joutsen et al., "Ecg signal quality in intermittent long-term dry electrode recordings with controlled motion artifacts," *Scientific Reports*, vol. 14, no. 1, Dec. 2024. DOI: `https://doi.org/10.1038/s41598-024-56595-0`.

[20] M. Boyer, L. Bouyer, J.-S. Roy, and A. Campeau-Lecours, "Reducing noise, artifacts and interference in single-channel emg signals: A review," *Sensors*, vol. 23, no. 6, Mar. 2023. DOI: `10.3390/s23062927`.

[21] B. Karlström, *Kretsanalys*, 3rd ed. Lund, Sweden: Studentlitteratur, 2022.

[22] L. P. H. Donald P. Havens, *Electric filter*, in AccessScience, New York, NY, USA: McGraw-Hill. [Online]. Accessed: Apr. 15, 2026, Sep. 2025. DOI: `10.1036/1097-8542.216100`.

[23] B. K. Lars Bengtsson, *Transformer och filter*, 2nd ed. Lund, Sweden: Studentlitteratur, 2016.

[24] H. Lindberg, *Trådlösa nätverk -WLAN, WEP och Wi-Fi*. Lund, Sweden: Studentlitteratur, 2002.

[25] Espressif Systems, *ESP-NOW*, 2026. Accessed: Apr. 9, 2026. [Online]. Available: `https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/api-reference/network/esp_now.html`.

[26] S. Gancarčík, J. Šumský, and M. Kubaščík, *Exploring wireless communication protocols on esp32 platform for outdoor applications*, Espressif Developer

Portal, Dec. 20, 2024. Accessed: Apr. 9, 2026. [Online]. Available: https://developer.espressif.com/blog/esp-now-for-outdoor-applications/.

[27] A. Abdelouahad, A. Belkhou, A. Jbari, and L. Bellarbi, "Time and frequency parameters of semg signal – force relationship," in *2018 4th International Conference on Optimization and Applications (ICOA)*, Apr. 2018, pp. 1–5. DOI: 10.1109/ICOA.2018.8370547. Accessed: Apr. 10, 2026.

[28] L. A. Kallenberg and H. J. Hermens, "Behaviour of motor unit action potential rate, estimated from surface emg, as a measure of muscle activation level," *Journal of NeuroEngineering and Rehabilitation*, vol. 3, no. 15, pp. 1–2, Jul. 2006. DOI: 10.1186/1743-0003-3-15.

[29] J. Chang, D. Chablat, F. Bennis, and L. Ma, "Estimating the emg response exclusively to fatigue during sustained static maximum voluntary contraction," in *Advances in Intelligent Systems and Computing*, Walt Disney World, USA, 2016, pp. 29–39. DOI: 10.1007/978-3-319-41694-6_4. Accessed: Apr. 15, 2026.

[30] P. Bartuzi, D. Roman-Liu, and T. Tokarski, "A study of the influence of muscle type and muscle force level on individual frequency bands of the emg power spectrum," *International Journal of Occupational Safety and Ergonomics*, vol. 13, pp. 241–254, 2007. DOI: 10.1080/10803548.2007.11076725.

[31] C. H. Baker, D. A. Jordan, and P. M. Norris, "Application of the wavelet transform to nanoscale thermal transport," *Physical Review B - Condensed Matter and Materials Physics*, vol. 86, no. 10, Sep. 2012. DOI: 10.1103/PhysRevB.86.104306.

[32] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "Application of wavelet analysis in emg feature extraction for pattern classification," *Measurement Science Review*, vol. 11, no. 2, pp. 45–52, Jan. 2011. DOI: 10.2478/v10048-011-0009-y.

[33] M. R. Canal, "Comparison of wavelet and short time fourier transform methods in the analysis of emg signals," *Journal of Medical Systems*, vol. 34, no. 1, pp. 91–94, Oct. 2008. DOI: 10.1007/s10916-008-9219-8.

[34] M. Stéphane, "Time meets frequency," in *A Wavelet Tour of Signal Processing*, 3rd ed., Boston, MA, USA: Academic Press, 2009, ch. 4, pp. 89–153.

[35] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bulletin of the American Meteorological Society*, vol. 79, pp. 61–78, 1998. DOI: 10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2.

[36] F. Wefers, *Partitioned convolution algorithms for real-time auralization.* Aachen, Germany: Logos Verlag Berlin GmbH, 2014, vol. 20. Accessed: Apr. 23, 2026. [Online]. Available: https://books.google.de/books?id=IA-bCgAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Quicksort," in *Introduction to Algorithms*, 3rd ed., Cambridge, Massachusetts: MIT Press, 2009, ch. 7, pp. 170–190.

[38] S. Elbaz, S. Agounad, M. Ait Yous, and M. Moufassih, "Hand gesture recognition using temporal and spectral electromyography features and machine

learning," in *Knowledge-Based Systems*, Hangzhou, China, 2026, pp. 7–8. DOI: 10.1016/j.knosys.2025.115256. Accessed: Apr. 25, 2026.

[39] H. Li, J. Li, X. Zhang, T. Liu, J. Yin, and Y. Gu, "Feature selection and classification of electromyographic signals for improved motion recognition," in *2025 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Hangzhou, China, 2025, pp. 1–6. DOI: 10.1109/AIM64088.2025.11175703. Accessed: Apr. 25, 2026.

[40] Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016), OJ L 119/1. Accessed: May 13, 2026. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679.

[41] J. F. Grubb, *Stainless steel*, in AccessScience, New York, NY, USA: McGraw-Hill. [Online]. Accessed: Apr. 9, 2026, Jan. 2020. DOI: 10.1036/1097-8542.650600.

[42] British Stainless Steel Association, *Soldering stainless steels*, https://bssa.org.uk/bssa_articles/soldering-stainless-steels/, (accessed 23-04-2026).

[43] MG Chemicals, *8330d – silver conductive epoxy adhesive*, https://mgchemicals.com/products/adhesives/electrically-conductive-adhesives/conductive-glue/, (accessed 23-04-2026).

[44] *INA121 FET-Input, Low Power Instrumentation Amplifier*, Burr-Brown Corporation, Tucson, AZ, USA, 1998. Accessed: Apr. 13, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina121.pdf?ts=1776022105409&ref_url=https%253A%252F%252Fwww.google.com%252F.

[45] *MCP6021/1R/2/3/4*, Microchip Technology Inc., Chandler, AZ, USA, 2016. Accessed: Apr. 28, 2026. [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/20001685e.pdf.

[46] *ESP32-S3 Series Datasheet Version 2.2*, Espressif Systems, Shanghai, China, 2026. Accessed: Apr. 10, 2026. [Online]. Available: https://documentation.espressif.com/esp32-s3_datasheet_en.pdf.

[47] *LD1117 Adjustable and fixed low drop positive voltage regulator*, STMicroelectronics NV, Plan-les-Ouates, Geneve, Switzerland, 2025. Accessed: May 5, 2026. [Online]. Available: https://www.st.com/resource/en/datasheet/ld1117.pdf.

[48] J. D. H. Ashok P. Nedungadi Donnell D. MacCarthy, *Voltage regulator*, in AccessScience, New York, NY, USA: McGraw-Hill. [Online]. Accessed: May. 5, 2026, Jan. 2020. DOI: 10.1036/1097-8542.736000.

[49] *Arduino-ESP32 Timer API*, Espressif Systems, Shanghai, China, 2026. Accessed: Apr. 10, 2026. [Online]. Available: https://docs.espressif.com/projects/arduino-esp32/en/latest/api/timer.html#arduino-esp32-timer-api.

[50] P. S. Foundation. "Thread-based parallellsim," Accessed: May 11, 2026. [Online]. Available: https://docs.python.org/3/library/threading.html.

[51] J. Carpenter. "Simd and vectorization using avx intrinsic functions (tutorial)," [Video]. YouTube. [Online]. Available: https://youtu.be/AT5nuQQO96o?si=GxgYG1zNNZvPMojV.

[52] scikit-learn. "Sklearn.dicriminant_analysis.lineardiscriminantanalysis — scikit-learn 0.24.1 documentation," Accessed: May 3, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html.

[53] D. A. Forman et al., "Characterizing forearm muscle activity in young adults during dynamic wrist flexion–extension movement using a wrist robot," *Journal of Biomechanics*, vol. 108, pp. 1–3, Jul. 2020. DOI: 10.1016/j.jbiomech.2020.109908.

[54] I. M. L. C. Volges, "Detection of temporal delays in visual-haptic interfaces," *Human Factors: The Journal of the Human Factors and the Ergonomics Society*, vol. 46, no. 1, pp. 118–134, Mar. 2004. DOI: 10.1518/hfes.46.1.118.30394.

[55] T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectronic prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 111–118, Mar. 2007. DOI: 10.1109/TNSRE.2007.891391.

A

Appendix 1

The link for the GitHub with the code for this bachelor's thesis can be access here. The name of the github is: Wireless_sEMG_band.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS