HAND-GESTURE RECOGNITION BASED HUMAN-MACHINE INTERFACE FOR ASSISTIVE TECHNOLOGY VIA ELECTROMYOGRAPHY

Amedeo Leproni

3rd Year Project Final Report

Department of Electronic &
Electrical Engineering
UCL

Supervisor: Dr Nasrollahy
Advisor: Dr Nasrollahy

6 March 2022

I have read and understood UCL’s and the Department’s statements and guidelines concerning plagiarism.

I declare that all material described in this report is my own work except where explicitly and individually indicated in the text. This includes ideas described in the text, figures, and computer programs.

This report contains 32 pages (excluding this page and the appendices) and 10059 words.

Signed: Amedeo Leproni Date: 06/03/2022

(Student)

Table of contents

List of figures

List of tables

Hand gesture recognition via electromyography (EMG) for assistive technology.

Amedeo Leproni

This report explores the development of a hand gesture recognition system via EMG for use in assistive technology. It includes the development of various hardware stages implemented on a printed circuit board (PCB) and analog to digital conversion (ADC), followed in software by the extraction of root mean square (RMS) feature and classification via a support vector machine (SVM). The aim of the investigation is to determine if a bespoke electrode layout with purposefully selected electrode positions on the forearm muscles leads to a higher classification accuracy than using a comparatively basic ('baseline') electrode layout widely found in literature. The bespoke layout yielded a classification accuracy of 83.3% and the baseline layout yielded a classification accuracy of 68.8%, supporting the hypothesis that placing electrodes on muscles related to executed hand gestures leads to higher classification accuracy. Closer analysis demonstrates the bespoke layout could be improved for the chosen hand gestures.

1 Introduction

1.1 Human-machine interface

Human beings each come with a unique set of physical characteristics, and each progresses through daily life, overcoming its various challenges, in his or her own unique way. Some people are faced with greater daily challenges than others, especially so when they suffer from a medical condition. Though the list of medical conditions is countless, so are the ways in which technology could be developed to assist those afflicted. As machines rapidly become essential in the lives of people all around the globe, the ability to interact with these machines becomes more and more crucial. Hand and wrist related amputations are just one of the many conditions which severely hinders the ability of a person to interact with machines, which are usually manually controlled. As investigated by Crowe et al. [1], the global incidence of hand and wrist fractures is 0.179%, and the incidence of hand digit amputations is 0.08%, combining to over 200 million people worldwide. The development of human-machine interfaces (HMI) via EMG geared towards use in assistive technology has the potential to help millions of people, afflicted not only by amputations as seen in Fig. 1, but a myriad of other medical conditions as well.

Figure 1: Control of a forearm prosthesis via EMG using MyoBand (image by Thalmic labs)

An HMI is a system which allows individuals to interact with machines, and it can be composed of a combination of physical devices and software. The objective of developing HMIs is to diminish the boundary between humans and machines, thus enabling people to interact with these tools more intuitively and intimately. The characteristics of HMIs can vary greatly based on the machine they are used to interface with and the user they are targeted towards. They are widespread in the industrial sector, where operators use them to interact with typically complex and expensive machinery. In contrast, some HMIs are geared towards mass-market consumers, such as keyboards interfacing individuals with their computer, or touchscreens interfacing individuals with automated teller machines (ATMs). Regardless of the application, some key characteristics emerge for a successful HMI. First and foremost, they must be accurate. This means that the response to a known user input must be consistent. For example, pressing the ‘a’ key on a computer keyboard will consistently send an ‘a’ character to the computer. They must also be intuitive, meaning most people will be able to learn to use them very quickly, and the process of using them becomes ‘second nature’. Once again referring to a keyboard, the user does not need think about the fact they are pressing keys when they are writing, the process becomes unconscious. Lastly, HMIs should also be comfortable for the user to interact with, meaning they do not bring about unpleasant user-experience when utilised in the intended way. [2]

The examples of HMIs mentioned so far have involved a manually controlled physical device for interaction to occur, such as a keyboard or a touch-screen display. The need to familiarize oneself with the physical object and to learn how to use it creates a boundary between human and machine. There are some forms of activity which are universally relatable for humans, which individuals carry out their entire lives, making them so intuitive and intimate that they become unconscious. Hand gestures are an example of this. A potential improvement of HMI technology rests in sidestepping manually controlled physical devices and relying instead on such a familiar method of communication. Research in this field creates a myriad of new possibilities for what machines could do in the novel context of more intuitive HMIs.

1.2 Hand gestures for HMI

The key requirements of accuracy, intuitiveness and comfort are very relevant to HMIs based on hand gesture recognition. In this case, accuracy specifically refers to correctly recognising a hand gesture when it is made and being capable of differentiating between different hand gestures. A hand-gesture based HMI has the potential to be incredibly intuitive, by virtue of human familiarity with gestures to communicate. However, the pairing of gestures with their respective commands or actuations must be consistent with the meaning(s) of the gestures themselves. Lastly, comfort while using a hand-gesture HMI refers both to the physical wellbeing of the user and the logistical feasibility of deploying the HMI in the context where the machine needs to operate. The two main candidate technologies to carry out hand gesture recognition can be classified as either endogenous or exogenous to the user: the data is collected from the body of the user, or the data is collected externally from the body of the user. Endogenous technologies refer to HMIs which the user can wear on their hands, wrists, and arms to gather data related to hand gestures. Exogenous technologies refer to HMIs which use computer vision and image processing, requiring recording devices (such as a camera) to be located near but not worn by the user.

To gather the necessary visual information, exogenous HMIs can make use of a few different tools. Depth-aware cameras and stereo cameras may be used to approximate a 3D representation of the user’s hand [3] [4]. Then, 3D model-based algorithms or skeletal-based algorithms are used to classify the hand gesture, though the former requires significant computational power and is therefore inappropriate for real-time use [5]. Another method is to use conventional cameras to record a 2D image of the hand gesture. These can be classified using parameters of the recorded image, which are comparable to template databases of possible hand gestures. Although the literature demonstrates these methods are successful at accurately classifying hand gestures, the necessity for a recording setup vastly limits the contexts in which the HMI can be deployed. The recording setup will require its own space and equipment, it will require appropriate lighting conditions, meaning a portable implementation is logistically impractical [6].

Endogenous HMIs on the other hand, are not tied down by this same limitation. Because they are worn by the user, they can be developed as a portable product, unlocking countless more applications. There are a few types of information that can be gathered from the body of the user which allows for hand gesture classification. The main options explored in literature are force sensitive resistors (FSRs), a combination of inertial, gyro and magnetic sensors, and electromyography (EMG). All these options function in the same basic way: the sensor records numerical data, representative features are extracted from the data, and finally the features are input to a classification algorithm which estimates hand-gestures with some degree of accuracy. The difference between these methods lies principally in the type of data collected. FSRs provide variable voltage values, changing with the resistance of the FSR, which itself changes with the force exerted on it by the contracting muscles [7] [8]. The combination of inertial, gyro and magnetic sensors provides voltage values which change with the acceleration and orientation of the subject’s hand and arm [9] [10]. Lastly, EMG provides a signal which closely resembles the action potential travelling through muscle fibres during their activation [11] (other biopotentials viable for hand gesture classification also exist, such as electroencephalographs [12]). Although all methods have been demonstrated to yield high classification accuracies, EMG is explored in this report. This is because both FSR and inertial, gyro and magnetic sensors utilise information that is created only after a hand gesture is carried out. In contrast, EMG signals are produced 2-3ms before the hand gesture is physically manifested due to the phenomenon of electromechanical delay, as explained by Fortune et al. [13]. This makes EMG a particularly desirable area of research when considering the potential for real-time applications permitted by the lower delay in signal acquisition. The study of EMG may also lead to a more thorough understanding of human muscles and the how they function in harmony to effectuate complex hand gestures.

1.3 EMG signals and EMG acquisition systems

The EMG signals most strongly associated to the execution of hand gestures are those present in the skeletal muscles of the forearm [14] [11]. These muscles are consciously controlled (contracted) by the brain through nerve impulses. Once a nerve impulse reaches the neuromuscular junction, which is the boundary between the end of the nerve and the start of the muscle tissue, a biochemical process occurs which leads to the depolarisation of the muscle cell membrane. Depolarisation signifies a sudden change in the membrane potential due to the uneven distribution of ions either side of the membrane. This uneven charge distribution generates an action potential in the muscle, which propagates from the site of the neuromuscular junction through to all the muscle fibres via specialised organelles called the T-tubules [15]. EMG is the process of recording the propagation of this action potential, which is a voltage pulse travelling through the muscle. It becomes evident that EMG does not provide a perfect picture of muscle contraction. Rather, EMG is an imperfect image of the biopotentials propagating through muscle fibres on their way to initiate a contraction.

EMG is an imperfect image of the true action potential due to the acquisition systems needed for recording. An overview is given for the system used in this report (see Fig. 2), though others will vary. First an array of bipolar surface electrodes is used to detect electrical signals from the skeletal muscles of the forearm – but it introduces significant interference to the signal in the process. Then, a circuit must amplify the detected signal and clean it [16]. Lastly, the signal is digitized using an ADC so that it can be transferred to a computer for analysis.

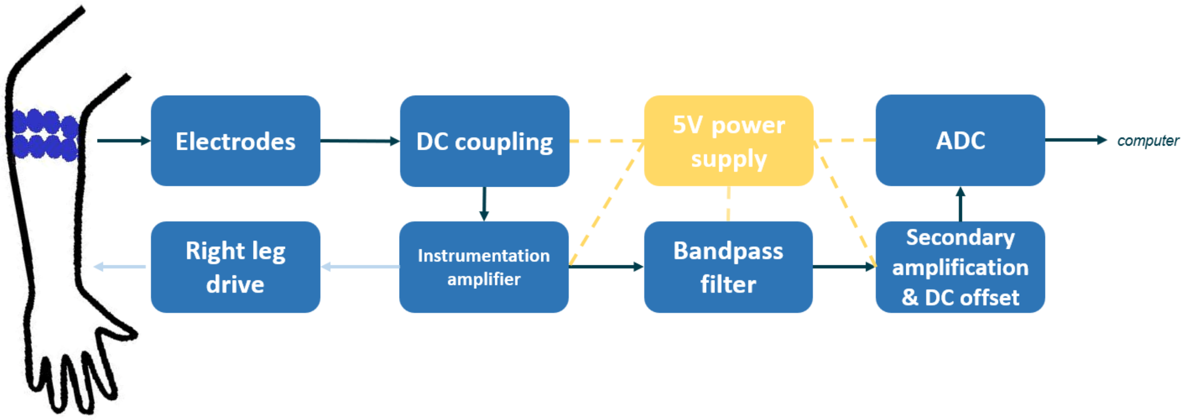


Figure 2: Block diagram of the multimodal hand gesture recognition system

This report investigates the position of the electrode array on the forearm. It may be composed of multiple channels, and the location of each channel on the forearm can vary. The pairs of electrodes are usually placed 1-2 cm apart, and a reference electrode shared between all channels is also used [17]. A system with more channels is better able to solve problems with multiple degrees of freedom (DoF). The DoF correspond to the number of hand gestures the system must be able to differentiate between. In literature, a recurrent electrode layout configuration is the equidistant placement of electrodes around the circumference of the forearm [18] [19] [14] [20] [21], as can be seen in Fig. 3. However, judiciously placing electrodes on muscles that play an important role in hand gestures could improve the classification accuracy of the system. The electrode configuration seen in Fig. 3 will be referred to as the 'baseline layout', whilst the configuration investigated in Section 5 will be referred to as the 'bespoke layout'.

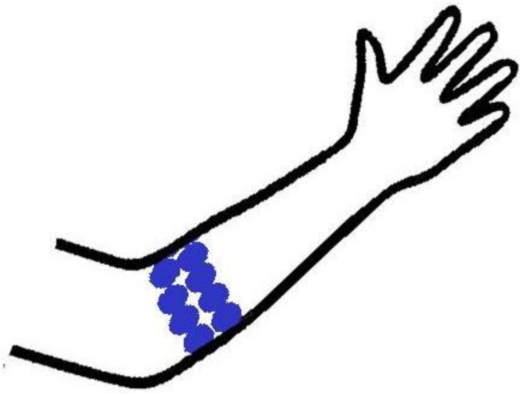


Figure 3: Baseline electrode layout

1.4 EMG for hand gesture recognition

A robust HMI does not restrict itself to just one application. The focus of this report is hand gesture recognition for assistive technology, but many other fields benefit from developments in EMG technology, such as competitive sports, commercial computers, and the video games industry [14]. Within the field of assistive technology, advances are being made on several fronts. Researchers have successfully implemented real-time sign language translation [22] and prosthetic limb control [20] [8] through hand gesture recognition via EMG. As explored by Farina et al. [21] characteristics that make these HMIs valuable to users are their response times, classification accuracy, and comfort, however the lack of widespread commercial success for these technologies demonstrates the need for marked improvements on all these aspects. In this report, to make the EMG system less bulky (so more comfortable), only 4 electrode pairs are used (compared to the usual ≥8 found in baseline layouts) and an attempt to improve accuracy is made through better electrode positioning. Though 4-channel systems are less frequent, they still appear in literature [22] [20].

2 Report structure

Having established sufficient context for the research carried out, the reader can now understand the roles of the different report sections. Firstly, the improvement to be made in EMG systems for hand gesture recognition are clearly stated in Section 3 (Objective), and the tasks that have been carried out to create all necessary components and build an experimental procedure are listed in Section 4 (Tasks). In Section 5 (Background knowledge), background knowledge is provided so that the reader may understand the design choices made in all parts of the manufactured system. In Section 6 (Literature review), a review of literature concerning electrode position on the forearm for EMG systems is made, showing the reader how the bespoke layout was decided upon. In section 7 (Methods), the reader can see how the background knowledge was acted upon for the implementation of the system. In Section 8 (Results), the results of classification accuracy for the baseline layout and the bespoke layout are presented, so that they can be referenced in Section 9 (Discussion) to evaluate the hypothesis and discuss valuable findings. Lastly, a summary of the entire project is given in Section 10 (Conclusion), as well as a list of potential system improvements to help future reproductions of this work.

3 Objective

The objective of this report is to determine if the positioning of electrodes on specific muscles of the forearm in a bespoke layout, constructed based on an analysis of literature and an exploration of forearm anatomy, influences hand gesture classification accuracy. A baseline electrode layout popular in EMG literature is used in comparison.

4 Tasks

4.1 Hardware

- Research literature to investigate EMG acquisition circuits and gain an understanding of the necessary components and performance requirements.
- Design an EMG acquisition circuit in simulation software Multisim and verify appropriate performance.
- Implement the designed circuit on a breadboard to test that it performs appropriately.
- Implement the circuit on a printed circuit board (PCB)

4.2 Data acquisition

- Configure the ADC of a microcontroller to have appropriate sampling frequency and multichannel capabilities, as well as appropriate interfacing with a computer via USB.
- Design an appropriate testing routine to perform hand gestures during recording, to be used with both the baseline layout and the bespoke layout.
- Write a MATLAB script to store and index the recorded EMG signal.

4.3 Feature extraction

- Label recorded EMG with gesture labels 1-4, representing the four hand gestures carried out during the experiment.
- Create windows of EMG signal for each hand gesture carried out.
- Extract RMS voltage for each window of EMG signal and export the datapoints to an appropriately indexed matrix.

4.4 Classification

- Import the feature data matrix into a Python script and randomly allocate 90% of the datapoint as train data and 10% of the datapoints as test data.
- Normalize the RMS values for each channel to be normally distributed (mean of 0 and standard deviation of 1)
- Build an SVM model and train it with the scaled RMS data.
- Test the classification accuracy of the trained model using the test data and create a confusion matrix for data examination.

4.5 Gesture selection

- Investigate literature to find which gestures are most often classified.
- Evaluate the ideal gestures based on intuitiveness for HMI and differentiability via EMG.

4.6 Electrode placement selection

- Investigate literature to find how electrodes have been positioned on the forearm for EMG acquisition.
- Investigate forearm anatomy for hand gesture execution to determine what the most relevant muscles are.
- Investigate the literature to determine the ideal electrode placement on the muscle.

5 Background knowledge

5.1 EMG Electrodes

There are two widely used electrodes to measure EMG signals: surface electrodes and intramuscular electrodes. These record biopotentials of the same nature, but with different strengths and weaknesses [23]. In the case of surface electrodes, the signal they record is called surface electromyography (sEMG). As the name suggests, these electrodes are used to measure superficial muscles. Depending on the size of the electrode, they can record unwanted signals from an adjacent muscle, or even from deep muscles. This phenomenon is called cross talk and can be misleading in the classification of EMG signals [24]. sEMG electrodes can also suffer from movement artefact, which occurs because muscles move under the skin during contraction while the electrodes remain fixed in place [24]. In contrast, intramuscular electrodes overcome both issues. They are needle shaped electrodes inserted into the muscle of interest. This makes it possible to measure deep muscles as well as superficial muscles, it severely reduces cross talk, and reduces movement artefact as the needle electrode follows the movement of the muscle [25]. However, intramuscular electrodes are invasive, potentially causing the user pain and discomfort and requiring specialised skills to insert them in the muscle of interest. This renders them unusable for the ubiquitous HMI this report seeks to advance towards. Furthermore, the muscles of interest for hand gesture classification are superficial. This means sEMG electrodes are the best option (beyond this point, EMG will be used interchangeably with sEMG).

If the skin-electrode interface has a high impedance, this will negatively affect the EMG signal quality. Silver/silver chloride electrodes are widely accepted as having acceptable signal to noise ratio (SNR) and low impedance. An electrode with a larger surface area will have a decreased impedance, but an acceptable middle ground between crosstalk due to large area and high impedance due to small area must be found [26].

Surface electrodes work as biopotential sensors by transducing ionic flow in the tissues into an electronic current, via a chemical reaction between an electrode and electrolyte. The ionic current causes the oxidisation of the electrode to release electrons into the wires, but this simultaneously produces cations which move to the electrolyte-electrode interface [27]. The potential difference at this interface causes a DC bias in the EMG signal, which may be removed with AC coupling.

5.2 EMG signals

As mentioned in the introduction, EMG signals are an imperfect image of the actual biopotentials travelling through the muscles. This concept is further explored in this section Fig. 4 demonstrates some examples of recorded EMG signal. EMG as it is measured by bipolar electrodes may be considered as the sum of three parts: the differential EMG signal we desire, the common mode interference caused by the body or external factors, and a DC offset component generated by the skin-electrode interface.

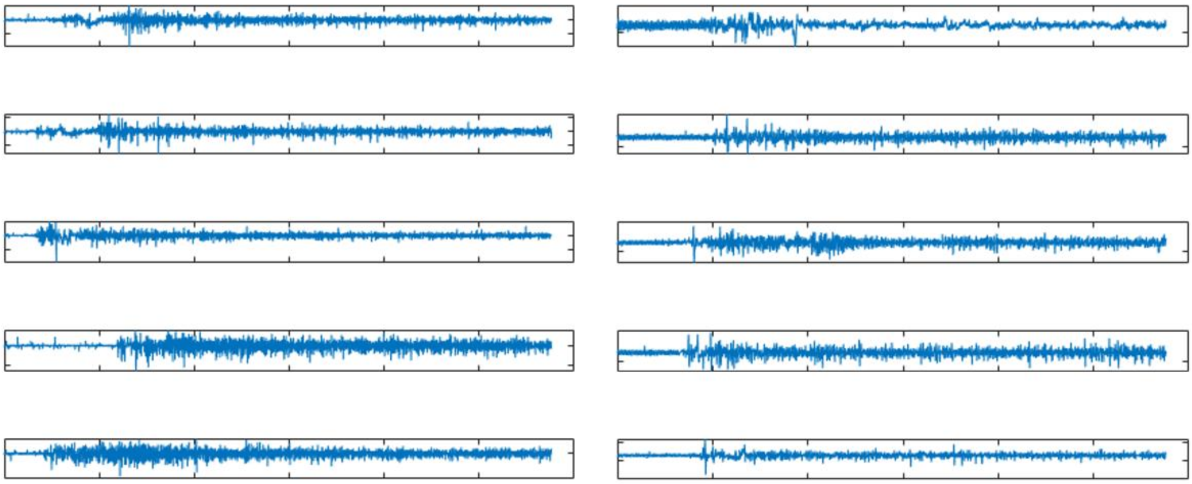


Figure 4: Examples of recorded EMG signals

The DC offset in the EMG signal is the first difference from the muscular action potential, and it is not desired. As outlined in Section 5.1, it is due to the oxidation reaction that takes place in the electrodes for the biopotential to be transduced. The common mode interference is another undesirable component of EMG signal, which is not found in the original action potential. The human body constantly carries out other biological processes which leave their biopotential footprints around the body and interfere with muscular action potential. Furthermore, the human body as well as the electrode wires act as an antennae to surrounding electromagnetic signals, which are abundant in the average laboratory. These electromagnetic signals induce currents in the body which are often larger in amplitude than the recorded biopotential. One is likely to find a signal saturated with a 50 Hz component due to mains electricity.

It is the differential signal which contains valuable information concerning the propagating action potential, and so we aim to preserve only this part of the recorded EMG. Bipolar electrodes are ubiquitously used in EMG acquisition because they allow for common mode interference to removed and differential signal to be preserved [17]. Each electrodes records a voltage signal, which becomes differential signal following the equation [28]:

$$V_{diff} = \frac{V_2 - V_1}{2} \qquad (1)$$

If we consider Eqn. 1, it is obvious that the differential signal is zero when no EMG is going through the muscle, because V_1 and V_2 are equal in magnitude and cancel to zero. If an action potential (V_p) is propagating through the tissue, it will take a discrete amount of time to move from the first to the second electrode. During this time, the difference in voltage between the two electrodes is recorded. It is important to understand that the differential signal is not the same as the action potential, as illustrated in Fig. 5 below. For a positive action potential pulse (left), the differential voltage reading (right) is not an image reflecting the pulse but shows instead first a negative peak and then a positive peak, resulting from Eqn. 1.

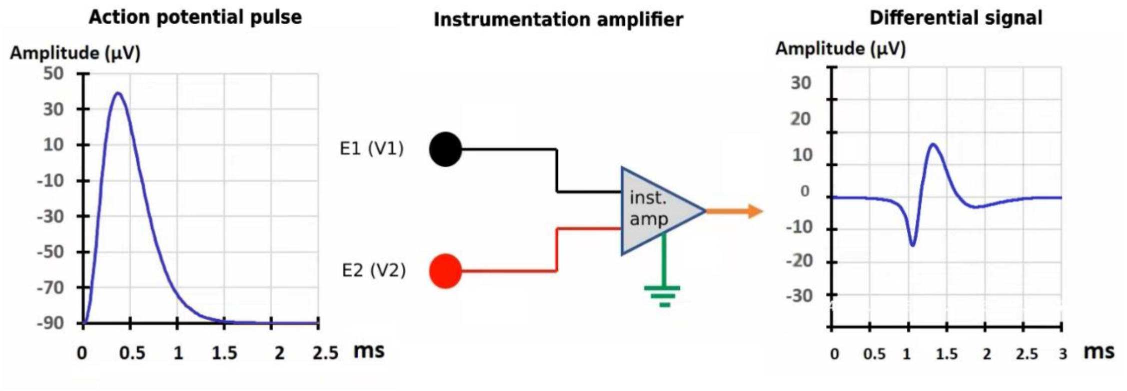


Figure 5: Effect of action potential pulse on differential voltage reading

The reason a bipolar electrode setup removes common mode noise can be explained with the help of Eqn. 1. The voltage reading V_1 and V_2 are composed of a differential signal V_{1diff} and V_{2diff} which are not equal due to the time taken for action potentials to propagate. They are also composed of a common mode signal V_{CM} which is equal in both V_1 and V_2 because the two points are subject to the same sources of interference. Thus V_{CM} is eliminated following:

$$V_{diff} = \frac{(V_{2diff}+V_{CM})-(V_{1diff}+V_{CM})}{2} \qquad (2)$$

$$V_{diff} = \frac{V_{2diff}+V_{CM}-V_{1diff}-V_{CM})}{2}$$

$$V_{diff} = \frac{V_{2diff}-V_{1diff}}{2}$$

It is important to note that V_{CM} can never be fully removed by the acquisition system but can be significantly attenuated.

5.3 Building blocks of hardware

The circuitry necessary in an EMG system is shown in Figs. 4,5, and 6. The first role of hardware is to remove the DC bias present in the signal due to the electrodes, which is achieved by AC coupling. The simplest way to implement AC coupling is to pass the signal through a capacitor and a grounded resistor, but as explained by Spinelli [29] this configuration would lower the common mode rejection ratio (CMRR) of the subsequent instrumentation amplifier stage. The better choice is to instead follow his proposed design for a balanced AC coupling circuit (see Fig. 6, blue). After the removal of the undesired DC offset, the instrumentation amplifier carries out the amplification of the differential signal between the two electrodes, for which the gain is set with an external resistance R_g (see Fig. 6, orange).

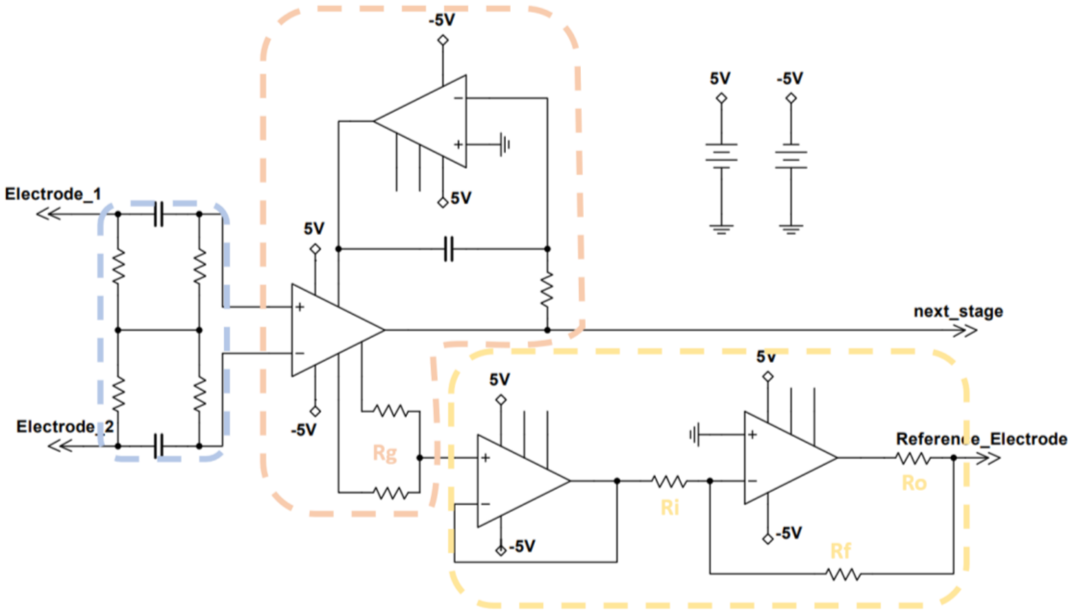


Figure 6: EMG acquisition circuit AC coupling, differential gain stage, feedback integrator and RLD

Biopotential acquisition systems require an amplifier with a high CMMR, as it will determine the gain ratio seen in Eqn. 3:

$$CMMR = \frac{differential\ mode\ gain}{common\ mode\ gain} = \frac{biopotential\ signal\ gain}{powerline\ \&\ noise\ gain} \qquad (3)$$

The higher the CMMR, the more strongly the undesired common mode signal is attenuated. This is why a balanced AC coupling circuit is implemented in the preceding stage. There is a factor of op amp input offset voltage (in the order of mV) present at the buffer stage of the instrumentation amplifier, which following the differential gain stage can cause substantial interference in the signal (in the order of V). To reduce the offset, Spinelli [29] proposes a feedback loop around the differential gain stage consisting of an integrator (see Fig. 6, orange). The next stage, connected to the instrumentation amplifier, is the right leg drive (RLD) circuit. The purpose of the RLD is to remove the common mode from the signal before it reaches the electrodes at all. The common mode filtered by the instrumentation amplifier is inverted, amplified, and fed back into the body through the reference electrode, where it destructively interferes with existing interference before it reaches the input electrodes. A negative feedback loop is created so that even when the common mode changes, the circuit adapts to the change and continues to remove it. The feedback circuit is designed with resistors (R_f and R_i) to determine the gain at the reference electrode, and a current limiting resistor (R_o) to regulate how much current is fed back to the patient's body [13] (see Fig. 6, yellow).

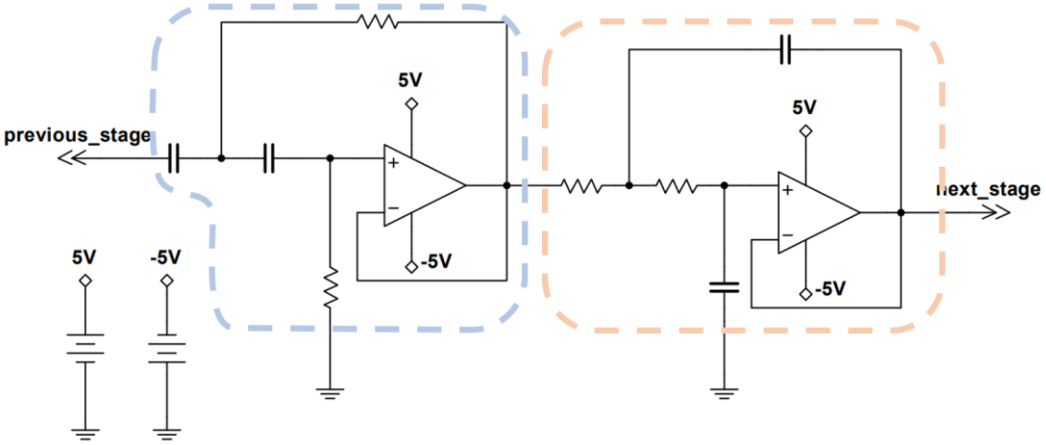


Figure 7: EMG acquisition circuit filtering stage

The output signal of the instrumentation amplifier contains the desired biopotential signal, however, the majority of EMG biopotential signal is found in the 10-500 Hz frequency band [30]. By using a bandpass filtering stage, signals outside this desired frequency range are removed. A good design for bandpass filters is a second order high-pass filter (see Fig. 7, blue) followed in series by a second order low-pass filter [31] (see Fig. 7, orange), as it provides steeper attenuation than a passive filter.

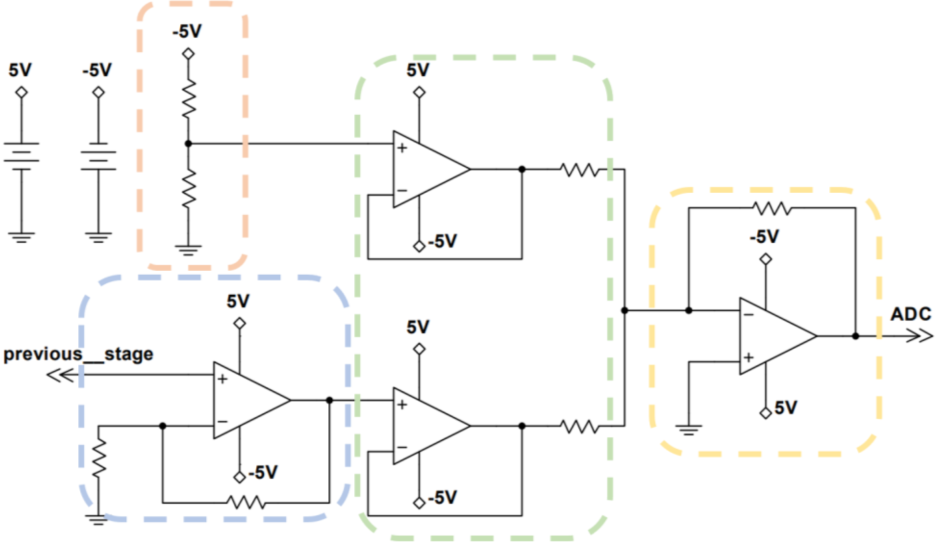


Figure 8: EMG acquisition circuit secondary amplification and DC offset stage

Following the filtering stage, the signal is clean and immune to further interference thanks to its high magnitude, so later circuit stages only pertain to making the signal compatible with the ADC. This involves a secondary amplification stage to make full use of the ADC voltage range (see Fig. 8, blue) followed by a DC offset to ensure the signal only contains positive voltages (see Fig. 8, orange). The secondary gain stage must be finetuned based on the p-p voltage of the EMG signal, so that it comes as close as possible to the ADC voltage range but no higher to avoid clipping. The final DC offset stage inputs both the EMG signal and a DC offset to a summing amplifier configuration (see Fig. 8, yellow), buffering both beforehand (see Fig. 8, green). The effect of this stage is to move the midpoint of the EMG signal to the midpoint of the ADC input range.

5.4 Data acquisition

When recording an EMG signal using the ADC of a microcontroller, there are two main characteristics to consider. The first one is the sampling frequency. As mentioned in section 5.3, the highest frequencies of interest in EMG signal are 500 Hz. Applying the Nyquist-Shannon sampling theorem, it becomes evident that the minimum frequency at which the EMG signal can be sampled is 1 kHz, though a higher sampling frequency than this ensures high signal quality. The second important characteristic of an ADC is its resolution. This is determined by the number of bits used to store each recorded value. Since the values are stored in binary, the resolution will follow Eqn. 4:

$$resolution = 2^{number\ of\ storage\ bits} \qquad (4)$$

However, the ADC is not the bottleneck for bit resolution. The transfer of the sampled EMG signal to a computer is carried out via universal serial bus (USB) connection. This employs the universal asynchronous receiver/transmitter (UART) protocol which has a limit of 8 information bits, meaning a maximum resolution of $2^8 = 256$. Therefore, an ADC with 8-bit resolution can be used. The UART protocol is executed between the microcontroller and a computer software via a serial port. This software must be able to receive and store data from the microcontroller at a sufficiently high baud rate. Due to overhead involved in the execution of the microcontroller code, computer software code, and the UART protocol through a serial port, the sampling frequency of the stored EMG signal can be smaller than the ADC's. This overhead should be kept to a minimum, or alternatively a higher sampling frequency can be used.

5.5 Feature extraction

To carry out classification, raw data is rarely ever used. Instead, features are extracted from the raw dataset. There are many different types of features, and each represents some characteristic(s) of the data. EMG features are usually divided into time-domain features and frequency-domain features (further explored by Nazmi et al. [30]). By using features rather than raw data, the information contained in the raw data can be thought of as being 'concentrated'. Features must characterize gestures and be repeatable across trials. This means that extracted features can distinguish different hand gestures from each other and make those distinctions consistently across trials [21].

Before feature extraction is carried out, the raw data is segmented. This process ensures that only the valuable parts of the data are used. In the case of EMG, this means eliminating the beginning and final parts of a signal which may contain the transitions between different gestures and a rest state. This can be implemented automatically using thresholding algorithms, or manually such as in Motoyama et al. [17]. Another important aspect of segmentation is windowing. This is the process of splitting a single original data signal into multiple fragments or windows. The key characteristics for the windows are their length (usually given in milliseconds for EMG) and whether they overlap with each other. In Phinyomark et al. [32], it was found that windows of length 500 ms performed strongly, and that overlapping between windows did not impact classification significantly. After segmentation, feature extraction is carried out on each individual window rather than on the whole signal, which allows more datapoints to be input to the classification algorithm.

One of the most common time domain features found in literature is root mean square (RMS) voltage. It has been used in various studies [14] [17], and is found by squaring the values in a dataset, taking the arithmetic mean of the squares, and then taking the square root of the mean. The operation can be seen in Eqn. 4:

$$RMS = \sqrt{\frac{1}{T}\sum_{t=0}^{T}(x_n(t))^2} \qquad (4)$$

Where $x_n(t)$ is the signal at all times t and T is the number of samples in the window.

5.6 Classification

The final stage of a hand gesture classification system is the classifier itself, which must learn to distinguish between different classes of data by comparing their features. There are several classifiers used in literature, such artificial neural networks, Bayesian classifiers, support vector machines and K-nearest neighbour [30]. This investigation makes use an SVM. SVMs are powerful supervised machine learning algorithms. They are given a set of training data, from which they can construct a model of the classes. They do this creating a hyper plane (see Fig. 9) between the different classes of data, maximising the margin between the hyperplane and the datapoints. They can also carry out data transformations into a space with more dimensions to classify less distinguishable data. They then label new testing data based on which side of the hyperplane it falls.

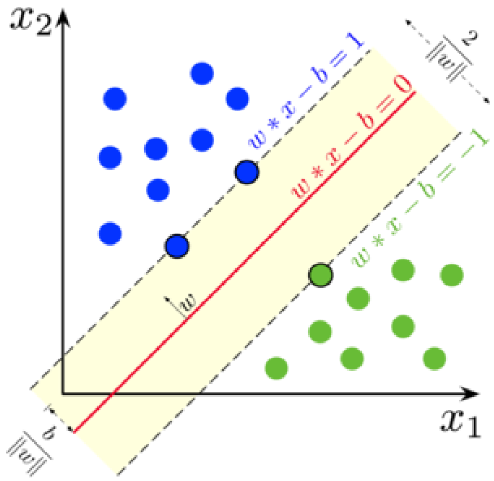


Figure 9: A hyperplane separating binary data classes in an SVM

5.7 Hand gesture selection

The number of gestures to be used in this investigation must be chosen. As mentioned in sections 1.2 and 1.3, more channels will allow for a higher classification accuracy on classification problems with higher DoFs, and the system used in this investigation has 4 channels. A problem with too many DoFs will yield bad classification results, which is undesirable, but a problem with too few DoFs risks eclipsing the effects of electrode positioning on classification accuracy. Because literature varies in number of channels used, a ratio of number of EMG channels to DoF is evaluated for a range of papers in Tab. 1 below.

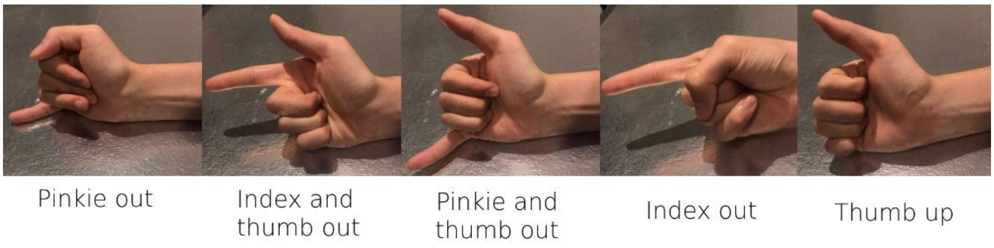
Channel number to degrees of freedom ratio	Decimal	Reference
8:5	1.6	[14]
10:12	0.83	[20]
8:5	1.6	[19]
8:10	0.8	[33]
7:10	0.7	[17]
4:10	0.4	[32]
	mean: 0.988	

Table 1: Ratio of EMG channels to degrees of freedom

The mean ratio (in decimal representation) of channels to number of electrodes was found to be about 1, which would suggest that a 4 DoFs problem is reasonable for this investigation. However, there is a lot of variance between the research papers so the ratio may change in future investigations.

The selection of hand gestures to use for HMI must be made with the user and the application in mind. Firstly, this means selecting hand gestures which are simple and intuitive to make, which the average person is likely to have already used. This includes for example, gestures used for greeting, saying goodbye, or playing games. Intuitive gestures can also relate to how humans interact with objects, such as turning a page in a book, writing with a pen, or holding cutlery. Furthermore, the gestures should be related to the application of the HMI. For example, an HMI used to navigate the pages of an e-book should make use of pronation and supination, as to the average person, these gestures are already associated with navigating books. Although most users would be able to carry out a tip pinch grasp, this does not make it a good selection for said application. Lastly, the gestures should be conducive to a high classification accuracy. This is achieved by selecting gestures which are anatomically distinct, using varied muscles during execution. Since this report does not restrict the acquisition system to only one application, intuitive and generically applicable gestures will be prioritised.

The first stage of gesture selection was carried out by reviewing four reputable papers which investigated hand gesture recognition via EMG. From each of these papers, 5 intuitive hand gestures were chosen and reproduced in Tab. 1 below.



[14]

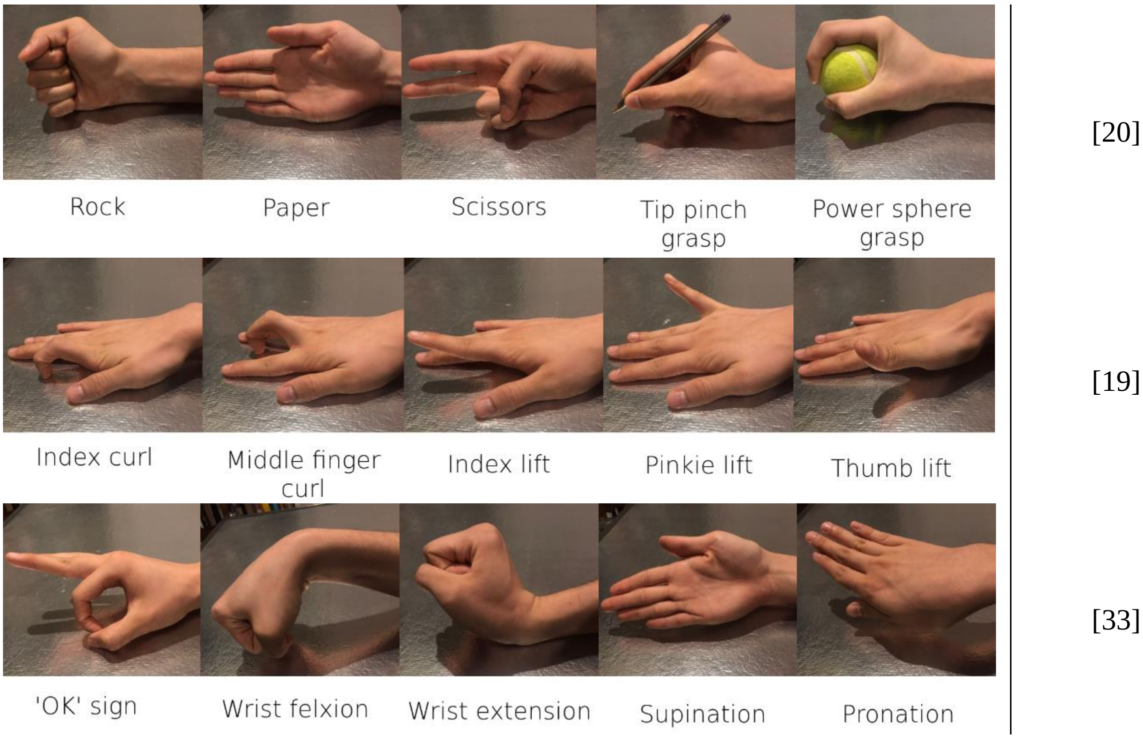


Table 2: Collection of hand gestures used in literature

After careful consideration of the above gestures, the four most intuitive, applicable, and differentiable gestures were selected to be ‘Rock’, a closed fist, ‘Paper’, and open hand, ‘Supination’, a medial rotation of the palm to face up, and ‘Pronation’, a lateral rotation of the palm to face down, all shown in Fig. 10.

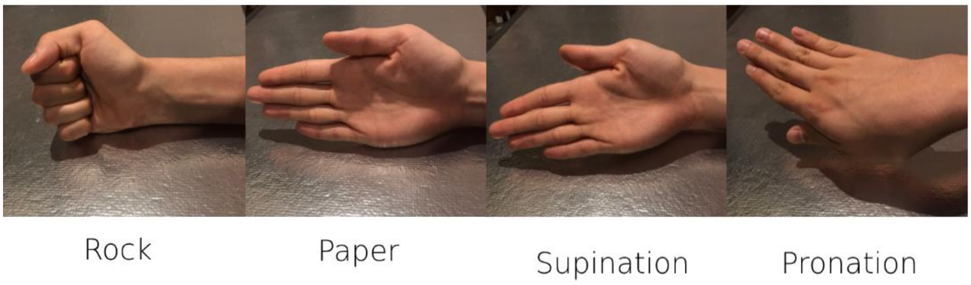


Figure 10: Selection of ideal hand gestures for classification

6 Literature review

Many valuable studies exist investigating different elements of EMG classification systems, seeking to improve all aspects of current software and hardware. For the most part however, there has been little investigation comparing different electrode layout positions on classification accuracy. Research presented by Rao et al. in [34] investigated the different positions electrodes can take on a muscle to optimise recorded EMG signals, but this is not the same as investigating which combination of muscles lead to higher classification performance.

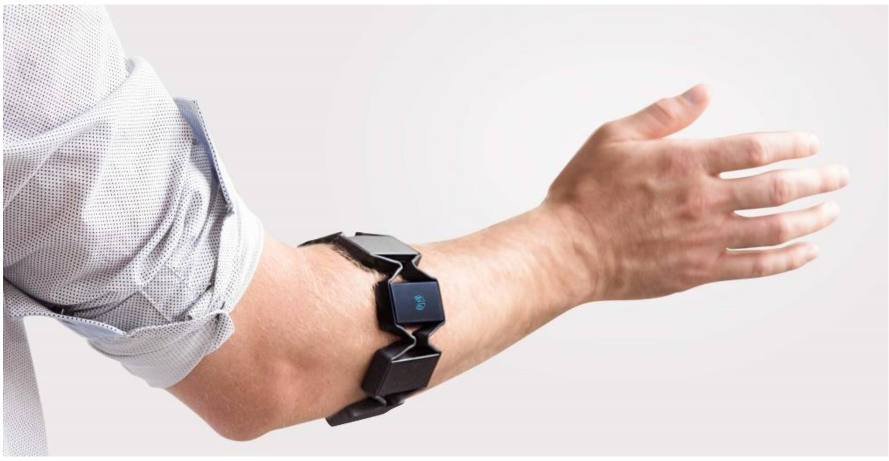


Figure 11: MyoBand in baseline electrode configuration (image by Thalmic Labs)

The present focus on software and hardware components has led researchers to overlook the question of electrode positioning. Many studies opt for commercially available acquisition systems, some with pre-determine electrode positioning, allowing more resources to be invested into the specifics of their investigation. Some examples of these commercially available acquisition systems are the MyoBand by Thalmic labs, the MyoBock 13E200-50 by Otto Bock, or the Trigno electrodes by Delsys. Of these, the MyoBand restricts electrode positioning to the baseline layout seen in Fig. 11, with electrodes equidistantly placed around the forearm, below the radio humeral joint. However, despite the freedom to place purchased electrodes anywhere on the forearm, many studies opt for the MyoBand baseline layout. The various acquisition systems, numbers of channels and the placements of reference electrodes for several studies are displayed in Tab. 3 below. Despite these systems being very diverse, all of them opt for the baseline layout. With this layout, they all achieved high classification accuracies. For this reason, the baseline layout has been selected in this investigation, to act as a benchmark for the classification performance of the whole system against which the bespoke electrode performance can be compared.

Acquisition system	Nº of ch.	Electrode layout	Reference electrode location	Ref
MyoBand by Thalmic labs	8	Baseline	Between bipolar electrodes	[14]
Custom build	7	Baseline around wrist	On the ulna	[17]
MyoBock 13E200-50	10	Baseline	Not given	[20]
Trigno by Delsys	12	Baseline	Not given	[20]
Bio-Semi Active Two	8	Baseline	Between bipolar electrodes	[19]
Custom build	2	Baseline	Between bipolar electrodes	[18]

Table 3: List of investigations using the baseline electrode layout

Despite the popularity of the baseline layout, many researchers also decide to create an original electrode layout. Unfortunately, the classification accuracy of different layouts used in different papers cannot be directly compared to evaluate the quality of the layout, due to the innumerable variables which change with different test subject, experimental tools and procedures, extracted features, and classification models. However, researchers select muscles based on a combination of existing literature and their understanding of forearm anatomy. This is trustworthy expertise which can be drawn from to create a bespoke electrode setup. A table of candidate muscles used in literature can be constructed to facilitate the selection of key muscles.

First, the reader must acquire a basic understanding of the key forearm muscles. The superficial and deep muscles of the anterior forearm (palm of the hand facing up) are presented in Fig. 12.

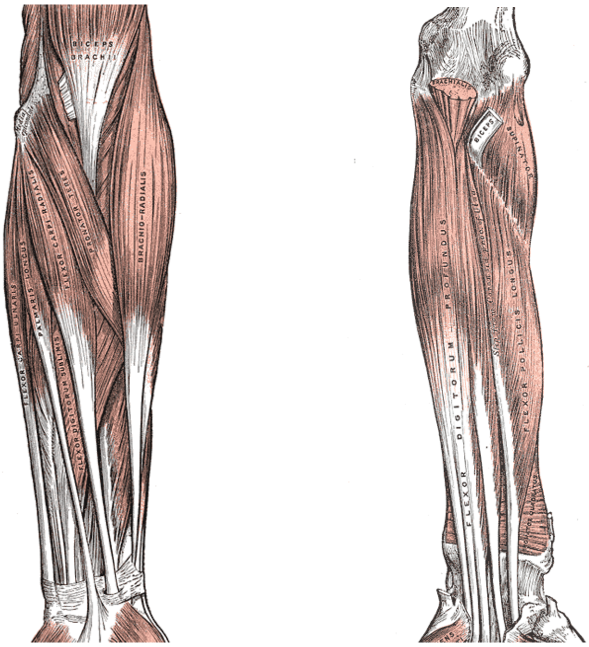


Figure 12: Superficial (left) and deep (right) muscles of the left anterior forearm (image by Gray, [35])

The superficial muscles of the posterior forearm (knuckles facing up) are presented in Fig. 13.

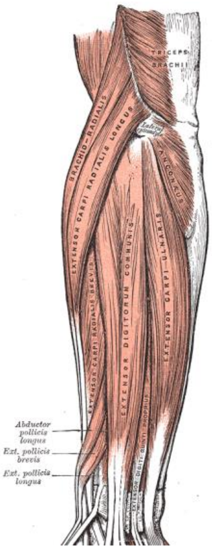


Figure 13: Superficial muscles of the right posterior forearm (image by Gray, [35])

Although the muscles presented in Figs. 7 and 8 are in the forearm, they are responsible for the execution of hand gestures, as they connect to various structures in the hand through tendons. The hand and wrist are capable of complex gestures, and these can be understood as the combination of different muscles contracting. The muscles in the anterior forearm are responsible for hand and wrist flexion, which is why they are named *flexor carpi radialis* and *flexor carpi ulnaris*. Flection moves the hand towards the anterior forearm. The *palmaris* longus also aids in flexion. Also in the anterior forearm are the *flexor digitorum superficialis* and *flexor digitalis profundus*, responsible for the flexion of the fingers. The muscles in the posterior forearm are responsible for hand and wrist extension, which is why they are named *extensor carpi radialis* and *extensor carpi ulnaris*. Extension moves the hand towards the posterior forearm. Also in the posterior forearm is the *extensor digitorum*, which is responsible for the extension of the fingers, and the *brachio radialis* is responsible for wrist pronation and supination [36]. The locations of these muscles are difficult to discern on Figs. 7 and 8 due to the small font in the illustrations but are clearly labelled in Tabs. 4 and 5, where the diagrams by Gray [35] are adapted to show which muscles have been most used in literature.

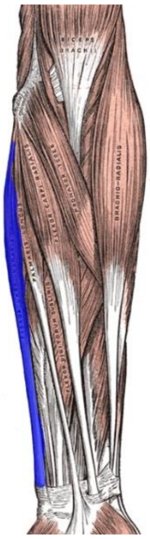
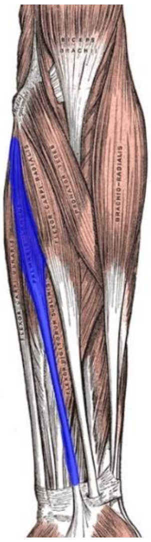
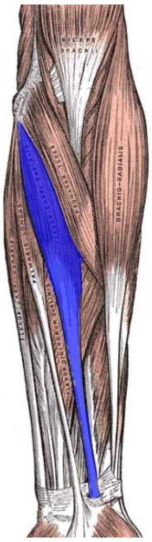
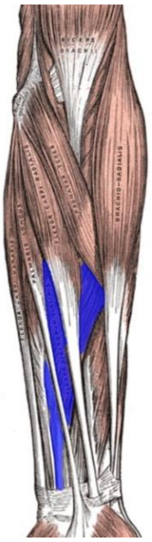
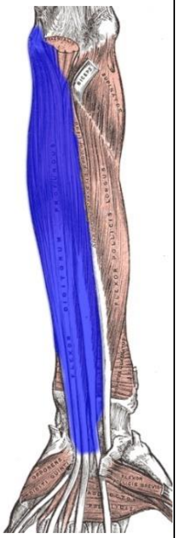
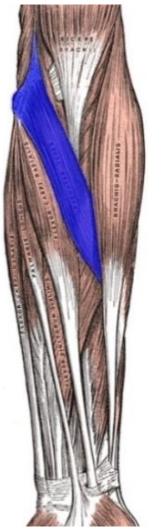
Muscle name	Flexor carpi ulnaris	Palmaris longus	Flexor carpi radialis	Flexor digitorum superficialis	Flexor digitorum profundus	Pronator teres
Muscle position						
Used in	[37] [38] [33]	[37]	[37] [39] [34] [33] [40]	[40]	[38]	[32] [37] [33]

Table 4: The most widely used anterior forearm muscles for hand gesture classification via EMG

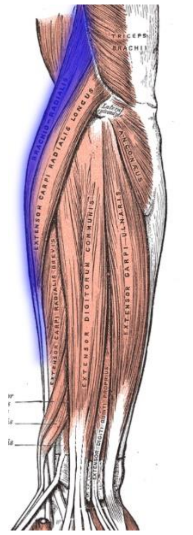
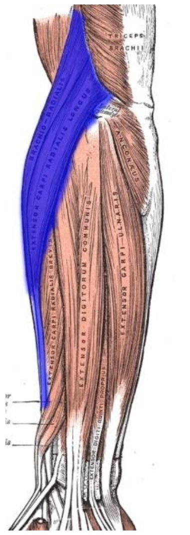
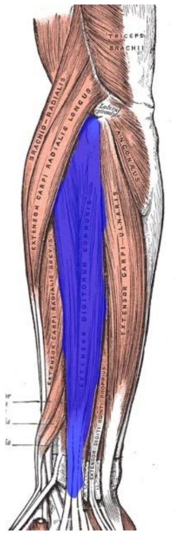
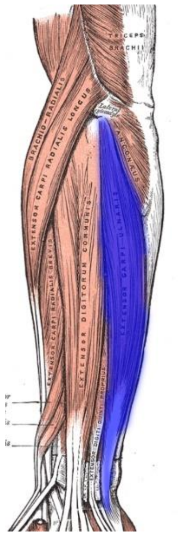
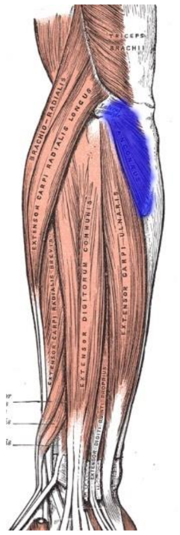
Muscle name	Brachio radialis	Extensor carpi radialis	Extensor digitorum	Extensor carpi ulnaris	Anconeus
Muscle position					
Used in	[32] [37] [33]	[38] [34] [40]	[32] [37] [38] [33] [40]	[32] [37] [40]	[33]

Table 5: The most widely used posterior forearm muscles for hand gesture classification via EMG

Because the paper hand gesture requires the extension of the fingers, the *extensor digitorum* is an ideal candidate. Furthermore, it is the muscle which has been used most frequently in the tabulated studies (5 different studies), giving further confidence of its good fit. The *brachioradialis* is closely linked to the hand gestures of pronation and supination, as well as having been used in 3 different studies. The *flexor digitorum superficialis* has only been used in 1 study, but its use in the flexion of the fingers caters well to the rock hand gesture. The last remaining channel will be positioned on the *flexor carpi radialis,* which generally plays a role in all the gestures and has been used in 5 different studies.

A note should be made concerning the proper positioning of the electrodes on these muscles. As has been found by both Farina et al. [41] and Hermens et al. [42], there are two zones of the muscles which should be avoided when placing EMG electrodes. These are the innervation zone and the tendon zone, found at the proximal (near the elbow) and distal (near the wrist) extremities of the forearm muscles. Placing electrodes on these zones leads to inconsistent recordings with considerable variations in amplitude. The muscle belly, which is found between the innervation zone and tendon zone, is the ideal location for the electrodes.

It is expected that the baseline layout will perform less well than the bespoke layout as it does not take into consideration the specific muscle activation described above, and it is located near the innervation zone of most muscles it records.

7 Methods

7.1 System design and implementation

7.1.1 Hardware: Simulation

The first stage in implementing the hardware for the EMG acquisition system was simulation. The circuit diagrams presented in Figs. 3, 4, and 5, were implemented in the electronic schematic capture and simulation program NI Multisim. In the case of passive components, virtual components of appropriate resistive or capacitive values were chosen since there was no concern around the availability of resistors and capacitors in the production stages of the acquisition system. In the case of active components, a selection was made based on a multitude of factors which impact the production stage of the system, including component performance, price, availability, and reliability. The chosen active components were the INA128 precision, low power instrumentation amplifier and the UA741 general-purpose operational amplifier by Texas Instruments. At this stage, a virtual ± 5 V power supply for the active components was used. The complete circuit with two virtual EMG signal inputs was simulated to verify the characteristics of its output. The original waveform undergoes an initial amplification of gain 65, common mode rejection, a secondary amplification of gain 11, and a DC offset of 1.15 V (see section 7.2 for ADC specifications). The result of these stages can be observed in Fig. 14, where the waveform occupies the 3.3 V range, has no negative values, and no evident 50 Hz frequency component left from the EMG input.

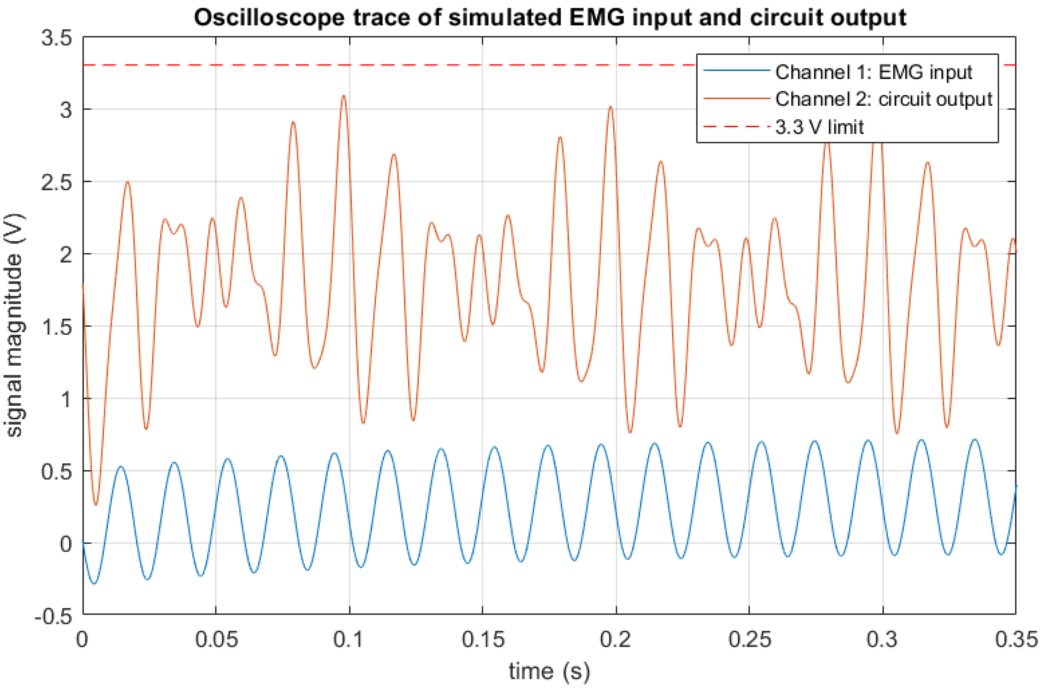


Figure 14: Circuit simulation in NI Multisim

In Fig. 15 is the bode plot demonstrating the bandpass filter stage successfully implementing -3 dB corner frequencies of 7.7 Hz and 724 Hz. This is not the ideal 10-500 Hz EMG band, but it is the closest values attainable with the resistor and capacitor values available in the PCB stage.

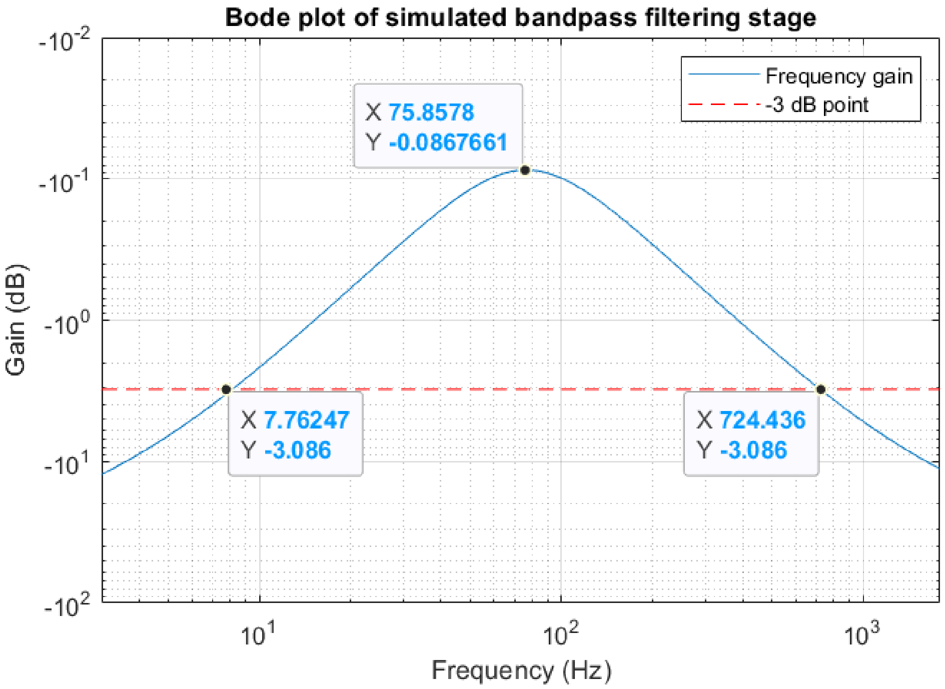


Figure 15: Frequency response of simulated bandpass filtering stage

See Appendix 1 for full circuit diagrams and active component selection candidates.

7.1.2 Hardware: Breadboard prototyping

The second stage in hardware development was the testing of a physical prototype assembled on a breadboard. During the prototype phase some choices were made to facilitate assembling, taking apart, reassembling, and testing the circuit. Firstly, the Keithley 2231A-30-3 triple channel DC power supply set to ±5V was used rather than a battery power supply. A Rigol DG1062z function generator was used to generate a sinusoidal wave of amplitude 10 mV and frequency 100 Hz, to replace the positive electrode. This was chosen over setting up electrodes on the skin every time the circuit had to be tested as it is not time efficient. The negative electrode and the reference electrode were both grounded.

The stages of hardware were implemented and tested in succession to facilitate troubleshooting, rather than assembling the entire circuit before testing. The importance of learning to intelligently troubleshoot circuits cannot be understated, as most of the time invested in hardware implementation was spent troubleshooting. The issues can be of various gravity and scope, though this does not always correlate to the time taken in resolving them. For example, a cable which was not plugged into the breadboard properly and a malfunctioning UA741 took a similar amount of time to resolve.

The Rohde & Schwarz RTB2004 oscilloscope was used to test the output of the breadboard and verify it matched expectations. The output of the breadboard prototype can be seen in Fig. 16, a 3.3 V peak-to-peak sinusoidal wave with no negative voltage values.

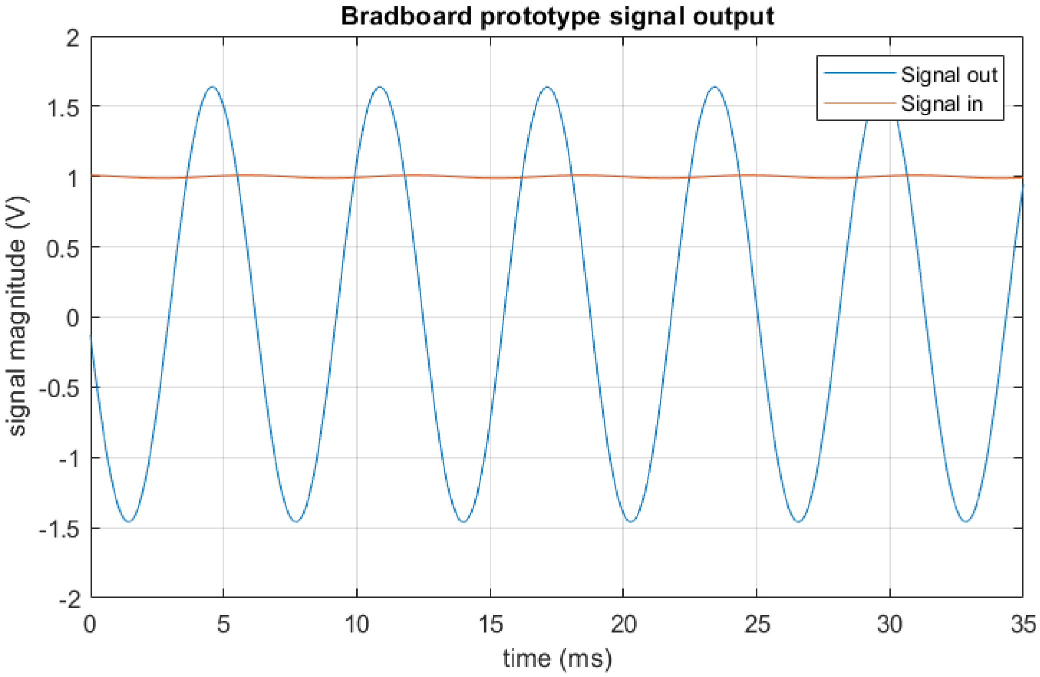


Figure 16: Output signal of breadboard prototype

Fig. 17 presents the bode plot for the bandpass filtering stage with corner frequencies of 8 Hz and 600 Hz.

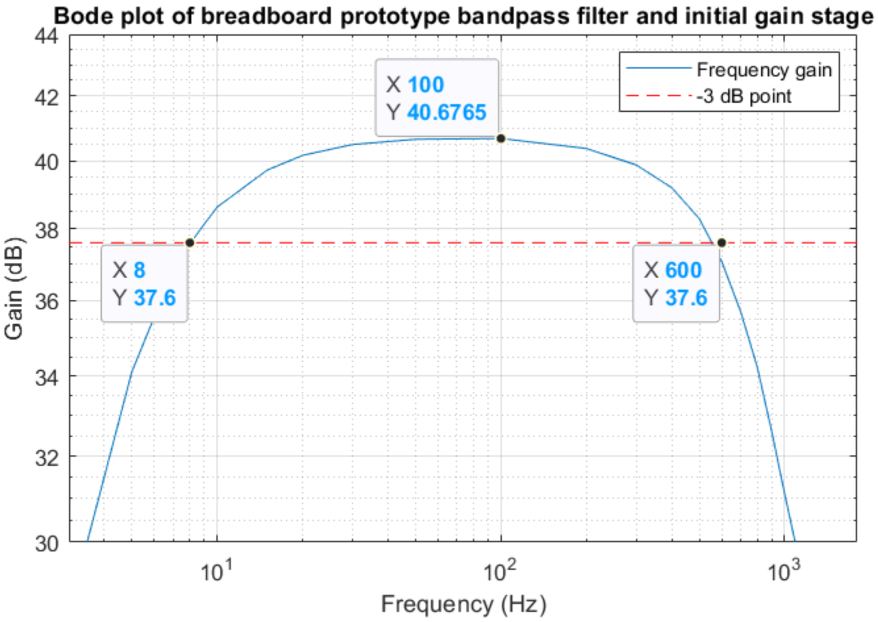


Figure 17: Frequency response of breadboard prototype bandpass filter and instrumentation amplifier

At this stage, actual EMG signal was not tested because of the very noisy nature of breadboards, which would likely saturate the system output. Common mode rejection was not yet tested, and it was expected that minor changes to the values of gain resistors would need to be made when using actual EMG signals. See Appendix 2 for a picture of the breadboard prototype.

7.1.3 Hardware: PCB design

The first stage of PCB implementation involved the production of a prototype PCB, tested to finalise the gain resistor values of the system, and verify appropriate common mode rejection. A power supply circuit was also designed at this stage to provide ± 5 V to the active components of the circuit (see Fig. 18). The prototype led to the final gain of the circuit being finetuned to 64 in the initial amplification and 19 in the secondary amplification.

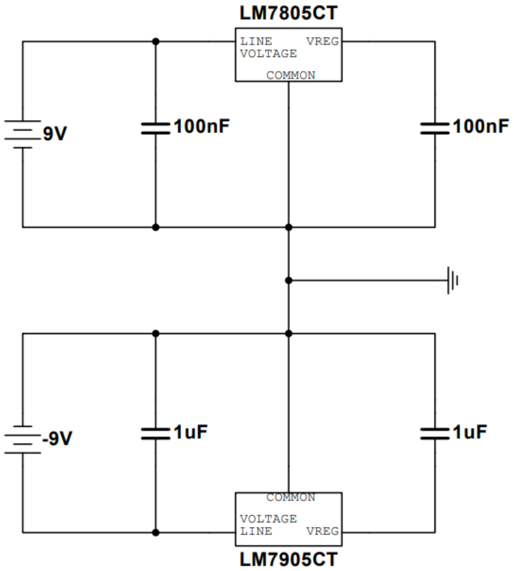


Figure 18: Power supply circuit for +5 V and -5 V

The final stage of hardware implementation was the design and production of a PCB to acquire four channels of EMG signal, meaning the circuits presented in Figs. 3, 4, and 5 were manufactured four times. The design of the four channels was identical, and the final product can be seen in Fig. 19.

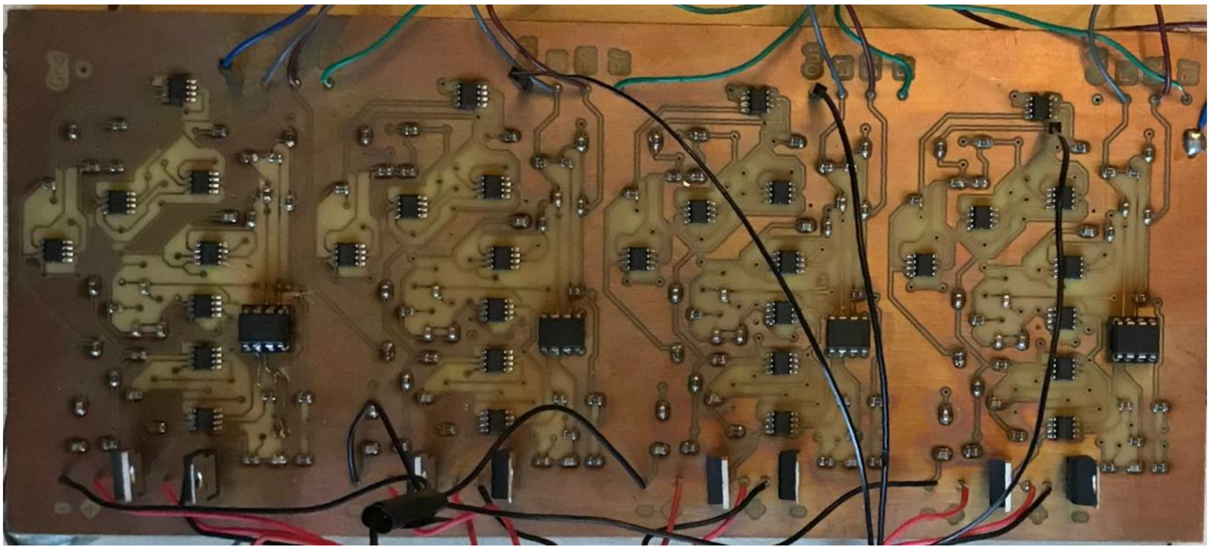


Figure 19: 4 channel PCB implementation

7.2 Data acquisition

For the acquisition of data, the MSP432P401R (MSP) by Texas Instruments was chosen. This microcontroller has a single ADC, but it can be used for four channels by using an input multiplexer. This sequentially switches the input to the ADC between four analog input pins of the microcontroller to which the PCB channels are connected. This sequential switching loops back to the first channel once the fourth channel has been sampled, allowing for continuous sampling. The switching happens fast enough to simulate four parallel ADCs working because the time difference between subsequent samples is negligible [43].

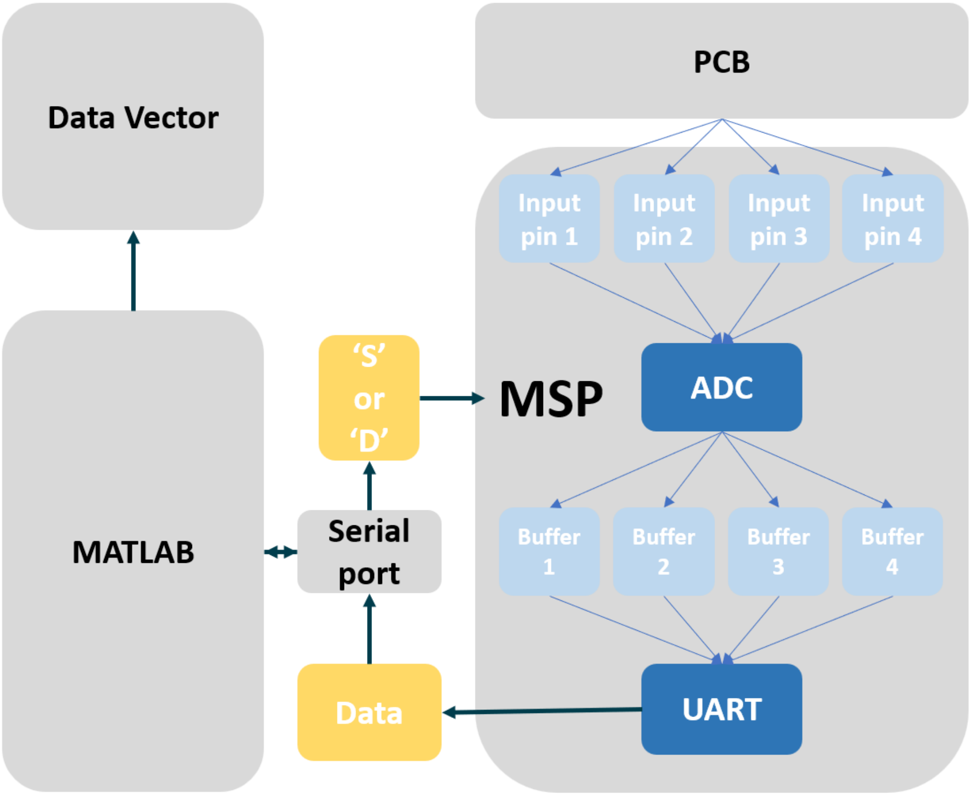


Figure 20: High level block diagram of ADC setup

Fig. 20 illustrates the key components in the process of sampling and recording. First, the MSP and a MATLAB script running on a computer must be interfaced by creating a shared serial port object. Then, to begin sampling, the MSP waits to receive an 'S' character from the serial port, which is sent by the MATLAB script. The ADC then converts the analog value at the first analog input pin into an 8-bit unsigned integer and stores it in a temporary buffer. It does this with the next three analog values, storing each in a different buffer. Then, the integers stored in each buffer are sequentially sent to the serial port via UART protocol. There, MATLAB stores the integers into a vector as they arrive. Once MATLAB received as many integers as it has asked the MSP for, it sends a 'D' character to end the sampling cycle.

The sampling frequency of the MSP was set to 46080 Hz. However, because multichannel sampling is carried out, the sampling frequency per channel is a fourth of this value, or 11520 Hz. This is considerably larger than the minimum sampling frequency of 1 kHz, so it is certain that the overhead does not pose sampling problems. See Appendix 3 for the MATLAB and MSP code.

With the PCB, ADC, and computer all connected, the experiment could begin. The test subject was a 20-year-old male, who's forearm skin was prepared by thoroughly rubbing it with alcohol wipes. Then the electrodes were attached to his forearm in the locations described in section 6 and connected to the PCB input pins using insulated crocodile clips, as shown in Fig. 21. Channel 1 was connected to the *brachioradialis*, channel 2 was connected to the *flexor carpi radialis*, channel 3 was connected to the *flexor digitorum superficialis,* and channel 4 was connected to the *extensor digitorum*. Conductive electrode gel was used to decrease the impedance between the skin and the electrodes. A test routine was designed so that all trials were consistent. The steps of the test routine were displayed on the computer screen so that the right gestures were made at the right time. First, the test subject was warned that the experiment was beginning. Then, he was instructed to rest his hand for 5 seconds. Once the 5 seconds had passed, the test subject was asked to perform the first gesture, 'rock'. This occurred three more times, interchanging 5 seconds of rest with 5 seconds of gesture until all four selected gestures were carried out. Finally, the test subject was informed that the trial was over, and the MATLAB program saved the recorded data to a matrix. In total, 30 trials were carried out for each electrode layout configuration.

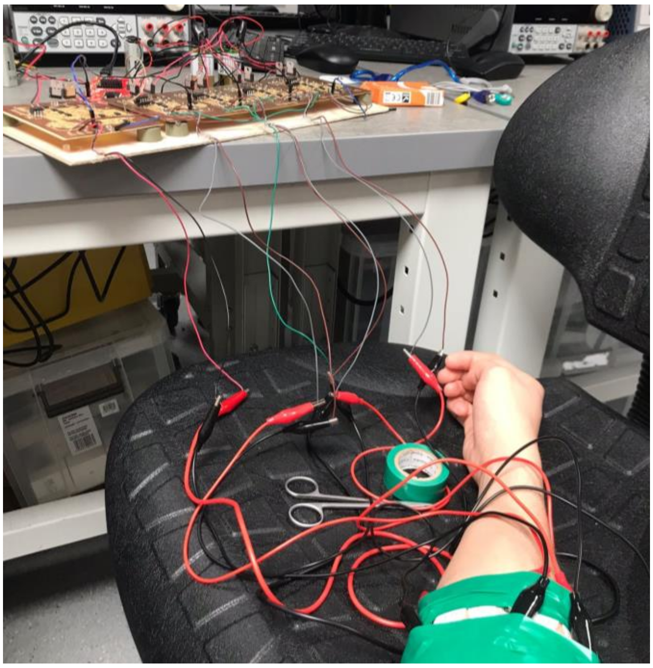


Figure 21: Process of connecting electrodes to the PCB for EMG recording

7.3 Data pre-processing and classification

The initial data pre-processing was carried out in MATLAB. Following the methods of Motoyama et al. [17], the raw EMG recordings were manually trimmed. The first second of recording was recurrently highly variable in comparison with the following four seconds. This is due to the variability in reaction time of the test subject to the instructions appearing on the computer screen, as well as the differences in transition between a rest state and a gesture. Then, following Phinyomark et al. [32], 500 ms windows were used, without overlapping. In total, this produced 8 RMS values for each four second EMG signal, yielding a total of 240 RMS values for each gesture, for both electrode layout configurations.

The classification algorithm was trained and tested using Python and the scikit-learn software. First, the data was normalised using a standard scaler, then a random selection of training and testing data was made in the ratio of 9:1. The SVM was trained using the training data, and the model's classification accuracy was tested on the testing data. Normalised confusion matrices were made to provide more insight into individual gesture classification. See Appendix 4 for MATLAB and Python code.

8 Results

When the classification algorithm was executed using data collected with the baseline electrode layout, it achieved an accuracy of 68.8%, while the classification accuracy achieved when using data collected with the bespoke electrode layout was 83.3%. This means the bespoke electrode layout yielded a higher classification accuracy.

The normalised confusion matrix for the baseline and bespoke electrode layouts can be seen in Figs. 22 and 23 respectively. For the baseline electrode layout, the most frequently misclassified gesture was paper, only classified correctly 58% of the time. The most consistent misclassification was mistaking rock for paper, which occurred 23% of the times that rock was executed.

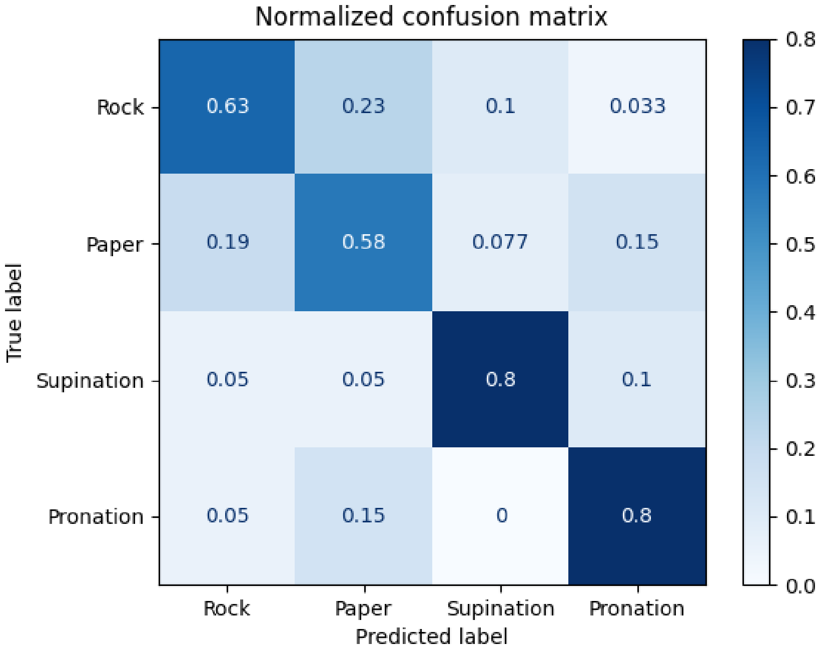


Figure 22: Normalized confusion matrix demonstrating individual hand gesture classification accuracies for the baseline electrode layout

For the bespoke electrode layout, the most frequently misclassified gesture was pronation, classified correctly 77% of the time. The most consistent misclassification was mistaking pronation for rock, which occurred 14% of the times that paper was executed.

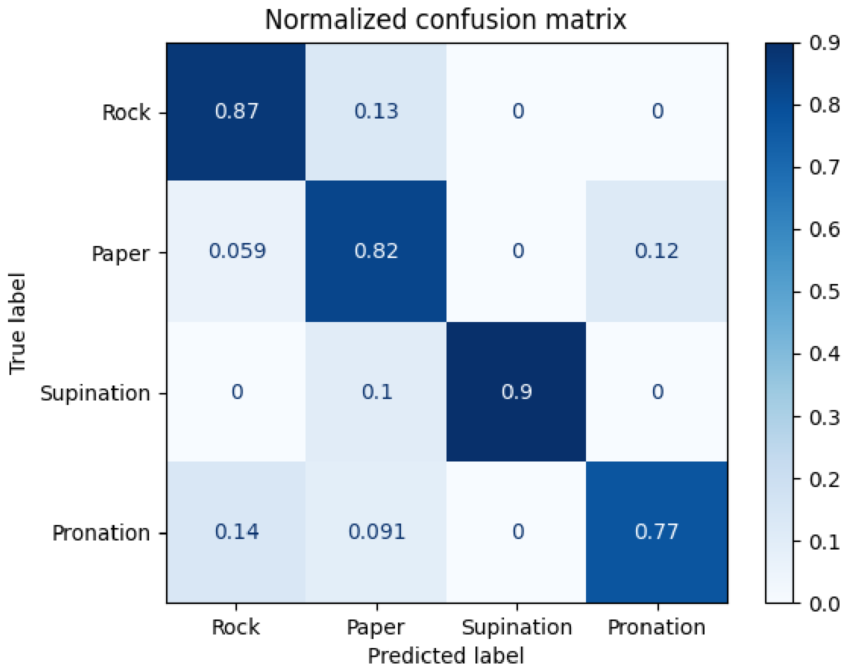


Figure 23: Normalized confusion matrix demonstrating individual hand gesture classification accuracies for the bespoke electrode layout

9 Discussion

Based on the results obtained, the hypothesis that the bespoke electrode layout would lead to higher classification accuracy has been confirmed. This may be due to a combination of factors. Firstly, the muscles relevant to the gesture effectuation were specifically targeted by the bespoke layout, while the baseline layout may have collected useless signals from unrelated muscles. Furthermore, the baseline electrode layout does not account for the ideal placement of electrodes on the muscle belly since it is positioned below the radio-humeral joint, where many forearm muscles are innervated.

Examining Figs. 22 and 23 reveals more insights on the effect of electrode positioning. The baseline electrode layout performed particularly badly when trying to differentiate the rock and paper gestures. The placement of an electrode on the *flexor digitorum superficialis* is the most likely cause of this successful rock classification. This muscle is responsible for the flexion of the digits, which is the only anatomical difference between the rock and paper gestures. Therefore, the bespoke layout has a strong ability to differentiate the two gestures by verifying if the digits are flexed or not. The baseline electrode layout has no such ability because the *flexor digitorum superficialis* is found lower down on the forearm, and so rock and paper are frequently misclassified.

Another interesting point from the confusion matrices is that the pronation gesture is misclassified more frequently with the bespoke layout than the baseline layout, in contrast with all the other gestures. This suggests that one or a combination of the electrodes in the baseline layout are positioned in a way which captures more information of the pronation gesture. This is consistent with the understanding of forearm anatomy, since muscles such as the *flexor carpi ulnaris* and the *palmaris longus* are strongly involved in the flexion of the wrist, and their bellies are located at the site of the baseline electrodes on the anterior forearm. This could be implemented into the bespoke electrode to improve its performance

These results confirm that placing electrodes to maximise gesture differentiability based on the understanding of muscle involvement for specific hand gestures increases the classification accuracy. Furthermore, the results improve said understanding of which muscles differentiate gestures the most. However, they also demonstrate that there are many aspects of muscle involvement which need to be studied further to be able to better design a bespoke setup, and they demonstrate that more differentiable hand gestures can be chosen.

10 Conclusion

This investigation demonstrated that a careful selection of both hand gestures and electrode placements leads to a more accurate classification of hand gestures. Through the experiment carried out, the successful electrode positions within the bespoke layout have been highlighted, while useful positions for a future improved layout have been suggested. In relation to the broader field of EMG for assistive technology, the knowledge created in this paper can be implemented in future assistive products. Based on the specific application of their products, developers will select a set of intuitive hand gestures. As was done in this report, they can compare the classification accuracy of different gestures and electrode layouts, to evaluate the electrode placement best suited to their application. It has been found that if their hand gestures contain digit flexion, a good electrode placement to contribute to classification is the *flexor digitorum superficialis*, while if they contain wrist pronation, a good electrode site is the *flexor carpi ulnaris* or the *palmaris longus*.

This investigation also showed how a hand gesture classification system via EMG can be constructed from start to end, including all stages of hardware and software. Although the development of the system was not the focus of the report, reproducible methods have been laid out for anyone to replicate this system. Potential improvements to this system are explained in the next section.

11 Future work

11.1 Improving methods

Various design decisions were made at every step of hardware and software development. To facilitate future improvements upon this work, some key improvement areas are outlined.

One place of improvement in hardware is to use shielded cables to connect the electrodes to the PCB. When the EMG signal travels from the electrodes to the PCB through a regular copper wire, the wire acts as an antenna to induce 50 Hz mains noise into the wire, which considerably increases the common mode noise in the signal. Active shielding protects cables from stray electromagnetic waves.

The testing routine used in the experiment can be changed to improve the robustness of the classification algorithm. In the methods, the hand gestures were always separated by a 5 second rest, and always executed in the order rock, paper, supination, and pronation. Randomizing the order in which the test subject carries out the gestures would improve the robustness of classification.

In data processing and analysis, a rich combination of time domain and frequency domain features should be used, to provide more types of valuable information to the classification algorithm. Another widely used classifier in literature is a convolutional neural networks, which can be implemented to verify consistent electrode layout performance.

11.2 Broadening the scope of investigation

The scope of this investigation was confined to comparing the baseline electrode layout commonly used in literature with a bespoke electrode layout, designed to maximise classification accuracy of the selected hand gestures. Many interesting areas for further investigation have been encountered during this report.

- The placement of the reference electrode could be changed to determine where it is conducive to highest classification accuracy. A location of particular interest is between the bipolar electrodes, as this placement is also widely found in literature employing the baseline configuration.
- A selection of gestures should be made to optimize the performance of the baseline electrode layout, to determine if it can match the classification accuracy of the bespoke electrode layout.
- A wider selection of gestures can be used with the baseline and bespoke electrode layouts, to determine the effect of hang gesture specific-muscle-activation on classification accuracy for different layouts.
- The *biceps brachii* should be included as an electrode location to determine the change in classification accuracy when including EMG from the upper arm.

12 References

[1] C. S. Crowe, B. B. Massenburg, S. D. Morrison, J. Chang, J. B. Friedrich, G. G. Abady, F. Alahdab, V. Alipour, J. Arabloo, M. Asaad, M. Banach, A. Bijani, A. M. Borzì, N. I. Briko, C. D. Castle, D. Y. Cho, M. T. Chung, A. Daryani and Demoz, “Global trends of hand and wrist trauma: a systematic analysis of fracture and digit amputation using the Global Burden of Disease 2017 Study.,” *Injury prevention : journal of the International Society for Child and Adolescent Injury Prevention,* vol. 26, pp. i115-i124, 2020.

[2] P. Gruhn, “Human Machine Interface (HMI) Design: The Good, The Bad, and The Ugly (and what makes them so),” in *66th Annual Instrumentation Symposium for the Process Industries,* Houston, 2011.

[3] L. Yang and J. Yunde, “A Robust Hand Tracking and Gesture Recognition Method for Wearable Visual Interfaces and Its Applications,” in *ICIG '04: Proceedings of the Third International Conference on Image and Graphics*, 2004.

[4] L. Kue-Bum, K. Jung-Hyun and H. Kwang-Seok, “An Implementation of Multi-Modal Game Interface Based on PDAs,” in *Fifth International Conference on Software Engineering Research, Management and Applications,* 2007.

[5] V. I. Pavlovic, R. Sharma and T. S. Huang, “Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Visual Interpretation of Hand Gestures for Human-Computer Interaction,” A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.

[6] M. Panwar, “Hand gesture recognition based on shape parameters,” in *International Conference on Computing, Communication and Applications*, 2012.

[7] Y. Chen, X. Liang, M. Assaad and H. Heidari, “Wearable Resistive-based Gesture-Sensing Interface Bracelet,” *2019 UK/ China Emerging Technologies (UCET),* pp. pp. 1-4, 2019.

[8] C. Ahmadizadeh, L. Merhi, B. Pousett, S. Sangha and C. Menon, “Toward Intuitive Prosthetic Control: Solving Common Issues Using Force Myography, Surface Electromyography, and Pattern Recognition in a Pilot Case Study,” *IEEE Robotics & Automation Magazine,* vol. 24, no. 4, pp. 102-111, 2017.

[9] J.-H. Kim, N. D. Thang and T.-S. Kim, “3-d hand motion tracking and gesture recognition using a data glove.,” in *IEEE International Symposium on Industrial Electronics,* Seoul, 2009.

[10] P. Kumar, J. Verma and S. Prasad, “Hand data glove: a wearable real-time device for human-computer interaction.,” *International Journal of Advanced Science and Technology,* vol. 43, 2012.

[11] M. B. I. Raez, M. S. Hussain and F. Mohf-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications,” *Biological Procedures Online,* vol. 8, no. 163, pp. 11-35, 2006.

[12] A. Macy, The Handbook of Human Physiological Recording, 2017.

[13] B. C. Fortune, A. Stewart, E. Hansenne, L. McKenzie, L. Chatfield and C. G. Pretty, “Design and Testing of a low-cost Electromyogram,” in *International Conference on Mechatronics and Machine Vision in Practice,* Auckland, 2017.

[14] E. Ceolini, C. Frenkel, S. B. Shrestha, G. Taverni, L. Khacef, M. Payvand and E. Donati, “Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing,” *Frontiers in Neuroscience,* vol. 14, 2020.

[15] T. D. Pollard, W. C. Earnshaw, J. Lippincott-Schwartz and G. T. Johnson, “Cell

Biology," Elsevier, 2017, pp. 671-697.

[16] B. Porr, "The Biosognal How-To," Attys, 15 January 2017. [Online]. Available: https://biosignals.berndporr.me.uk/. [Accessed 12 December 2021].

[17] H. Motoyama, T. Inoue, Y. O. Kato and J. Ozawa, "Electrode Layout of Wristband EMG Measuring Device," in *IEEE International Conference on Consumer Electronics,* Kyoto, Japan, 2015.

[18] T. S. Saponas, D. Tan, D. Morris, J. Turner and J. Landay, "Making Muscle-Computer Interfaces More Practical," in *28th International Conference on Human Factors in Computing Systems,* Atlanta, 2010.

[19] T. S. Saponas, D. Tan, D. Morris and R. Balakrishnan, "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces," in *Conference on Human Factors in Computing Systems,* Florence, 2008.

[20] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo and A. Hager, "Electromyography data for non-invasice naturally-controlled robotic hand prosthes," *Scientific Data,* vol. 1, no. 140053, 2014.

[21] D. Farina, "The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 22, no. 4, pp. 797-809, 2014.

[22] J. Wu, L. Sun and R. Jafari, "A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors," *IEEE Journal of Biomedical and Health Informatics,* vol. 20, no. 5, pp. 1281-1290, 2016.

[23] A. Péter, E. Andersson, A. Hegyi, T. Finni, O. Tarassova, N. Cronin, H. Grundström and A. Arndt, "Comparing Surface and Fine-Wire Electromyography Activity of Lower Leg Muscles at Different Walking Speeds," *Frontiers in Physiology,* vol. 10, 2019 .

[24] C. J. De Luca, "The Use of Surface Electromyography," *JOURNAL OF APPLIED BIOMECHANICS,* vol. 13, pp. 135-163, 1997.

[25] A. Chapman, B. Vicenzino, P. Blanch, J. Knox and P. Hodges, "Intramuscular fine-wire electromyography during cycling: repeatability, normalisation and a comparison to surface electromyography," *Journal of electromyography and kinesiology,* vol. 20, no. 1, pp. 108-117, 2010.

[26] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan and T. Chang, "Surface Electromyography Signal Processing and Classification Techniques," *Sensors,* vol. 13, no. 9, p. 12431–12466, 2013.

[27] B. Taji, S. Shirmohammadi, V. Groza and I. Batkin, "Impact of Skin–Electrode Interface on Electrocardiogram Measurements Using Conductive Textile Electrodes," *IEEE Transactions on Instrumentation and Measurement,* vol. 63, no. 6, pp. 1412-1422, 2014.

[28] A. Macy, "Electrodes for Physiological Recording," BIOPAC, 6 August 2015. [Online]. Available: https://blog.biopac.com/electrodes-for-physiological-recording/. [Accessed 11 December 2021].

[29] E. M. Spinelli, R. Pallas-Areny and M. Mayosky, "AC-Coupled Front-End for Biopotential Measurements," *IEEE transactions on bio-medical engineering,* vol. 50, no. 3, pp. 391-5, 2003.

[30] N. Nazmi, M. A. A. Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri and S. A. Mazlan, "A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions," in *Sensors*, Basel, 2016.

[31] S. A. Lopez, *Design and construction of an EMG multichannel acquisition system*

prototype", Torino: Politecnico di Torino, 2012.

[32] P. A., Q. F., C. S., S. C., T.-B. F. and L. Y., "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Expert Systems,* vol. 40, p. 4832–4840, 2013.

[33] V. Lantz, X. Chen, Q. Li, J. H. Yang and K. Q. Wang, "Test-Retest Repeatability of Surface Electromyography Measurement for Hand Gesture," in *2nd International Conference on Bioinformatics and Biomedical Engineering,* 2008 .

[34] G. Hossein, A. Siti and J. I. Asnor, "Recommended Surface EMG Electrode," in *IEEE Student Symposium in Biomedical Engineering & Sciences,* 2015 .

[35] H. Gray, Anatomy of the human body, New York: Bartleby, 2000.

[36] R. R. Seeley, T. D. Stephens and P. Tate, Anatomy and Physiology, McGraw Hill, 2008.

[37] A. Balbinot and G. Favieiro, "A neuro-fuzzy system for characterization of arm movements," *Sensors,* vol. 13, no. 2, pp. 2613-30, 2013.

[38] M. Takamitsu and M. Jun, "Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 60, no. 8, 2013.

[39] S. Kavya, M. Dhatri, R. Sushma and B. Krupa, "Controlling the hand and forearm movements of an," in *IEEE INDICON,* Bangalore, 2015.

[40] E.-P. Takala and R. Toivonen, "Placement of forearm surface EMG electrodes in the assessment of hand loading in manual tasks," *Ergonomics,* vol. 56, no. 7, pp. 1159-1166, 2013.

[41] D. Farina, R. Merletti, M. Nazzaro and I. Caruso, "Effect of joint angle on EMG variables in leg and thigh muscles," *IEEE Engineering in Medicine and Biology Magazine,* vol. 20, no. 6, pp. 1937-4186, 2001.

[42] 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, 2000.

[43] M. Pachchigar, "Demystifying High-Performance Multiplexed Data-Acquisition Systems," *AnalogDialogue,* vol. 48, 2014.

[44] K. Ogino and W. Kozak, "Spectrum analysis of surface electromyogram (EMG)," in *IEEE International Conference on Acoustics, Speech, and Signal Processing,* Boston, 1983.

[45] Arduino, "Arduino Uno R3 Prpduct Reference Manual," Arduino, Torino, 2021.

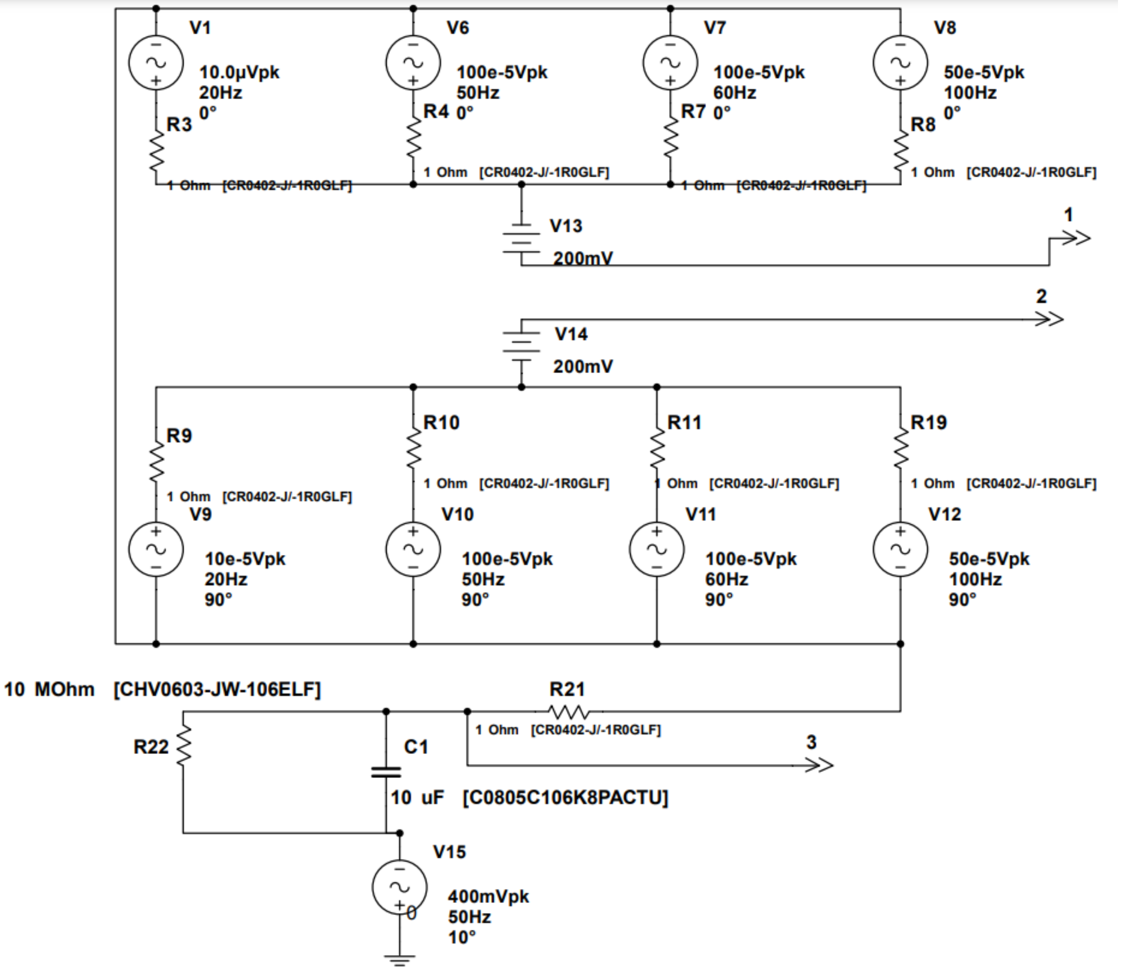
[46] Y. Chen, B. Luo, Y.-L. Chen, G. Liang and X. Wu, "A real-time dynamic hand gesture recognition system using kinect sensor," in *IEEE International Conference on Robotics and Biomimetics (ROBIO),* 2015.

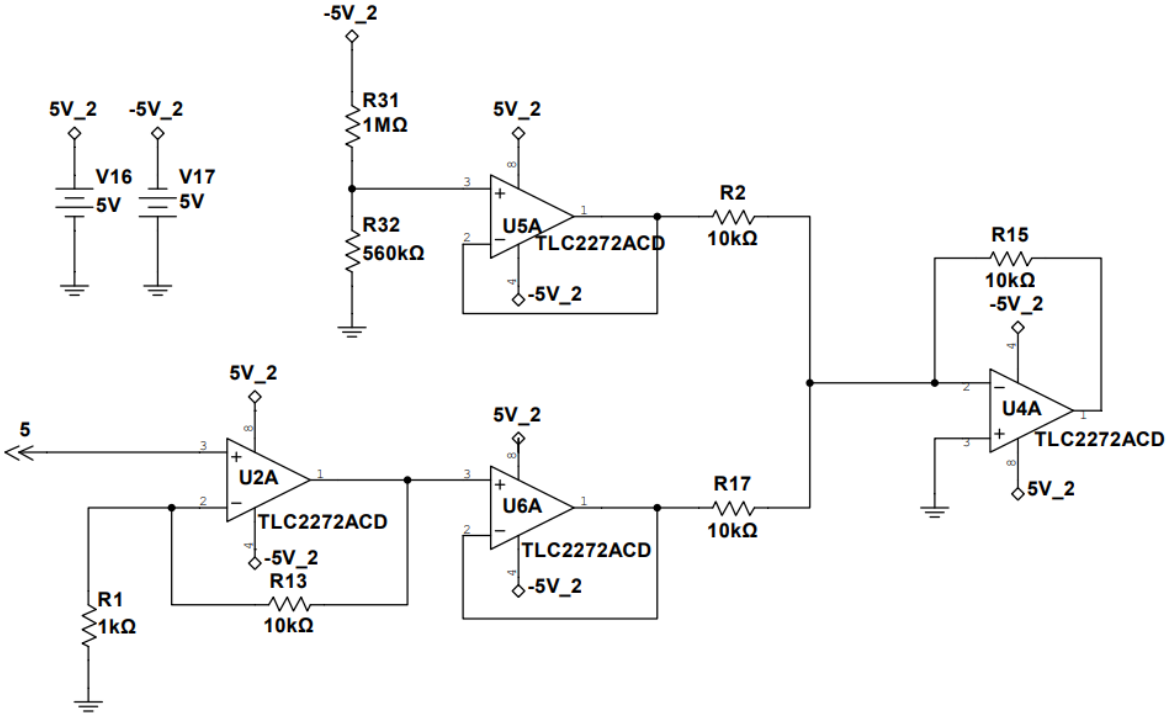
[47] R. Shodhan, C. Raffaella and S. Stefano, "Stiffness and position control of a prosthetic wrist," in *32nd Annual International Conference of the IEEE EMBS,* Buenos Aires, 2010.

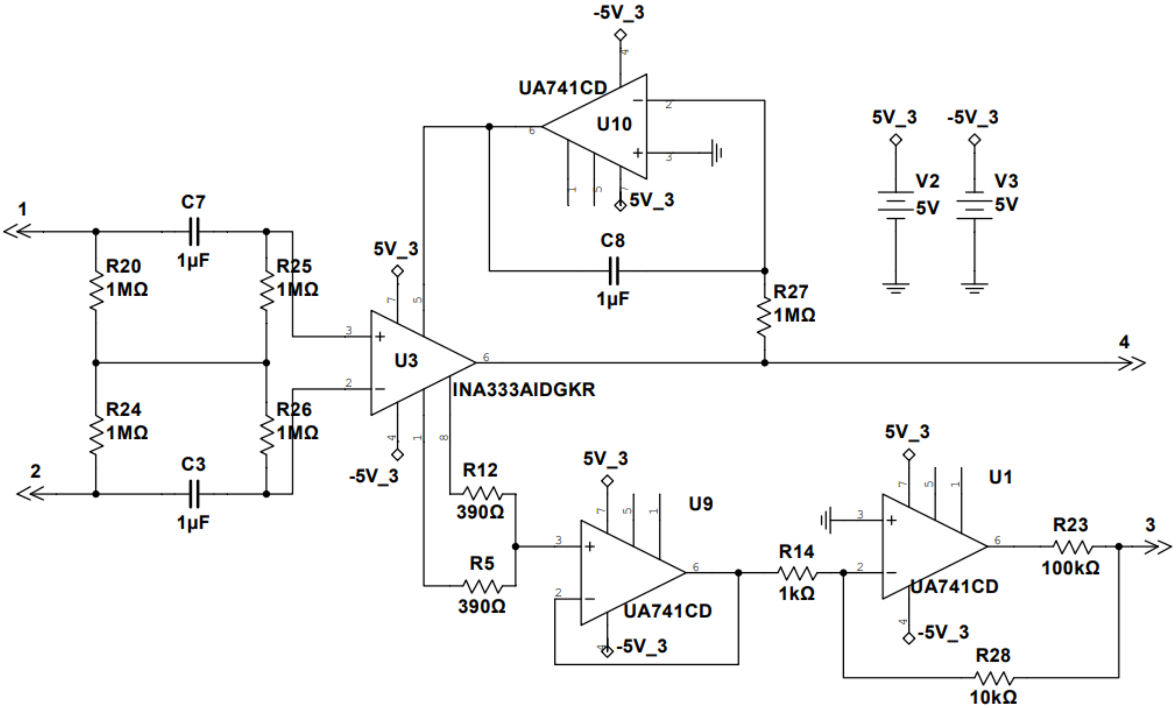
[48] S. A. A. a. A. J. I. H. Ghapanchizadeh, "Developing Multichannel Surface EMG Acquisition System by Using Instrument OpAmp INA2141," in *IEEE Region 10 Symposium,* 2014.

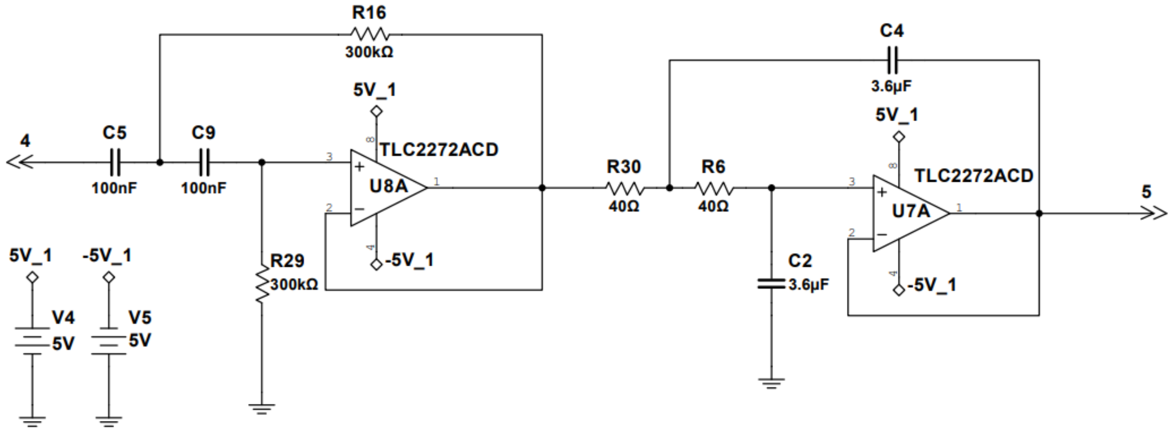
13 Appendices

13.1 Simulation






-5V_3
UA741CD
U10
5V_3
C8
1μF
R27
1MΩ
5V_3
-5V_3
V2
5V
V3
5V
1
C7
1μF
R20
1MΩ
R25
1MΩ
5V_3
U3
INA333AIDGKR
-5V_3
4
R24
1MΩ
R26
1MΩ
2
C3
1μF
R12
390Ω
R5
390Ω
5V_3
U9
UA741CD
-5V_3
R14
1kΩ
5V_3
U1
UA741CD
-5V_3
R23
100kΩ
3
R28
10kΩ


R16
300kΩ
5V_1
TLC2272ACD
U8A
-5V_1
4
C5
100nF
C9
100nF
R29
300kΩ
5V_1
-5V_1
V4
5V
V5
5V
C4
3.6μF
5V_1
TLC2272ACD
U7A
-5V_1
R30
40Ω
R6
40Ω
C2
3.6μF
5

13.2 Breadboard

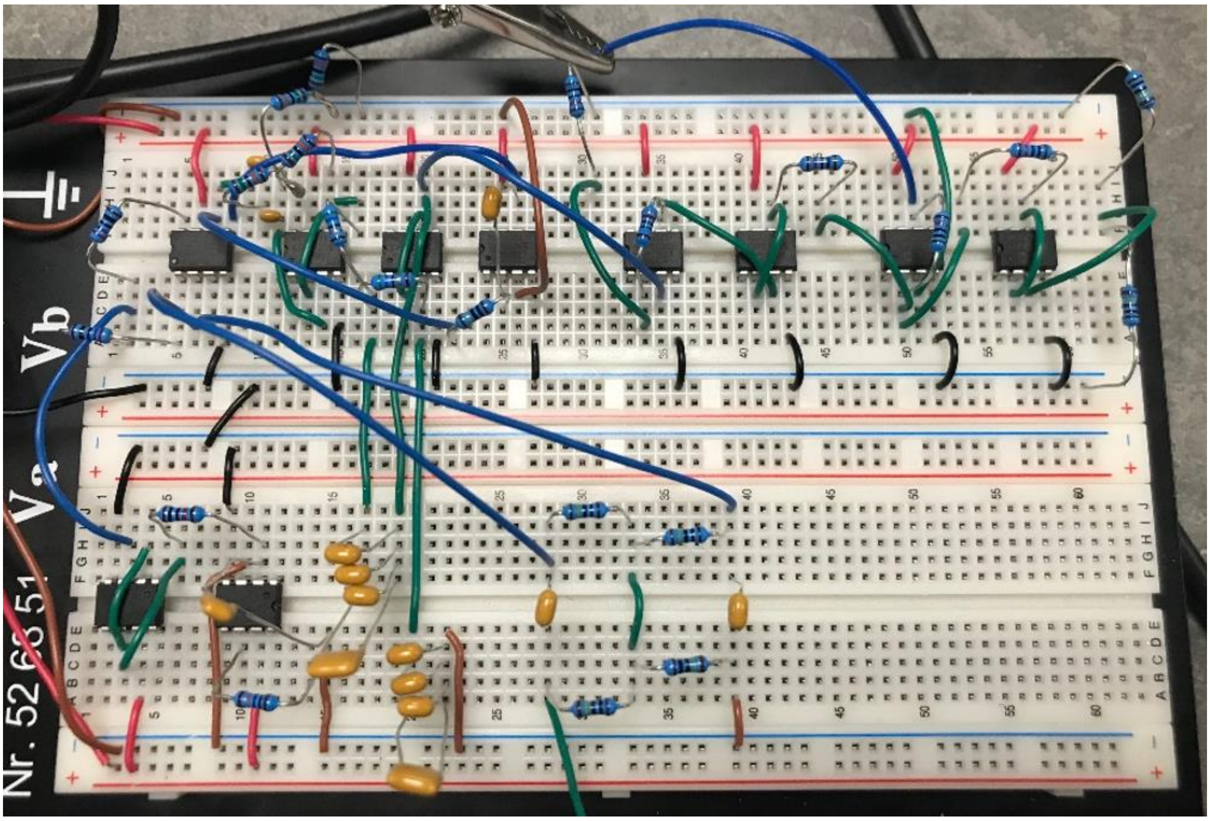


Figure 6: Breadboard prototype of designs seen in figures 3, 5, & 7

13.3 MATLAB and MSP code for ADC

MATLAB code:

```
%% Setup serial communication
clear all
clc
x = [0];
numOfSamplesToRecord = 5*46080;
delete(instrfindall);
s = serial('COM9');
set(s,'BaudRate',460800,'StopBits',1,'Parity','none');
s.InputBufferSize = numOfSamplesToRecord;
disp('The experiment is beginning... get ready')

%% Record First Rest
fopen(s);
disp('rest your hand for 3 seconds')
tic
fprintf(s,'S')
rest1 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D')
time = toc;
disp('Finished in (s):')
disp(time)
%% Record first gesture
disp('rock for 3 seconds')
tic
fprintf(s,'S');
gesture1 = fread(s,numOfSamplesToRecord,'uint8');
```

```
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Second Rest
disp('rest hand 3 seconds')
tic
fprintf(s,'S');
rest2 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Second gesture
disp('paper 3 seconds')
tic
fprintf(s,'S');
gesture2 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Third Rest
disp('rest hand 3 seconds')
tic
fprintf(s,'S');
rest3 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Third gesture
disp('supination (medial) for 3 seconds')
tic
fprintf(s,'S');
gesture3 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Fourth Rest
disp('rest hand 3 seconds')
tic
fprintf(s,'S');
rest4 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
disp('Finished in (s):')
disp(time)
%% Record Fourth gesture
disp('pronation (lateral) for 3 seconds')
tic
fprintf(s,'S');
gesture4 = fread(s,numOfSamplesToRecord,'uint8');
fprintf(s,'D');
time = toc;
fclose(s);
disp('Finished in (s):')
disp(time)
disp('Thank you, experiment finished.')

%% Create gesture matrices
Rest1 = zeros(numOfSamplesToRecord/4,5);
Gesture1 = zeros(numOfSamplesToRecord/4,5);
Rest2 = zeros(numOfSamplesToRecord/4,5);
Gesture2 = zeros(numOfSamplesToRecord/4,5);
Rest3 = zeros(numOfSamplesToRecord/4,5);
Gesture3 = zeros(numOfSamplesToRecord/4,5);
Rest4 = zeros(numOfSamplesToRecord/4,5);
Gesture4 = zeros(numOfSamplesToRecord/4,5);


%% 1 Sort first rest
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Rest1(n, 1) = rest1(i, 1);
Rest1(n, 2) = rest1(i+1, 1);
Rest1(n, 3) = rest1(i+2, 1);
Rest1(n, 4) = rest1(i+3, 1);
Rest1(n, 5) = 0;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Rest1(:,1))
ylim([0 300])
title('Rest 1 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Rest1(:,2))
ylim([0 300])
title('Rest 1 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Rest1(:,3))
ylim([0 300])
title('Rest 1 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Rest1(:,4))
ylim([0 300])
title('Rest 1 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Rest1(:,5))
ylim([0 5])
title('Rest 1 label')
xlabel('samples')
ylabel('label')

%% 2 Sort first gesture
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Gesture1(n, 1) = gesture1(i, 1);
```

```
Gesture1(n, 2) = gesture1(i+1, 1);
Gesture1(n, 3) = gesture1(i+2, 1);
Gesture1(n, 4) = gesture1(i+3, 1);
Gesture1(n, 5) = 1;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Gesture1(:,1))
ylim([0 300])
title('Gesture 1 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Gesture1(:,2))
ylim([0 300])
title('Gesture 1 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Gesture1(:,3))
ylim([0 300])
title('Gesture 1 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Gesture1(:,4))
ylim([0 300])
title('Gesture 1 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Gesture1(:,5))
ylim([0 5])
title('Gesture 1 label')
xlabel('samples')
ylabel('label')
%% 3 Sort second rest
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Rest2(n, 1) = rest2(i, 1);
Rest2(n, 2) = rest2(i+1, 1);
Rest2(n, 3) = rest2(i+2, 1);
Rest2(n, 4) = rest2(i+3, 1);
Rest2(n, 5) = 0;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Rest2(:,1))
ylim([0 300])
title('Rest 2 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Rest2(:,2))
ylim([0 300])
title('Rest 2 Ch.2')
xlabel('samples')
```

```
ylabel('magnitude')
subplot(5,1,3)
plot(Rest2(:,3))
ylim([0 300])
title('Rest 2 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Rest2(:,4))
ylim([0 300])
title('Rest 2 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Rest2(:,5))
ylim([0 5])
title('Rest 2 label')
xlabel('samples')
ylabel('label')
%% 4 Sort second gesture
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Gesture2(n, 1) = gesture2(i, 1);
Gesture2(n, 2) = gesture2(i+1, 1);
Gesture2(n, 3) = gesture2(i+2, 1);
Gesture2(n, 4) = gesture2(i+3, 1);
Gesture2(n, 5) = 2;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Gesture2(:,1))
ylim([0 300])
title('Gesture 2 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Gesture2(:,2))
ylim([0 300])
title('Gesture 2 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Gesture2(:,3))
ylim([0 300])
title('Gesture 2 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Gesture2(:,4))
ylim([0 300])
title('Gesture 2 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Gesture2(:,5))
ylim([0 5])
title('Gesture 2 label')
xlabel('samples')
```

```
ylabel('label')
%% 5 Sort third rest
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Rest3(n, 1) = rest3(i, 1);
Rest3(n, 2) = rest3(i+1, 1);
Rest3(n, 3) = rest3(i+2, 1);
Rest3(n, 4) = rest3(i+3, 1);
Rest3(n, 5) = 0;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Rest3(:,1))
ylim([0 300])
title('Rest 3 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Rest3(:,2))
ylim([0 300])
title('Rest 3 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Rest3(:,3))
ylim([0 300])
title('Rest 3 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Rest3(:,4))
ylim([0 300])
title('Rest 3 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Rest3(:,5))
ylim([0 5])
title('Rest 3 label')
xlabel('samples')
ylabel('label')
%% 6 Sort third gesture
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Gesture3(n, 1) = gesture3(i, 1);
Gesture3(n, 2) = gesture3(i+1, 1);
Gesture3(n, 3) = gesture3(i+2, 1);
Gesture3(n, 4) = gesture3(i+3, 1);
Gesture3(n, 5) = 3;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Gesture3(:,1))
ylim([0 300])
title('Gesture 3 Ch.1')
xlabel('samples')
ylabel('magnitude')
```

```
subplot(5,1,2)
plot(Gesture3(:,2))
ylim([0 300])
title('Gesture 3 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Gesture3(:,3))
ylim([0 300])
title('Gesture 3 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Gesture3(:,4))
ylim([0 300])
title('Gesture 3 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Gesture3(:,5))
ylim([0 5])
title('Gesture 3 label')
xlabel('samples')
ylabel('label')
%% 7 Sort fourth rest
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Rest4(n, 1) = rest4(i, 1);
Rest4(n, 2) = rest4(i+1, 1);
Rest4(n, 3) = rest4(i+2, 1);
Rest4(n, 4) = rest4(i+3, 1);
Rest4(n, 5) = 0;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Rest4(:,1))
ylim([0 300])
title('Rest 4 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Rest4(:,2))
ylim([0 300])
title('Rest 4 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Rest4(:,3))
ylim([0 300])
title('Rest 4 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Rest4(:,4))
ylim([0 300])
title('Rest 4 Ch.4')
xlabel('samples')
ylabel('magnitude')
```

```
subplot(5,1,5)
plot(Rest4(:,5))
ylim([0 5])
title('Rest 4 label')
xlabel('samples')
ylabel('label')
%% 8 Sort fourth gesture
n = 1;
for i = 1:4:(numOfSamplesToRecord - 4)
Gesture4(n, 1) = gesture4(i, 1);
Gesture4(n, 2) = gesture4(i+1, 1);
Gesture4(n, 3) = gesture4(i+2, 1);
Gesture4(n, 4) = gesture4(i+3, 1);
Gesture4(n, 5) = 4;
n = n + 1;
end
figure
subplot(5,1,1)
plot(Gesture4(:,1))
ylim([0 300])
title('Gesture 4 Ch.1')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,2)
plot(Gesture4(:,2))
ylim([0 300])
title('Gesture 4 Ch.2')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,3)
plot(Gesture4(:,3))
ylim([0 300])
title('Gesture 4 Ch.3')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,4)
plot(Gesture4(:,4))
ylim([0 300])
title('Gesture 4 Ch.4')
xlabel('samples')
ylabel('magnitude')
subplot(5,1,5)
plot(Gesture4(:,5))
ylim([0 5])
title('Gesture 4 label')
xlabel('samples')
ylabel('label')

%% exporting the data
allData = [Rest1
    Gesture1
    Rest2
    Gesture2
    Rest3
    Gesture3
    Rest4
    Gesture4];

save("record30.mat", 'allData')
```

MSP code:

```
#include "msp.h"
#include <stdint.h> /* library for exact-width integers */
#include <stdio.h> /* library to manipulate integers */

int result1;
int result2;
int result3;
int result4;

int main(void)
{
WDT_A->CTL = WDT_A_CTL_PW | // Stop watchdog timer
WDT_A_CTL_HOLD;

// Configure ports as inputs to ADC
P5->SEL1 |= BIT5 | BIT4 | BIT2 | BIT1; // Enable A/D channel
P5->SEL0 |= BIT5 | BIT4 | BIT2 | BIT1;
//configuring p5.6 and p5.7 as voltage reference for adc
P5->SEL1 |= BIT7 | BIT6;
P5->SEL0 |= BIT7 | BIT6;

// Enable global interrupt
__enable_irq();

NVIC->ISER[0] = 1 << ((ADC14_IRQn) & 31);// Enable ADC interrupt in NVIC module

// Turn on ADC14, extend sampling time to avoid overflow of results
ADC14->CTL0 = ADC14_CTL0_ON |
ADC14_CTL0_MSC |
ADC14_CTL0_SHT0__192 |
ADC14_CTL0_SHP |
ADC14_CTL0_CONSEQ_3|
ADC14_CTL0_SSEL_4;

//setting reference voltage for every adc channel
ADC14 -> MCTL[0] = ADC14_MCTLN_VRSEL_14;
ADC14 -> MCTL[1] = ADC14_MCTLN_VRSEL_14;
ADC14 -> MCTL[2] = ADC14_MCTLN_VRSEL_14;
ADC14 -> MCTL[3] = ADC14_MCTLN_VRSEL_14;

//ADC resolution = 8 bit
ADC14 -> CTL1 = ADC14_CTL1_RES_0;

ADC14->MCTL[0] = ADC14_MCTLN_INCH_0; // ref+=AVcc, channel = A0
ADC14->MCTL[1] = ADC14_MCTLN_INCH_1; // ref+=AVcc, channel = A1
ADC14->MCTL[2] = ADC14_MCTLN_INCH_3;
ADC14->MCTL[3] = ADC14_MCTLN_INCH_4 | ADC14_MCTLN_EOS;

ADC14->IER0 = ADC14_IER0_IE2; // Enable ADC14IFG.2

//configure clocks
CS->KEY = CS_KEY_VAL; // Unlock CS module for register access
CS->CTL0 = 0; // Reset tuning parameters
CS->CTL0 = CS_CTL0_DCORSEL_3; // Set DCO to 12MHz (nominal, center of 8-16MHz
range)
CS->CTL1 = CS_CTL1_SELA_2 | // Select ACLK = REFO
CS_CTL1_SELS_3 | // SMCLK = DCO
CS_CTL1_SELM_3; // MCLK = DCO
CS->KEY = 0; // Lock CS module from unintended accesses

// Configure UART pins
P1->SEL0 |= BIT2 | BIT3; // set 2-UART pin as secondary function

EUSCI_A0->CTLW0 |= EUSCI_A_CTLW0_SWRST; // Put eUSCI in reset
EUSCI_A0->CTLW0 = EUSCI_A_CTLW0_SWRST | // Remain eUSCI in reset
EUSCI_B_CTLW0_SSEL__SMCLK; // Configure eUSCI clock source for SMCLK
EUSCI_A0->BRW = 1;
EUSCI_A0->MCTLW = (10 << EUSCI_A_MCTLW_BRF_OFS) |
(0x0 << EUSCI_A_MCTLW_BRS_OFS)|
EUSCI_A_MCTLW_OS16;
EUSCI_A0->CTLW0 &= ~EUSCI_A_CTLW0_SWRST; // Initialize eUSC
//UCA0MCTLW = 6;

SCB->SCR &= ~SCB_SCR_SLEEPONEXIT_Msk; // Wake up on exit from ISR

// Ensures SLEEPONEXIT takes effect immediately
__DSB();

while(1)
{
if(EUSCI_A0->RXBUF == 'S'){
// Start conversion when char S is received
ADC14->CTL0 |= ADC14_CTL0_ENC |
ADC14_CTL0_SC;
__sleep();
__no_operation();
}
if(EUSCI_A0->RXBUF == 'D'){
//Stop conversion when char D is received
ADC14->CTL0 &= ~ADC14_CTL0_ENC;
ADC14->CTL0 &= ~ADC14_CTL0_SC;
}
}
}

// ADC14 interrupt service routine
void ADC14_IRQHandler(void)
{
if (ADC14->IFGR0 & ADC14_IFGR0_IFG2)
{
result1 = ADC14->MEM[0];
result2 = ADC14->MEM[1];
result3 = ADC14->MEM[2];
result4 = ADC14->MEM[3];
while(!(EUSCI_A0->IFG&0x02)) { } /* wait for transmit buffer empty */
EUSCI_A0->TXBUF = result1;
while(!(EUSCI_A0->IFG&0x02)) { }
EUSCI_A0->TXBUF = result2;
while(!(EUSCI_A0->IFG&0x02)) { }
EUSCI_A0->TXBUF = result3;
while(!(EUSCI_A0->IFG&0x02)) { }
EUSCI_A0->TXBUF = result4;
}
}
```

13.4 Feature extraction and data classification

```
clc
clear
resultsPerTrial = 8;
trials = 30;

RMS1 = zeros(resultsPerTrial*trials,5);
RMS2 = zeros(resultsPerTrial*trials,5);
RMS3 = zeros(resultsPerTrial*trials,5);
RMS4 = zeros(resultsPerTrial*trials,5);


for t = 1:30;
dir = ['C:\Users\amede\Desktop\bespoke\record' num2str(t) '.mat'];
values{t} = importdata(dir);


[rows,columns]=size(values{t});

G1 = 0;
G2 = 0;
G3 = 0;
G4 = 0;

for i=1:rows
x = values{t}(i,columns);

if x == 1
G1 = G1 + 1;
for j = 1:columns
gesture1(G1,j) = values{t}(i,j);
end
end

if x == 2
G2 = G2 + 1;
for j = 1:columns
gesture2(G2,j) = values{t}(i,j);
end
end

if x == 3
G3 = G3 + 1;
for j = 1:columns
gesture3(G3,j) = values{t}(i,j);
end
end

if x == 4
G4 = G4 + 1;
for j = 1:columns
gesture4(G4,j) = values{t}(i,j);
end
end

end
for i = 1:resultsPerTrial
    %start = 10000 + ((i-1)*2000);
    %stop = 14000 + ((i-1)*2000);
    start = 10000 + ((i-1)*5760);
    stop = 15760 + ((i-1)*5760);
    index = i + ((t-1)*resultsPerTrial);

    RMS1(index,1) = rms(gesture1(start:stop,1));
    RMS1(index,2) = rms(gesture1(start:stop,2));
    RMS1(index,3) = rms(gesture1(start:stop,3));
    RMS1(index,4) = rms(gesture1(start:stop,4));
    RMS1(index,5) = 1;


    RMS2(index,1) = rms(gesture2(start:stop,1));
    RMS2(index,2) = rms(gesture2(start:stop,2));
    RMS2(index,3) = rms(gesture2(start:stop,3));
    RMS2(index,4) = rms(gesture2(start:stop,4));
    RMS2(index,5) = 2;


    RMS3(index,1) = rms(gesture3(start:stop,1));
    RMS3(index,2) = rms(gesture3(start:stop,2));
    RMS3(index,3) = rms(gesture3(start:stop,3));
    RMS3(index,4) = rms(gesture3(start:stop,4));
    RMS3(index,5) = 3;


    RMS4(index,1) = rms(gesture4(start:stop,1));
    RMS4(index,2) = rms(gesture4(start:stop,2));
    RMS4(index,3) = rms(gesture4(start:stop,3));
    RMS4(index,4) = rms(gesture4(start:stop,4));
    RMS4(index,5) = 4;

end
% subplot(5,1,1)
% plot(RMS4(:,1))
%
% subplot(5,1,2)
% plot(RMS4(:,2))
%
% subplot(5,1,3)
% plot(RMS4(:,3))
%
% subplot(5,1,4)
% plot(RMS4(:,4))

end

RMSfinal = [RMS1
    RMS2
    RMS3
    RMS4];

%% - plot
% subplot(5,1,1)
% plot(RMSfinal(:,1))
%
% subplot(5,1,2)
% plot(RMSfinal(:,2))
%
```

```
% subplot(5,1,3)
% plot(RMSfinal(:,3))
%
% subplot(5,1,4)
% plot(RMSfinal(:,4))
%
% subplot(5,1,5)
% plot(RMSfinal(:,5))
```

```
import numpy as np
import pandas as pd

# Now load in the dataset with Pandas
from matplotlib import pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay

df = pd.read_csv('rmsDataBaseline.csv')

# Print out part of the dataset to ensure proper loading
y = df['class']
x = df[['ch1', 'ch2', 'ch3', 'ch4']]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1,
random_state=31)

from sklearn.preprocessing import StandardScaler
# Create an instance of the scaler and apply it to the data
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)


from sklearn import svm

classifier = svm.SVC(kernel='linear', C=1, random_state=200)

classifier.fit(X_train, y_train)

print(classifier.score(X_test,y_test))


np.set_printoptions(precision=2)

# Plot non-normalized confusion matrix
titles_options = [
    ("Confusion matrix, without normalization", None),
    ("Normalized confusion matrix", "true"),
]
for title, normalize in titles_options:
    disp = ConfusionMatrixDisplay.from_estimator(
        classifier,
        X_test,
        y_test,
        display_labels=['Rock', 'Paper', 'Supination', 'Pronation'],
        cmap=plt.cm.Blues,
        normalize=normalize,
    )
    disp.ax_.set_title(title)

    print(title)
    print(disp.confusion_matrix)

plt.show()
```