
**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**A Major Project Report**

**On**

# Human-Computer Interaction using Neuromuscular Signals

**Submitted By:**

Rabin Nepal (073/BEX/331)

Rhimesh Lwagun (073/BEX/333)

Sanjay Rijal (073/BEX/342)

Upendra Subedi (073/BEX/347)

**Submitted To:**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

KATHMANDU, NEPAL

October, 2020

**ACKNOWLEDGEMENT**

We would like to use this opportunity to express our deepest gratitude to our project supervisor, Mr. Dinesh Baniya Kshatri, for all the guidelines and motivation he has provided. We would also like to expand our gratitude to our lecturers and classmates who have helped us directly and indirectly.

Finally, we are very thankful to the Department of Electronics and Computer Engineering, IOE Thapathali Campus for providing us this opportunity to conduct this project.

Rabin Nepal (073/BEX/331)
Rhimesh Lwagun (073/BEX/333)
Sanjay Rijal (073/BEX/342)
Upendra Subedi (073/BEX/347)

**ABSTRACT**

Subvocal speech or internal articulation is a form of non-voiced speech that is voluntarily spoken. It is generated alongside the micromovement of the articulatory muscles that is imperceptible to others. However, the faint sEMG (surface Electromyography) signals can still be detected and analyzed to predict the internally articulated speech. This research project attempts to study this very phenomenon and its possible use case in human computer interaction. By extracting sEMG signals from several of these articulators and processing the extracted signals, prominent features of a particular utterance can be isolated and can be used to train a machine learning model. After training the model on several such utterances, accurate predictions of the utterances can be made which can be further utilized to perform a predefined action on a remote computer. This research also explores on improving traditional speech recognition models by possible augmentation of both approaches.

*Keywords: Human computer interaction, Internal articulation, Speech recognition, Surface electromyography*

## TABLE OF CONTENTS

**List of Tables**

**List of Figures**

**List of Abbreviations**

| | |
|---|---|
| ADC | Analog to Digital Converter |
| ALS | Amyotrophic Lateral Sclerosis |
| ANN | Artificial Neural Network |
| AR | Auto regression |
| BCI | Brain Computer Interaction |
| CMRR | Common Mode Rejection Ratio |
| CN | Cranial Nerve |
| CNN | Convolutional Neural Network |
| CTC | Connectionist Temporal Classification |
| DAC | Digital to Analog Converter |
| DFT | Discrete Fourier Transform |
| DTCWT | Dual Tree Complex Wavelet Transform |
| ECG | Electrocardiography |
| ECoG | Electrocorticography |
| EEG | Electroencephalography |
| EMG | Electromyography |
| EMS | Electrical Muscle Stimulation |
| FFT | Fast Fourier Transform |
| FIR | Finite Impulse Response |
| GBP | Gain-Bandwidth Product |
| GMM | Gaussian Mixture Model |
| HPF | High Pass Filter |
| IDE | Integrated Development Environment |
| IIR | Infinite Impulse Response |
| LPF | Low Pass Filter |
| LSB | Lower Significant Bit |
| MFCC | Mel Frequency Cepstral Coefficients |
| MLP | Multi-Layer Perceptron |
| MUAP | Motor Unit Action Potentials |
| NEMS | Neuromuscular electrical stimulation |
| PCA | Principle Component Analysis |

| | |
|---|---|
| PCB | Printed Circuit Board |
| PNS | Peripheral Nervous System |
| PWM | Pulse Width Modulation |
| RCA | Radio Corporation of America |
| RNN | Recurrent Neural Networks |
| SAR | Successive Approximation Register |
| SMD | Surface Mount Devices |
| SMT | Surface Mount Technology |
| SNR | Signal to Noise Ratio |
| STFT | Short-Time Fourier Transform |
| UART | Universal Asynchronous Receiver/Transmitter |
| WER | Word Error Rate |
| ZCR | Zero Crossing Rate |

# 1. INTRODUCTION

Communication is an integral part of humans. Communication enables humans to exchange information and messages. There are different means of communication but the very intriguing and primitive form of communication is speech.

## 1.1 Background

Speech is human vocal communication, where information is exchanged with different sounds produced. This is a very simple method and is implemented by many of the modern gadgets taking it as an input from the user. This kind of interaction between a human and a computing device has existed for decades now but such interface has just recently been reliable and fast enough to be used in real world applications. The more traditional approach for human computer interface is peripheral devices like mouse, keyboard, touchpad, stylus, joysticks, etc. Despite the advancement in verbal human computer interface, such traditional approaches are yet to be replaced.

Nevertheless, the advancement of neuroscience has introduced new methods of interacting with a computer. These kinds of interfaces are termed as brain computer interface. The input signal for the computer in such cases are extracted from the brain through process such as EEG. Operating a brain-controlled device then becomes as simple as thinking about controlling the device. This interaction method, however, does come with some major hindrances. The user must train themselves to concentrate on doing a specific task and not wander along with thoughts. There is also an ethical dilemma associated with such interface which questions if peeking inside someone's brain and their thoughts should be allowed. Another major drawback of such interface is the large number of input channels that needs extensive technical analysis to interpret any useful information.

An intermediate method between voiced interface and brain-controlled interface could be "Silent Speech Interface". Unlike voiced interface, this method is completely silent and unlike brain computer interface, this method is voluntary; a user has to deliberately

convey information through unvoiced form of speech. Silent Speech Interface is more seamless than voiced interface and requires no special user training at all. It utilizes the sEMG signals extracted from articulatory muscles (during voluntary speech production) using special electrodes and uses these signals to predict what the user is speaking internally. Besides sEMG there also exist many other techniques for studying internal speech articulation. They can be listed as follows:

- Electroencephalography (EEG)
- Electrocorticography (ECoG)
- Permanent Magnet Sensors
- Ultrasound
- Optical Camera
- Vocal Tract Resonant Signals

### 1.2 Motivation

There were some amazing researches and projects which are similar to the project. The idea of this project is motivated mostly by ‘Backyard Brains’ and ‘AlterEgo’ which are described as follows:

Backyard Brain is a team of engineers, scientists and researchers that have been constantly researching on inner working of nervous system. And have designed some interface and hardware with cheap electronics that can help students in providing insight into the working of the nervous system. The equipment for conducting such experiments costs much but they aim to provide students below graduation to do experiments on nervous system maintaining ethics regarding experimentation on living beings. [1]

AlterEgo is a personalized wearable silent speech interface that allows its users to silently converse with a computing device without any voice or any discernible movements - thereby enabling the user to communicate with devices, AI assistants, applications or other people in a silent, concealed and seamless manner. A user's intention to speak and internal speech is characterized by neuromuscular signals in internal speech articulators that are captured by the AlterEgo system to reconstruct this speech. This interface is used

to facilitate a natural language user interface, where users can silently communicate in natural language and receive aural output (e.g.: - bone conduction headphones), thereby enabling a discreet, bi-directional interface with a computing device, and providing a seamless form of intelligence augmentation. [2]

This project is selected as it has interfacing of brain signals with computer. Brain computer interface (BCI) is new field for research and product development. In this context, this project provides platforms for research on this field. It is also intended to help on one of the evolving fields of science and technology i.e. bio-technology.

### 1.3 Problem Definition

Since the development of the very first computer, human-computer interaction has always required to have some form of physical action as an input to the computer. These traditional input devices include keyboards, mouse, joystick, cameras, microphones etc. Although these methods possess high accuracy and convenience, they suffer from lack of privacy. And it may not be possible for everyone to use such traditional means for interacting with a computing device. The traditional system has also been proved to be very slow in today's world. Speech interaction somewhat tackles this issue but is still subjected to privacy problems. The proposed system in this project tackles all these problems and provides a secure and faster interaction between a human and a computing device. And the proposed system works the same for everyone disregarding their disabilities.

### 1.4 Project Objectives

The main objectives of the project are:

- To extract and transfer EMG signals from articulatory muscles to a computer
- To process and convert the articulated speech signals to text

### 1.5 Project Scope and Applications

This project explores EMG signals generated in the speech articulatory muscles as a foundational element to human speech and tries to achieve a simple human computer

interaction through it. This project does not include the decoding of signals produced during articulation of sentences and special characters. It is not compatible for languages other than English. The interface is unidirectional and does not include the control of any devices.

This project carries with it an extensive potential in regards to both aid to human potential and apprehension of the speech processes in humans. It possesses the ability to directly help speech impaired people and improve upon existing communication methods. Complex process of speech production can be simplified and studied further to understand hidden patterns and thus refine existing speech recognition models.

The real-life applications of this research project can further be illustrated using the following points:

### 1.5.1 Silent Means of Communication

Internal articulation is inaudible mode of speaking that is only perceptible to the speaker themselves. Lack of any perceptible sound makes it perfect for any cohort applications where privacy is crucial. Similarly, this means of communication is also suitable in situations where voiced speech is not appropriate such as in libraries, meetings, etc.

### 1.5.2 Novel Human Computer Interface

Traditional human computer interface takes place through peripheral devices like mouse, keyboards, touchpads, etc. These traditional interfaces are often slow and inconvenient to people of all backgrounds, especially for people with disabilities. There exist voiced interfaces for differently abled peoples but those with speaking disabilities cannot utilize even this interface. Such is the case for people suffering from ALS (Amyotrophic Lateral Sclerosis) and other articulation disorders. Exploiting sEMG signals from speech articulators as an input to computing devices can help people suffering from such disabilities. In addition to this, controlling robotics limbs, IOT and other remote devices truly help it make a novel means of human computer interface.

### 1.5.3 Improvement of Speech Recognition Models

Current Speech Recognition models require huge amounts of data and tremendous computing power to obtain usable accuracy in recognition tasks. These models still cannot generalize well to different scenarios such as background noises, difference in speaking rate, difference in pronunciation, etc. sEMG signals from targeted articulatory muscles could provide new perspective to Speech Recognition tasks. These signals along with acoustic speech data could help improve the existing Speech Recognition models.

## 1.6 Report Organization

The material presented at the report is organized into eleven chapters. Chapter 1 is an introduction section which mainly describes the background, objective, scope and application of the project. It also focuses on the need of the project. Chapter 2 presents brief summaries of all existing works that have already been carried out in the related field. It describes the activities related to a project that have been carried by some specific people. Chapter 3 provides the information on generation and working of neuromuscular signals, explains the role of these signals in controlling the facial muscles during speech and how the internal articulation works along with the muscles involved. Chapter 4 illustrates the mathematical models required in the project. Chapter 5 explains about the effect of noise and the types of noise that the designed system is susceptible to. Chapter 6 provides an account of the hardware and software requirements of the project. This chapter also summarizes the implementation process of the used hardware and software. Chapter 7 explains in detail a particular sequence in which the work has been carried out along with detailed procedures, block diagram or data flow diagram which illustrate the explanation of how the hardware and software are used to accomplish the project. Chapter 8 also contains the details of the implementation of the things that have been explained in the methodology. In short, it describes how the methodology is implemented. Chapter 9 contains the result of the project. In other words, it shows the total progress done on the project till date. The output is shown in graphical form as well. Chapter 10 signifies the problems faced during the course of design of the system in the project. It also briefly states the analysis of the result. Chapter 11 gives the information about the tasks in the project that are still remaining. Chapter 12 contains the additional topics like cost estimate, project schedule. Chapter 13 includes the references used for the project.

## 2. LITERATURE REVIEW

There has been a number of attempts in electrophysiology for analysis of neural activities. “AlterEgo: A Personalized Wearable Silent Speech Interface”, a research accomplished by Arnav Kapur, Shreyas Kapur and Pattie Maes which was published by MIT Media Lab in 2018, presents a natural extension of the user's own cognition by enabling a silent, discreet and seamless conversation with machines and people. It presents a wearable silent speech interface that allows users to provide arbitrary text input to a computing device or other people using natural language, without discernible muscle movements and without any voice command i.e. without explicitly saying anything. The nerve impulses were sourced as seven channels from laryngeal region, hyoid region, levator anguli oris, orbicularis oris, platysma, anterior belly of the digastric mentum using electrodes on the outer skin [3].

According to neuroprosthetics experiment, “Control Machines with your Brain” done from 2009-2017 by Backyardbrains, a team of researchers and engineers, the EMG signals were extracted from Muscle SpikerShield which was interfaced with micro-controller and the data obtained was visualized in Spike Recorder App developed by the team. The research was further extended as Human-Machine Interfaces, which included control of robotic arm, video games and voiceless communication. [4]

A research paper by Michael Wand and Tanja Schultz named “SESSION-INDEPENDENT EMG-BASED SPEECH RECOGNITION” describes the method of speech recognition by surface EMG signals. By recording the electric active potentials of human articulatory muscles, it can be decoded into a speech that person is vocalizing. Speech recognition using EMG signals dates back to 1980s. 93% accuracy was observed on 10-word vocabulary. It suggests that good result can be obtained even for the signals taken when words are silently articulated. [5]

“Electrical Stimulated as a Modality to Improve Performance of the Neuromuscular System”, a research paper by Vanderthommen Marc and Duchateaus Jacques in October 2007 transcutaneous neuromuscular electrical stimulation (NEMS) can modify the order of motor unit recruitment and has a profound influence on the metabolic demand

associated with producing a given muscular force. Tetanic contractions elicited by pulses of high intensity and short duration induce a high metabolic stress in the muscle, contribute to the reversal of motor unit recruitment, and improve the maximal capability of the neuromuscular system primarily not only through increased force-generating capacity of the muscle but also through intensified voluntary activation. [6]

A research paper “End-to-end neural networks for subvocal speech recognition” written by Pol Rosello, Pamela Toman and Nipun Agarwala attempts to perform session independent subvocal speech recognition by leveraging character-level recurrent neural networks (RNNs) and the connectionist temporal classification loss (CTC). They utilized EMG-UKA trial coprus’s two hours of data to train their CTC models. Although the accuracy of their model is not mentioned, they did express some measures to improve the field to silent speech recognition through EMG signals in their paper. [7]

Munna Khan and Mosarrat Jahan wrote a paper “The Application of AR Coefficients and Burg Method in Sub-vocal EMG Pattern Recognition” which showcases successful recognition of Hindi phonemes (Ka, Kha, Ga, and Gha) with accuracy of about 75.5% to 80%. They studied burg algorithm techniques for EMG spectral analysis and used reflection coefficients and AR coefficients as features of sub-vocal EMG signal to recognize the patterns of sub-vocal phonemes. They concluded that the pattern recognition in EMG signal using reflection coefficients and AR coefficients is highly efficient and can be used to develop a real time module. [8]

In “Development of sEMG sensors and algorithms for silent speech recognition”, a research paper published by Geoffrey S. Meltzner, James T. Heaton et al., a new system capable of recognizing silently mouthed words and phrases based completely on surface EMG signals has been described. They tested a system of sensors and algorithms during a series of subvocal speech experiments involving more than 1,200 phrases generated from a 2,200-word vocabulary and obtained 91.1% recognition rate i.e. word error rate (WER) of 8.9%. They prepared their dataset performing experiments on a total of 19 subjects (11 males and 8 females) ranging from 20-42 years in age with no speech or hearing disabilities. They had applied discrete-cosine Fourier transform in order to obtain coefficients of the signal for the training set. [9]

Chuck Jorgensen and Kim Binsted in 2005 published a paper “Web Browser Control Using EMG Based Sub Vocal Speech Recognition” which describes they had trained six subvocally pronounced control words, 10 digits, 17 vowel phonemes and 23 consonant phonemes using a scaled conjugate gradient neural network. They had recorded the surface EMG signals of frequency range 30-500 Hz from the larynx and sublingual areas below the jaw, filtered them, sampled them at 2000 Hz and transformed into features using a Kingsbury’s Dual Tree complex wavelet transform (DTCWT) and short time Fourier transform (STFT). They had also designed a notch filter to eliminate line noise at 60 Hz. They had obtained an average of 92% accuracy. Using the trained control words, they performed sub vocal web browsing. [10]

## 3. ELCTROPHYSIOLOGY OF SPEECH PRODUCTION

Movement of body parts of living beings, voluntary as well as involuntary, is fully coordinated and controlled by the brain. Brain performs this controlling and coordinating activity through electrical signals.

### 3.1 Neuromuscular Signals

Electrophysiology is a branch of neuroscience that studies the electrical properties of biological cells and tissues. It also includes the measurements of electric current or voltage changes in biological cells [11]. Common types of electrical bio-signals are: EEG (Electroencephalogram), ERG (Electroretinogram), EMG (Electromyography), EOG (Electrooculography) and EGG (Electrogastrogram).

Normally, in biological cells the concentration gradient of potassium is greater from inside towards outside of the cell membrane so there is a strong tendency of extra $K^+$ ions to diffuse outward through the membrane. Diffusion of $K^+$ ions outside the cell membrane creates electro-positivity outside the membrane and electronegativity inside the membrane due to negative ions that remain behind and do not diffuse outward with the potassium ions. Within a millisecond the potential difference between the inside and outside of the cell membrane, called the resting or diffusion membrane potential becomes high enough to resist the further $K^+$ ions diffusion. In normal human nerve fiber, the resting potential is about -90 mV. The membrane is said to be polarized at this stage. Membrane then suddenly becomes very permeable to the $Na^+$ ions allowing a large number of $Na^+$ ions to diffuse to the interior of the membrane. Again, the membrane potential rises high enough within milliseconds and blocks further diffusion of sodium ions inside the membrane. The normal resting potential, -90 mV is neutralized; this is known as depolarization of the membrane. In large nerve fibers excess of $Na^+$ ions diffusion inside cause membrane potential to overshoot beyond the neutral point which is known as action potential. Within a $1/10000^{th}$ of a second, sodium channels begin to close and the potassium channels open more than normal. There is continuous pumping of three $Na^+$ ions outside the membrane for each two $K^+$ ions pumped inside. Rapid

diffusion of $K^+$ ions re-establishes the normal resting potential. This process is called repolarization. [12]

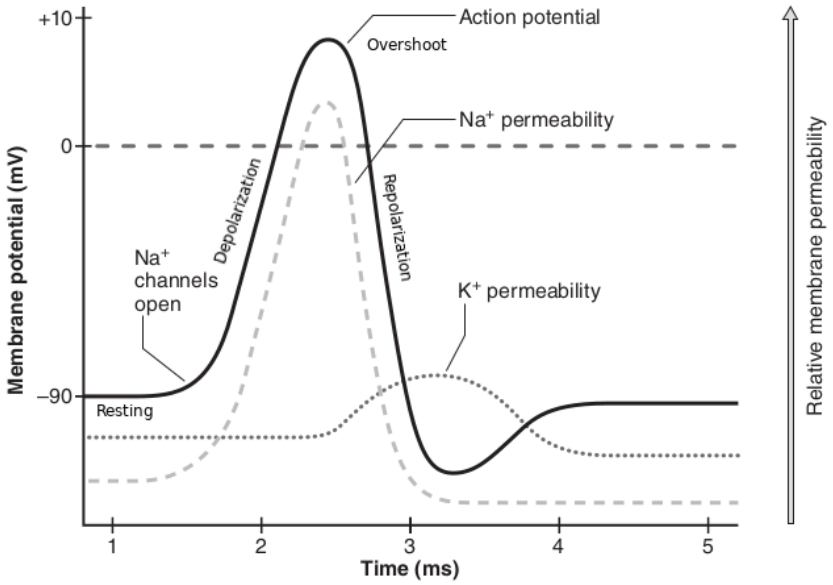



Figure 3-1: Time Course of the Muscle Fiber Action Potential [13]

At neuromuscular junctions, the nerve fibers form a complex of branching nerve terminals that invaginates into the vicinity of the muscle fiber. In the axon terminal are many mitochondria that supplies Adenosine Triphosphate (ATP) for the synthesis of an excitatory neurotransmitter, 'acetylcholine' which excites the muscle fiber membrane. The neuromuscular junction (skeletal muscle fiber) is as shown in figure below.

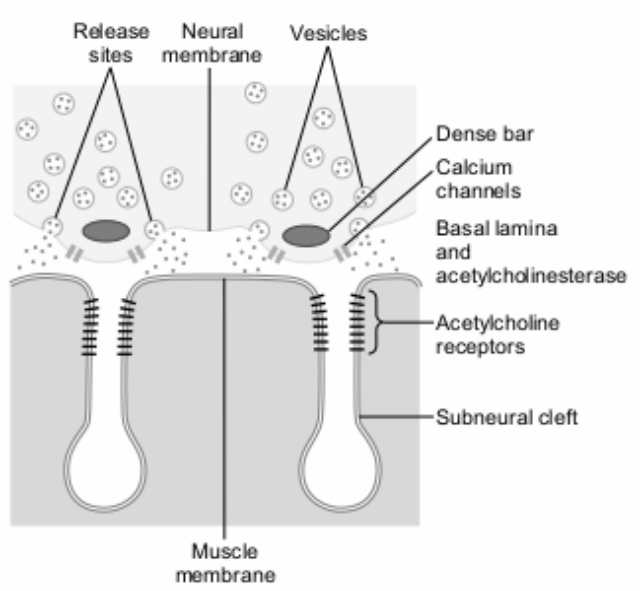



Figure 3-2: Neuromuscular Junction

When a nerve impulse reaches the neuromuscular junction, about 125 vesicles of acetylcholine are released from the terminals into the synaptic space and then continue to activate the acetylcholine receptors as long as it persists in the synaptic space, which is at most a few milliseconds. The elicitation of the acetylcholine receptors opens the acetylcholine gated channels at the muscle fiber membrane. The principal effect of opening these channels is to allow the large number of $Na^+$ ions to diffuse inside the fiber, carrying with them a large number of positive charges. This induces electrical potential inside the fiber at the local area of the end plate which is called the end plate potential (50-75 mV). The end plate potentials if not weakened by the toxins are strong enough to initiate the action potential as shown in figure 1-3. The end plate potential A and C are recorded from the muscles weakened by toxins; curare and botulinum respectively. The action potential then spreads along the muscle fiber membrane. After the generation of action potential, there is a brief refractory period during which membrane can't be stimulated, this prevents the message from being transmitted backward. [12]

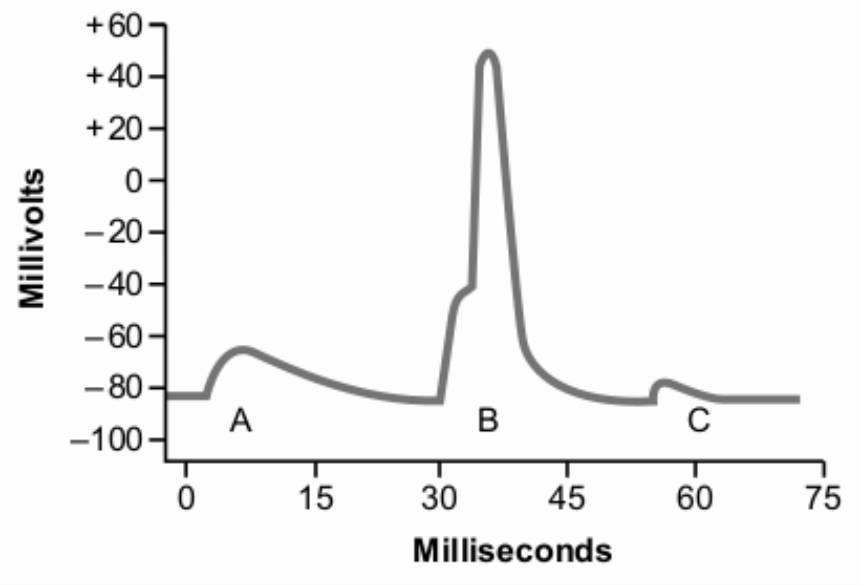



Figure 3-3: End Plate Potentials A, B and C with Action Potential at B

The skeletal muscle fiber is so large that the action potential spreading along its surface membrane causes almost no current flow in the fiber. Yet, to cause maximum muscle contraction, current must penetrate deeply into the muscle fiber to the vicinity of the separate myofibrils. This becomes possible due to transmission of action potential along the transverse tubules (T-tubules) that penetrate deep all the way through the muscle fiber.

The T-tubules action potential results in opening of the voltage gated calcium channels located at each side of the dense bar as shown in figure 1-2. The $Ca^{++}$ ions then diffuse from the synaptic space inside the muscle fiber in the immediate vicinity of the myofibrils and cause the muscle contraction. This overall process is called excitation-contraction coupling. The signal thus generated on the muscle fibers is known as Electromyography (EMG) signals which can be measured using electromyography. [12]

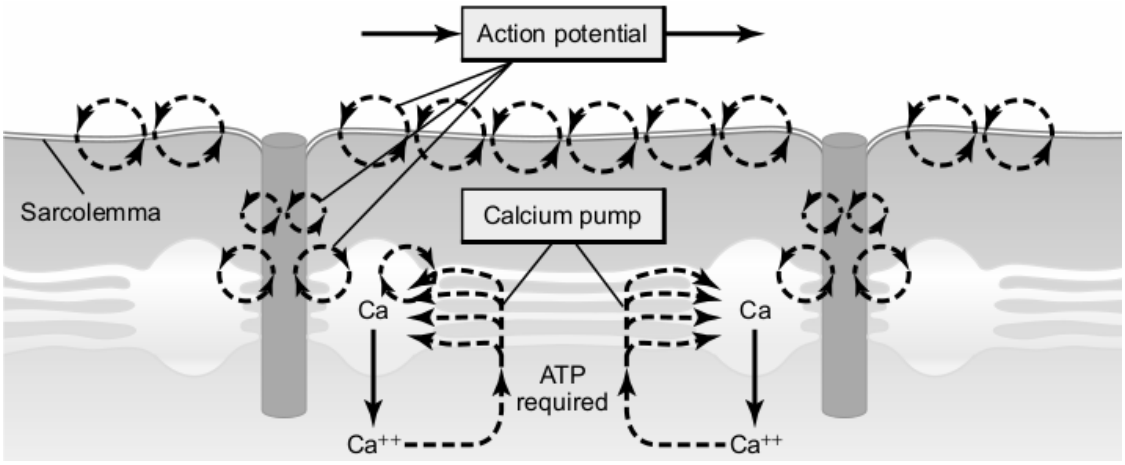



Figure 3-4: Excitation-Contraction Coupling in the Muscle [12]

The $Ca^{++}$ ions exert attractive force on the acetylcholine vesicles drawing them to the neural membrane. These vesicles then fuse with the neural membrane and empty their acetylcholine into the synaptic space by the process of exocytosis. During this process acetylcholine is rapidly removed by the acetylcholinesterase which decomposes acetylcholine into acetate and choline. The short time during which acetylcholine remains in the synaptic space (few milliseconds) is enough to excite the muscle fiber. The rapid removal of acetylcholine prevents continued muscle contraction. The number of vesicles present in the nerve endings is sufficient to allow transmission of only a few thousand nerve-to-muscle impulses. So, for continued transmission neuromuscular signals, new vesicles need to be reformed rapidly. Within a few seconds after each action potential is over, coated-pits appear in the terminal membrane caused by the contractile proteins in the nerve ending. Within a few seconds the protein contracts and causes the pits to break away into the interior of the membrane thus forming new vesicles. Extracellular fluids (ECF) reuptake choline from decomposed acetylcholine which then combines with the

free chlorine, acetyl coenzyme A, ATP and glucose to form acetylcholine in cytoplasm and then transferred to the vesicles within another few seconds. [14]

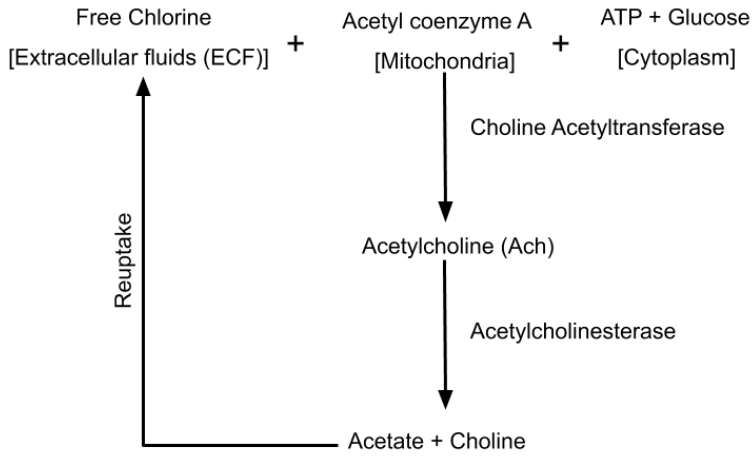



Figure 3-5: Formation and Decomposition of Acetylcholine (neurotransmitter)

For the analysis of superimposed motor unit action potentials (MUPAs) generated from several motor units, the EMG signals can be decomposed into their constituents MUPAs distinguished by their characteristic shapes. The shape and size depend on where the electrode is located with respect to the fibers and are different if the electrode position is altered slightly. [13]

### 3.2 Speech Production Mechanism

Human speech is a very complex process. It can be exemplified using a three-stage model of Conceptualization, Formulation and Articulation. The first two stages occur in the brain itself whereas the last stage occurs in the motor system under the control of the brain. Articulation again can be subdivided into three stages: Respiration, Phonation and Articulation.

#### 3.2.1 Respiration

Respiration, also known as breathing, refers to the inhalation and exhalation of air by the contraction and expansion of diaphragm. During inhalation, the lungs expand, causing the air to flow from the mouth to the lungs with the glottis relatively open. During exhalation, the lungs contract, pushing the air from the lungs toward the mouth, which provides energy for human speech. The energy is given as a stream of air coming from

the lungs which passes through trachea and the vocal fold as shown in figure 3-6, where the phonation occurs. For most languages, the production of sound occurs during exhalation which is why humans cannot generally speak while inhaling.

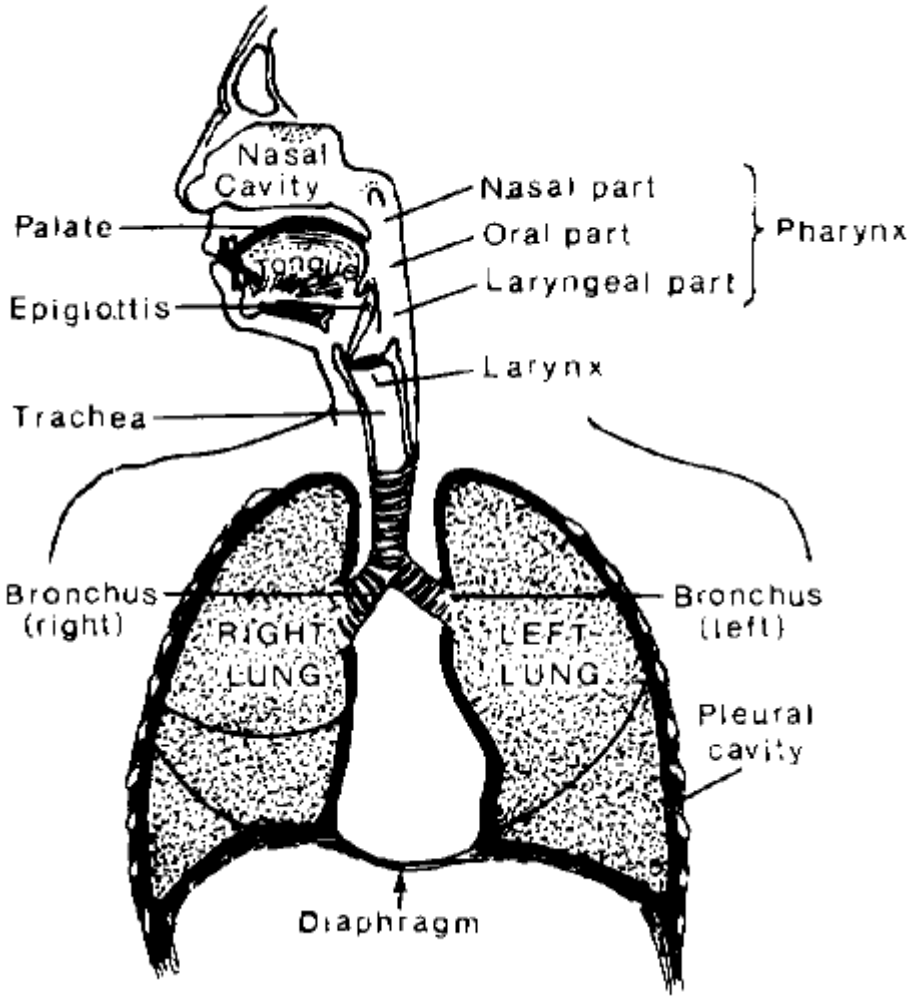



Figure 3-6: Vocal Apparatus in Humans

### 3.2.2 Phonation

Phonation is the process that modifies the pulmonic air in such a way that it produces acoustic signals. During phonation, the vocal folds vibrate causing a change in air pressure generating acoustic waves which then gets amplified internally through resonance. The voice source is also used to change the sentence melody and the tonal form of words by varying the subglottal pressure as well as the tension of the vocal folds.

This leads to changes in the rate of vibration of the vocal folds, which are in turn perceived by the listener as modifications in pitch and/or in loudness. [15]

### 3.2.3 Articulation

Articulation is the term used for all actions of the organs of the vocal tract that affect modifications of the signal generated by the voice source. This modification results in speech events which can be identified as vowels, consonants or other phonological units of a language [15]. The air stream is manipulated by several mobile organs called active articulators. The major active articulators are lower lip, tongue, glottis and uvula. These active articulators are supported by a number of passive articulators which are: palate, nasal cavity, epiglottis, lower teeth, alveolar ridge etc.

During the production of speech, a complex series of finite and coordinated neuromuscular communication is associated. For the complete generation of sound more than 100 muscles are involved. When a person articulates a word internally without acoustic vocalization and no significant movements in facial muscle and tongue, more than 15 muscles which are parts of the speech system, are neurologically activated. These particular muscles receive feeble electrical signals from the PNS. Nerves involved in the articulatory system and respective muscles movement are tabulated below.

Table 3-1: Nerves and Muscle Movements in the Articulatory System

| S.N. | Nerve | Movements | Sensory Functions |
|---|---|---|---|
| 1 | Trigeminal Nerve (CN V) | Biting and chewing | Sensory data from palate, teeth and anterior tongue |
| 2 | Facial Nerve (CN VII) | Facial muscle | Sensation to the external ear |
| 3 | Glossopharyngeal Nerve (CN IX) | Elevation of Pharynx and larynx | Sensation to posterior tongue and upper pharynx |
| 4 | Vagus Nerve (CN X) | Elevation of the palate phonation | Sensory data from external ear, tongue and larynx |
| 5 | Hypoglossal Nerve (CN XI) | Movement of the tongue | Sensory data from the tongue |

#### 3.2.3.1 Articulatory Muscles

Different facial muscles are involved in the activation of the active articulators for articulation. They are divided into different regions which are listed below:

- Labial region
- Lingual region
- Mandibular region
- Palatal region
- Pharyngeal region

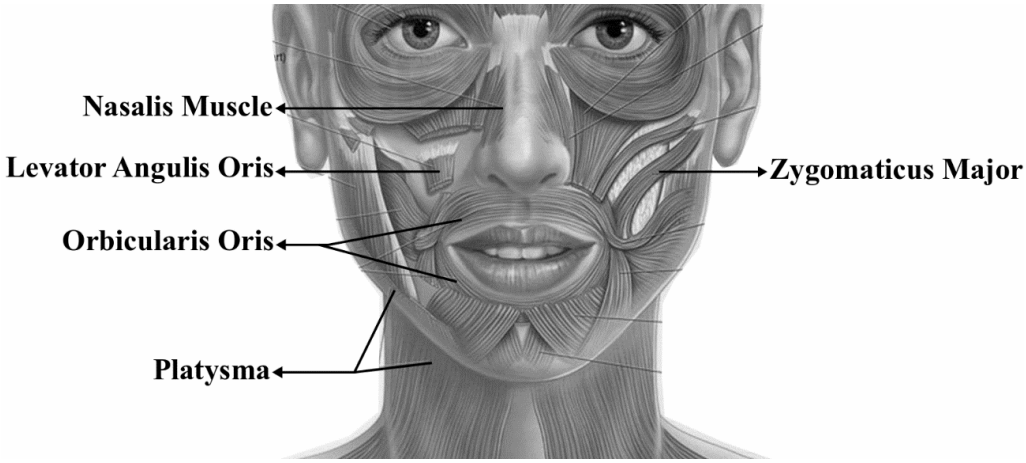



Figure 3-7: Active Facial Muscles

Selecting the right muscle for extracting the EMG signal is very essential. Many factors such as noise susceptibility, signal strength, cross-talk, signal frequency, etc. depend on the selection of the muscle. Some of the basic criteria to be followed while selecting the muscles are as follows:

- Select the muscle where it is convenient to place the electrodes.
- Distance from the electrode to the muscle should be minimum.
- Size of muscle should be large enough to avoid cross-talks.
- Configuration to be followed for signal extraction (bipolar or monopolar) also determines which muscle to select.
- Size of the electrode and the type of electrode should be considered before selecting the muscles.

Some of the major muscles fulfilling the above criteria are listed as below:

#### 3.2.3.1.1 Levator Anguli Oris

Levator Anguli Oris is a facial muscle close to the mouth opening that lifts the angle of the mouth. This muscle is innervated by the buccal branch of the facial nerve (CN VII). When activated, the levator anguli oris lifts the angle of the mouth, thus participating in creating a smile. Contractions of this muscle produce a facial expression associated with self-confidence.

#### 3.2.3.1.2 Zygomaticus Major

The zygomaticus major muscle is a paired facial muscle that extends between the zygomatic bone and the corner of the mouth. It is one of the two zygomatic muscles (major and minor) that lie next to each other in the cheek area. An activated zygomaticus major muscle is involved in creating an expression in the human face known as a smile. The nerve supply of the zygomaticus major is received from the zygomatic and buccal branches of the facial nerve (CN VII).

#### 3.2.3.1.3 Platysma

The platysma (also platysma muscle, Latin: platysma) is a wide, flat, superficial neck muscle extending from the lower part of the face to the upper thorax. The platysma is a paired thin and superficial muscle arising from the upper parts of the shoulders and inserting into the mouth area. Contractions of the platysma depress and wrinkle skin of the lower face and the mouth. The platysma also contributes to forced depression of the mandible. The platysma is innervated by the cervical branch of the facial nerve (CN VII).

#### 3.2.3.1.4 Acnterior Belly of Digastric

The anterior belly of the digastric (Latin: venter anterior musculo digastrico) is one of the two bellies of the digastric muscle. The anterior belly is smaller than the posterior belly, and it develops from the first pharyngeal arch. Upon contraction the anterior belly of the digastric muscle elevates the hyoid bone. The anterior belly of the digastric is innervated by the mylohyoid nerve, which arises from the mandibular division of the trigeminal nerve (CN V3).

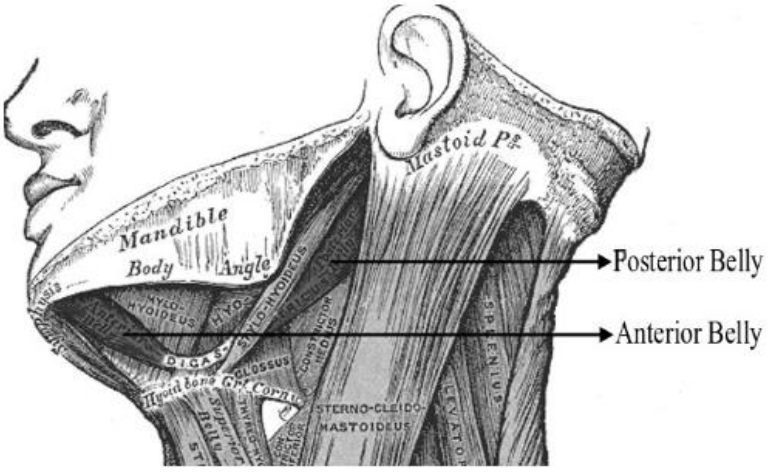



Figure 3-8: Anterior and Posterior Belly of Digastric

### 3.3 Internal Articulation

Internal articulation is a significantly attenuated form of speech that does not engage the vocal folds or produce any acoustic output and is indiscernible to an external observer. In simple words, any form of speech in absence of the first two steps of speech production that is characterized by miniscule movements in larynx and articulatory muscles is termed as internal articulation. This mode of speaking naturally occurs while reading that helps the human brain to comprehend what is being read and potentially reduces the cognitive load on the brain.

It should be clearly established that the internal articulation process is completely voluntary and occurs in the articulatory system upon receiving the stimulus from the brain. It is often confused with the thought process involved in the production of speech that occurs in the brain itself. The brain is responsible for the selection of words, organization of relevant grammatical forms and control of the motor system associated with the vocal apparatus (shown in figure 3-6). The motor system actuates the vocal apparatus accordingly and produces speech whether it be acoustic, whispered or silent (internally articulated). Internal articulation occurs due to the cumulative action of the brain and the peripheral nervous system but is very different from the speech conception happening in the brain.

# 4. MATHEMATICAL MODELING

Development of precise hardware as well as software in any system embodies proper calculation of design parameters which can be done using ideal mathematical models.

## 4.1 Circuit Modeling

This section includes the mathematical models for the parametric calculations of hardware components along with circuit implementation specifications.

### 4.1.1 Differential Amplifier

Differential amplifier is the amplifier which amplifies the difference between the two input signals. In an ideal differential op-amp, the output signal $V_{out}$ is given by

$$V_{out} = A_d(V_1 - V_2) \quad 4.1$$

Where, $V_1$ = input in non-inverting terminal
$V_2$ = input in inverting terminal
$A_d$ = Differential gain of op amp

From the above equation it is clear that for ideal op amp, any signal common to both the inputs have no effect on the output. This is known as common mode rejection. In non-ideal op amps output depends not only on differential input (Vd) but also on common mode signal (Vc).

$$V_d = (V_1 - V_2) \quad 4.2$$

$$V_c = \frac{(V_1 + V_2)}{2} \quad 4.3$$

Let Ad denotes differential gain of op amp and Ac denotes its common mode gain then the common mode rejection ratio (CMRR) of a differential amplifier is calculated as

$$\text{CMRR} = \left|\frac{Ad}{Ac}\right| \tag{4.4}$$

$$\text{CMRR} = 20\log\left|\frac{Ad}{Ac}\right| dB \tag{4.5}$$

CMRR depends on a number of design choices within the amplifier, but basically depends on the loop gain of the amplifier. With high loop gain, the error signal across the input terminals of the op amp is driven to zero and CMRR is high. At low frequency where the loop gain is high, the error signal is very low and CMRR is high. With low loop gain, the error signal across the input terminals of the op amp is high and CMRR is low. As frequency increases and loop gain decreases, the error signal across the input terminals of the op amp increases. The larger error signal across the input terminals of the op amp intern leads to lower CMRR. [16]

### 4.1.2 Bandwidth of an Operational Amplifier

Real op amps cannot operate in all frequency ranges as they have finite bandwidth. Let $A_0$ be the open-loop gain of an op amp and $w_0$ be the cut-off frequency of the op amp then the open loop voltage-gain transfer function of the op amp is given by

$$\text{A(s)} = \frac{A_O}{1 + \frac{s}{w_O}} \tag{4.6}$$

$$\text{Or, A(jw)} = \frac{A_O}{1+\frac{jw}{w_O}} \tag{4.7}$$

$$\text{Or, } |\text{A}| = \frac{A_O}{\sqrt{1+\left(\frac{jw}{w_O}\right)^2}} \tag{4.8}$$

For frequencies much greater than $w_0 = 2\pi f_0$, $w >> w_0$ open loop gain of op amp scales as

$$|A| = \frac{A_O}{\sqrt{1 + \left(\frac{jw}{w_O}\right)^2}} \quad 4.9$$

$$\text{Or } |A| = A_O\left(\frac{w_O}{w}\right) \quad 4.10$$

$$\text{Or } |A|f = A_O f_O \quad 4.11$$

Where, $A_0\, f_0$ is Gain-Bandwidth Product (GBP)

Thus, the product of open-loop gain and bandwidth of an op amp is constant which is known as Gain-Bandwidth Product (GBP). This product is given in the manufacturer dataset for each op amp and sometimes is denoted as "unity gain bandwidth". [17]

### 4.1.3 Butterworth Filter

Butterworth filter is a signal processing filter designed to have a frequency response as flat as possible in the passband i.e. ideally no ripples in the pass band. The transfer function of an ideal n-order Butterworth filter is given by

$$or, |T_n(\text{jw})^2| = \frac{1}{\left(1 + \frac{w_0}{\text{w}}\right)^{2\text{n}}} \quad 4.12$$

Where, $w_o$ = cut-off frequency of filter

The frequency response of the Butterworth filter is obtained from above equation which is as shown in figure below

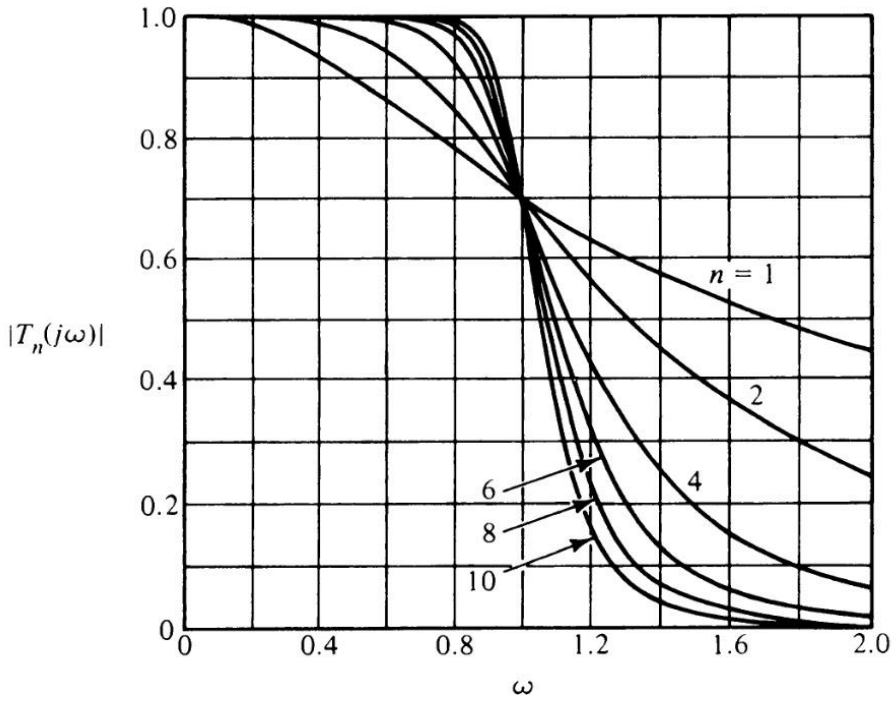



Figure 4-1: Frequency Response of Different Orders of Butterworth Filter [18]

The attenuation (A) introduced by a Butterworth filter is given by

$$A = 20\log|T(jw)|dB \tag{4.13}$$

For the pass band extending from w = 0 to w = $w_p$ the attenuation should not exceed Amax. From $w_p$ to $w_s$ lies transition band and for a stop band beyond $w_s$ the attenuation should not be less than $A_{min}$.

The order of a Butterworth filter with maximum attenuation for passband ($A_{max}$), minimum attenuation for stop band ($A_{min}$) is calculated using the following equation

$$T(jw) = \frac{T_0}{\left[1 + \varepsilon^2 \left(\frac{w_s}{w_p}\right)^{2n}\right]} \tag{4.14}$$

Where ε = maximum pass band gain,

$w_s$ = stop band frequency,

$w_p$ = pass band frequency.

The order of the Butterworth filter also determines its roll-off characteristics. For a Butterworth filter of order 'n' the roll-off rate is 20n dB/decade or 6n dB/octave. The design and circuit implementation of such higher order Butterworth filters with all the above parameters can be done in different filter design topologies such as Cauer topology, Sallen and Key topology, etc. [18]

#### 4.1.3.1 High Pass Filter

High-pass filter passes signals above cut-off frequency (fc) and attenuates signals lower than the cut-off frequency. Based on the design requirements the order of a Butterworth high-pass filter can be determined from equation 4.14. General high-pass passive filter of the second order is as shown in figure below.

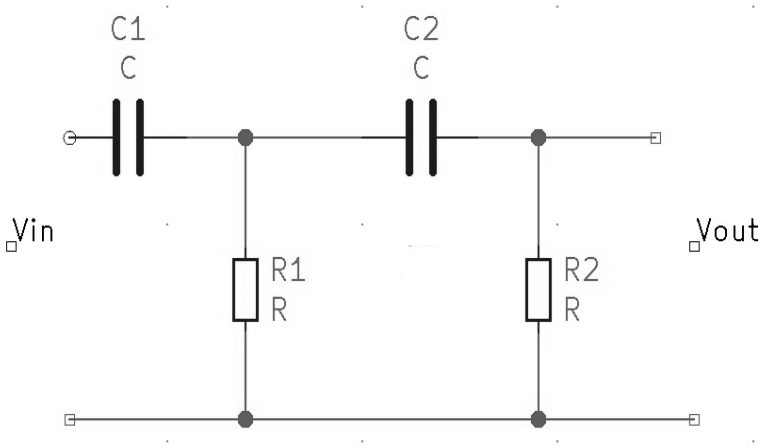



Figure 4-2: Second Order Passive Butterworth HPF

The cut-off frequency of the above filter circuit is given by

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \qquad 4.15$$

Let $R_1 = R_2 = R$ and $C_1 = C_2 = C$ the above equation simplifies into

$$f_c = \frac{1}{2\pi RC} \qquad 4.16$$

The value of C is determined as per the choice or given cut-off frequency and the corresponding value of R is determined from the above equation.

### 4.1.3.2 Low Pass Filter

Low-pass filter passes signals below cut-off frequency (fc) and attenuates signals higher than the cut-off frequency. Based on the design requirements the order of a Butterworth low-pass filter can be determined from equation 4.14. Sallen and Key topology of active analog filter design is very common in practice. Basic 2nd order LPF based on this topology as shown in figure below.

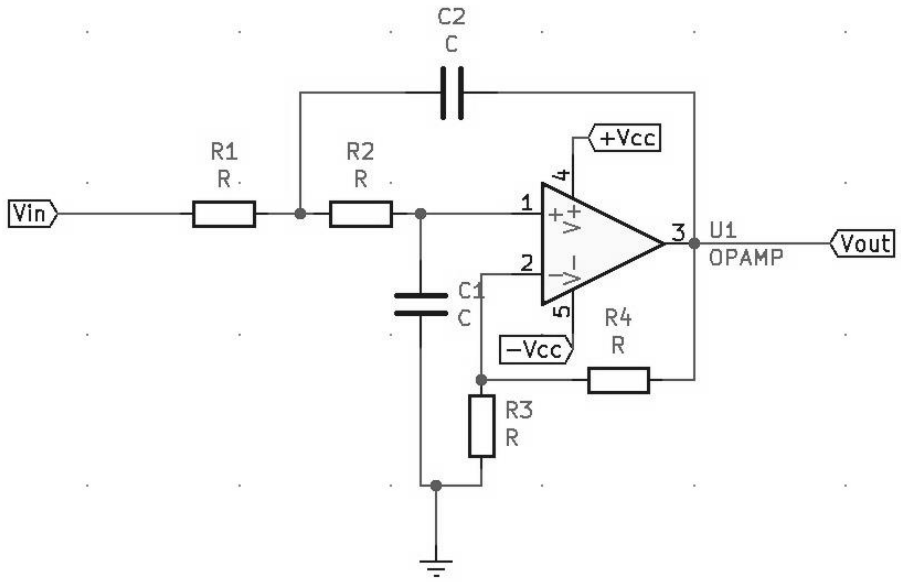



Figure 4-3: General Second Order Sallen-Key LPF

The transfer function of the above circuit is given by

$$\mathrm{A(s)} = \frac{A_0}{1 + w_c\{C_1(R_1 + \mathrm{R2}) + (1 - A_0)R_1C_2\}\mathrm{s} + {W_C}^2 R_1R_2C_1C_2 2S^2} \qquad 4.17$$

Where $w_c$ is the cut-off frequency

Since $R_1 = R_2 = R$ and $C_1 = C_2 = C$ we have

$$\mathrm{A(s)} = \frac{A_0}{[1 + w_c \mathrm{RC}(3 - A_0)s + (w_c RC)]} \quad 4.18$$

$$A_0 = 1 + \frac{\mathrm{R4}}{\mathrm{R3}} \quad 4.19$$

$$\mathrm{a} = w_c RC(3 - A_0) \quad 4.20$$

$$\mathrm{b} = (w_c RC)^2 \quad 4.21$$

Here a and b are considered

The value of C is set as per the choice and the corresponding values of R and $A_0$ is determined using the following equations.

$$\mathrm{R} = \frac{\sqrt{b}}{2\pi \mathrm{f}_c \mathrm{C}} \quad 4.22$$

$$\mathrm{A}_0 = 3 - (\frac{\mathrm{a}}{\sqrt{b}}) \quad 4.23$$

$$\mathrm{A}_0 = 3 - \frac{1}{\mathrm{Q}} \quad 4.24$$

Where Q = pole quality of the filter

On comparing the denominator of transfer function with 2nd order Butterworth polynomial which is,

$$B_2(s) = s^2 + 1.414s + 1 \quad 4.25$$

Which gives,

$$1 = \frac{1}{w_c RC} \quad 4.26$$

$$w_c = \left(\frac{1}{RC}\right) \quad 4.27$$

$$f_c = \frac{1}{2\pi RC} \quad 4.28$$

Where $f_c$ is the cutoff frequency of the filter, $w_c = 2\pi f_c$ and,

$$1.414 = \frac{3 - A_0}{w_c RC} \quad 4.29$$

From equation 4.26,

$$1.414 = 3 - A_0 \quad 4.30$$

$$A_0 = 1.586 \quad 4.31$$

From equations 4.24 and 4.31

$$Q = 0.707 \quad 4.32$$

Therefore the value of Q must be 0.707 and $A_0$ be 1.586, with equation 3.19 the value of $A_0$ is set. The value of Q must be equal or near to 0.707 otherwise if the value of Q is greater in a 2nd order filter, it will respond to a step input by quickly rising above, oscillating around, and eventually converging to a steady-state value. With a very low quality factor it will respond to a step input by slowly rising toward an asymptote. The response being similar to 1st order low pass filters.

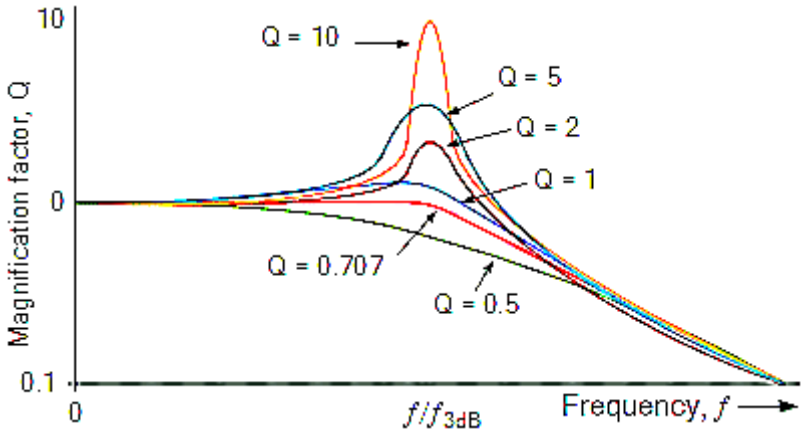



Figure 4-4: Q factor vs Frequency

### 4.1.4 Circuit Board Routing Traces

The design of the layout of printed circuit boards can be as demanding as the design of the electrical circuit especially when dealing with low amplitude and low frequency signals such as EMG. The selection of proper components, placement of power rails and each component, width of copper traces and even fabrication methods determine the performance of a PCB. The amount of current that a trace can conduct is heavily influenced by thickness of the plated copper, resistance of copper trace, width of the trace, length of the trace and many other parameters. To calculate the width of traces, the following formula can be used:

$$W = \frac{A}{(t * 1.378)} \tag{4.33}$$

Where W is the PCB trace width, in thousands of an inch (mil), A is the cross-section area, in $mils^2$ and t is the trace thickness, in $oz/ft^2$.
A is calculated as:

$$A = \left(\frac{l}{k * (T_{RISE})b}\right)^{\frac{1}{c}} \tag{4.34}$$

Where I is the maximum current through the PCB trace, $T_{RISE}$ is the maximum temperature rise of the PCB trace (in °C over ambient temperature) and b = 0.44, and c = 0.725 are fixed parameters.

The value of k depends on the location of the trace, for internal traces k = 0.024, and for external PCB traces k = 0.048. The resistance, R, of the trace is calculated as:

$$R = \left(\rho * \frac{L}{A'}\right) * \left(1 + \alpha * (T_{TEMP} - 25\,°C)\right) \tag{4.35}$$

Where L is the length of the trace, in cm, A' is the cross-section area, in $cm^2$, $\rho = 1.7 * 10^{-6}$ Ω cm is the resistivity of copper, $\alpha = 3.9 * 10^{-3}$ $°C^{-1}$ is the resistivity temperature coefficient for copper and $T_{TEMP}$ is the temperature of the PCB trace.

## 4.2 Signal Preprocessing Theory

This portion includes digitized data preprocessing along with machine learning model parameters under ideal considerations.

### 4.2.1 Windowing

There are different types of windowing functions that can be applied depending on the characteristics of a signal. Hamming, Hanning, Blackman-Harris and Gaussian are some examples. The frequency domain plot of a window is a continuous spectrum with a main lobe centered at each frequency component of time domain signal and side lobes approaching zero. Lower amplitude of the side lobes reduces the spectral leakage, a phenomenon in which the spectrum is smeared due to leakage of energy from the frequency components nearby. It can further be reduced by increasing the roll-off rates of side lobes. [19]

Let us consider X(t) be a time domain signal which is to be fed to the window characterized by a transfer function W(t), then the output of the window is given by,

$$Y(t) = X(t)W(t) \tag{4.36}$$

Here W(t) truncates the signal beyond its window size resulting in only a part of the input signal as Y(t). The cut-off points of W(t) introduce high frequency components at the beginning and at the end of the output Y(t). The input signal applied with the Hamming window is as shown in the figure below.

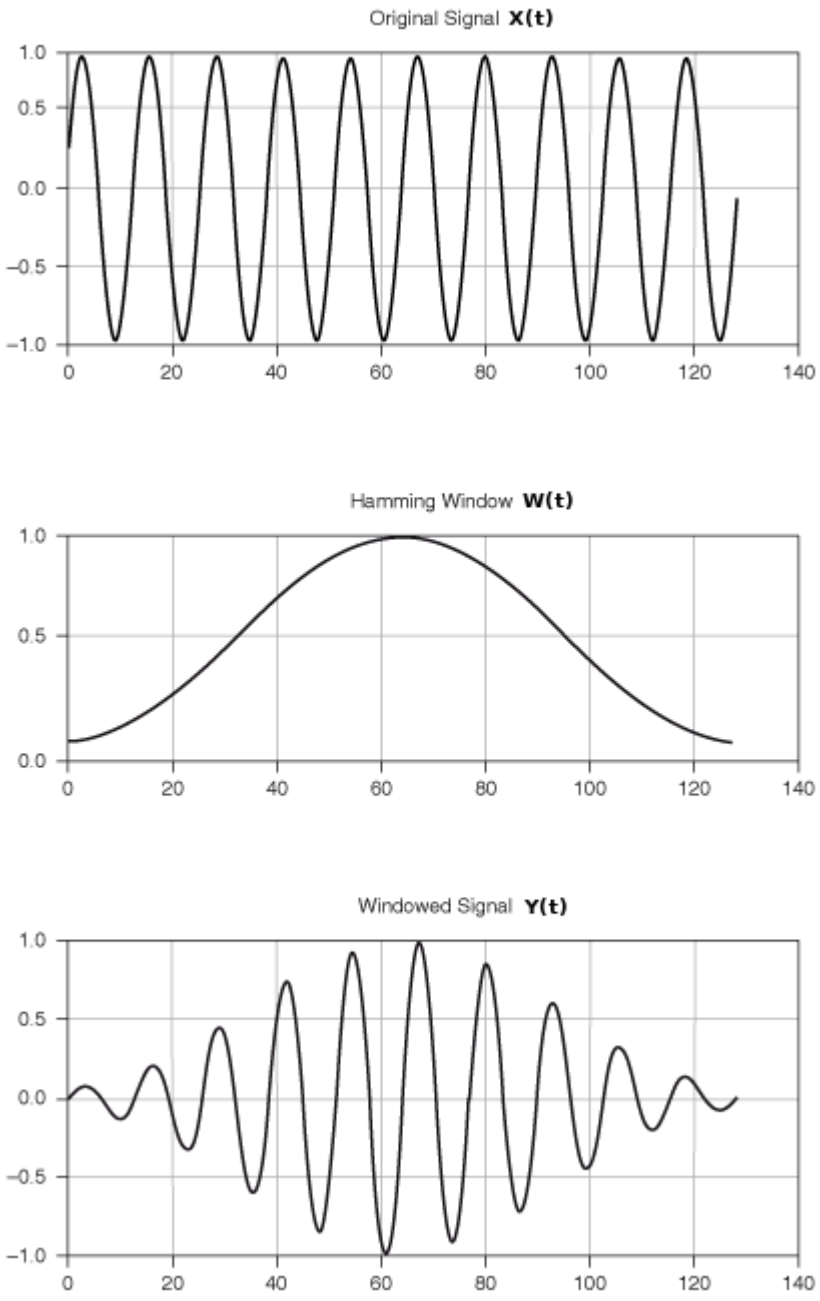



Figure 4-5: Hamming Window Applied to an Input Signal

### 4.2.2 Temporal and Spectral Features

Most of the suitable modern neural networks are able to learn the features from the provided EMG signals, but the black-box nature of deep learning does not reveal much information learned by the network and how it relates to handcrafted features. The high variability of EMG recording between participants causes neural network to generalize

poorly across subjects using standard training methods. Therefore, a hybrid approach of providing handcrafted features for a deep learning model stands to be more suitable to for training a neural network in this regard [20]. These handcrafted features include both the temporal and spectral features.

#### 4.2.2.1 Short Time Fourier Transform

Short Time Fourier Transform (STFT) is obtained by introducing a sliding window to the time variant signal so it is also known as time-dependent Fourier transform. This window adds a new dimension of time to the frequency response by suppressing the input signal outside a certain region. Let us consider an input signal x(t) is introduced to the sliding window with window function $\gamma(t)$ then the discrete time STFT of the signal is given by,

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)\gamma(t-\tau)e^{-j\omega n}dt \qquad 4.37$$

Here $\tau$ is the window interval centered on zero.

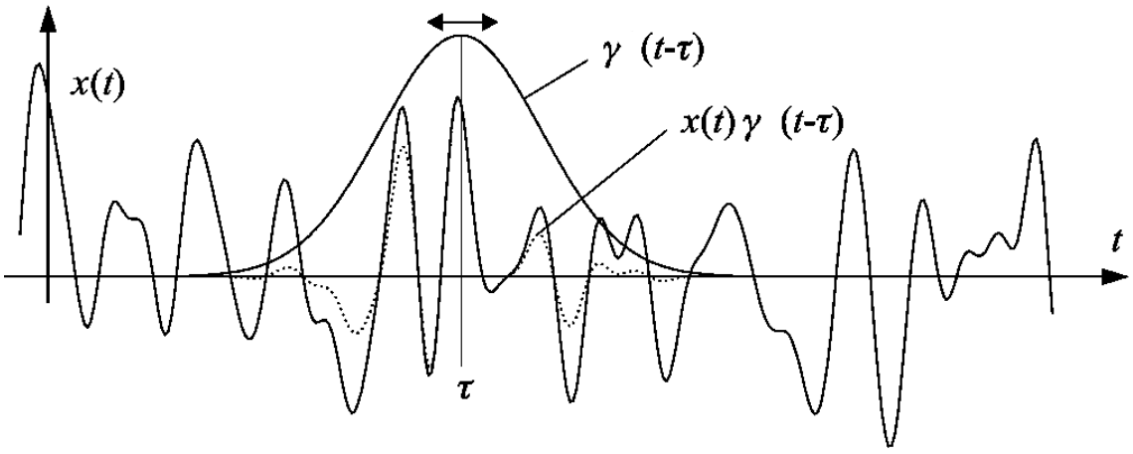



Figure 4-6: STFT of a Continuous Time Signal

Let us consider, Δt be the radius of the time window γ(t) centered at τ(0) while Δω be the frequency window **Γ** (ω) centered at w_0. Form Heisenberg uncertainty principle we obtain,

$$\Delta t * \Delta\omega \geq \frac{1}{2} \tag{4.38}$$

This relation shows that size of time-frequency windows cannot be made arbitrarily small and that a perfect time-frequency resolution cannot be achieved [21]. Thus, the selection of window size should be done considering equation 4.35. The window function is assumed to be non-zero only within the window interval. The time-frequency resolution of the spectrogram will be dependent on the chosen value of window size. Large window size results in very low time-frequency resolution while very small window size cannot locate the time domain so the selection of appropriate window size is essential. The selection of appropriate window size depends on the type of signal [22]. Also, the type of window selection depends on the type of signal.

The squared magnitude of STFT is known as a spectrogram. Generally, STFT is complex-valued so spectrograms are often used for further processing of the signal. The equation for spectrogram $S(\tau, \omega)$ can be obtained by rectifying and then squaring equation 4.36 as,

$$S(\tau, \omega) = \left| \int_{-\infty}^{\infty} x(t)\gamma(t-\tau)e^{-j\omega n}dt \right|^2 \tag{4.39}$$

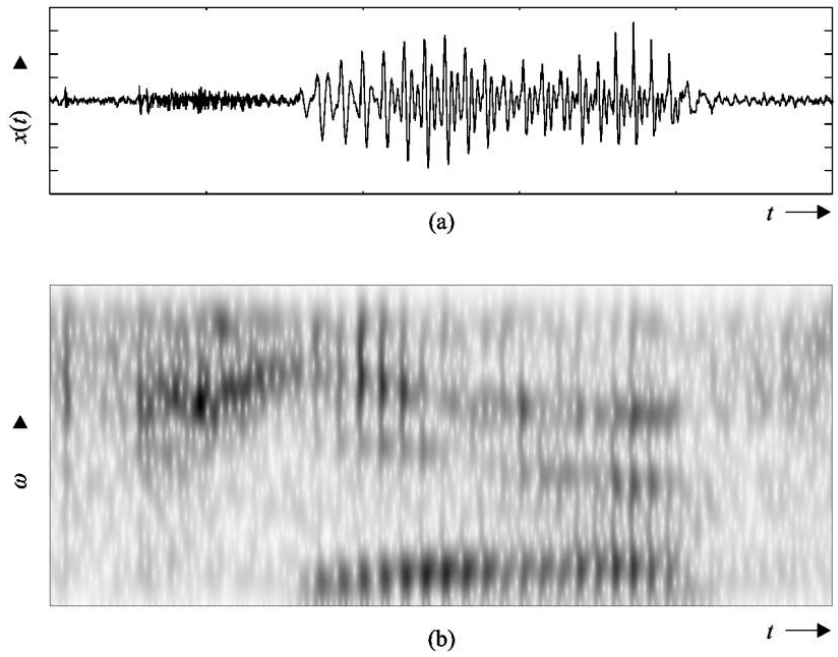



Figure 4-7: a) Input Speech Signal b) Spectrogram of the Input Signal

#### 4.2.2.2 Mel-Frequency Cepstral Coefficient

Mel-frequency Cepstral Coefficient is an analytical tool that is based on human perception of sound. It combines the advantages of the cepstrum analysis with a perceptual frequency scale based on human ear's critical bands. Human hearing can be modeled using a non-linear scale called the Mel-scale as shown in figure 4.37. In other words, MFCC is a representation of the STFT power spectrum of sound based on linear cosine transform of a log power spectrum on a non-linear Mel-scale frequency [23]. The formula to convert a frequency f in Hertz to Mel-scale is given as:

$$m = 2595 \ log\ (1 + \frac{f}{700}) \qquad 4.40$$

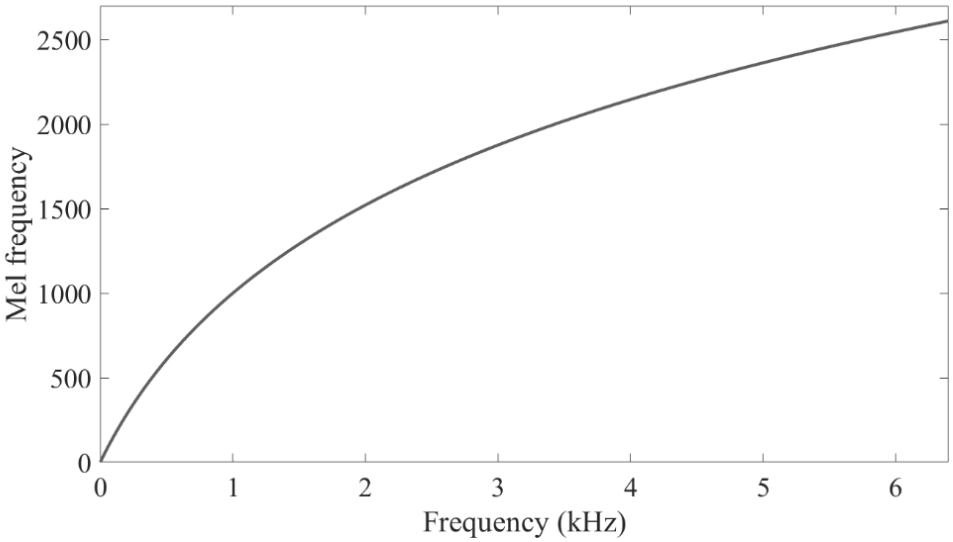



Figure 4-8: Mel Scale Frequency Plot

To calculate the MFCC, a section of windowed signal as shown in figure 3-10 is first ported to frequency domain using Fourier transform and mapped to a Mel-scale using Mel-filter banks to obtain Mel-spectrum. Logarithm of the Mel-spectrum gives the log Mel-spectrum and the discrete cosine transform of the obtained Log Mel-spectrum gives the Mel-Frequency Cepstral Coefficients as shown in figure 4-10. [24]

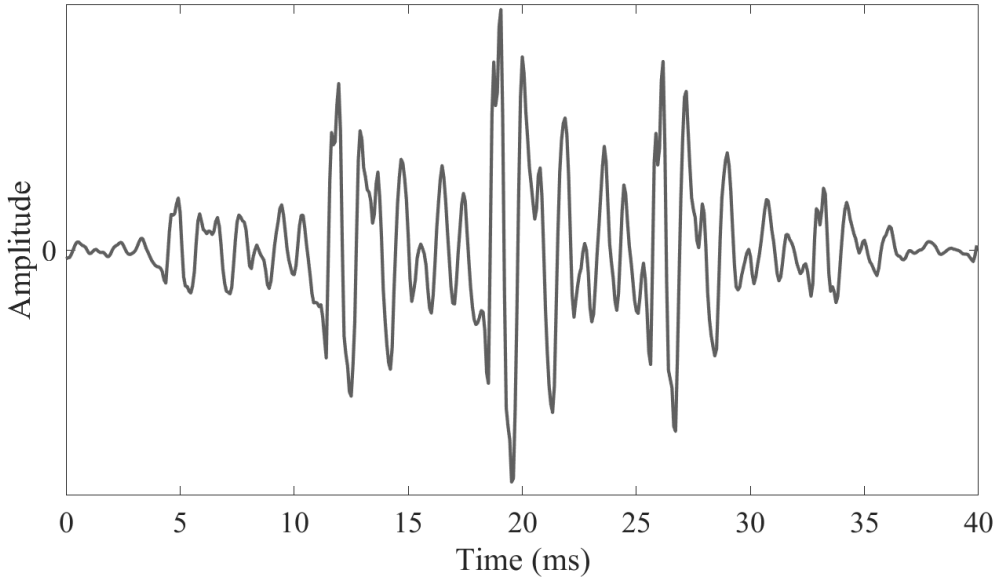



Figure 4-9: Windowed Audio Signal

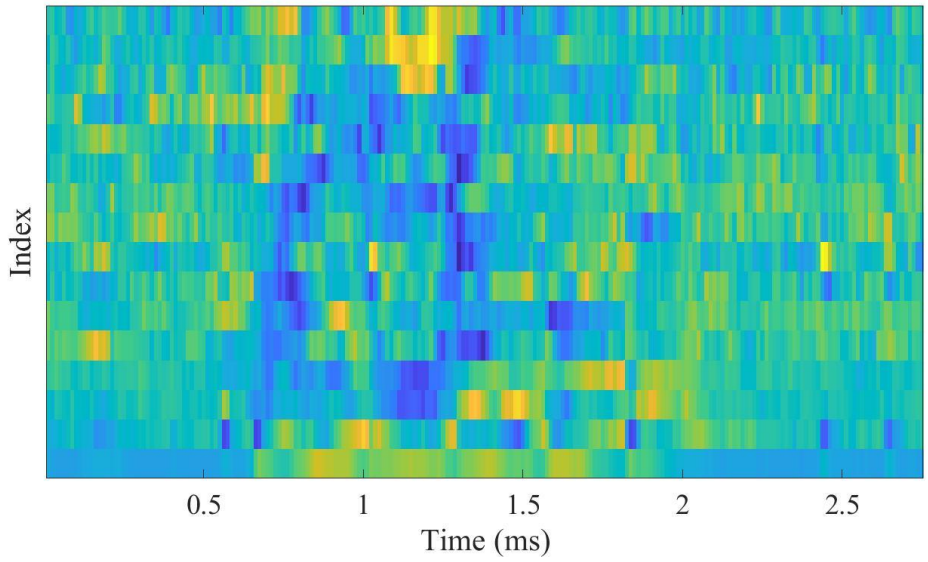



Figure 4-10: MFCC Plot of a Segment of Audio Signal

#### 4.2.2.3 Zero Crossing Rate

Zero Crossing Rate (ZCR) is the rate of sign changes of a signal i.e. the rate of which the signal changes from positive to zero to negative or from negative to zero to positive. This is a temporal feature used in both speech recognition and music information retrieval. ZCR is defined formally as:

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{R<0}\,(S_t S_{t-1}) \qquad 4.41$$

Where **S** is the signal of length **T** and is $\mathbf{1_{R<0}}$ an indicator function.

#### 4.2.2.4 Nine Point Double Average

The nine-point double-averaged signal w[n] is defined as

$$w[n] = \frac{1}{9}\sum_{k=-4}^{4} x[n+k] \qquad 4.42$$

Where v[n] is given by,

$$v[n] = \frac{1}{9}\sum_{k=-4}^{4} x[n+k] \quad 4.43$$

Where x[n] is the mean of an EMG signal.

#### 4.2.2.5 High Frequency Signal

The high frequency signal is obtained by subtracting the nine point double average from normalized mean. Formula for calculating high frequency and rectified high frequency is given by 4.41 and 4.42 respectively.

$$p[n] = x[n] - w[n] \quad 4.44$$

$$r[n] = |p[n]| \quad 4.45$$

#### 4.2.2.6 Frame Based Power

The Frame based power is the sum of squares of the signal in the frame. Which is given as,

$$P_f = \sum_{i=0}^{k-1} (a[i])^2 \quad 4.46$$

Where k is the frame size a[i] is the segment of the signal.

### 4.3 Neural Network Parameters

Machine learning models, especially in supervised learning, are meaningless unless the parameters are tuned or optimized. The reason for doing this is to maximize the algorithm objective, optimize classification or regression process and enhance the model processing time as well. Some of the model optimizing parameter are:

### 4.3.1 Activation function

Activation function is the most crucial part of a neural network as they determine the output of a node given a set of inputs along with the accuracy and computational efficiency of the model. Activation function is a mathematical function and is attached to each neuron in the network. It determines whether a neuron should be activated (fired) or not based on the relevance of the neuron input for the prediction of the model. Activation functions should be differentiable. The activation function can be linear, rectified linear or non-linear. Some of the commonly used activation functions are ReLU, Sigmoid, hyperbolic tangent and softmax. Activation function layer can be applied after each convolution layer for better mapping and classification. [36]

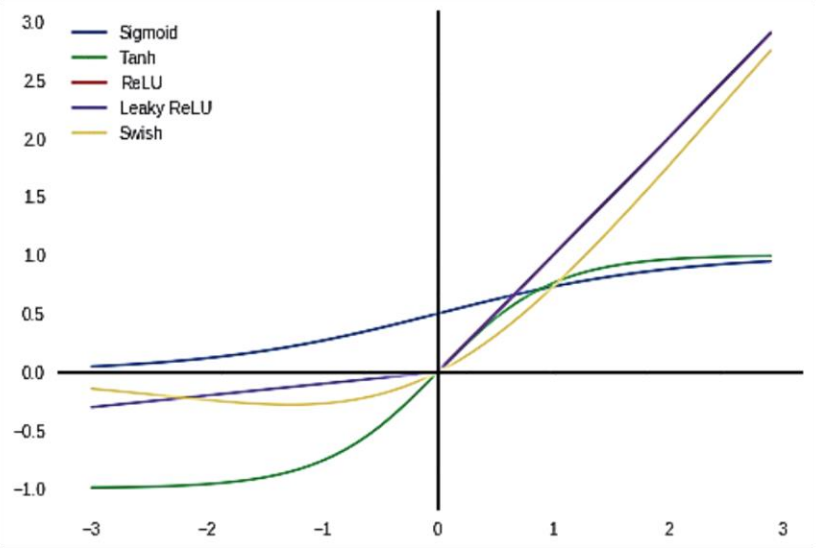



Figure 4-11: Comparison of Some Commonly Used Activation Functions

**A. Sigmoid**

Sigmoid activation function is mathematically defined as

$$F(x) = \frac{1}{1 + e^{-x}} \qquad 4.47$$

And its first derivative is given by,

$$F'(x) = \big(1 - F(x)\big) * F(x) \qquad 4.48$$

Sigmoid activation has smooth gradient and output value is normalized between zero and one being an effective activation function for binary classification. The main problem with using this function is vanishing gradient. For very large and small values of input, the output of this function saturates more towards one or zero. So, it takes more time to predict the result, which makes the computation expensive. Also, the output of sigmoid is non zero-centric which causes problems because later layers would receive only positive inputs which makes optimization harder.

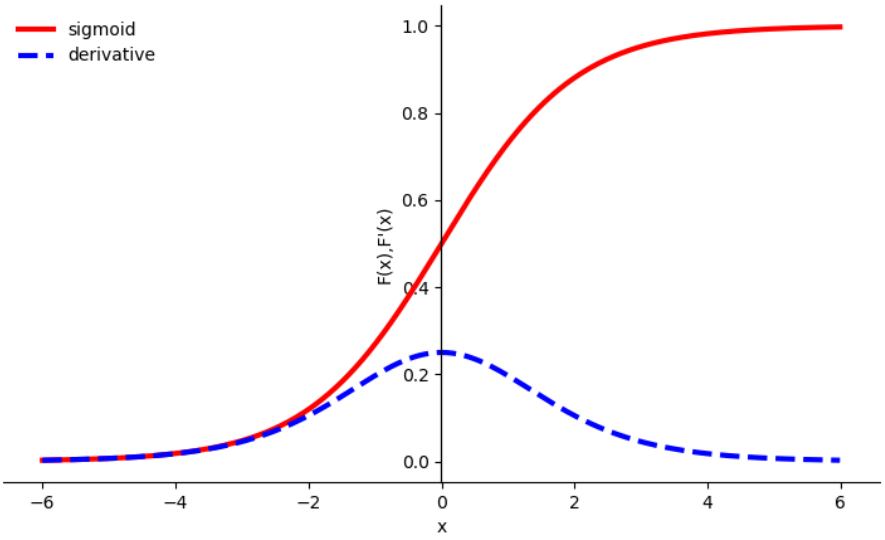



Figure 4-12: Sigmoid Activation Function and its Derivative

**B. Hyperbolic Tangent**

Hyperbolic tangent, as stated by its name is hyperbolic of trigonometric function ‘tan’. It is mathematically represented as

$$F(x) = \frac{e^{-x} - e^{+x}}{e^{-x} + e^{+x}} \quad 4.49$$

First derivative of tanh gives,

$$F'(x) = 1 - F(x)^2 \quad 4.50$$

Hyperbolic tangent, Tanh is an activation function which has a similar nature of curve with sigmoid. Its output lies between -1 and 1. Unlike sigmoid, it has zero centered output which is suitable for models that have strongly negative, neutral and positive values. Tanh activation function also suffers vanishing gradients like sigmoid.

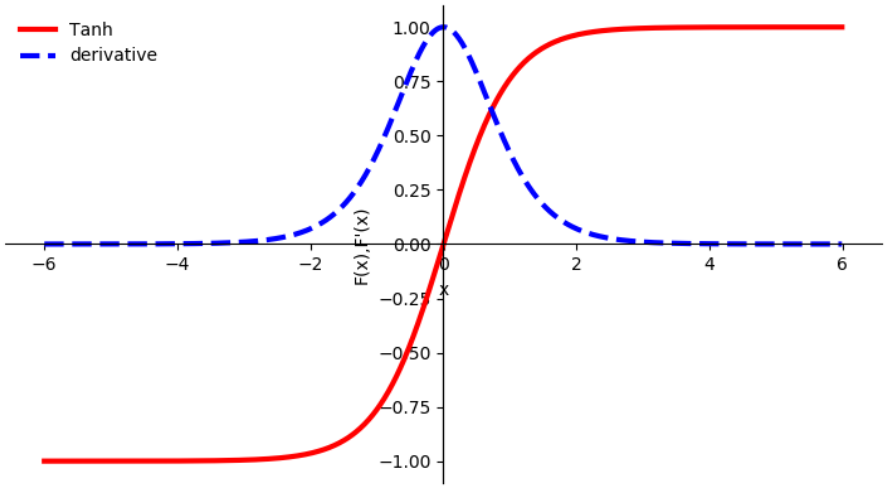



Figure 4-13: Tanh Activation Function and its Derivative

**C. ReLU**

Rectified Linear Unit (ReLU) is defined as,

$$F(x) = 0 \ for\ x \leq 0, x\ for\ x > 0 \quad 4.51$$

Derivative of ReLU is given as,

$$F'(x) = 0 \ for\ x \leq 0, 1\ for\ x > 0 \qquad 4.52$$

ReLU activation is another effective activation function commonly used in machine learning. It simply converts all negative value to zero and positive value remains as it. Compared with other activation functions, it is computationally efficient and the output is also non-saturating. But, when the inputs are negative, gradient of the function becomes zero, due to which backpropagation cannot be performed and the model cannot learn further. This situation is known as 'dying ReLU'. Another problem with ReLU is for positive values, output does not saturate so there will be no vanishing neurons on further nodes.

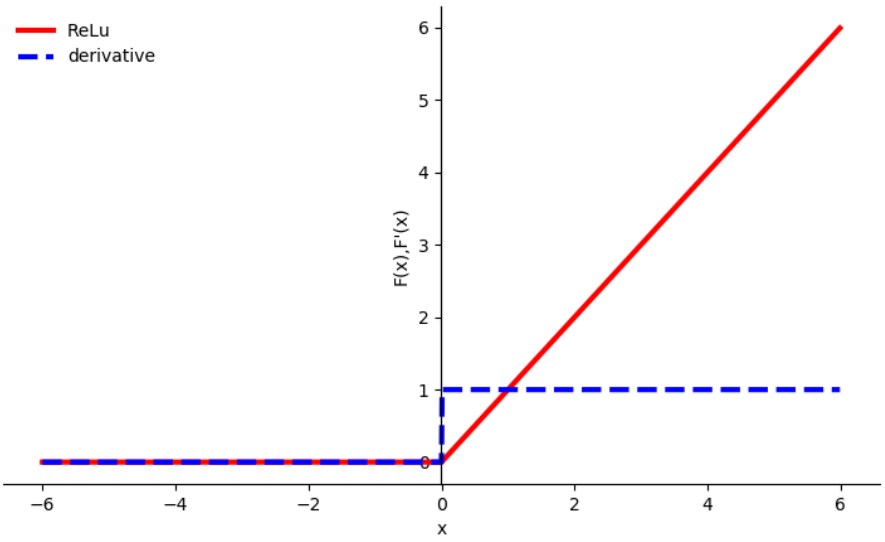



Figure 4-14: ReLU Activation Function and its Derivative

**D. Leaky ReLU**

To overcome the problem of dying ReLU, a small slope is introduced in ReLU, in its negative region. The resulting function is known as Leaky ReLU. Because of slope in the negative region, it includes the value of the negative axis too during back propagation.

Leaky ReLU is defined as

$$F(x) = 0 \ for \ x \leq 0, 0.01x \ for \ x > 0 \quad 4.53$$

Derivative of ReLU is given as,

$$F'(x) = 0 \ for \ x \leq 0, 0.01 \ for \ x > 0 \quad 4.54$$

Despite being easy computations, Leaky ReLU gives inconsistent output for negative inputs.

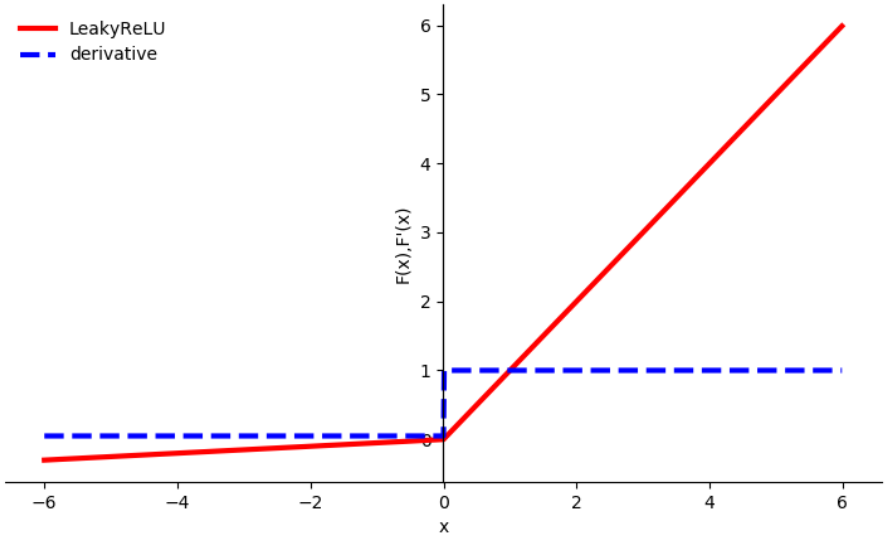



Figure 4-15: Leaky ReLU Activation Function and its Derivative

### 4.3.2 Loss function

A loss function or cost function maps values or events of the variables onto the real domain representing some cost associated with the events or values. In other words it is an objective function to evaluate how well an algorithm models the given data. If prediction of the algorithm deviates from the actual output then loss function computes a value which is termed as cost or loss. This value is to be minimized as much as possible during the training process under subjective constraints. The loss function may be broadly classified as a regression and classification loss function. Regression loss function includes mean square error, mean bias error, etc. while classification includes hinge loss, cross-entropy loss, and so on.

**A. Sum of Squared Error**

Error in statistics is defined as the difference between individual data and the estimation of the sample. The sum of the squares of these errors can be used to determine the variance within a cluster. The SSE of a sample with (x1,x2,...xn) datapoints and sample mean $\overline{x}$ is given by,

$$SSE = \sum_{i=1}^{n} (x_i - \overline{x}) \quad 4.55$$

**B. Hinge Loss**

Hinge loss is a classification loss which is used to train classifiers for maximum margin classification. For a feature space X = ($x^{(1)}$, $x^{(2)}$ ,..., $x^{(n)}$) with labels Y = ($y^{(1)}$, $y^{(2)}$ ,..., $y^{(n)}$) the hinge loss can be computed as,

$$Loss_h(z) = 0 \ if \ z \geq 1, (1 - z) \ if \ z < 1 \quad 4.56$$

Where $z = y^{(i)} (\theta . x^{(i)} + \theta_o)$.

Here $\theta$ is the parameter vector which controls the orientation of the boundary while $\theta_o$ is the offset parameter.

**C. Cross-Entropy Function**

The cross entropy of the distribution q relative to a distribution p, H(p, q) is given by the expectation of log as,

$$H(p, q) = -E(log q) \quad 4.57$$

For a discrete random variable x in X discrete states with probability p(x), entropy can be calculated as,

$$H(p,q) = -\sum_{x \epsilon X} p(x) log q(x) \quad 4.58$$

The selection of the loss function depends on the type of prediction which the model is supposed to perform. The type of dissimilarity between the output of the idealized system and the actual data points is termed as error. This error is then modeled using a probabilistic distribution function. The nature of error such as Gaussian decides the type of loss function appropriate to use.

# 5. SIGNAL ARTIFACTS AND NOISE

Noise is any unwanted disturbance that hinders or interferes with a desired signal. To put it differently, everything that is not part of the signal wanted to be measured is considered noise. However, a differentiation can be made between disturbances or interferences and the word noise. Disturbances often come from sources external to the circuit under study, and result from electromagnetic or electrostatic coupling with the power lines, fluorescent lights, cellphones and cross-talk between adjacent circuits, even mechanical vibration could also cause disturbances. Most of these types of disturbances and possible sources of interference are “man-made” and can be minimized or eliminated.

## 5.1 Types of Noise

The identity of an actual EMG signal that originates in the muscle is lost due to the mixing of various noise signals or artifacts. The attributes of the EMG signal depend on the internal structure of the subject, including the individual skin formation, blood flow velocity, measured skin temperatures, the tissue structure (muscle, fat, etc.), the measuring site, and more. These attributes produce different types of noise signals that can be found within the EMG signals.

### 5.1.1 Electrode Noise

Electrode noise occurs due to the electrolyte–skin and electrolyte–metal interfaces. Once the electrolyte–metal electrochemical reaction stabilizes, this source of noise is negligible (0.3 μV P-P). The amplitude of the electrolyte–metal noise for Ag-AgCl electrodes decreases dramatically within the first minute of application and stabilizes. The electrolyte-skin interface is more problematic. The noise voltage can range from 5 to 60 μV P-P.

The elimination of such noises can be achieved with good skin preparation (but it is subject dependent) and using specific types of electrolyte for the specific type of electrode.

### 5.1.2 Inherent Noise

The amplitude of EMG is random in nature. EMG signal is affected by the firing rate of the motor units, which, in most conditions, fire in the frequency region of 0 to 20 Hz. The numbers of active motor units, motor firing rate and mechanical interaction between muscle fibers can change the behavior of the information in the EMG signal. This kind of noise is considered as unwanted, and the removal of the noise is important.

### 5.1.3 Cross-Talk

Cross-Talk refers to the signal that is detected over a certain muscle but is generated by another, mostly nearby muscle. It is mostly prevalent in surface electrodes where the distance of the detection points from the sources is of the same order of magnitude for the sources in different muscles. Crosstalk also depends on the many physiological parameters, and can be minimized by choosing electrode size and inter-electrode distances (typically 1–2 cm) carefully which actually improves the selectivity of the electrodes.

Cross-talk also occurs in between the electrode leads that are carrying the detected sEMG signals from the electrodes to the acquisition device. The sEMG signals are however weak and cause minimal distortions in the adjacent leads. Nonetheless, use of shielded leads and maintaining proper distance in between the electrode leads should help eliminate any unwanted effects in the signals.

### 5.1.4 Movement Artifacts

It is vital to maintain a steady and secure connection at the skin-electrode interface to eliminate any artifact associated with the movement of cables and displacement of electrodes. Movement artifacts cause irregularities in the signal. This can be reduced by proper design of the electronic circuitry and maintaining proper set-up.

### 5.1.5 ECG Artifacts

EMG signal extract is bound to be contaminated by the electrical activity from the heart. The placement of EMG electrodes, which is conducted by a selection of the pathological

muscle group, often decides the level of ECG contamination in EMGs. Due to an overlap of frequency spectra by ECG and EMG signals and their relative characteristics, it is very difficult to remove the ECG artifacts from the EMG signal.

ECG contamination in EMGs may be kept at a minimal level by common-mode rejection at the recording site, by the careful placement of bipolar recording electrodes along the heart's axis if possible. The electrode placement in the proposed design is mostly focused on muscles relating to speech and are localized at the facial region and thus not very susceptible to ECG artifacts.

### 5.1.6 Electromagnetic Noise

The human body behaves like an antenna—the surface of the body is continuously inundated with electric and magnetic radiation, which is the source of electromagnetic noise. Electromagnetic sources from the environment superimpose the unwanted signal, or cancel the signal being recorded from a muscle. The amplitude of the ambient noise (electromagnetic radiation) is sometimes one to three times greater than the EMG signal of interest.

The dominant concern for the ambient noise arises from the 50 Hz radiation from power sources, which is also called line noise. This is caused by differences in the electrode impedances and in stray currents through the patient and the cables. However, in order to remove the recorded artifact, off-line processing is necessary. Line noise (n(t)) with its harmonics can be mathematically represented as:

$$n(t) = cos(2\pi\, 50t) + cos(2\pi\, 100t) + cos(2\pi\, 200t) + cos(2\pi\, 300t) \qquad 5.1$$

A number of adaptive filter techniques have been proposed for the attenuation of the line noise, such as adaptive FIR notch filter, adaptive IIR notch filter, adaptive notch filter using Fourier transform and so forth. These filters improve the SNR of an EMG signal by eliminating the line noise from the system.

# 6. INSTRUMENTATION AND REQUIREMENT ANALYSIS

The hardware as well as software requirements are to be analyzed before implementation. The analysis includes specifications, responses and supportive environment of the component in the system.

## 6.1 Hardware Components

This section includes hardware component description along with the analysis of individual response in the circuit.

### 6.1.1 Electrodes

An electrode is a solid electric conductor through which an electric current enters or leaves an electrolytic cell. It is simply a transducer that converts ionic potentials to electric potentials. There exist two main types of electrodes for the extraction of EMG signals from the body. They are:

#### 6.1.1.1 Indwelling Electrode

Indwelling electrodes are invasive electrodes inserted through the skin directly over the muscle. Needle electrodes and fine wire electrodes are two commonly used indwelling electrodes used to measure action potential of a motor unit directly. Indwelling electrodes have two main advantages. One is that its relatively small pickup area enables the electrode to detect individual MUAPs during relatively low force contractions. The other is that the electrodes may be conveniently repositioned within the muscle (after insertion) so that new tissue territories may be explored. However, better selectivity and crosstalk immunity of indwelling electrodes comes at a price. They are painful and carry the risk of infections.

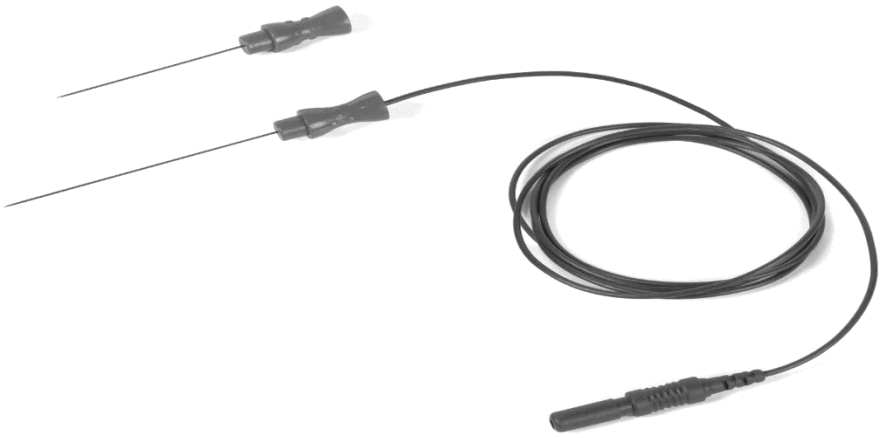


Figure 6-1: Monopolar Needle Electrode

#### 6.1.1.2 Surface Electrode

Surface electrodes are non-invasive electrodes placed on the skin directly over the muscle for measurement and detection of EMG signal. These electrodes are simple and very easy to implement and do not require medical supervision and certification. It is designed to selectively obtain the surface EMG signal while minimizing the artifacts, DC potentials and environment noise picking.

The theory behind the working of surface electrodes is that they form a chemical equilibrium between the detecting surface and the skin of the body through electrolytic conduction, so that current can pass from an electrolyte to a non-polarized electrode oxidizing the electrode atoms. The resulting cations and electrons flow in opposite directions: the electrons go through the metal cables attached to the electrodes meanwhile the cations go to the electrolyte. However, use of proper electrolytes with respective electrodes should be ensured for the electrolytic conduction to occur.

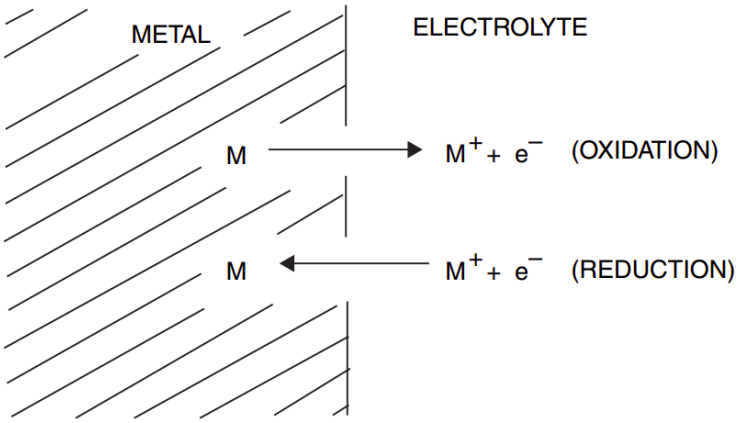



Figure 6-2: Electrode-electrolyte Interface

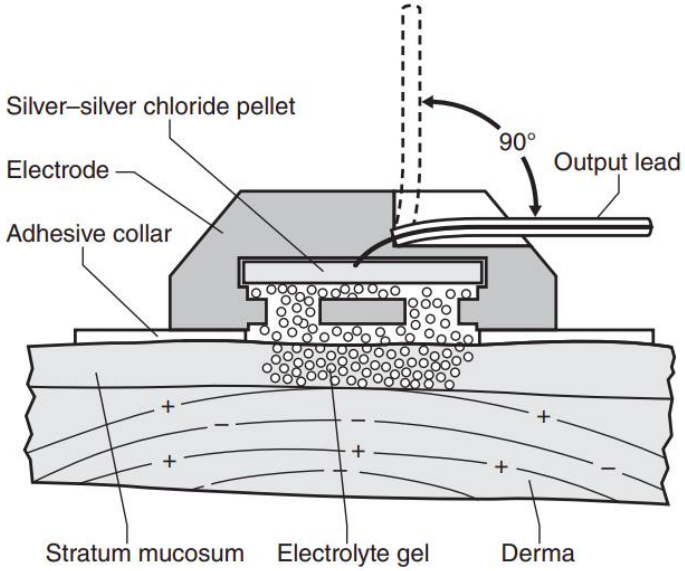



Figure 6-3: Skin-Electrode Interface (Ag-AgCl electrodes)

When the electrochemical reaction between the metal and the electrolyte stabilizes, a potential difference known as "half-cell potential" is formed between the negative electrode and the positive electrolyte which is determined by the Nernst equation as:

$$E = \frac{RT}{nF} ln(\frac{a_1}{a_2}) \qquad 6.1$$

Where $a_1$ and $a_2$ are ionic activities on each side of the membrane,

E= half-cell potential,

R= universal gas constant = 8.314 Joule per mole per kelvin,

T= absolute temperature,

n= the number of valence electrons in the metal,

F= Faraday Constant = 96485 C per mole.

The half-cell potential of a single electrode results in a DC offset in EMG signal. If two chemically identical electrodes make contact with the same electrolyte/body, the two interfaces should, in theory, develop identical half-cell potentials. When connected to a differential amplifier, the half-cell potentials of such electrodes would cancel each other out and the offset voltage would be zero. The electrode potentials would, therefore, make zero contribution to a bio-signal they were being used to detect. Unfortunately, slight differences in electrode metal or gel result in the creation of offset voltages, which can greatly exceed the physiological variable to be measured. Generally, a more significant problem is that the electrode offset voltage can fluctuate with time, thus distorting the monitored bio-signal.

The skin, gel, and electrode interfaces function as a complex physical system that is frequency dependent and affects the EMG signal in a deterministic way. It represents a complex impedance that can be modeled as a capacitor ($C_1$) in series with a resistor ($R_1$). This impedance may vary from a few kilo-ohms to a few mega-ohms, depending on electrode size and skin condition. There is an additional resistor ($R_2$) in parallel to denote the resistance of the chemical reaction (activation energies) that moves the charge at the interface to accurately model the skin-electrode interface.

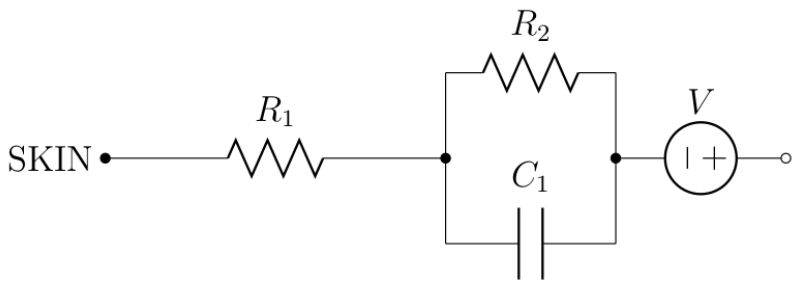



Figure 6-4: Skin-Electrode Circuit Model [17]

Where V = Half-cell potential

$C_1$ = capacitive effects of the electrolyte dipole layer at the electrode surface,
$R_1$ = Bulk resistance of the electrolyte gel,
$R_2$ = resistance of the chemical reaction (activation energies) that moves the charge at the interface.

Surface electrodes are usually made up of silver/silver chloride (Ag-AgCl), silver chloride (AgCl), silver (Ag) or gold (Au) or platinum (Pt). Surface area, utility, selectivity, sensitivity and many other parameters vary with the type of material used in the electrode. Selecting the proper type of electrodes that can result in having low electrode-skin impedance and can last longer for recording is important for EMG measurements.

Surface electrodes can be either polarizable or non-polarizable. The electrode where no actual charge crosses the electrode-electrolyte interface when a current is applied is a polarizable electrode. The current across the interface is a displacement current and the electrode acts like a capacitor. The electrode where the current passes freely across the electrode-electrolyte interface without any external energy to make the transition is a non-polarizable electrode. Platinum electrode is an example of polarizable electrode whereas Ag-AgCl electrode is an example of non-polarizable electrode.

#### 6.1.1.2.1 Silver−Silver Chloride Electrodes

Ag-AgCl electrodes are electrodes with a thin layer of silver coating on plastic substrates and the outer layer of silver is converted to silver chloride. Some of the important characteristic of Ag-AgCl electrodes are:

- Low half-cell potential of about 220 mV
- High conductivity of $6.30\times10^7$ Siemens per meter at 20℃
- High exchange current density of 10A/cm
- Low level of intrinsic noise
- Low contact impedance

Electrodes made of Ag-AgCl are often preferred over the others, as they are almost non-polarizable electrodes, which means that the electrode-skin impedance is resistive and

not capacitive. Low half-cell potential results in low DC offset in recordings and small redox potential facilitates the easier and fast exchange of ions. Therefore, the surface potential is less sensitive to relative movements between the electrode surface and the skin. Additionally, these electrodes provide a highly stable interface with the skin when electrolyte solution is interposed between the skin and the electrode. Such a stable electrode-skin interface ensures high signal to noise ratios, reduces the power line interference in bipolar derivations (50 Hz or 60 Hz frequencies and their harmonics) and attenuates the artifacts due to body movements.

Figure 6-5: Disposable Ag-AgCl Electrodes

#### 6.1.1.2.2 Gold Electrodes

Gold electrodes are electrodes with a thin layer of gold coating on metals like silver or copper. Some of the important characteristic of gold electrodes are:

- Half-cell potential of about 1.680 V
- Has high conductivity of 4.1×10^7 Siemens per meter at 20℃
- Higher contact impedance than Ag-AgCl
- Although expensive, they are reusable and durable
- High immunity to external noises

Typically, gold plated EMG electrodes have a 1.45 mm diameter conductive area on a disc of 10 mm. Smaller area provides high selectivity and thus is suitable for detection of EMG signals of a localized area or an individual muscle tissue.

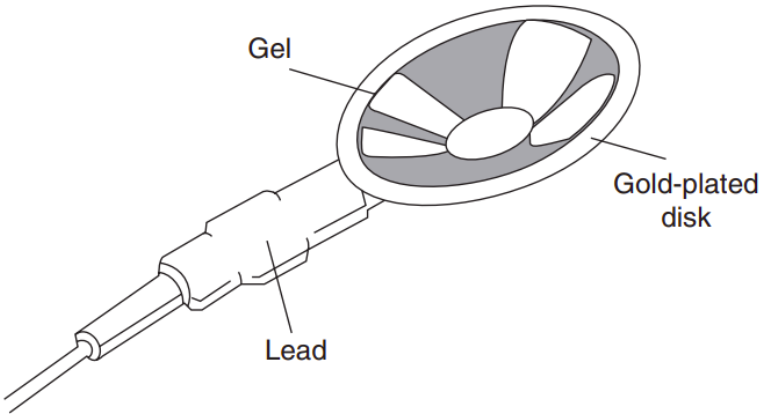



Figure 6-6: Gold Plated Cup Electrodes [25]

### 6.1.2 Electrode Configuration

Electrode configuration refers to the number of recording surfaces and their arrangement relative to muscle, tendon and bony surface. The two most common methods are:

#### 6.1.2.1 Monopolar Configuration

Monopolar uses three electrodes E1, E2 and Ground. E1 is placed over the muscle itself where the EMG signal is to be extracted and also referred to as “active recording surface electrode”. E2 is placed on an electrically neutral location such as tendon and also referred to as “reference electrode” and Ground is placed on a bony surface distant to E1 and E2. This configuration is called monopolar because only one electrode (E1) is used to record the muscle activity.

For monopolar configuration, select muscle on the skin surface where the lowest possible electrical stimulation will produce a minimal muscle twitch. The main drawback of this configuration is that it does not take full advantage of the differential amplifier design to reduce the unwanted noise in the EMG recordings.

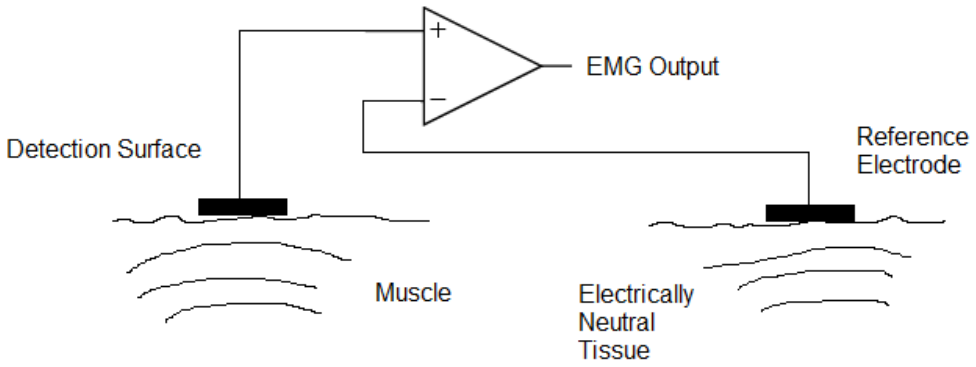



Figure 6-7: EMG Signal Extraction in Monopolar Configuration [26]

### 6.1.2.2 Bipolar Configuration

Bipolar also uses three electrodes E1, E2 and Ground. E1 and E2 are placed over the muscle at a certain distance of about 5 to 20 mm apart. Ground is placed on a bony prominence typically near E1 and E2.

For bipolar configuration, select a large enough muscle on the skin surface with lowest possible movement. This method overcomes the shortcoming of monopolar by taking the full advantage of amplifier circuitry that is designed to minimize unwanted interference signals from electromagnetic fields in the surrounding environment. However, the amplitude and frequency are largely dependent on the inter-electrode distance which sometimes is not easy to work with.

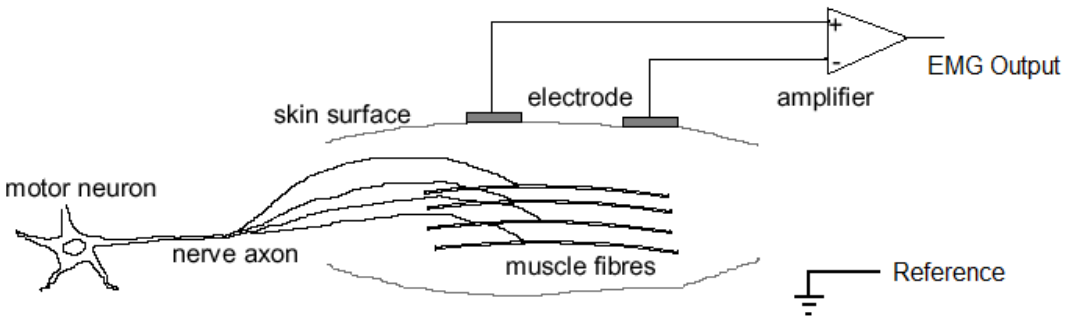



Figure 6-8: EMG Signal Extraction in Monopolar Configuration [26]

### 6.1.3 Electrode Leads

Electrode leads are any set of wires that has the sole responsibility to transfer the charges induced on the electrodes to a signal acquisition system which has an amplifier (normally an Instrumentation Amplifier) at the front end. Simply, electrode leads are specialized cables designed to conduct electrical signals with minimum losses and distortion. Since signal from the electrode is fed to the amplifier through electrode leads, they are also termed as input leads. As the input leads offer a finite resistance, there will be some degree of voltage drop between the electrodes and amplifier resulting in loss of signal.

From ohm's law,

$$V(drop) = I(cable) \times R(cable) \quad 6.2$$

Resistance is a function of the conductivity of the material (σ), length (l) and surface area (A) which is given by,

$$R = lA \quad 6.3$$

Of these three factors, length is the most critical because it can change to the greatest degree and is under control. Keeping the length of the input leads and all cables as short as possible will minimize the voltage drop.

A signal amplitude ($V_{in}$) is attenuated to differential voltage at the amplifier ($V_{out}$) by the electrode leads. Thus, the Attenuation (A) can be calculated as:

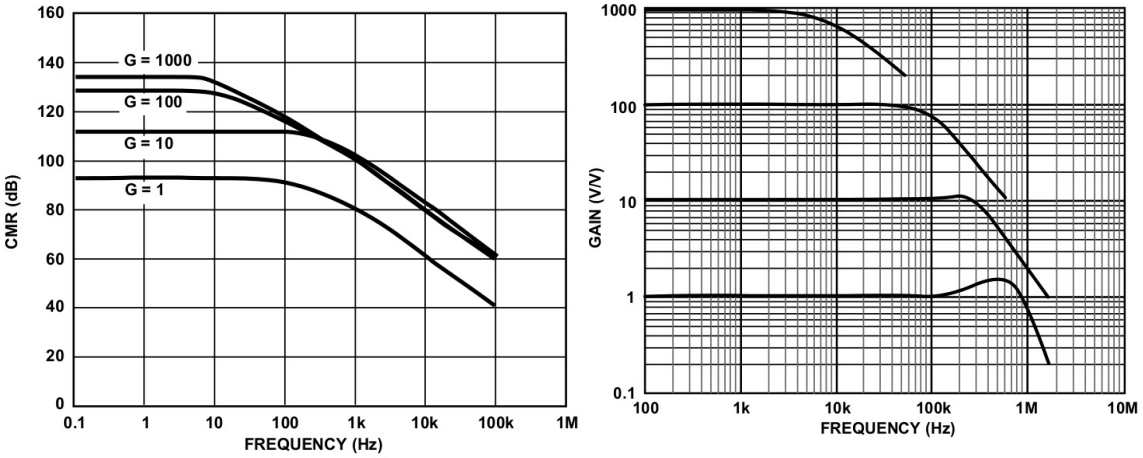
$$A = -20 \log\left(\frac{Vout}{Vin}\right) \quad 6.4$$

The most sensitive part in the EMG system design is the path between the electrodes and the amplifier because it is where the EMG signal has the lowest voltage level and is most vulnerable to noise and interference pickup. The longer the signal has to travel, the more interference and noise get coupled electromagnetically. EMG signal for intended purpose

varies from 1 Hz to 500 Hz in frequency and is not susceptible to attenuation loss due dielectrics at high frequencies (above 1MHz). However, electromagnetic interference does occur and hinders the quality of propagating signals. This can be avoided by the use of shielded cables. Shielded cables are composed of three layers. A signal-carrying conductor at the center is covered by a flexible insulating layer, which is then surrounded by a braided metal sheath. Shielded cable acts as a Faraday cage to reduce electrical noise from affecting the signals. It also minimizes capacitive coupled noise from other electrical sources.

### 6.1.4 Amplifier

Instrumentation amplifiers amplifies the weak signal from the muscles and make it detectable for the microcontroller which extract the data and send it for further processing. The voltage signal is of a magnitude range of 10 microvolts, which is very small for filtering and feeding to ADC. The instrumentation amplifier is a type of differential amplifier that eliminates the use of input impedance matching also rejects superimposed noise and interference noise. It provides very low DC offset voltage, low noise, very high open-loop gain, very high common mode rejection ratio (CMRR) and very high input impedance. AD620 has a bandwidth of 120 KHz with a gain range of 1 to 1000, settling time of 15 µs and 100 dB min Common-Mode Rejection Ratio (CMRR). The response of CMRR and gain of AD620 with respect to frequency is as shown in the figures below.




(a) (b)

Figure 6-9: a) CMRR vs. Frequency, b) Voltage Gain Vs. frequency [27]

The Gain of AD620 can be adjusted by placing a resistor of suitable value, which can be determined by the following equation:

$$R_G = \frac{49.4}{G - 1} K\Omega \qquad 6.5$$

Where G is the required gain or gain of the instrumentation amplifier and $R_G$ is the required resistor for the Gain.

AD620 and OP37AJ amplifier ICs are used along with resistors and capacitors for signal amplifying and filtering. The signal is first pre-amplified with AD620 instrumentation amplifier and low frequency noise is eliminated. Using filter after 1st stage amplification blocks low noises on further amplification. Amplifier OP37AJ amplifies the signal to higher strengths and then low pass filter is introduced to reject high frequency noises. Finally, another OP37AJ amplifier amplifies the signal to produce output in the level of volts. Gain of amplifiers and frequency of filters can be set using resistors and capacitors, which provides flexibility for similar kinds of varying signals. [9]

The response curve of amplifier OP37 is as shown in the figure below.

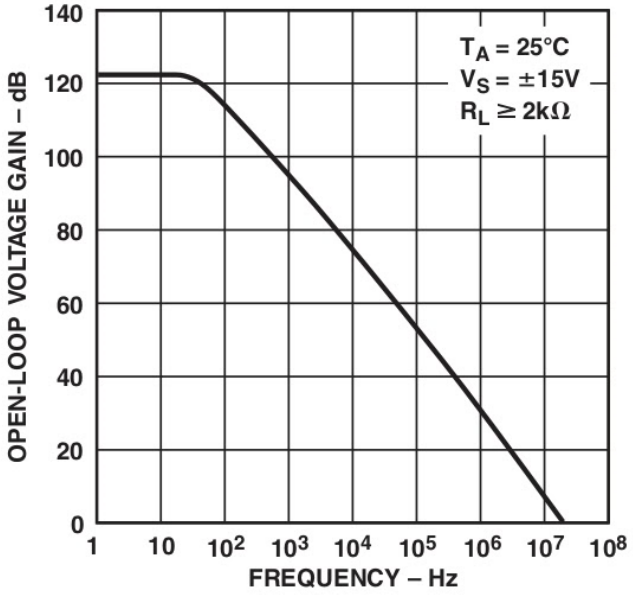



Figure 6-10: Frequency Response of OP37G [28]

### 6.1.5 Filter

EMG signals have wide range of frequency ranging from 1 Hz to 5,000 Hz [13]. However, the desired articulated EMG signals ranges from 1 Hz to 100 Hz which is associated with noises such as line noise, ECG artifacts, cross-talks, blood flow in the muscles, aberrant signals from the central nervous system and so on. Thus, a suitable band pass filter needs to be designed having cut-off frequencies at 1 Hz and 100 Hz. That said, the output response of the filters other than Butterworth filter contains ripples in the pass band which distorts the overall output of the filter. In expense of the flattest response of the pass band, it has a wide transition band. To overcome this drawback, the roll-off factor needs to be higher which ultimately implies higher filter order which can be calculated as given by the equation 4.14.

Along with the selection of the pass band, significant attenuation also occurs in passive filters. Active filters, in other hand, provide gain to the signal along with filtration which is most suitable for low frequencies. Band pass filtration is achieved cascading a passive high pass for higher frequencies and an active low pass filter for lower frequencies.

### 6.1.6 Arduino

The signal obtained is in analog form which needs to be converted to digital using the in-built ADC of Arduino. It consists of a 6 channel ADC with a resolution of 10 bit. The type of ADC in Arduino is Successive Approximation Register (SAR) which maps input voltage between 0 to reference volts into integer values between 0 and 1023 but does not sample in negative domain. By default, the reference voltage is 5 volts in Arduino.

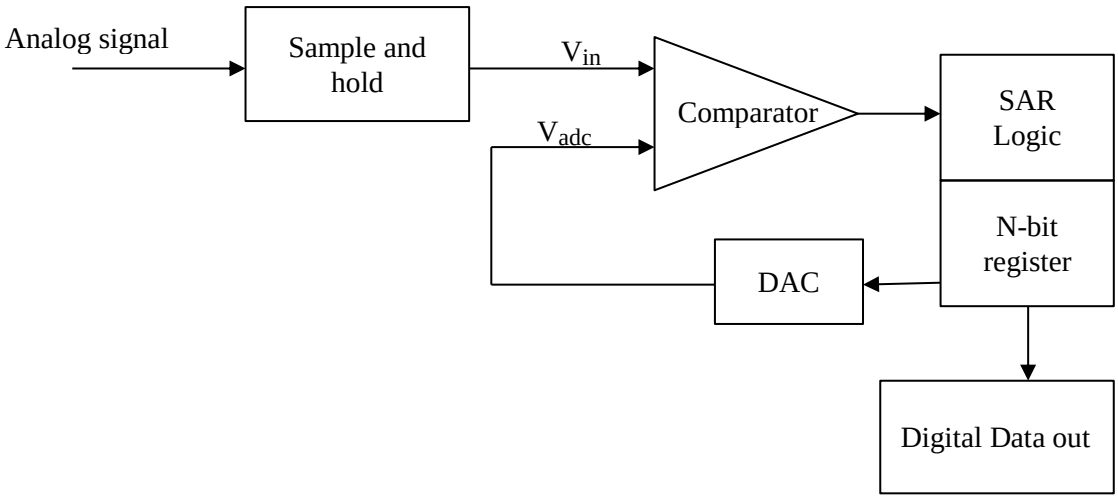



Figure 6-11: Block Diagram of SAR ADC

SAR ADCs have four sub-circuits; sample and hold, analog comparator, digital to analog converter and successive approximation register. The analog signal is stored by the sample and hold circuit on the hold capacitor. This voltage is compared with the voltage given by DAC. At start the MSB of the DAC is set. After the comparator output settles, the SAR resets the MSB if the DAC output is greater or keeps it set if the output is smaller. This binary search continues until every bit in the register is tested. The resulting DAC is a digital approximation of the sampled analog voltage. This value is then output by the ADC at the end of the conversion.

The voltage resolution of the ADC of the Arduino is given by,

$$Q = \frac{V_{ref}}{2^n} \tag{6.6}$$

Where $V_{ref}$ is the reference voltage or full scale voltage range and n is the ADC's resolution in bits. This value is equal to LSB. The maximum error is also 1 LSB which ranges from 0 to 1 LSB range known as "quantization uncertainty". As there are many ranges of analog input values for the given code. The maximum quantization uncertainty is the quantization error. The value sampled by the ADC is affected by the quantization error but with increased resolution it can be minimized.

The digital values are transmitted to the computer via Serial communication protocol. The Arduino UART is set to a certain baud rate which is also equal to the baud rate of the receiver. The Data is converted to a packet framed with the necessary signal or bits. The Packet consists of one start bit, data frame of 5 - 9 data bits, one parity bits and one stop bit.

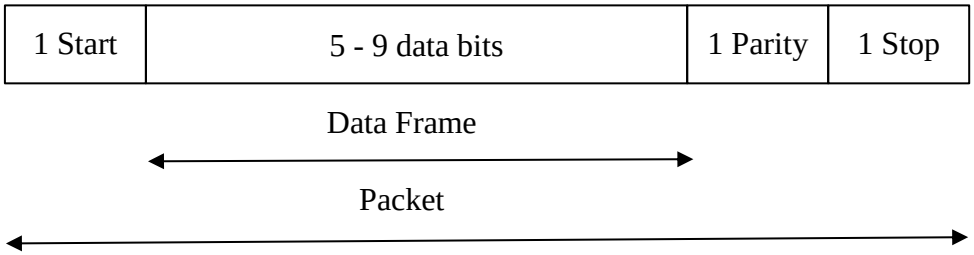



Figure 6-12: Packet of Serial Data

The data from the Arduino is stored in the serial buffer of the receiver which it uses by checking the parity of the received packet.

### 6.2 Software Platforms

This portion includes the requirement analysis of development environments, libraries and circuit designing platform.

#### 6.2.1 Arduino IDE

The Arduino IDE is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards but also, with the help of third-party codes, other vendor development boards. The source code for IDE is released under the GNU (General Public License).

#### 6.2.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python is also vastly used for data processing. Python has provided libraries like SciPy, pyEEG, pySpACE, bioSPPy for pre-processing of signals. Further processing of signals is done by using libraries like sklearn and librosa. Python also incorporates highly optimized open-source libraries like tensorflow, numpy to affiliate model development and implementation. Python also provides GUI features using PyQt, it is a cross-platform GUI toolkit Qt, implemented as a Python plug-in. It provides an easy and effective platform for building GUIs.

### 6.2.3 KiCad

KiCad is a free software suite for electronic design automation (EDA). It facilitates design of schematics for electronic circuits, converting them to printed circuit board (PCB) designs along with circuit simulation. It has built-in footprints of different electronic components. Footprint of new components can also be designed manually.

## 6.3 Speech EMG Dataset

The EMG-UKA is a corpus of synchronous EMG and acoustic recordings of continuous speech collected for the purpose of subvocal speech recognition [25]. The dataset contains citations of at least 50 different sentences from an English News Broadcast per session in three modes of speaking: Audible, Whispered and Silent.

Audible speech is normally spoken speech with normal voicing and intonation. It is recorded using standard close-talking USB microphone sampled at 16KHz. Whispered data is a speech emphasizing breath rather than vibration of the vocal tract. It is recorded using both the microphone similarly to the audible signal and EMG electrodes similarly to the silent speech. Silent speech is the speech with no sound while performing normal articulatory movements. It is recorded using the 7 channel EMG electrodes at a sampling frequency of 600Hz. Channel 7 is just a marker signal that is used to synchronize different speaking modes in the data. The articulation muscles for extraction of these EMG signals are Levator Anguli Oris, Zygomatics Major, Platysma, Depressor Anguli Oris, Anterior Belly of Digastric and Tongue. It should be noted that the data is collected from a group of 4 speakers in sessions that are either multi-modal or single-modal. Here, multi-model

specifies that the data is collected in two or three different modes at the same time using both the microphone and the EMG electrodes.

Table 6-1: Description of EMG-UKA Corpus

| Mode | Sampling Rate (Hz) | Channels | Length (hh:mm:ss) | Speaker Count | Session Count |
|---|---|---|---|---|---|
| Audible | 16000 | 2 | 01:52:24 | 4 | 6 |
| | 600 | 7 | 01:08:16 | 4 | 6 |
| Whispered | 600 | 7 | 00:21:47 | 4 | 6 |
| Silent | 600 | 7 | 00:22:21 | 4 | 6 |

The dataset arranges the audio data in ".wav" format and respective EMG data in ".adc" format. The corresponding sentence of the audio and EMG is transcribed in a sub-folder named as "Transcripts". Since there is some offset between audio and EMG signal, the dataset also provides the offsets and alignments of the uttered words. Moreover, the transcribed words are further broken down into their phonemes. The speakers for the dataset collection were all non-native speakers of English but were instructed well for clear pronunciation of each words and were between the age of 24 and 30 years old. Out of 4 speakers, 3 were male and 1 was female. [25]

## 7. SYSTEM ARCHITECTURE AND METHODOLOGY

The architecture of the designed system has been represented graphically using figures and functional blocks. Signal extraction, amplification and processing mechanisms are illustrated in details along with the extraction of features and machine learning algorithms.

### 7.1 System Block Diagram

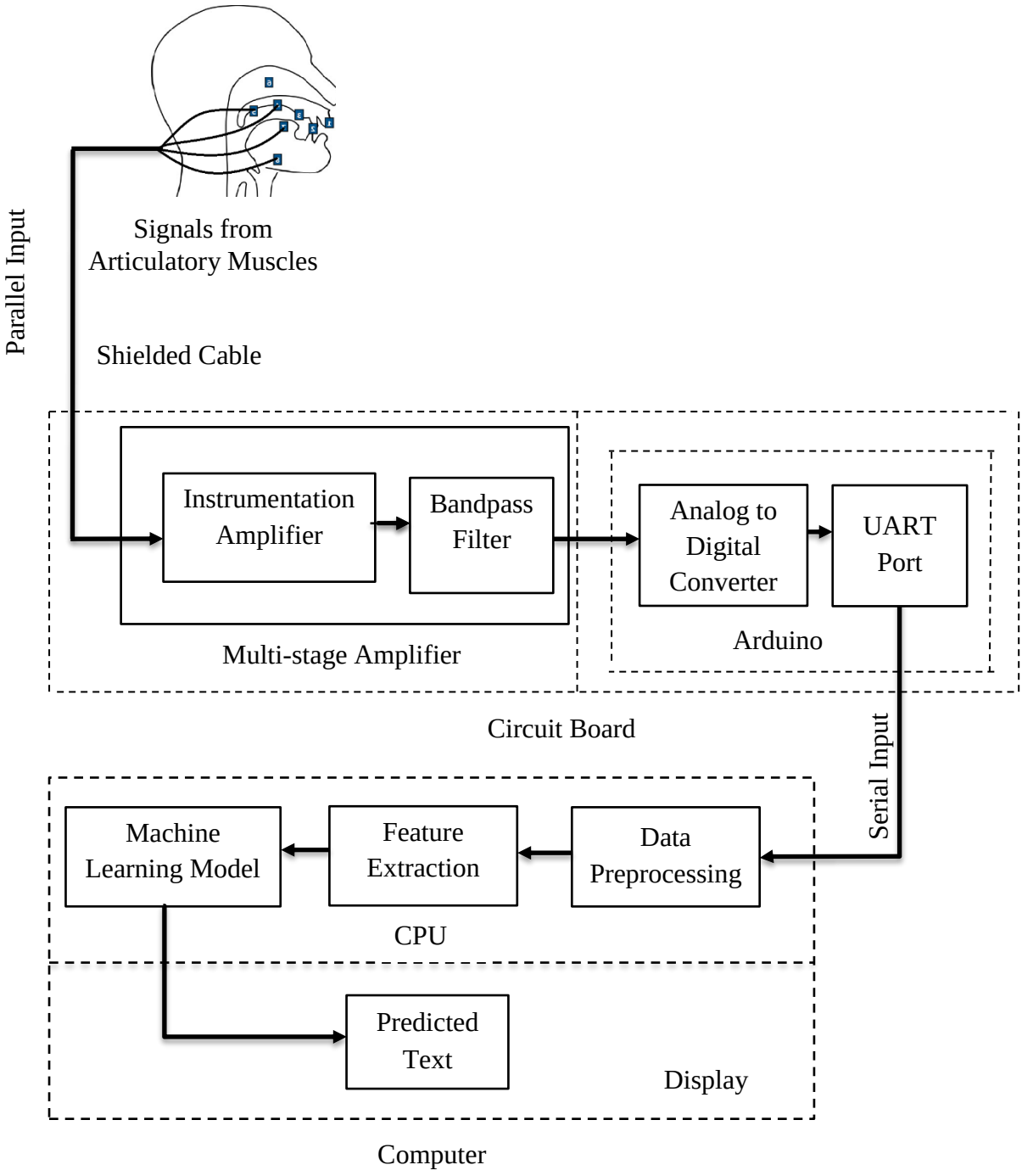



Figure 7-1: System Block Diagram

### 7.2 Electrode Placement and Signal Extraction

For the extraction of EMG signals, the electrode placement on the muscles are as shown in the figure 7-2.


Figure 7-2: Placement of Electrodes

The placement of electrode is based on the information from the table 3.1. The signals from the Ag-AgCl electrodes placed on the respective muscles were extracted and fed to the next stage.

Table 7-1: Electrode and Their Respective Muscle

| EMG Channel | Muscle name |
|---|---|
| 1 | Anterior belly of the digastric |
| 2 | Zygomaticus Major |
| 2,3 | Levator Angulis Oris |
| 4 | Platysma |
| 5 | Unused |
| 6 | Depressor Anguli Oris |

The channel 2 and 6 are derived bipolarly. The other channels are derived unipolar with reference electrodes on the nose at point 1 (for channel 1) and behind the ears (for channels 3, 4 and 5).

### 7.3 Signal Amplification and Filtering

The amplification and filtering of the signal that is extracted from the articulatory muscles is performed in the multi-stage amplifier block. Further classification of the block is shown in figure 6-3. The instrumentation amplifier AD620 amplifies the signal obtained from electrodes with low noise and high CMRR. The output is then fed to the second order high pass filter. As this filter has a cut-off frequency at 1 Hz, only the signals above 1 Hz are passed while other signals are filtered out. The signals are then forwarded to the input of amplifier OP37 which amplifies the signal with a calculated gain. The signal is finally passed to the second order low pass active filter having cut-off frequency at 100 Hz. This block passes the signals below 100 Hz along with their amplification. Thus, as a whole, the high pass and low pass filter in cascade give a response equivalent to a band pass filter with pass band of 1 Hz to 100 Hz.

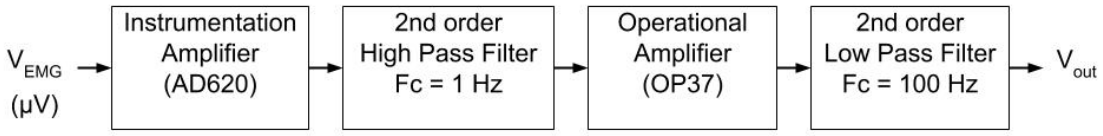



Figure 7-3: Block Diagram of Amplifier and Filter

### 7.4 Analog to Digital Conversion

The signal obtained from the multi-stage amplifier section is in analog form. This ADC block converts the analog signal to digital signal. The ADC samples the data at the sampling frequency of 600 Hz. It maps input voltage from 0 to 3.3 volts into integer values between 0 and 1023. The signal after being converted to digital is sent to a computer for further processing.

### 7.5 Serial Communication

The digitized signals are required to be transmitted to a PC for further processing. The Arduino sends the digitalized data through USART. The Arduino is connected to the USB port of a computer and the data is accessed serially. The communication is synchronized at a baud rate of 115200 and the latency at both the sending and receiving end is addressed

by the serial buffers. The accessed data is stored in the computer and further processing is done in the computer.

### 7.6 Signal Preprocessing

The digital signal converted by ADC needs to be further processed before it can be fed to a classifying neural network. The received signal may contain missing or erroneous data points and other inherent noises. Any faulty data points may cause the neural network to fail to generalize leading to a higher error in classification. The essence of preprocessing the signal is further illustrated in the following literature.

#### 7.6.1 Signal Smoothing

Signal smoothing helps eliminate the erroneous data points, some inherent noises and also help extrapolate missing data points in a signal. It does so by incorporating the previous data point and considers that the consecutive data point is representative of the previous data point. It is a type of mathematical convolution and typically implemented on a single dimensional signal model. It generates a time series data constructed by taking averages of several sequential data points of another time series. Moving average algorithms namely Simple Moving Average, Exponential Moving Average, Weighted Moving Average and Cumulative Moving Average are applied to achieve signal smoothing.

#### 7.6.2 Windowing

When FFT is used to analyze the frequency domain of a signal, the signal is biased on a finite set of data. When the number of periods is not an integer, the end-points of the FFT become discontinuous due to sharp transitions. These discontinuities show up in the FFT as high-frequency components, sometimes much greater than the Nyquist frequency, which are not present in the original signal. Thus, the spectrum of the signal will be smeared due to the spectral leakage. This causes fine spectral lines to spread into wider signals. Windowing reduces the amplitude of discontinuities at the boundaries of each finite sequence. It multiplies the time record by a finite-length window with an amplitude

that varies smoothly and tapers towards zero at the edges. This results in continuous waveform without sharp transitions.

Furthermore, large signals are difficult to analyze statistically as statistical calculations require all the points to be available for analysis. So small subsets of the whole data are analyzed through the process of windowing. It splits the input signal into sufficiently small segments such that the properties of the signal are time-invariant within that segment. It reduces the time domain information and thus resolution in the frequency domain is reduced which implies that there is reduced leakage of spectrum. Thus, before extracting any features in the frequency domain, the time variant data is windowed. It alters the spectral properties of the signal, but the change is designed such that its effect on signal statistics is minimized. All the data points outside the window is truncated while the cut-off points at the ends of the sample will introduce high-frequency components [31]. Based on the different mathematical implementations, windows may be of various types such as Rectangular, Hamming, Blackman, Flattop and Gaussian and so on.

## 7.7 Extraction of Signal Features

After the EMG signals are processed, they need to be further translated to features that are the actual data contained in a multi-channel EMG signal. Since EMG signal is very different from the speech signal, it is necessary to explore feature extraction methods that are suitable for EMG to text conversion. However, due to lack of discrete word level dataset, the continuous speech signals from the EMG-UKA Corpus needed to be trimmed to form unique words. Again, before working with raw EMG signals, verifying that discrete word creation from the continuous speech stayed valid was essential. Thus, for this reason and the fact that there is no publicly available speech EMG dataset, the parsed word EMG data from the EMG-UKA Corpus was used. Many techniques can be followed for extracting features suitable for audio signals which can be carried over to the EMG signal also and these methods can be described as follows:

### 7.7.1 Temporal Features

Temporal features are the time domain features calculated over a window of fixed size that traverses over all the samples in the time domain. Typically, for a sEMG signal,

window size of greater than 100 ms and less than 250 ms is used [13] but in cases such as in this project, the EMG signals somewhat follow auditory temporal properties and thus features like Zero Crossing Rate, High Frequency Signals, Rectified High Frequency Signals, Frame Based Power and Double Nine Point Average with a window size of 10ms to 60 ms can be used [29]. From the mentioned features, zero crossing rate, average rectified value, average power and root mean square were selected as they showed significant improvement in the performance of the Neural Network.

### 7.7.2 Short Time Fourier Transform

Time-frequency analysis of a signal is typically required to characterize the non-stationary phenomena of signals. The frequency components can be revealed by Fourier transform in chunks of data using sliding windowing technique which is known as Short Time Fourier Transform (STFT). Each transformed complex chunk is added to the matrix which records the magnitude and phase for each point in time and frequency but in doing so all the time related information will be lost. Due to this significant shortcoming of STFT, it is not desirable for wide-band and ultra-wide-band signals, where low spectrogram resolution is observed. However, the selection of appropriate window size for narrow-band signals such as audio signals, EMG signals, etc. can ideally ensure that the input signal falling within the window remains stationary. But use of very small window size cannot localize the frequency domain. For wide-band signals constant Q transform (CQT) can be used which gives a frequency resolution that depends on the geometrically spaced center frequencies of the analysis window.

### 7.7.3 Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients (MFCC) are the coefficients that are based on a cepstral representation of audio signals in frequency bands that are equally spaced on a Mel scale that actually approximates the response of a human auditory system. Although they are mostly used in Automatic Speech Recognition, music genre classification, and many other fields, the fact that the MFCC represent differences in different frequencies on a scale that is easily relatable by auditory system, they can have very significant contribution to represent the EMG signals more accurately. Moreover, MFCC also takes

into account the dynamic aspect of an EMG signal and the variability in the length of EMG signal in time domain. [30]

Table 7-2: Table of Extracted Features

| **Temporal Features** | **Spectral Features** |
|---|---|
| Average Rectified Value | Mel Frequency Cepstral Coefficients (MFCC) |
| High Frequency Signals | Short Time Fourier Transform (STFT) |
| Double Nine Point Average | |
| Frame Based Power | |
| Zero Crossing Rate | |

### 7.8 Machine Learning Models

After the extraction of features, they need to be fed to a recognition model which classifies the features to their corresponding word (or letter) labels.

#### 7.8.1 Supervised Models

Supervised machine learning model includes Multi-Layer Perceptron, Convolutional Neural Network and K-Nearest Neighbor algorithms whose model architectures are described below:

##### 7.8.1.1 Multi-Layer Perceptron

For fast prototyping and verifying the credibility of the extracted features, a simple Artificial Neural Network (ANN) stands to be very viable option for most machine learning projects. Multilayer perceptron (MLP) is a class of feedforward ANN that learns a function f(X): $R^m \rightarrow R^n$ by training on a dataset, where m is the input dimension and n is the output dimension. Given a set of features $X = x_1, x_2, ..., x_m$ and target y, it can learn a non-linear approximator for either classification or regression. It consists of at least

three layers of nodes: an input layer, a hidden layer and an output layer with each neuron with linear or non-linear activation function. MLP usually means fully connected network i.e. each neuron in one layer is connected to every neuron on the next neuron.

The architecture of the designed MLP network is as shown in Figure 6-4. It consists of an input layer with a number of neurons equal to the size of input feature tensor, three hidden layers with linear and ReLU activation functions consecutively and finally an output layer with 10 neurons. The feature tensors are 3 dimensional and need to be flattened into 2 dimensional tensors, which is done by flatten activation function of the input layer. The input to the first hidden layer (H1) with linear activation function must be a 2-dimensional tensor which was obtained previously from the flatten activation function. The output of H1 forms H2 with 64 neurons and has ReLU activation function. Similarly, H2 feeds the final hidden layer H3 (with ReLU activation function) and yields the output layer with 10 neurons mapped from 64 neurons by linear activation function of H3. All the layers are fully connected except the input and H1 which is mapped one-to-one by flatten activation function.

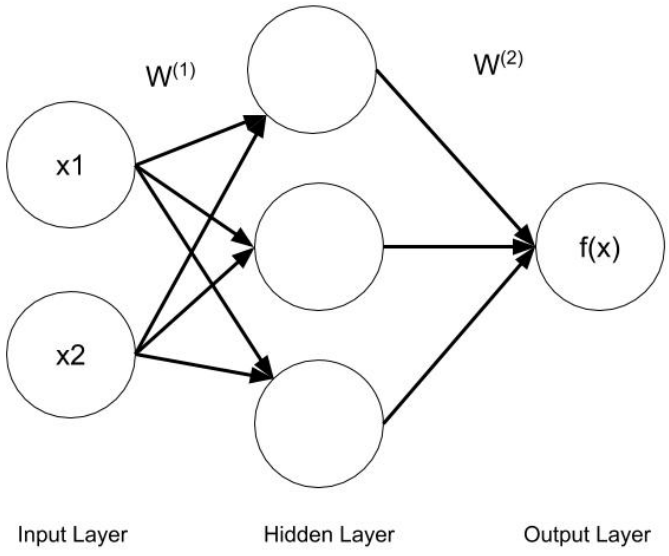



Figure 7-4 : A Simple Three Layer MLP Network

Let f(x) be the output vector of the network shown above, $w^{(1)}$ and $w^{(2)}$ be the weight matrices, $b^{(1)}$and $b^{(2)}$ be the bias vectors and G and s be activation functions then the output of the MLP network shown above can be given by,

$$f(x) = G\left(b^{(2)} + w^{(2)}\left(s\left(b^{(1)} + w^{(1)}x\right)\right)\right) \quad 7.1$$

$$f(x) = G(b^{(2)} + w^{(2)}h(x) \quad 7.2$$

The vector $h(x) = (s(b^{(1)} + w^{(1)}x))$ constitutes the hidden layer.

#### 7.8.1.2 Convolutional Neural Network

Convolution Neural Network (CNN) is basically a regularized version of a multilayer perceptron. MLPs are more prone to overfitting due to their fully connectedness and thus require regularization. CNNs regularize the data using convolution principle i.e. smaller and simpler patterns are used to assemble complex patterns over a hierarchical pattern of data. If f[n] and g[n] are two discrete time data, then the convolution of these two functions are given by:

$$f[n] * g[n] = \sum_{m \to -\infty}^{\infty} f[m]g[n-m] \quad 7.3$$

The set of smaller data points that is compared to input data is known as kernel. The concatenation of multiple kernels, each kernel assigned to a particular channel of input is known as a filter. The filter always has one dimension higher than that of the kernel. As shown in figure 6-5, during convolution operation, the kernel matrix slides over the input data as per the stride value. If the stride value is 1, the kernel moves by a single column of the input matrix. It then performs dot product within the sub-region of the input data and similarity between them is computed. Highest value of output, activation map is produced where the kernel is most similar with the portion of input that is being compared. Let, I(a) be the input and K(a) is the kernel, their convolution C(t) is mathematically defined as:

$$C(t) = \sum_{a} I(a)K(t-a) \quad 7.4$$

$$C(t) = \sum_{a} I(t-a)K(a) \tag{7.5}$$

Now, flipping can be done to get cross-correlation

$$C(t) = \sum_{a} I(t+a)K(a) \tag{7.6}$$

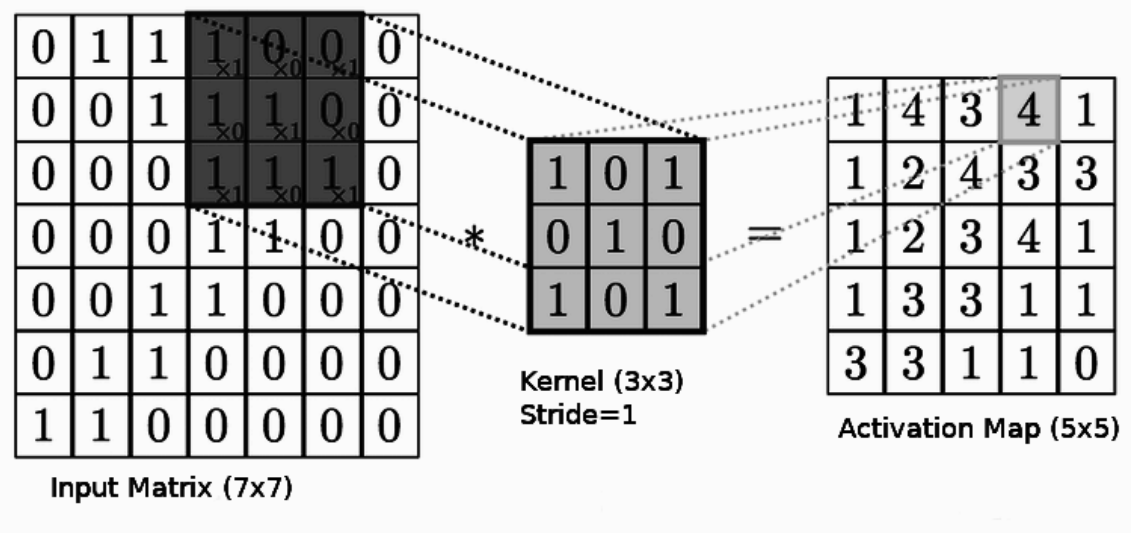



Figure 7-5: Convolution of 7x7 Input Matrix with 3x3 Kernel of Unit Stride

The convolution of these two data represent how the shape of one is modified by the other one. A basic CNN contains a convolution layer, a pooling layer, a fully-connected layer and sometimes dropout layer as well. Convolution layer computes the convolution between the input data based on the above equation 7.3. This layer can also be designed to compute cross-correlation instead of convolution which is basically the same as convolution, the only difference is they are opposite in signs. Cross-correlation being positive is preferred over convolution.

When dealing with high dimensional inputs, it is impractical to connect all the neurons of a layer to every neuron in the previous layer because such a layer wouldn't consider the spatial structure of the data. CNN exploits spatially local correlation by enforcing a sparse local connectivity pattern between adjacent layers which means each neuron is connected to only a small region of the adjacent layer. The extent of this connectivity is

determined by the hyperparameters; depth, stride and zero padding. They control the size of the output volume of the convolution layer. Depth of the output volume corresponds to the number of kernels to be used. The stride is the unit by which the kernel is to be slid over the input matrix. Zero padding allows to control the spatial size of the output volume by padding with zeros around the border [31]. The spatial size of the output volume (O) can be computed as,

$$O = \frac{W - K + 2P}{S} + 1 \tag{7.7}$$

Where W = input volume size or input dimension,
K = receptive field size of convolution layer or kernel size,
P = amount of zero padding,
S = stride

A Pooling layer (PL) is an effective way of nonlinear down-sampling. It has kernel size and stride as non-learnable parameters. The size of stride is the same as that of the kernel by default. The exact location of a feature is less important than its relative location with respect to other features. This layer progressively reduces the spatial size of the representation for latter layers, memory footprints and computation complexity of the network but adds no new parameters. It also contributes in controlling overfitting. Due to its destructiveness, PLs are very rarely used or discarded in case of very small dataset. Various pooling layers such as max-pooling, average-pooling, l2-norm pooling, etc. are used in a neural network. [32]

Mathematically, max-pooling function is defined as,

$$p_{i,m} = \max_{1 \le n \le G} q_{i,(m-1)*s+n} \tag{7.8}$$

Where G is pooling size and s is the shift size or stride size

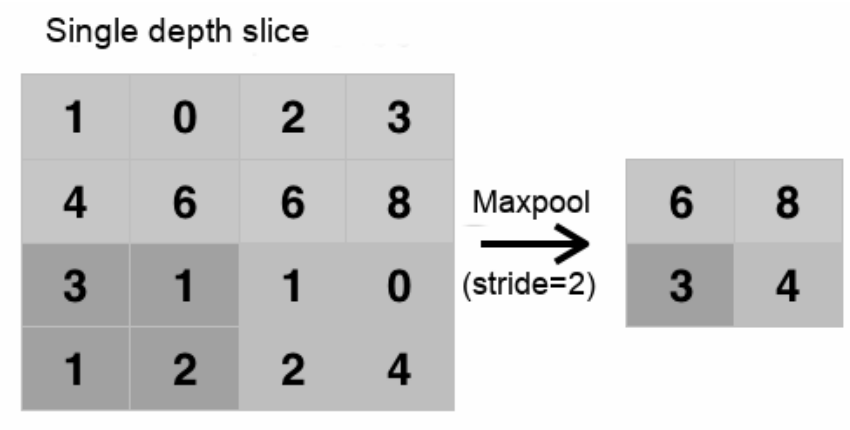



Figure 7-6: Max Pooling of a slice with a stride of 2 units

A fully connected layer is usually a linear layer while a dropout layer is a layer implemented for data regularization which randomly drops out some neurons of the layer on which it is implemented to avoid data overfitting. Neurons have connections to all the activations of the previous layer. Their activations can thus be computed as an affine transformation; preserving parallelism, with matrix multiplication followed by a biased offset.

#### 7.8.1.3 K-Nearest Neighbour

K-Nearest Neighbour (KNN) is a simplistic machine learning model that classifies a data instance according to its neighbours. This algorithm uses the similarity in features to predict the new data instances which is based on the plurality vote of its neighbours. The number of neighbours is assigned as K and is varied according to the data and classification task at hand. The data points for this algorithm needs to be normalized to obtain good classification as it is an instance based learning algorithm that predicts according to the local approximation.

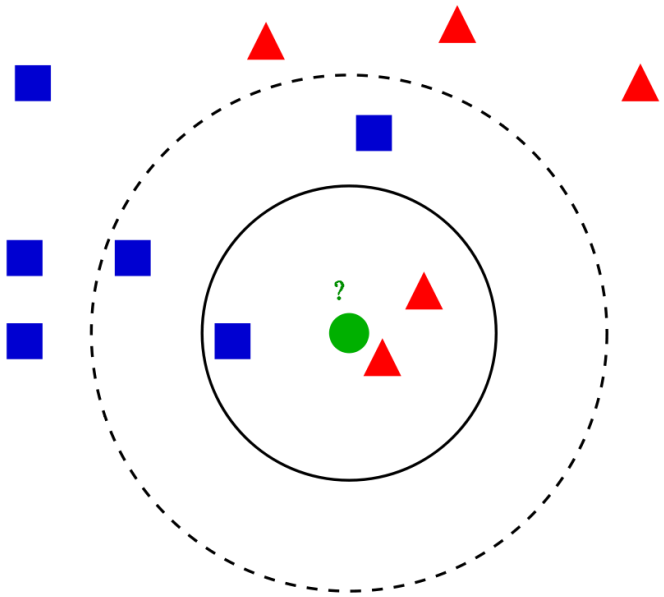



Figure 7-7: KNN Classification Model

In the figure 7-7 above, the working of KNN classification model is shown. A new data instance represented by the circle in the middle is to be classified in either of the two given classes represented by a triangle and a square. If the number of neighbours is set to 3, the algorithm looks at its three closest neighbours and since the two out of three neighbours belong to the class triangle, the algorithm classifies the input circle as a triangle. If the number of neighbours is set to 5, the algorithm looks at its five closest neighbours and since the three out of five neighbours belong to the class square, the algorithm classifies the input circle as a square.

### 7.8.2 Unsupervised Models

K-means Clustering is used as unsupervised learning model. The details about this model is described below.

#### 7.8.2.1 K-Means Clustering

K-means clustering is a method of vector quantization that segregates n feature spaces into k clusters in which each feature vector belongs to the cluster with the nearest mean or the cluster centroid acting as the cluster representative as shown in figure 7-10. K-

means is simply an Expectation Maximization (EM) algorithm applied to a particular Bayes model.

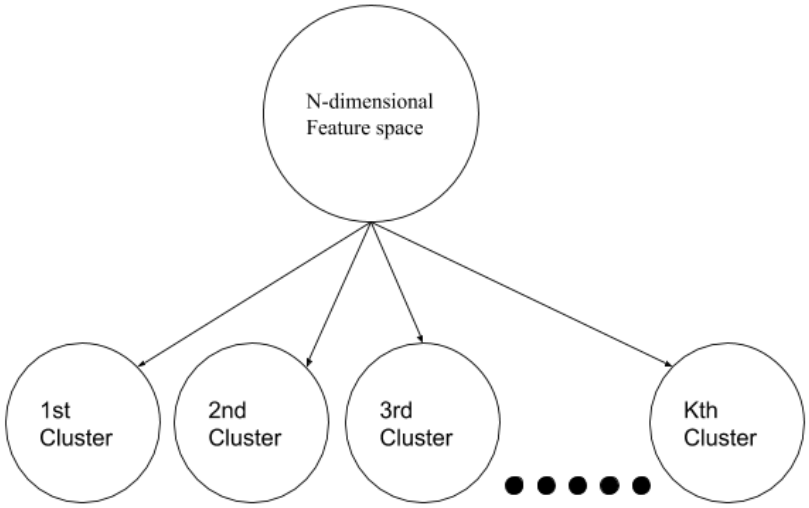



Figure 7-8: K-means Clustering Technique Applied to Bayes Net

**1. Expectation Step**

In estimation step, the two dimensional Gaussian centroid of the clusters are computed from the Gaussian mixture model (GMM) which is then utilized to calculate the sum of squared errors (SSE) as,

$$SSE = \sum_{i=1}^{n} ||x_i - \mu||^2 \qquad 7.9$$

This portion of the algorithm also calculates the posterior probability of the Bayesian net as

$$p(\theta|x) = \left[\frac{p(x|\theta)}{p(x)}\right] p(\theta) \qquad 7.10$$

**2. Maximization Step**

In this step the estimation computed in the E-step is locally optimized by minimizing the cost value. The Gaussian variance of the component is also calculated from the GMM.

$$Cost = \sum_{i=1}^{k} SSE \tag{7.11}$$

The overall flow of K-means algorithm can be recognized from the block diagram given below.

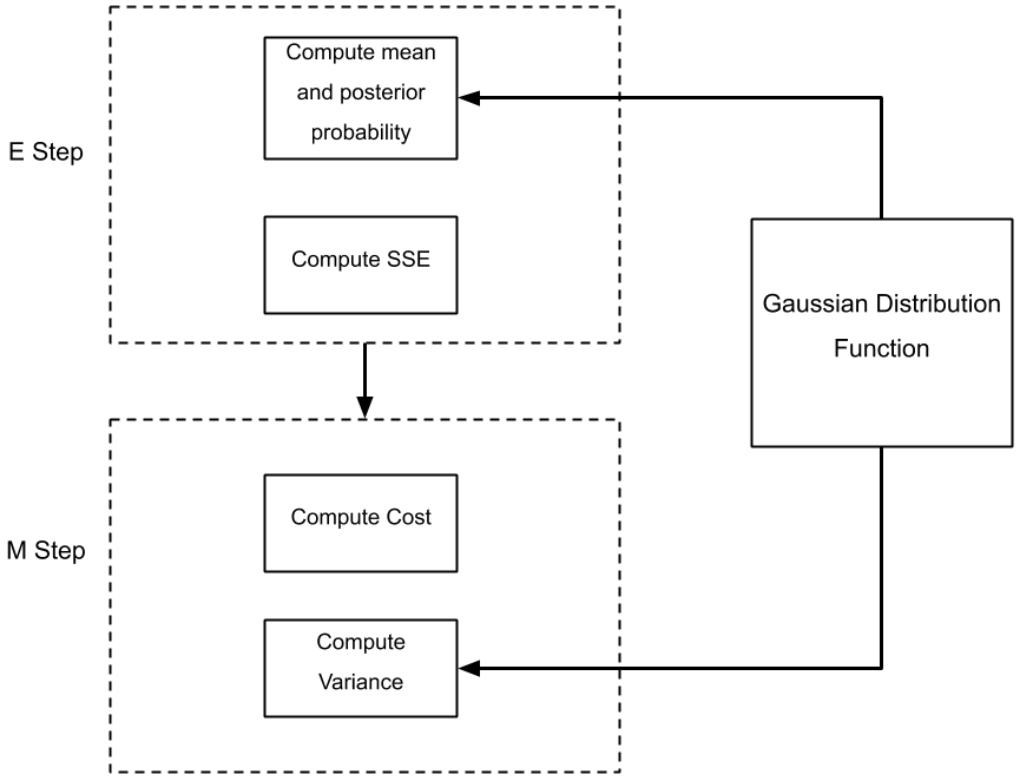



Figure 7-9: Block Diagram Showing Flow of K-means Algorithm

# 8. IMPLEMENTATION DETAILS

This chapter describes how the proposed systems were put into practice. The hardware and software discussed in the previous chapters explained how each of them worked using theoretical and mathematical approach but not how they were implemented in the project to meet the project goals.

## 8.1 Hardware Implementation

This section elaborates on how each of the components in hardware has been used, what kind of setup has been followed and how the designed circuit has been fabricated.

### 8.1.1 Parameter Calculation

The Instrumentation Amplifier AD620 is used as a pre-amplifier for the circuit. The EMG are weak and are in presence of common mode signals. So it is necessary to have a good CMRR to amplify the EMG signals and reject strong common mode signals. For EMG signals it is recommended to have CMRR greater than 90 dB. From the Figure 6-9: Typical CMRR vs. frequency curve of AD620, a gain of 10 provides CMRR greater than 90 dB steady within the frequency range of 1 - 100 Hz which is the working band frequency of the system. And from the Figure 8.1, voltage Gain Vs. frequency curve of AD620 the gain remains steady within the frequency range. The AD620 can be changed by changing the value of $R_G$. Equation 6.4 gives the relation of $R_G$ and gain of the instrumentation amplifier and the gain was set to 10.8 using a resistor ($R_G$) of value 5KΩ.

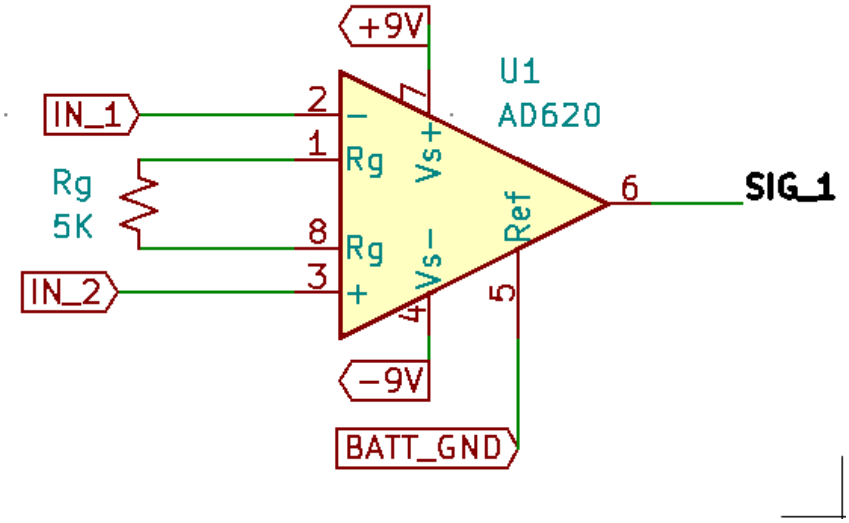



Figure 8-1: Instrumentation Amplifier

The small gain of about 10 was chosen so that noise signals of relatively low amplitude in comparison to the EMG signal do not get amplified to a significant level. This stage cannot avoid noise but on further filtering, the noise with less amplitude are suppressed very well or are more prone to attenuation. And upon choosing gain less than 10 the CMRR might not be sufficient to reject stronger common mode signals and also amplify the necessary signals.

The signal passes through a high pass filter with cutoff frequency of 1Hz. The equation 4-16 gives the relation of the frequency, resistors and capacitors from which the required values are obtained. But the Instrumentation Amplifier gives bipolar output so electrolytic capacitors are avoided which have the higher capacitance value instead ceramic capacitor is used which is non-polar. The ceramic capacitor has less value so a high value of resistor is chosen.

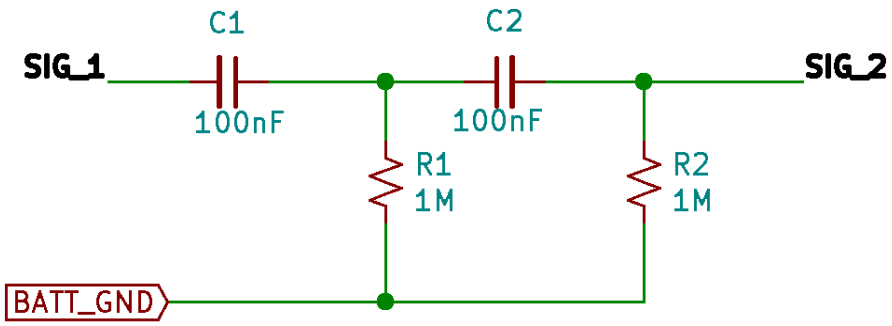



Figure 8-2: High Pass Filter

The signal from the high pass filter is then amplified by the non-inverting amplifier. The gain of the amplifier is controlled by the resistor R1 as shown below. The signal from the high pass filter is very low and with much experimentation the value of R1 was set to 470 KΩ resulting in a gain of about 471.

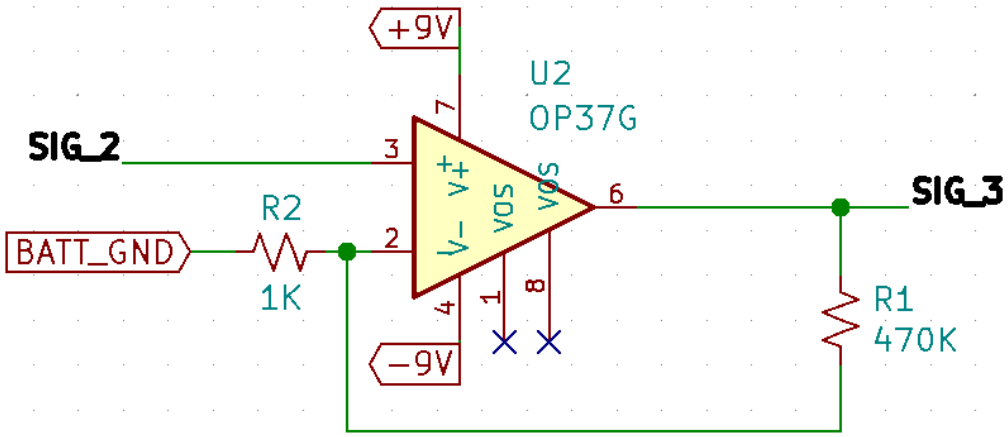



Figure 8-3: Amplifier

To create a band pass of 1 - 100 Hz, a low pass filter is added. The cut off frequency of the low pass filter is 100 Hz. It is an active low pass filter with a gain of 1.5. This is the value of $A_0$ referencing the equation 3.31. The gain is adjusted using the equation 3.19 which gives the relation of R4, R3 and $A_0$. The value of Q becomes 0.667 with the selected value, which is less than the value referenced at equation 3.32. The calculated values are tabulated in the table 8-1.

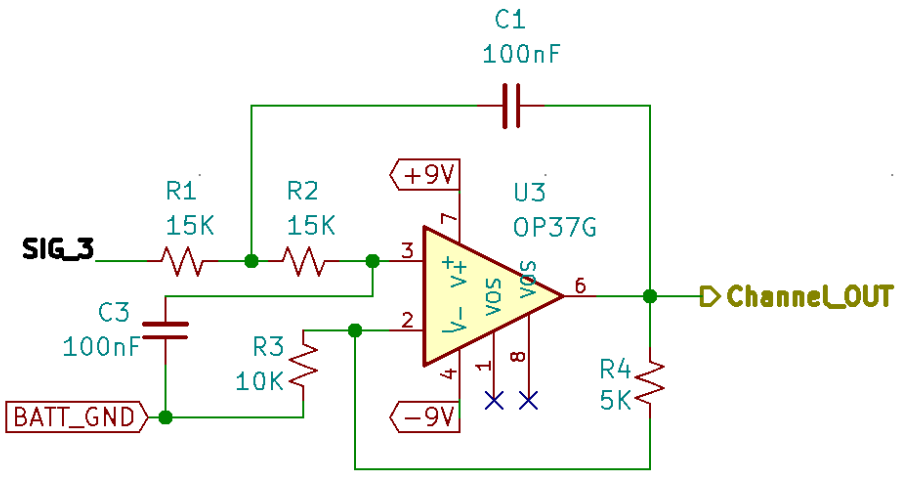



Figure 8-4: Low Pass Filter

The calculated values are tabulated below with the respective circuit above.

Table 8-1: Circuit Parameter Calculation

| Circuit Element | Resistor Values (KΩ) | Capacitor Values (nF) | Cut-off (Hz) | Gain |
|---|---|---|---|---|
| IA | Rg = 5 | - | - | 10.8 |
| HPF | R1 = 1000<br>R2 = 1000 | C1 = 100<br>C2 = 100 | 1 | 1 |
| Amplifier | R1 = 470<br>R2 = 1 | - | - | 471 |
| LPF | R1 = 15<br>R2 = 15<br>R3 = 10<br>R4 = 5 | C1 = 100<br>C2 = 100 | 100 | 1.5 |
| **Total Gain** | | | | 7630.2 |

### 8.1.2 Schematic Design

Schematic is a simple representation of a circuit design on a two-dimensional plane that shows the functionality and connectivity between the different components in the circuit. Utilizing the schematic, CAD software knows the inter-connection between the components, the types of components used and how the components have been used.

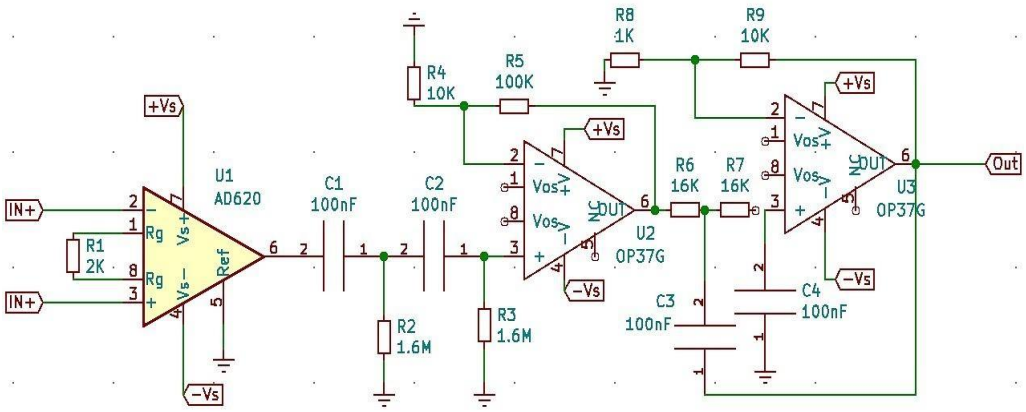



Figure 8-5: Branch Sheet of Designed Schematic

A multi-channel EMG data acquisition schematic has redundant circuit elements and circuit connections that makes schematic design unconventionally complicated and difficult to follow. To counter the redundancy and to keep the schematic design conventional, hierarchical sheet schematic design was followed. The multi-channel repetitive circuit elements were kept on a branch sheet that was referenced in the root sheet whenever it was needed as shown in figure 8-6.

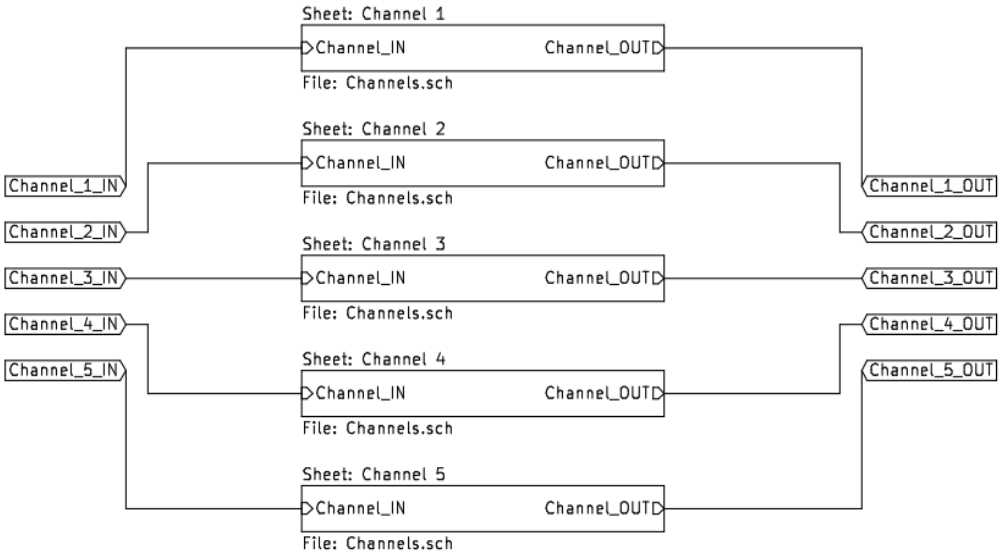



Figure 8-6: Root Sheet of Designed Schematic

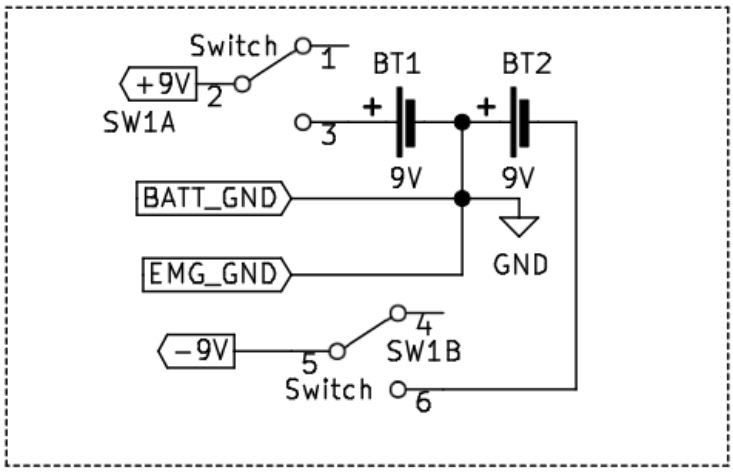



Figure 8-7: Schematic of Power Supply

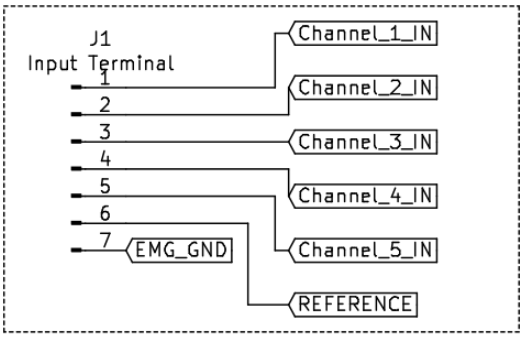



Figure 8-9: Input Channel Terminals

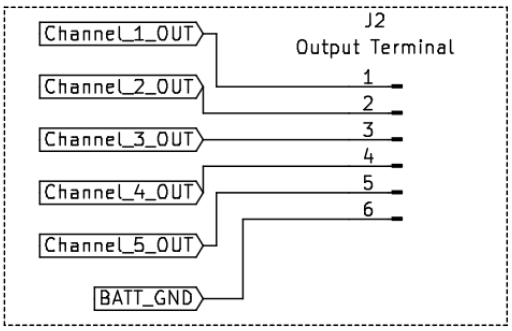



Figure 8-8: Output Channel Terminals

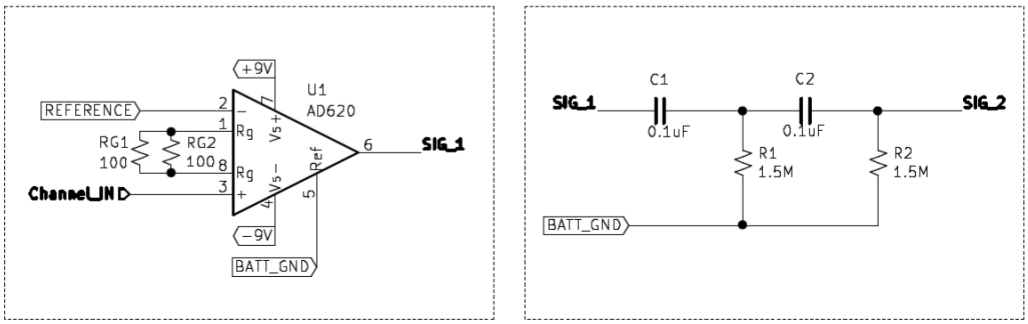



Figure 8-11: Instrumentation Amplifier and High Pass Filter Block

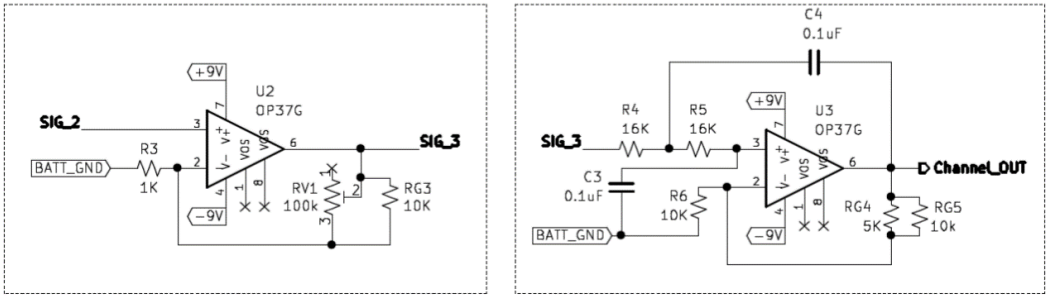



Figure 8-10: Amplifier (left) and Low Pass Filter (right) Block

### 8.1.3 Layout Design

Layout design refers to design of the printed circuit board taking in consideration all the aspects due to which the functioning of the system gets hindered. The layout design can be as important as the circuit design to the overall performance of the final system. The signals involved in a circuit could be analog or digital. It is advised to always isolate such circuitries to avoid interference between the two. The EMG acquisition hardware too has both the analog and digital part. The instrumentation amplifier circuit, the filter circuit and the amplifier circuit belong to the analog part whereas the Arduino belongs to the digital part. The designed PCB only hosts the analog circuits and the Arduino is kept separate with only necessary wires interconnecting them. Both the analog and digital part of the circuit is battery powered to avoid any ac coupling in the circuits.

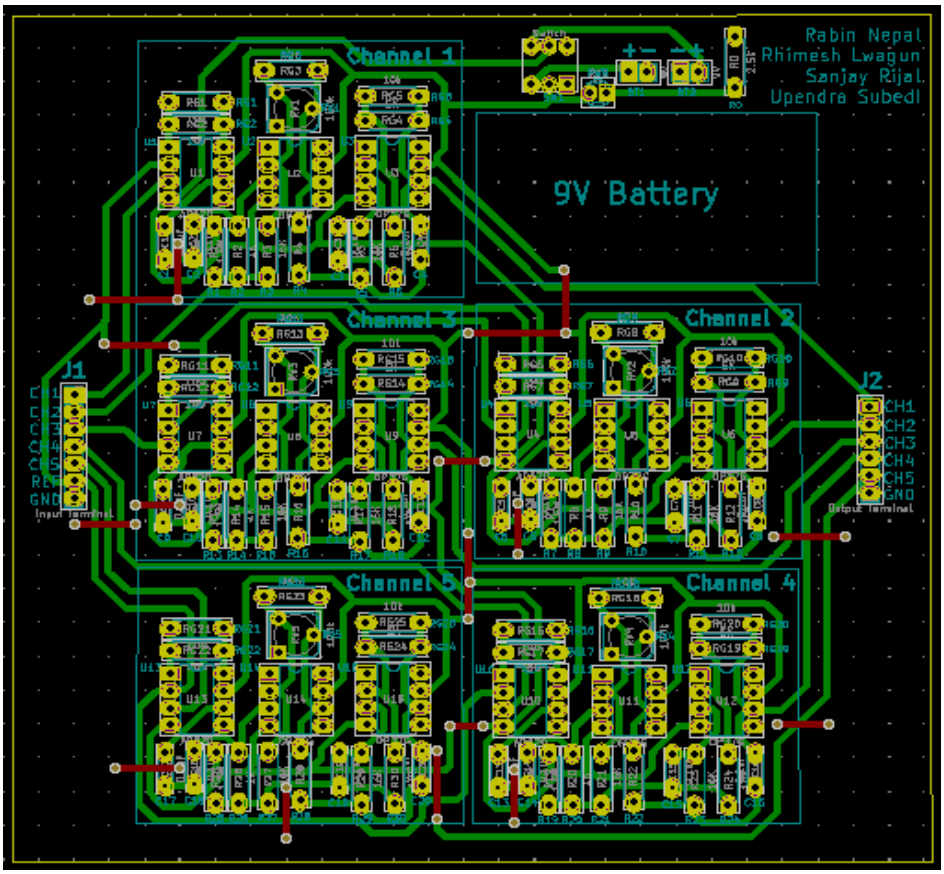



Figure 8-12: PCB Layout

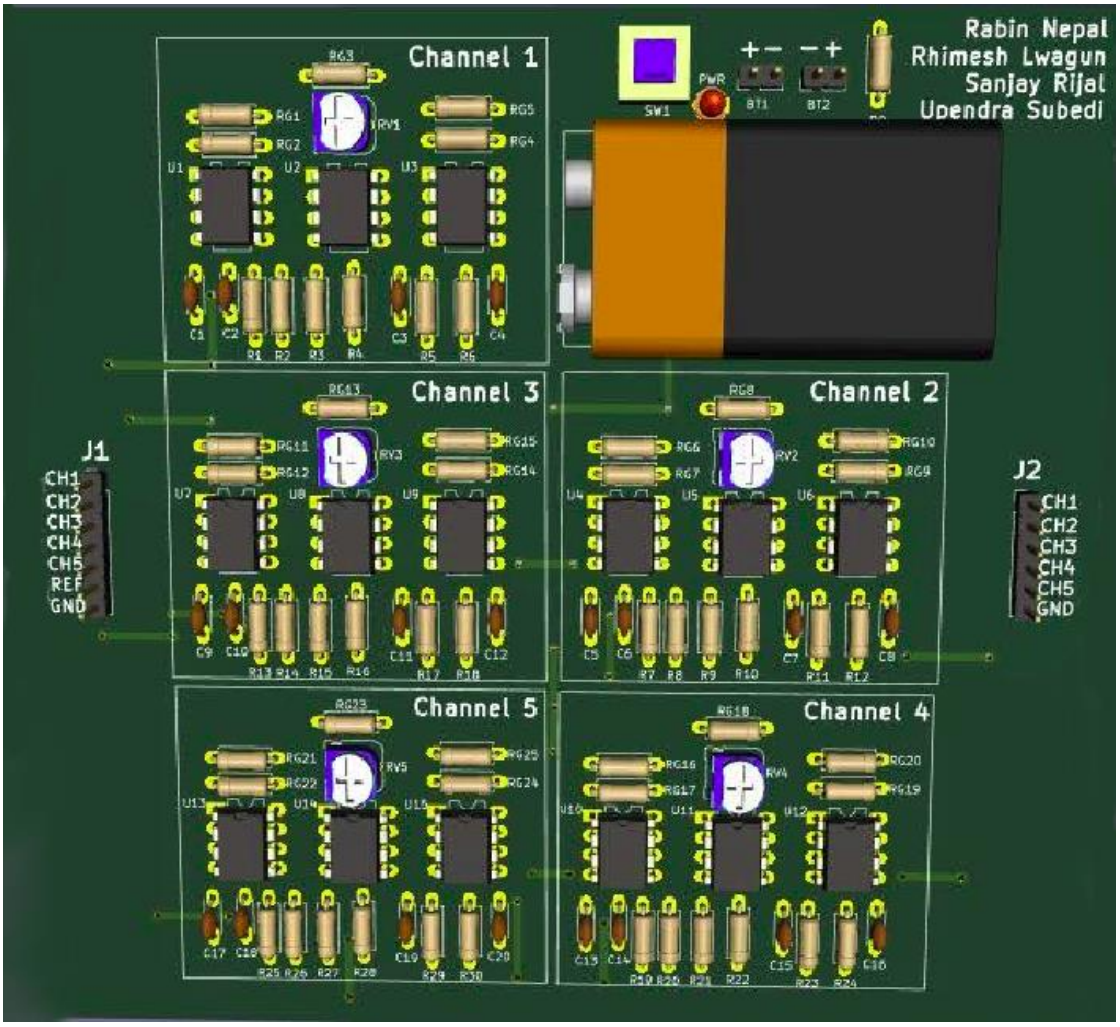



Figure 8-13: 3D View of Designed PCB

The copper trace width for the circuit is restricted to 1mm (40 mils) by the fabrication process involved. Any traces below 1 mm gets eaten away by the chemical solution along with the other unmasked copper regions. The fabrication process also restricts the design to a single sided PCB as the exact alignments of different layers in a PCB is difficult without computerized machineries. However, with available resources a fully functioning circuit was designed. The 3D view of designed PCB layout is as shown in figure 8-12.

### 8.1.4 Hardware Execution

The amplifier circuit and the filter circuit was initially tested individually by giving an input of low frequency signal from a function generator and the output was visualized in an oscilloscope. The gain of the amplifiers and the cut-off frequency of the filters were inspected during the test. The circuits were then tested with EMG signals from facial

muscles extracted using the Ag-AgCl electrodes in bipolar mode. The signal electrodes E1 and E2 were placed on the cheek muscle (Zygomaticus Major) and the ground electrode G1 was placed on the wrist as shown in figure 8-7. The user was then asked to twitch his cheek muscles and the EMG signals were observed in an oscilloscope

Figure 8-14: Initial Setup for Circuit Testing

After successfully testing the individual circuits of the EMG acquisition system, a throughout system with a tunable gain was designed for two channels as shown in the figure 8-9. The circuit was tested with bipolar signals from two articulatory muscles; Zygomaticus Major (Channel 1) and Platysma (Channel 2), as shown in figure 8-8. The signal electrodes E1 and E2 for Channel 1 were attached along the length of the muscle Zygomaticus Major keeping them at a certain distance apart. Similarly, the signal electrodes E1 and E2 for Channel 2 were attached across the length of the muscle Platysma with some distance in between them to avoid cross-talk. The ground electrode G for both the channels was placed on the wrist away from the signal electrodes.

The overall circuit gain was first set at 5000 and was gradually increased until the signals of proper amplitude were observed which was at about a gain of 7600. The signals were sampled with Arduino's ADC at a sampling rate of about 600 Hz. The signals were then recorded for a discrete word utterance for an interval of 3 seconds, which was the average time for the user to utter a word.

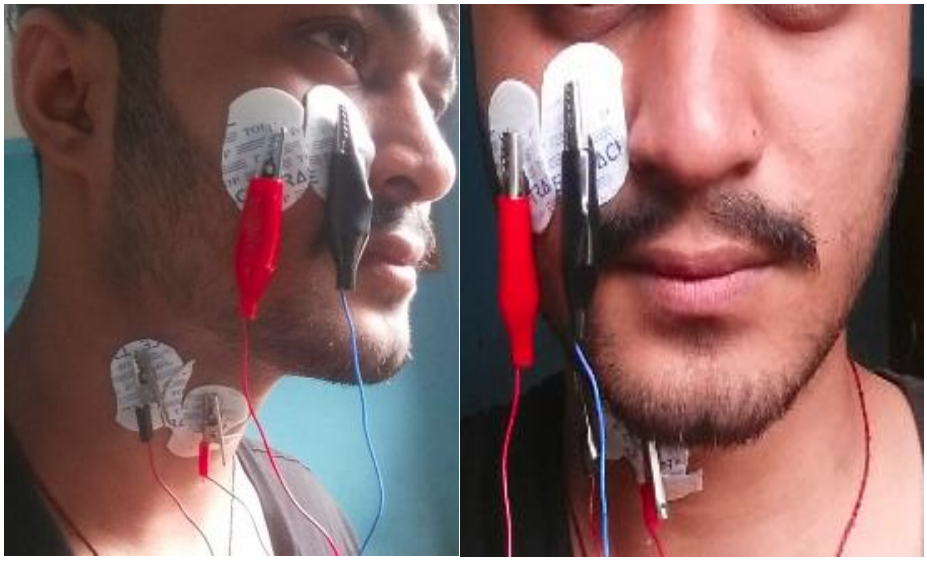


Figure 8-15: Electrode Placement on Facial Muscles

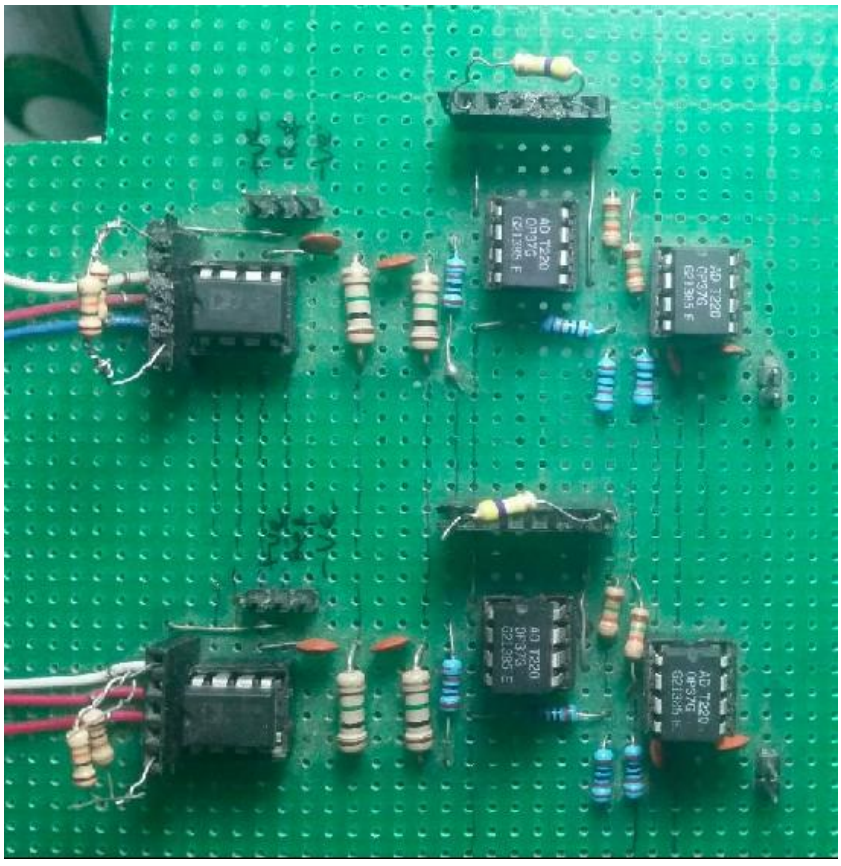


Figure 8-16: Dual Channel EMG Acquisition Circuit

The above figure shows the designed circuit implemented on a matrix board. The upper section is designated as channel 1 and the lower section is designated as channel 2. The pair of three wires going into both the sections are signal wires E1 and E2, and ground wire G1. The ground wire is shorted to circuit ground and the signal wires are fed to the differential input of the instrumentation amplifier AD620. The circuit has female headers in both sections for making the resistors swappable so that the gain of the system can be tuned accordingly. The filtered EMG signals are sent to Arduino using jumper connectors through the male headers present at the end of the circuit.

### 8.2 Software Implementation

This section explains how the dataset has been used, how the data has been obtained from the hardware system, how the data has been processed and also how the machine learning model has been developed.

#### 8.2.1 Data Segmentation

The EMG-UKA corpus has parallel recordings of speech in three different modes: Audible, Whispered and Silent. All the recordings are in the form of continuous speech with phoneme level and word level annotations in TIMIT format [25]. The continuous audio and sEMG data can be thus chopped according to the given annotations and made into corpus of discrete words. The distribution of words was analyzed and the top ten most occurring words were selected from the discrete words dataset and grouped according to the modes of speaking.

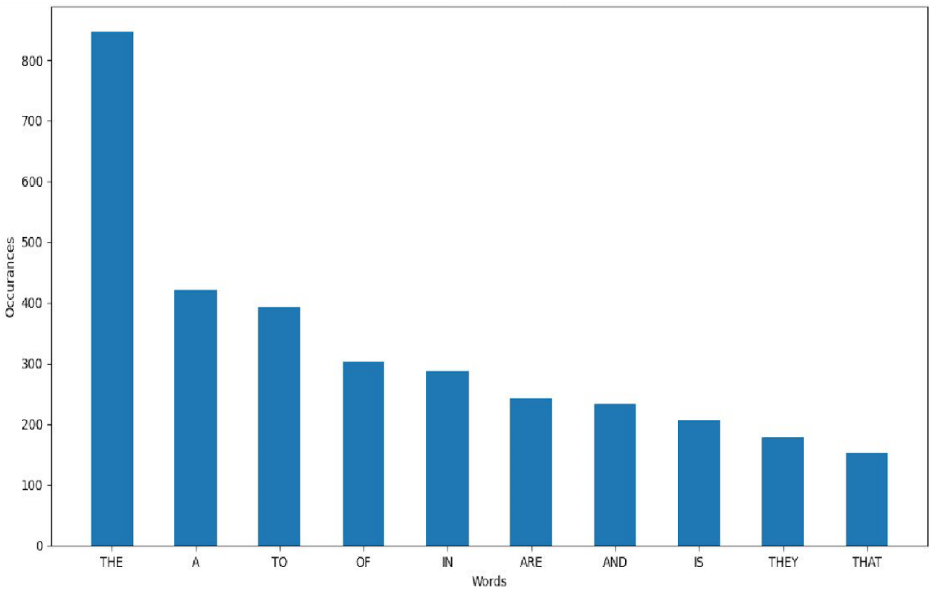



Figure 8-17: Occurrences of Top 10 Words

The figure 8-10 shows the distribution top 10 words in the devised dataset of discrete words. Evidently, the word “THE” has the highest occurrence and is almost double that of the occurrence of the second word “A”.

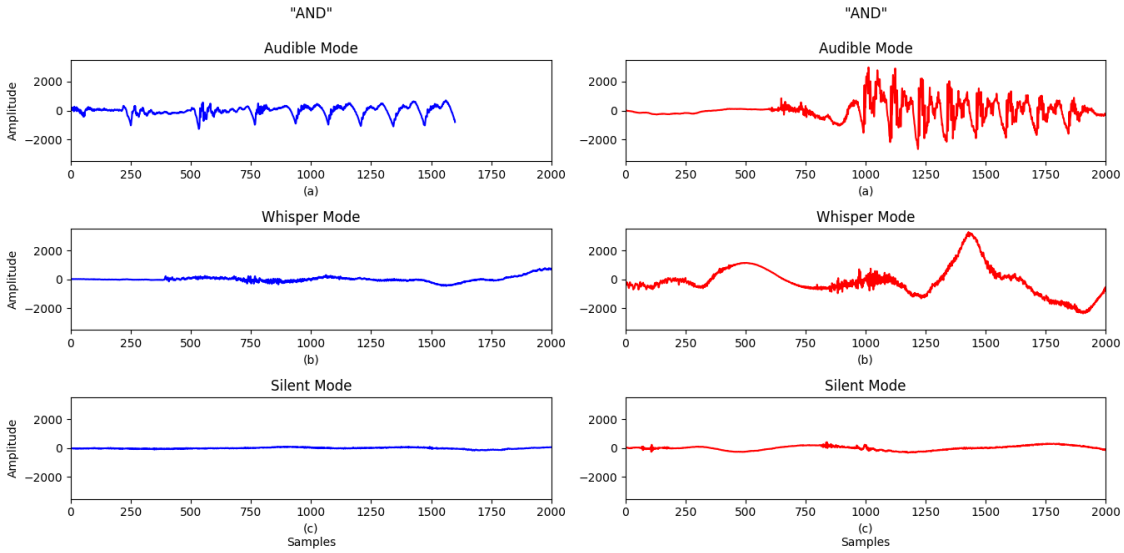



Figure 8-18: Utterance of Words “AND” (left) and “THAT” (right)
a) Audible, b) Whispered and c) Silent

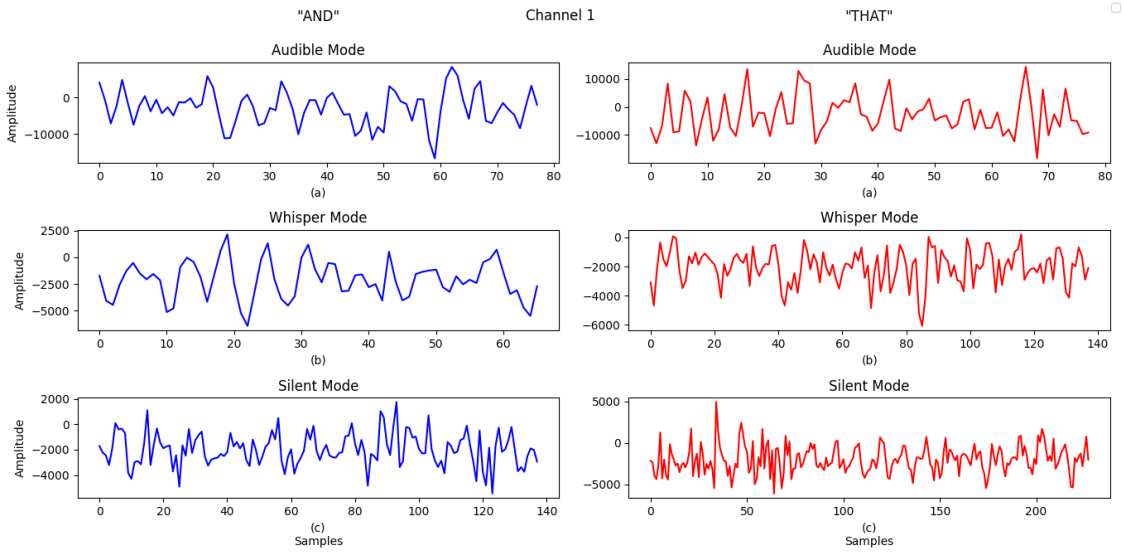



Figure 8-19: Channel 1 Signal of “AND” and “THAT

### 8.2.2 Data Acquisition

For data collection and training of the system a python script was written. It records Serial data for a certain fixed time. The serial data from the Arduino is stored to a CSV file under a suitable labelling. The columns in the CSV file indicates the channel and rows are the data of the respective channels.

For further convenience a GUI interface capable of visualizing and recording was developed. But the features are still under development. It is able to visualize the raw EMG signal from the Arduino. It has two threads, one for receiving the serial data and another for the update of the plot. The features FFT, save and record emits signal when triggered and respective event is performed.

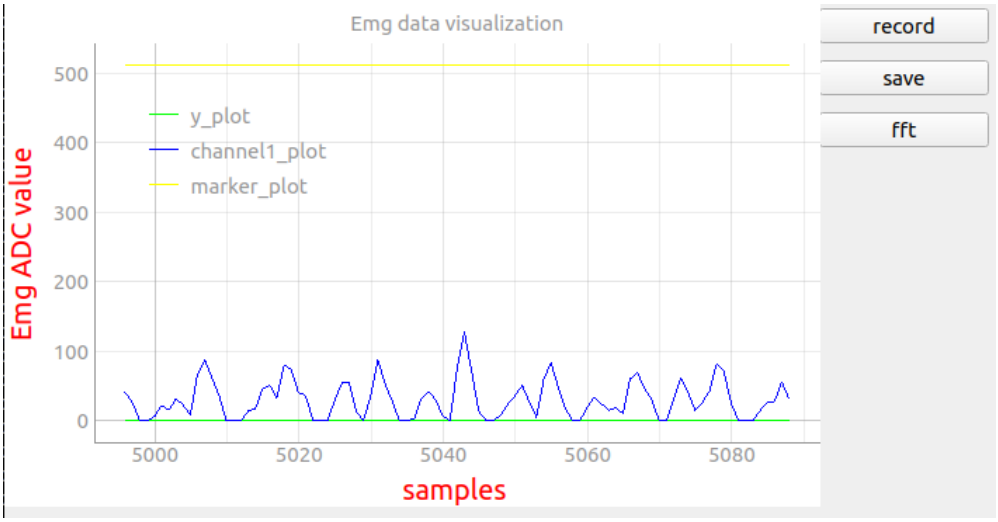



Figure 8-20: Custom Graphical User Interface

The y-axis gives the raw EMG ADC value from Arduino. The x-axis gives the number of samples. The line y-plot is used for testing and debugging the plot. The channel1_plot indicates the signal from the Arduino, number of EMG can be increased and set different colors for other plots. The marker plot is the ending value sent by the Arduino, which usually contains 1024, 512 or any value. As this value also helps in setting the maximum range of the graph plot.

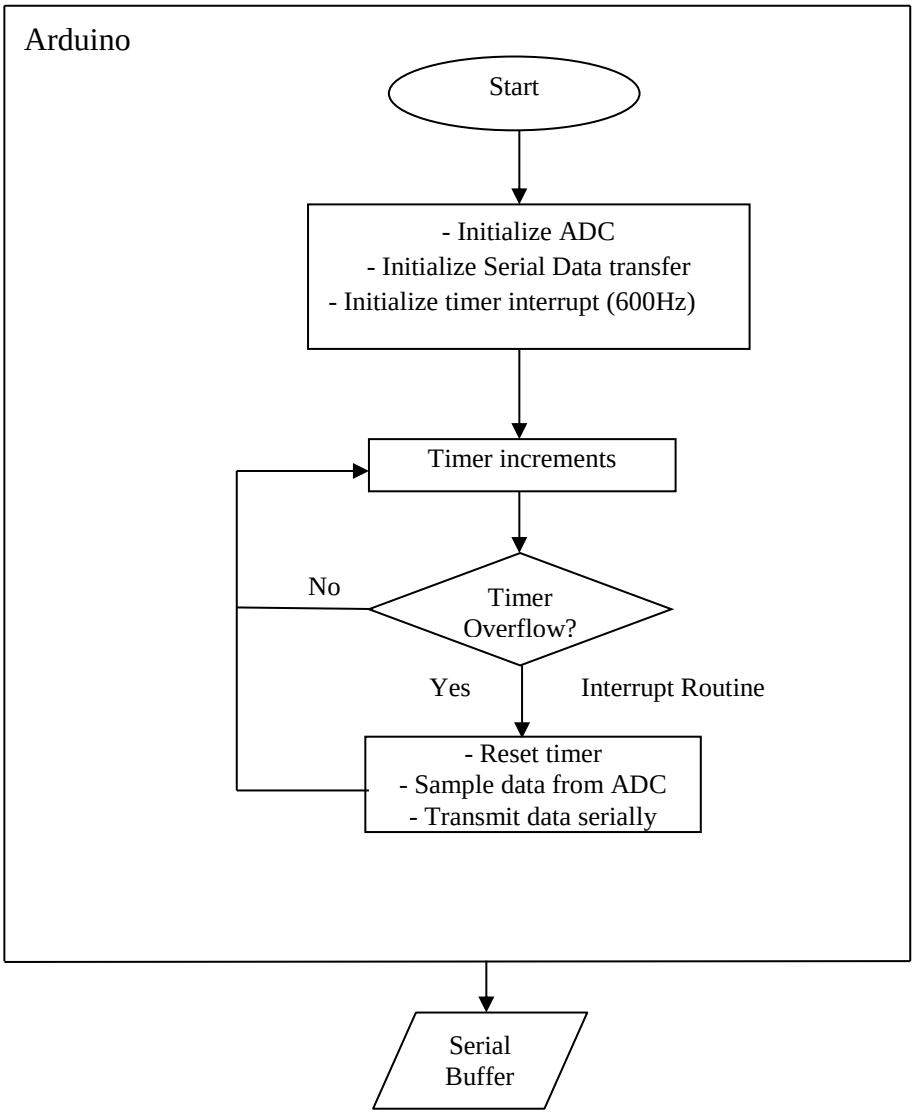



Figure 8-21: Flowchart of Serial Data Transfer from Arduino

The Arduino and the computer work in unison to record the EMG signal. The program execution and interaction of the Arduino and computer can be realized from the following flowchart. The Arduino samples data with frequency of 600 Hz which is set with the timer which is then transmitted to the computer through serial interface. The computer intermittently checks the availability of data in the serial buffer and continues to record the data till the given recording time period.

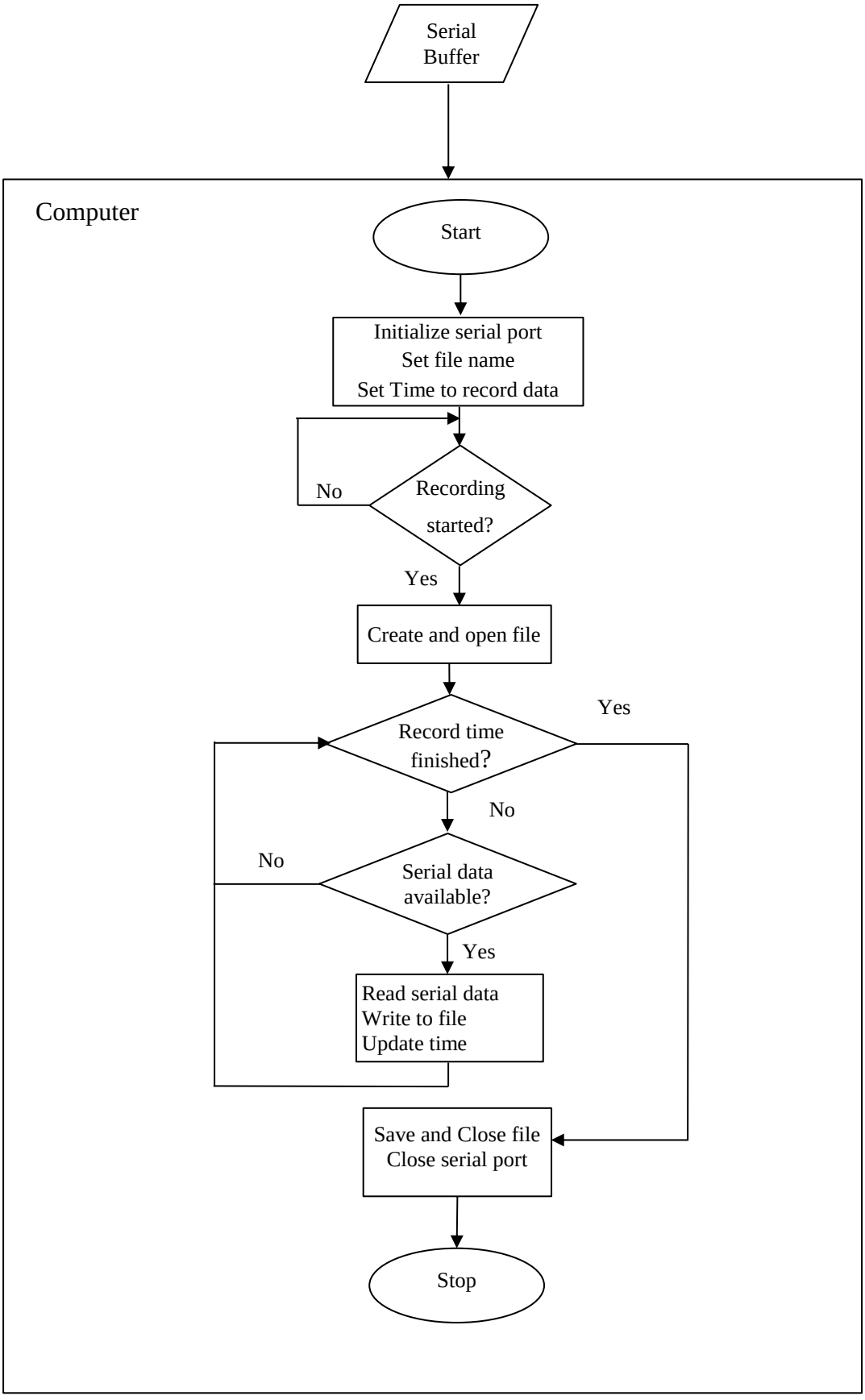



Figure 8-22: Flowchart of Serial Data Receiver in Computer

### 8.2.3 Data Processing

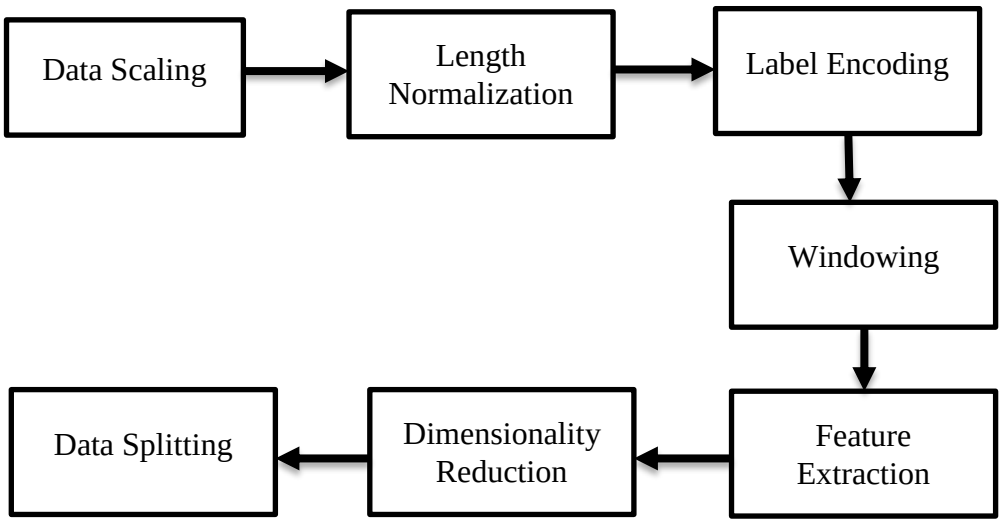



Figure 8-23: Flow of Data Processing Before Model Training

After segmenting continuous speech data into discrete words, the data was scaled to transform all the data to a common scale without distorting differences in the ranges of values or losing any information. Firstly, data from all the channels were converted to microvolts (μV) and a channel-wise standardization was performed by computing z-scores for all the respective channels. The mean and variance values for each channel were also stored to scale the data during the model deployment stage.

During the analysis of data, it was observed that the number of samples were different for different classes and even among the data belonging to the same class. This difference in sample length was compensated by trimming the data instances with large sample length and zero padding those data instances with smaller sample length. The cut-off length was selected so that most of the data is included and only the outliers are left out. To determine the cut-off length, a list containing the length of all the samples was generated and the 95th percentile of this list was calculated. The data instances beyond the 95th percentile was trimmed to this value and the rest were zero padded to this value. The 95th percentile for audible mode, whispered mode and silent mode were calculated as 148, 150 and 198 respectively.

The class labels are in string format which needs to be encoded into numeric format for classification tasks. The label encoding of the dataset was achieved using the Sklearn's LabelEncoder method.

Extracting the features of a whole session at a single lot is an inappropriate as well as vulnerable task so the data was chunked into small segments using windowing technique. The length of the segmented data depends on the nature of the window. Kaiser window with beta parameter 25 was used. The beta parameter of the Kaiser window controls the width of the main lobe.

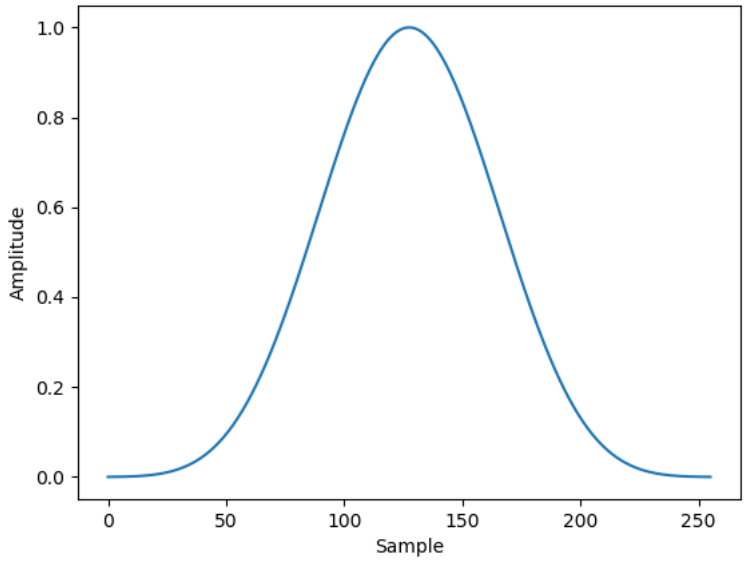



Figure 8-24: Kaiser Window (M=256, β=25)

After windowing the data, the temporal as well as spectral features of the data were extracted. Under temporal features, zero crossing rate, average rectified value, average power and root mean square were selected as they showed significant improvement in the performance of the Neural Network. STFT and MFCC were implemented to extract the spectral features. The output of STFT were frequency samples, time segments and complex transform features. The complex transform features were 3 dimensional tensors which were reshaped into 4 dimensional tensors before feeding into the 2-dimensional input layers of MLP and CNN.

With a higher number of input features it becomes difficult to visualize the training set and makes the prediction task complex. Most of the features are correlated and thus redundant due to which the n-dimensional data in a vector space may represent a small and non-representative sample. Dimensionality reduction eliminates the redundant feature spaces and makes the data visualization clearer. Principal Component Analysis (PCA) was implemented for dimensionality reduction. It first calculated the mean of the data and formed a covariance matrix then eigenvalues and eigenvectors were computed. The eigenvector with the highest eigenvalue is the principal component of the data space. The eigenvectors with lower eigenvalues were eliminated from the feature space in expense of loss of some information. The lost information however is compensated by the principal component. Finally, a n-dimensional feature space is reduced into k-dimensional feature space with a better data visualization aspect. The eigenvector with the highest eigenvalue is the principal component of the data space.

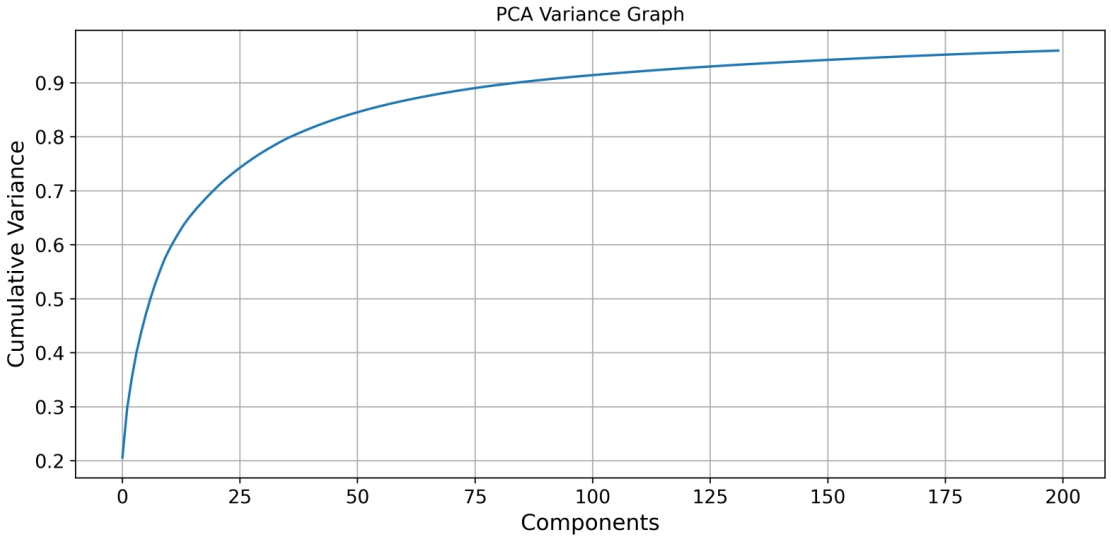



Figure 8-25: Variance Graph for PCA

From the figure above, it can be seen that 90% of the variance is conserved when taking 77 components from the k-dimensional feature space and 95% of the variance is

conserved when taking 200 components from the k-dimensional feature space. Thus, 200 components were selected keeping the 95% variance in the data.

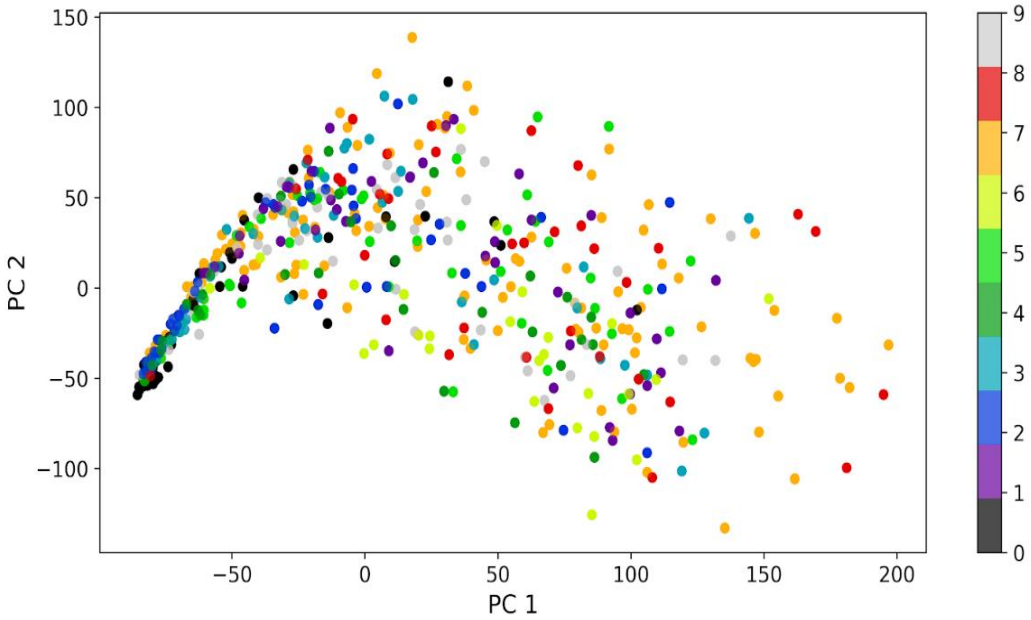



Figure 8-26: Component Scores of PCA

The comparison of two PCA components was plotted as shown in figure above. The more scattered the data points the more representative becomes the dimension reduced feature space. This means for the reduced feature space to be the best estimation of the total feature space or population, the data points or the eigenvalues should be taken from all the eigenvector arrays in the population. In the above figure, the vertical colored bar represents the labels of the feature space. The PCA 1 components of the data points or the vectors are scattered more between the range 50-150 and the PCA 2 components are scattered more between 0-100 ranges while the remaining scores are sparse. This shows that the reduced feature space is not the best estimation of the population however, the labels with dominant data are represented fairly enough by the reduced feature space.

The dataset was split into two subsets; train set and test set. The train set was used to fit the designed model while the test set was used for analyzing the performance of the model. The test set was given input to the model and the predictions made by the model were compared to the theoretical expectation of the model. Loss or cost value is obtained after the comparison which is performed using a loss function. Among various loss

functions, cross-entropy or log loss function was used which is a classification loss function and measures the cross-entropy or the differences between two probability distributions. Entropy is the average number of bits required to identify an event drawn from the dataset. A skewed distribution has low entropy and the distribution where events have equal probability has high entropy. The test data was repeatedly analyzed by the loss function to optimize the model. A portion of train set (10%) was extracted as development set (dev set) for cross-validation of the model and parameter tuning.

### 8.2.4 Machine Learning Model Development

This section describes the architectural structures of machine learning models MLP and CNN. As the size of layers and selection of activation function alters the result, they need to be properly selected for desirable output from the model.

#### 8.2.4.1 Architecture of MLP Model

The architecture of the designed MLP network is as shown in Figure 8-20. It consisted of an input layer with a number of neurons equal to the size of input feature tensor, two hidden layers with ReLU activation functions consecutively and finally an output layer with 10 neurons. The input is a 1D tensor of size 1x200 which was fed to a hidden layer H1 with 64 neurons that gave an output of 1D tensor with a size of 1x64. The next hidden layer H2 shares the same properties as the layer H1. The both hidden layers implemented ReLU activation function. Finally the output of size 1x64 from the hidden layer H2 was mapped to the output layer with 10 neurons using softmax regression.

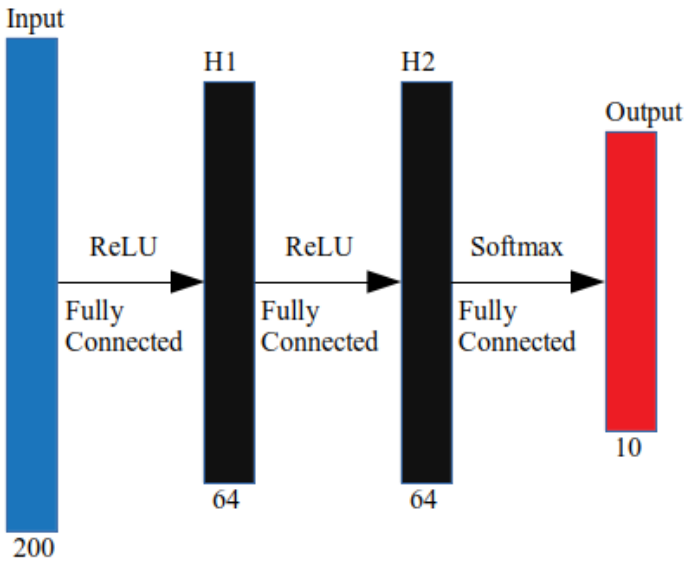



Figure 8-27: Designed Multilayer Perceptron (MLP) Model

### 8.2.4.2 Architecture of CNN Model

The architecture of designed CNN is as shown in the figure 8-21. The size of the input tensor was 1x200. The first hidden layer was a 1-dimensional convolution layer (Conv1D 1) with ReLU activation function. It consisted of 100 filters and a kernel/filter of size 3. It convoluted the data resulting output of size 100x198 which was then pooled with 1-dimensional max-pool layer (MaxPool-1D 1) with kernel size of 2. It resulted in a tensor of size 100x99. The data was again fed to Conv1D 2 with the same number of filters and same kernel size with ReLU activation function that resulted in the output of shape 100x97. This layer was then fed to MaxPool-1D 2 layer with the same properties as that of Maxpool-1D 1. The tensor size output from this layer was 100x48. The data was then flattened to one dimensional tensor of size 4800 using a Flatten layer. The next hidden layer was H1 with the number of nodes being 100. The final layer was the output layer with a resulting tensor size of 10 which was mapped from the hidden layer H1 using softmax regression. All the hidden layers along with the output layer beyond the Flatten layer were fully connected.

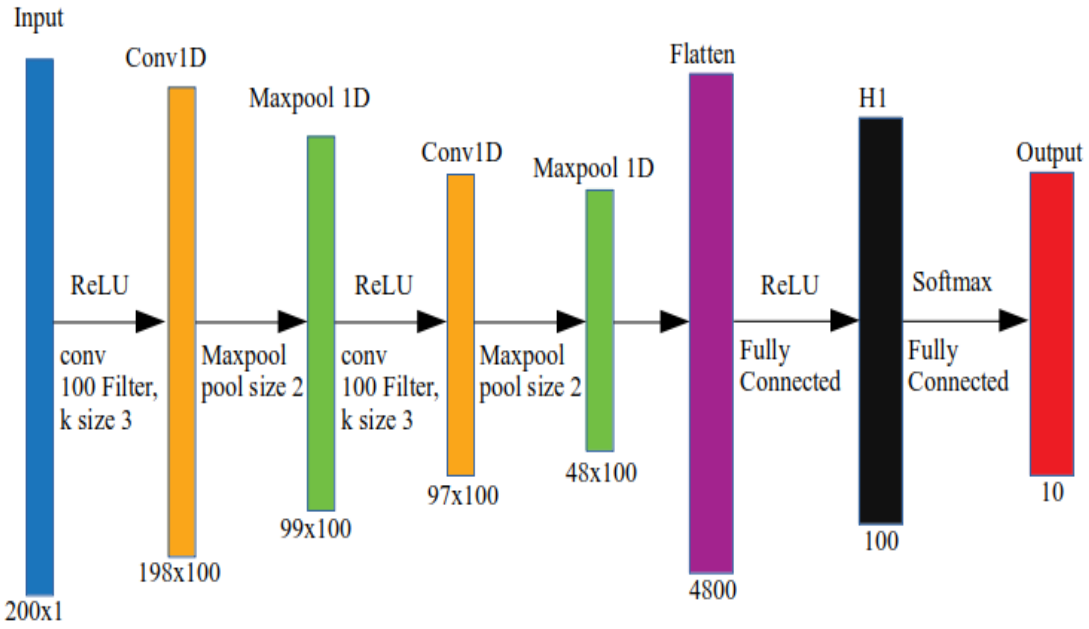



Figure 8-28: Architecture of Designed 1D CNN

# 9. RESULTS

This section includes responses of circuit and different machine learning models. Both supervised and unsupervised machine learning models are tested and relevant outputs are illustrated through figure and plots. Also, the extracted EMG signals has been visualized in real time and recorded using a custom interface.

## 9.1 Circuit Response

Using the python sketch, some of the words uttered by the subject was recorded. This data was then manipulated using Octave and FFT was visualized without applying any digital filter. The recorded EMG data of some words are as shown in figures below.

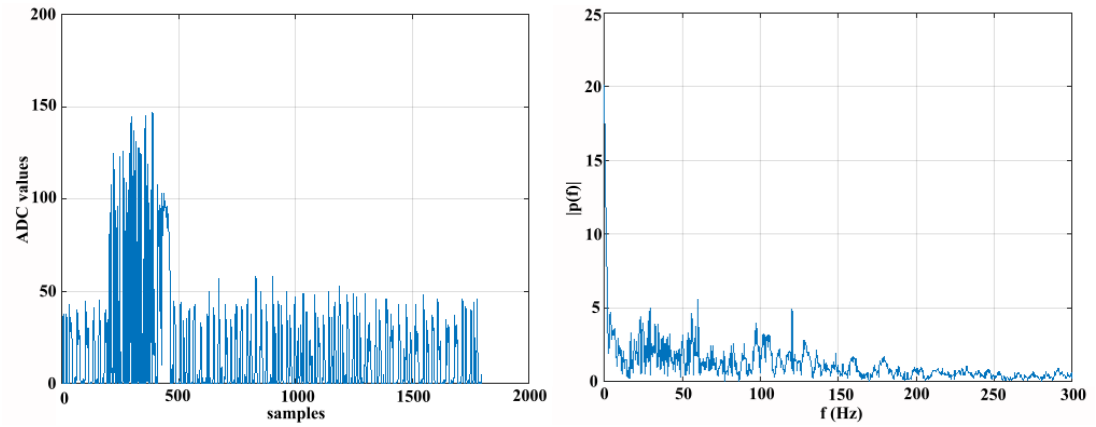



Figure 9-1: Raw EMG Data (Left) and FFT (Right) of Channel 1 Data for “AND”

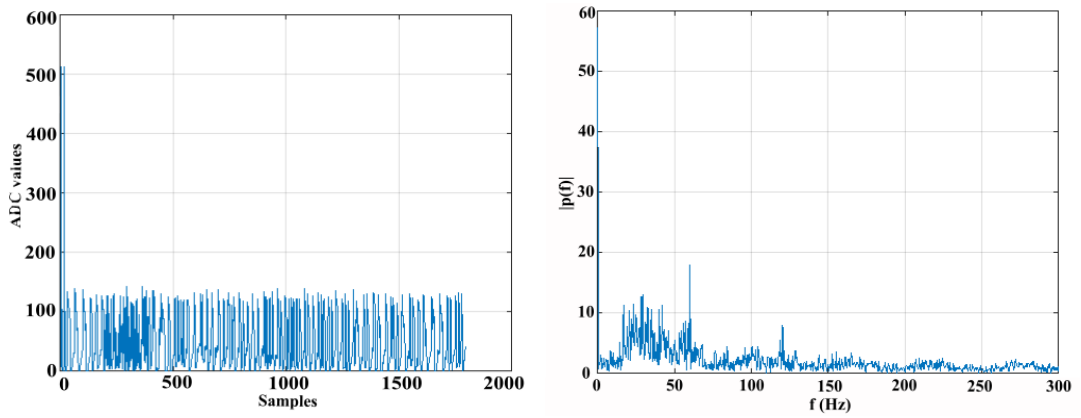



Figure 9-2: Raw EMG Data (Left) and FFT (Right) of Channel 2 Data for “AND”

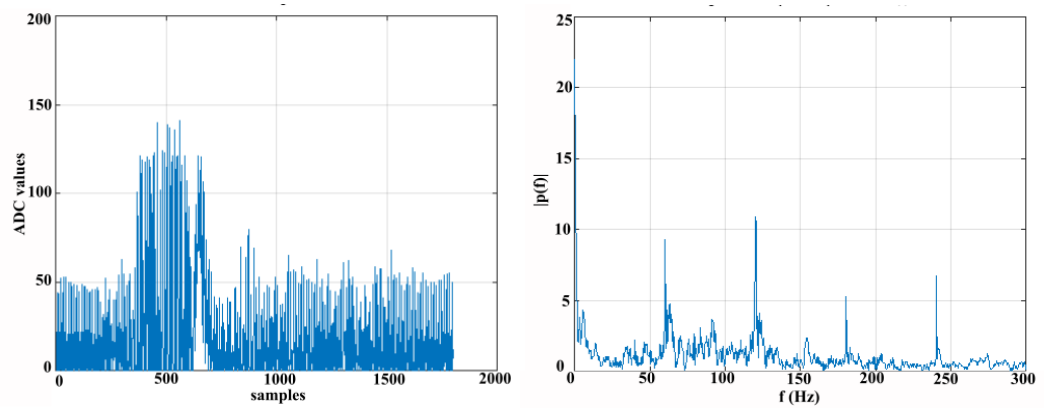



Figure 9-3: Raw EMG Data (Left) and FFT (Right) of Channel 1 Data for “THAT”

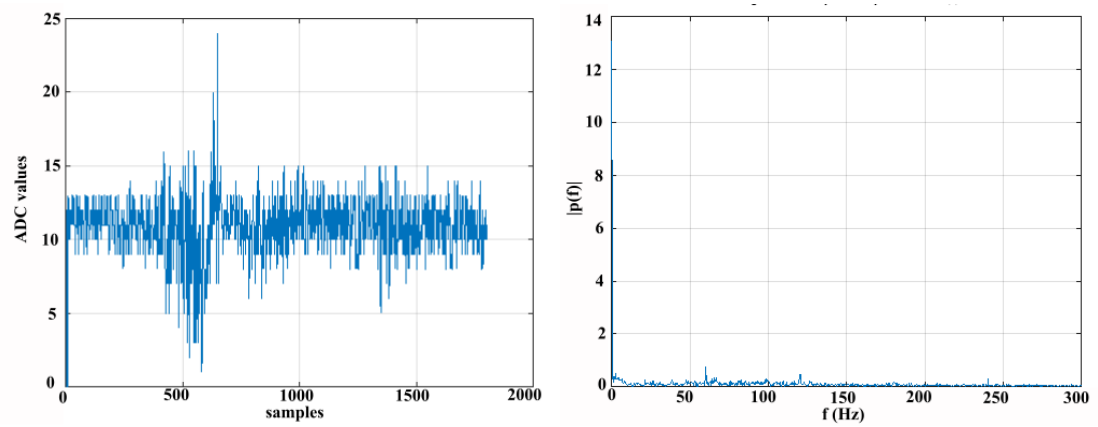



Figure 9-4: Raw EMG Data (Left) and FFT (Right) of Channel 2 Data for “THAT”

Prominent spikes on the above graphs signify the successful extraction of the facial EMG signals by the designed hardware. The regions of the highest amplitude represent the point of utterance. The spikes are dominant within the frequency range of 1 - 100 Hz which verifies the performance of the designed filters as per the calculation. However, implementation of digital filters still remains an optimization part.

The channel 1 located at cheek region shows prepotent EMG signals over channel 2 while uttering both words “AND” and “THAT”. Both of the plots have differentiable graphs yet other features are to be realised to distinguish the uttered words. Also the line noise was encountered at 60 Hz and its harmonics which can be clearly seen in the figure 9-6.

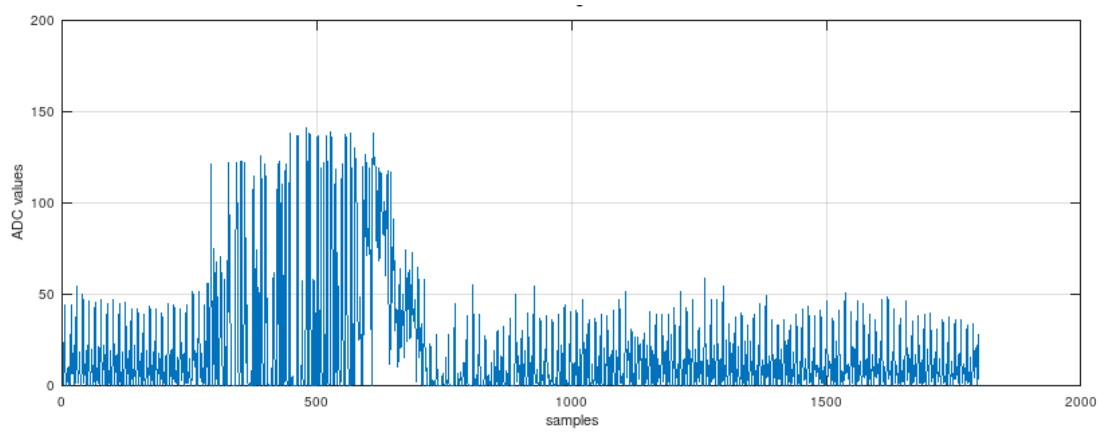



Figure 9-5: Raw Channel 1 EMG Signal of Word “THAT”

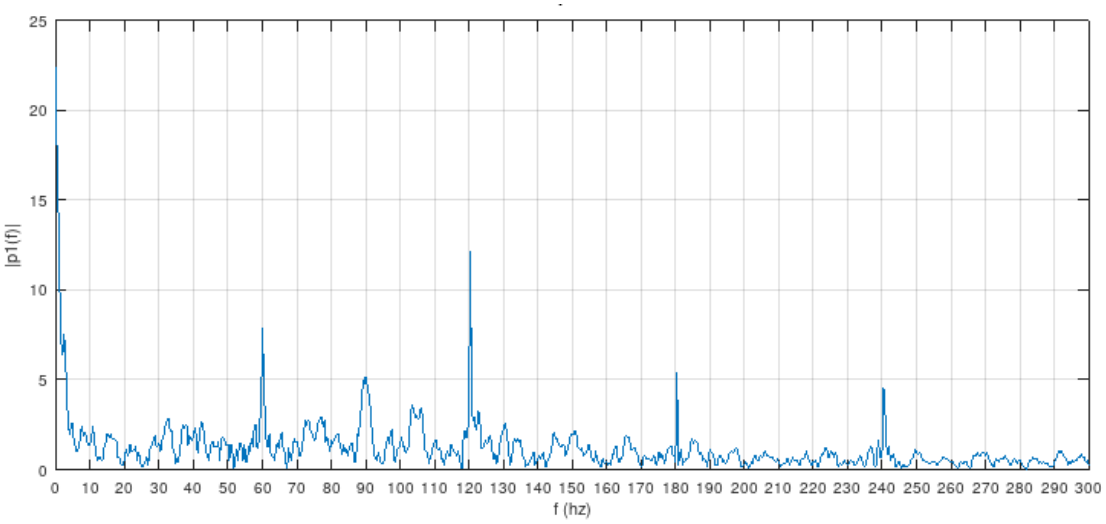



Figure 9-6: Frequency Spectrum of Word “THAT” with Line Noise

This does affect the data but can be avoided using a notch filter with higher order low pass filter. Since the circuit needs to be as compact as possible and additional components will only increase the size of the circuit board, digital filters should be implemented. A digital filter composed of a low pass filter with cut-off at 100 Hz and notch filter at 60 Hz was designed and the output of which is as shown in figure below.

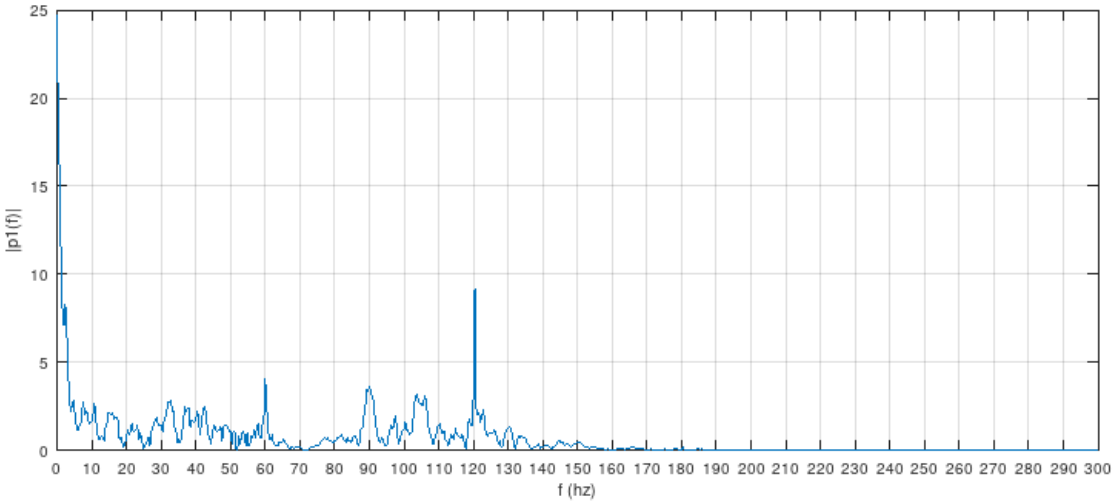



Figure 9-7: Filtered Frequency Spectrum of Word "THAT"

With the same hardware setup as shown in figure 8.8, the interface developed for visualizing and recording the facial EMG signals was tested. With only a single channel tested on, it did plot the raw EMG data but was unable to efficiently plot as in Arduino's Serial Plotter. The interface sometimes cannot decode the serial data from the Arduino which results in sparse EMG data points. While this does not affect the graph plot of the EMG data but the recording is severely affected.

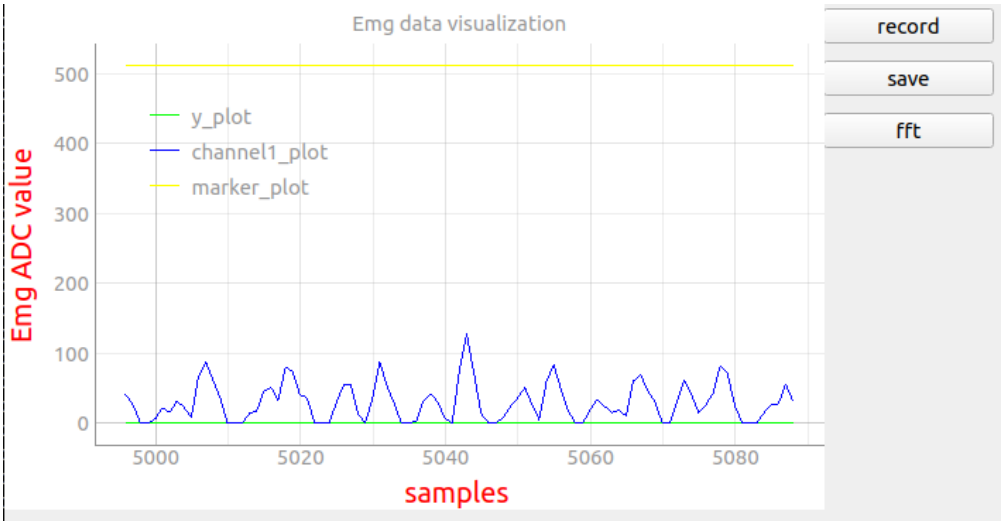



Figure 9-8: Visualization of EMG in Custom Interface

## 9.2 Learning Model Response

The extracted EMG signals are fed to different classifying machine learning models to predict the articulated word in all speaking modes.

### 9.2.1 Supervised Model Response

Firstly, the supervised machine learning models were trained using the labelled data parsed from the EMG-UKA corpus. The data was prepared according to the data preparation process mentioned in the software implementation section of implementation details. The models were tested using both temporal and spectral features extracted from the data. The temporal features extracted were: Zero Crossing Rate, High Frequency Signals, Rectified High Frequency Signals, Frame Based Power and Double Nine Point Average and the spectral features extracted was MFCC. The models selected for the classification tasks were KNN (K-Nearest Neighbours), MLP (Multi-Layer Perceptron) and 1D CNN (Convolutional Neural Network).

#### 9.2.1.1 KNN Model Output

The processed data was first used to train the KNN model with the number of nearest neighbours varying from 1 to 14 neighbours as shown in figure 9-9. The accuracy for the audible mode was observed to be highest with a value of 46.28% (at K = 7 Neighbours) and 48.79% (at K = 9 Neighbours) for temporal and spectral data respectively. Similarly, the accuracy for whisper mode observed was 35.88% (at K = 8 Neighbours) and 37.27% (at K = 5 Neighbours) for temporal and spectral data respectively. For silent mode, it was seen that accuracy was highest for spectral data at K = 8 Neighbours which was 29.74%. For the temporal data, the maximum accuracy was observed at N = 12 Neighbours which was only 30.97%.

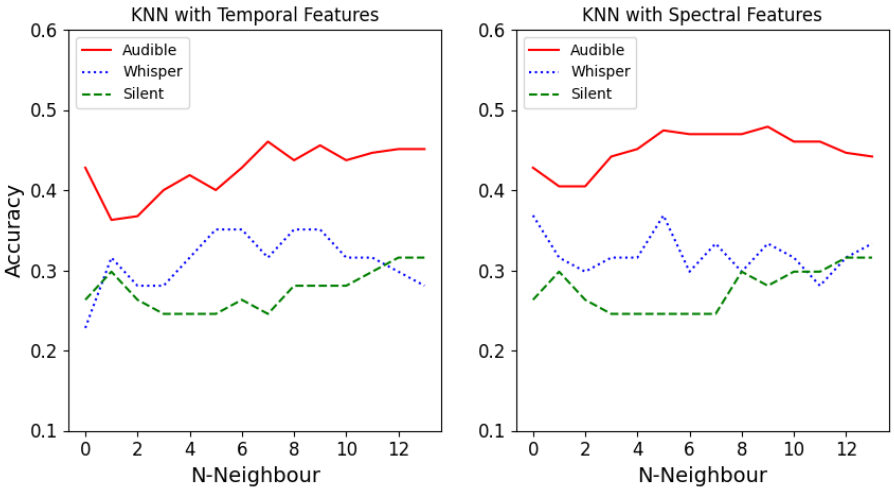



Figure 9-9: Accuracy vs K-Neighbours Plot

The figure below is the plotted heat map for the KNN model using temporal features in the left and using spectral features in the right. The generated heat map shows the imperfections in the prediction made by the KNN model. For both the feature types, the model cannot properly classify the phonetically similar words; “A” and “THE”. The word “THEY” also has been falsely classified as “THE” in many instances. Overall, the KNN model (for both and spectral features) predicts the phonetically dissimilar words “A”, “OF” and “TO” more correctly than any other classes.

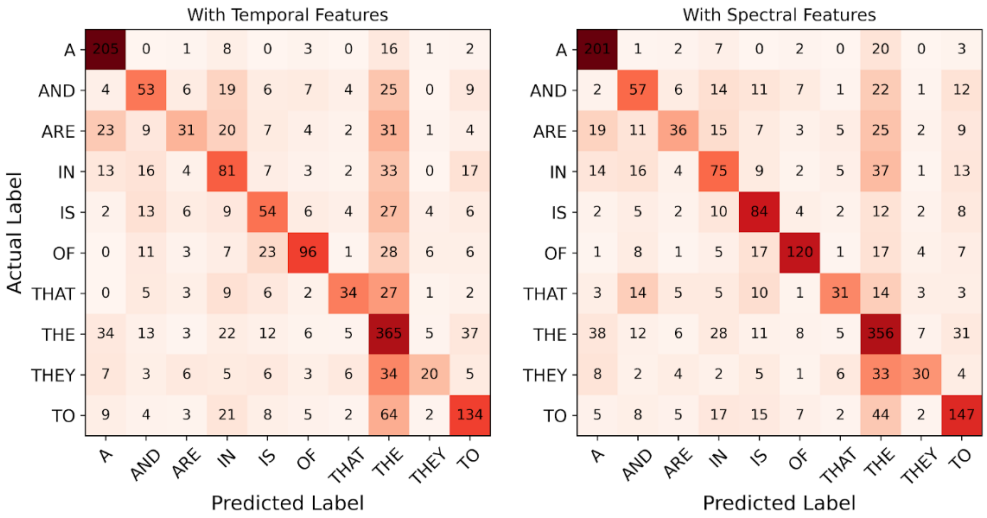



Figure 9-10: KNN Confusion Matrix for Audible Mode

From the generated heat map shown below in the figure 9-11, the KNN model performed worse for the whisper mode than in the audible mode. In both the spectral and temporal feature heat maps, the KNN model has failed to make much of the predictions except for the words “A” and “THE”. For these words, the model predicted as well as in the audible mode while for the rest of the words, the performance was worse. However, the model using the spectral features performed slightly better than the one using temporal features in the whisper mode itself.

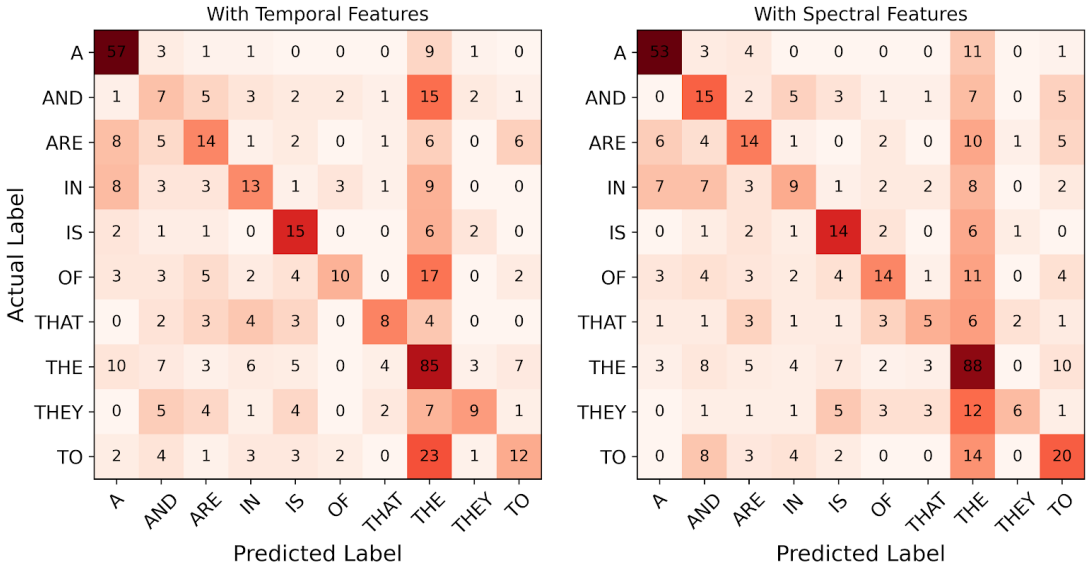



Figure 9-11: KNN Confusion Matrix for Whisper Mode

The figure 9-12, below shows the generated heat map for silent mode. Silent mode is the hardest to predict as it has relatively lower amplitudes as compared to other modes. This fact is well resonated in the figure too. For this mode, the prediction made by the model is only accurate 38% of the time as shown in the table 9-3. Despite the low precision, the model performed well for the phonetically similar words “A” and “THE” for both types of features.

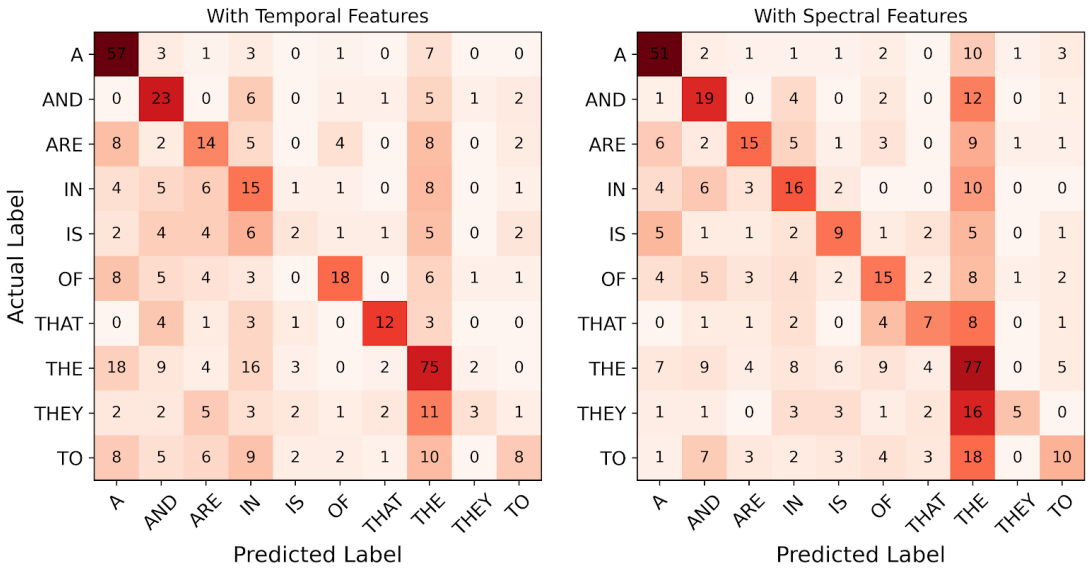



Figure 9-12: KNN Confusion Matrix for Silent Mode

### 9.2.1.2 MLP Model Output

The next classifier tested was MLP model with an input layer of 200 nodes, two hidden layers with 64 nodes each and an output layer of 10 nodes that represented the 10 words used in the dataset. The model implements ReLU as activation function and is trained over 200 epochs with a batch size of 50 using Adam optimizer function with the default learning rate of 0.001. The accuracy for each mode was plotted differently for different speaking modes as shown in the figure 9-13.

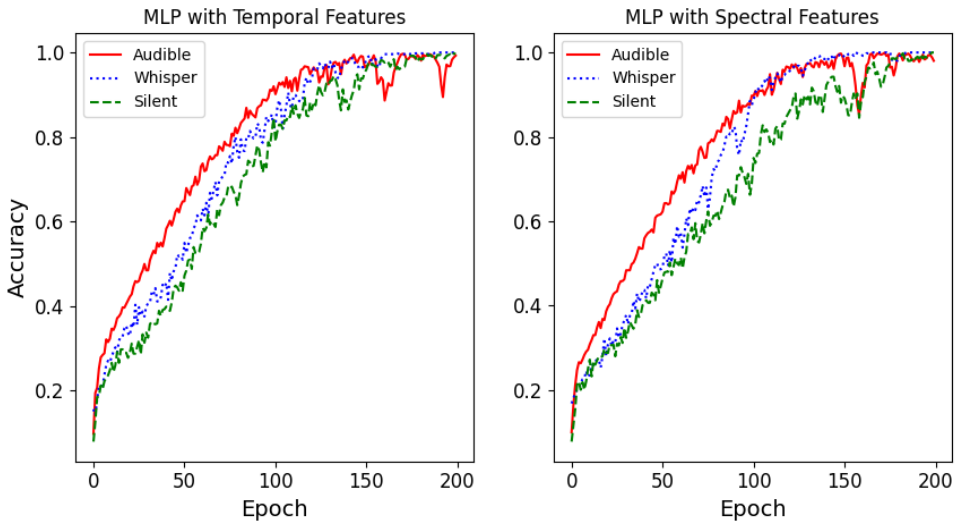



Figure 9-13: MLP Accuracy Curve

From the above figure, it can be deduced that the training accuracy is relatively high for all modes of speaking and for both feature types. The figure on the left is the accuracy plot using the temporal features for three modes represented by; solid line for audible mode, dotted line for whisper mode and dashed line for silent mode. The accuracies for this plot were found to be 99.56%, 98.25% and 97.17% for audible, whisper and silent mode respectively. The figure on the right is the accuracy plot using the spectral features with similar notations used in the figure on the left. The MLP model has the accuracies of 98.99%, 99.12% and 99.37% for audible, whisper and silent mode respectively. From the above figure, it seems that the model is overfitting the data for both spectral and temporal features.

The heat map shown in the figure 9-14, represents the confusion matrix for the MLP model. This model has a really high train accuracy for all the modes which leads to the suspicion of the model overfitting the data. The heat map also shows that the model has classified all the words correctly and only missing out a few instances while using the spectral features for training the model. Also, the precision and recall scores for the model is 98% and 97% (for temporal features), and 100% and 100% (for spectral features) respectively which further supports the suspicion.

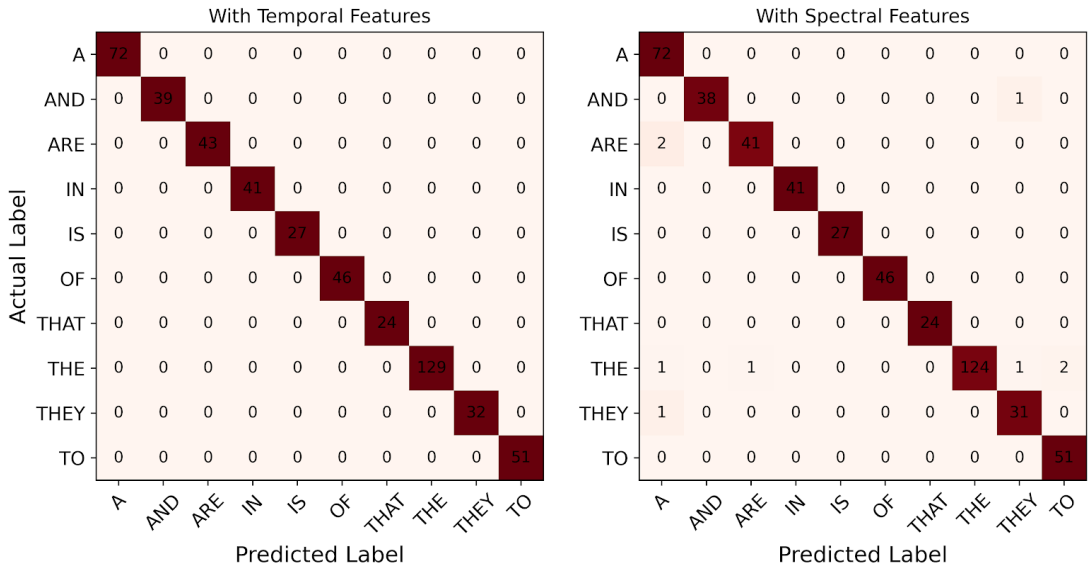



Figure 9-14: MLP Confusion Matrix for Audible Mode

The figure below is the confusion matrix's heat map generated for the MLP model in whisper mode. The performance of the model has slightly degraded as compared to the audible mode. The model has failed to correctly classify all the data instances of the words "A", "AND", "IN" and "TO" while using the temporal features whereas it has classified all the words correctly while using spectral features. This model too needs to be further studied to overfitting as it has high train accuracy but low test accuracy as observed from the table 9-2.

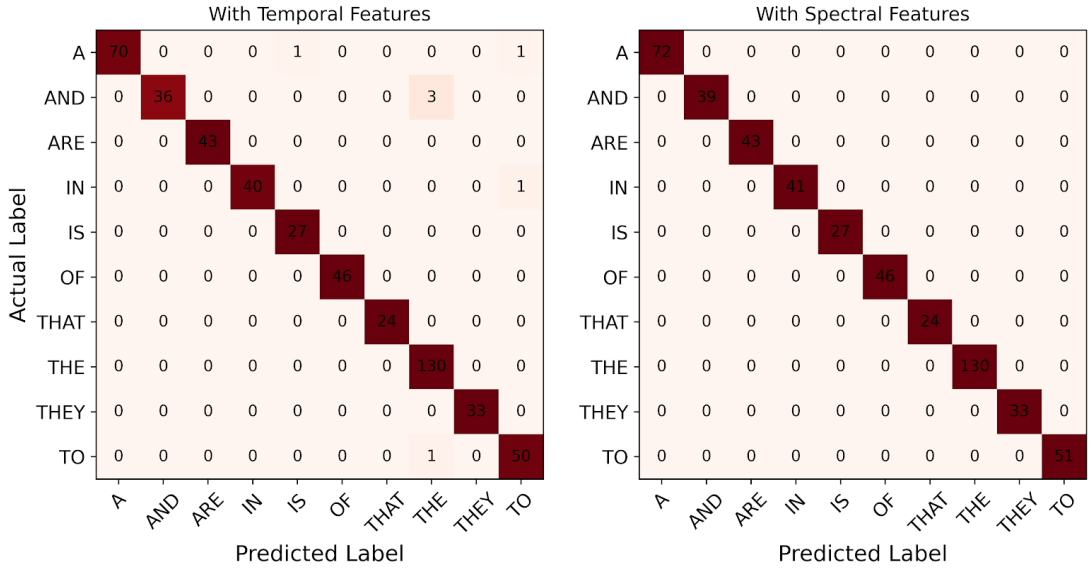



Figure 9-15: MLP Confusion Matrix for Whisper Mode

In the figure below, the MLP model output has been studied using the heat map generated from the confusion matrix. In comparison to the audible mode and the whisper mode, the performance has been degraded slightly. The data instances for the word "A" and "ARE" have been misclassified mostly as "IN" and many of the word "A" has also been misclassified as "THE" in the model with temporal features. The model implementing spectral features has performed better with respect to the one with temporal features.

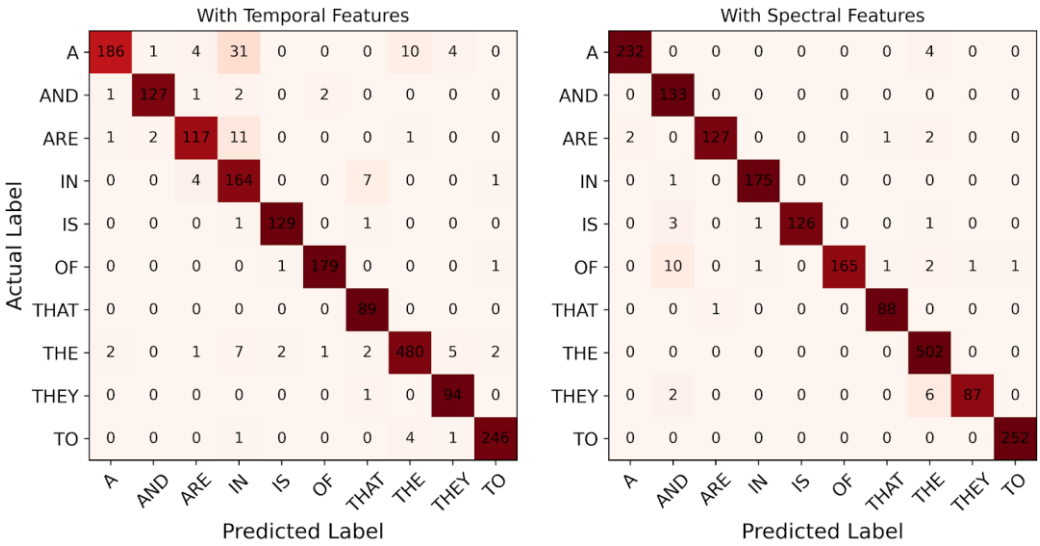



Figure 9-16: MLP Confusion Matrix for Silent Mode

#### 9.2.1.3 CNN Model Output

The final classifier model tested was a 1D CNN. The designed model has an input convolutional layer with 100 filters of size (1X3) that is followed by a max pooling layer with a pool size of (1X2). This convolutional layer and max-pooling layer is repeated once which is then followed by a fully connected layer with 100 nodes that is further connected to another fully connected layer with 10 nodes. The activation function that this network utilizes at the convolutional and hidden layers is ReLU and at the output layer is softmax. The model is optimized using Adam optimizer and is trained over 200 epochs with a batch size of 50 with the default learning rate of 0.001. The accuracy for this model is shown in the figure below.

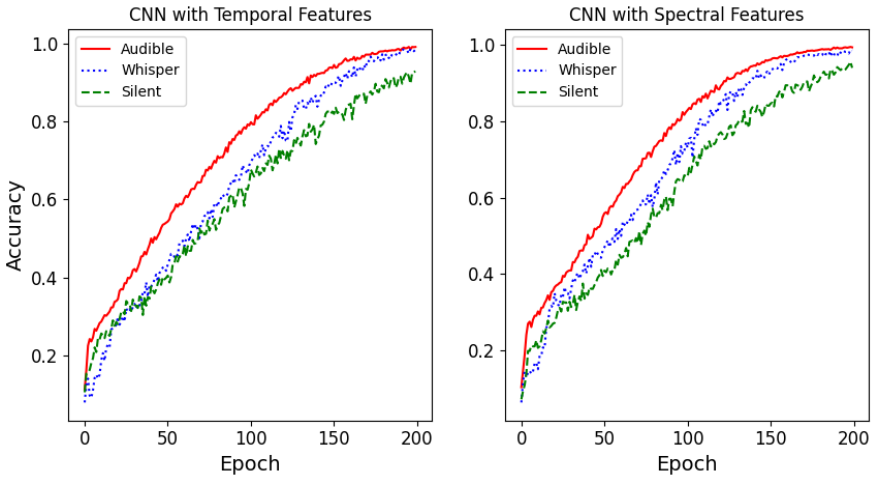



Figure 9-17: CNN Accuracy Curve

In the above figure, the training accuracies for all speaking modes with both temporal and spectral features used for training is shown. As seen similarly in the MLP model, the accuracies are high which leads us to the suspicion that the model is overfitting the data. The accuracies observed are 98.39%, 96.36% and 91.07% for the audible, whisper and silent mode respectively while using temporal features and 99.74%, 99.01% and 92.46% for the audible, whisper and silent mode respectively while using the spectral feature.

The CNN model for all three modes was analyzed using confusion matrices and their respective heat map. The figure 9-18, 9-19 and 9-20 are the generated heat maps for audible mode, whisper mode and silent mode respectively. From the generated heat maps, it can be inferred that the model using spectral features performed well in comparison to the model using temporal features in the whisper mode whereas in the silent mode, the model using temporal features performed the best and in the audible mode, both the models performed the same.

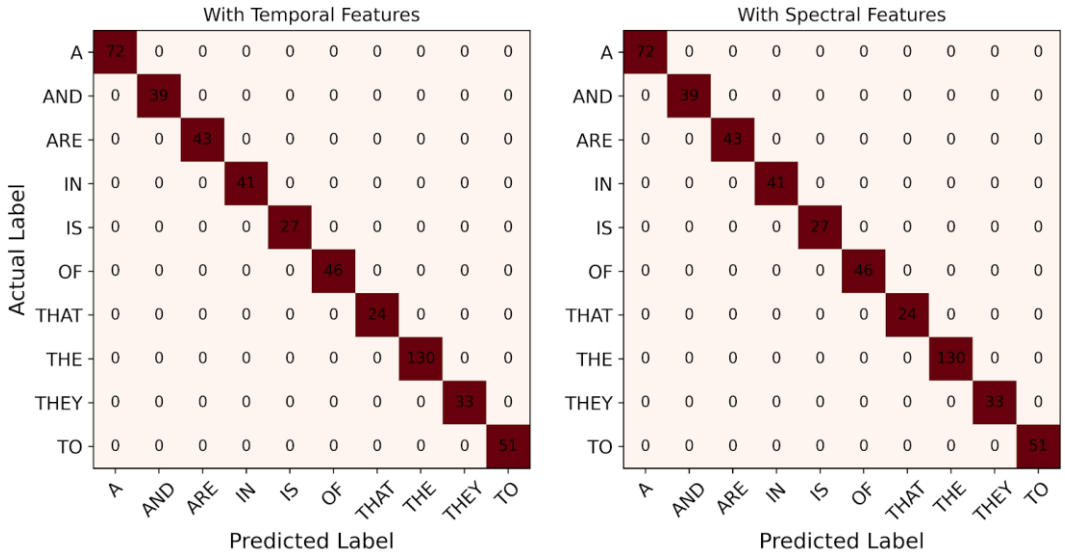



Figure 9-18: CNN Confusion Matrix for Audible Mode

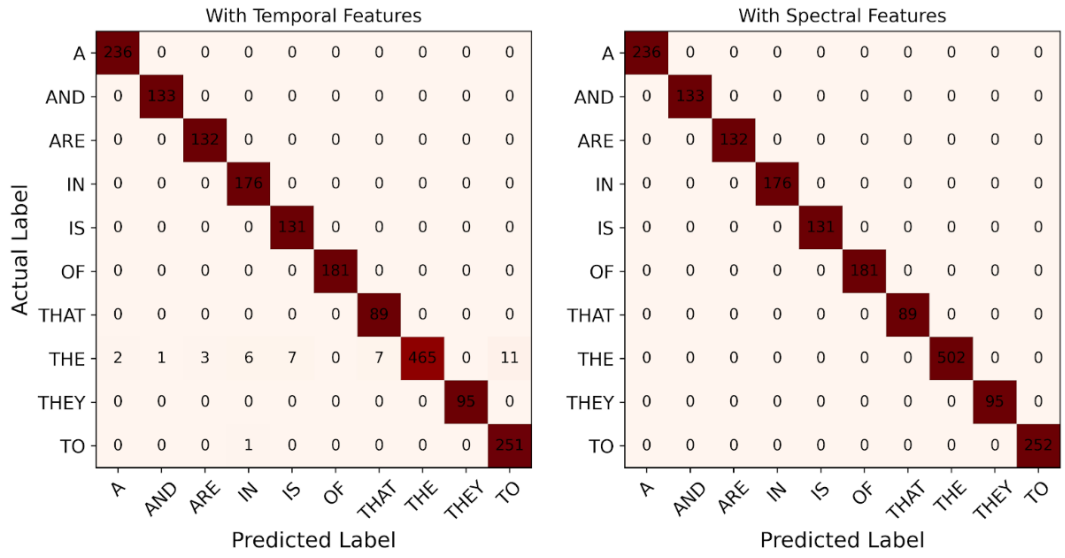



Figure 9-19: CNN Confusion Matrix for Whisper Mode

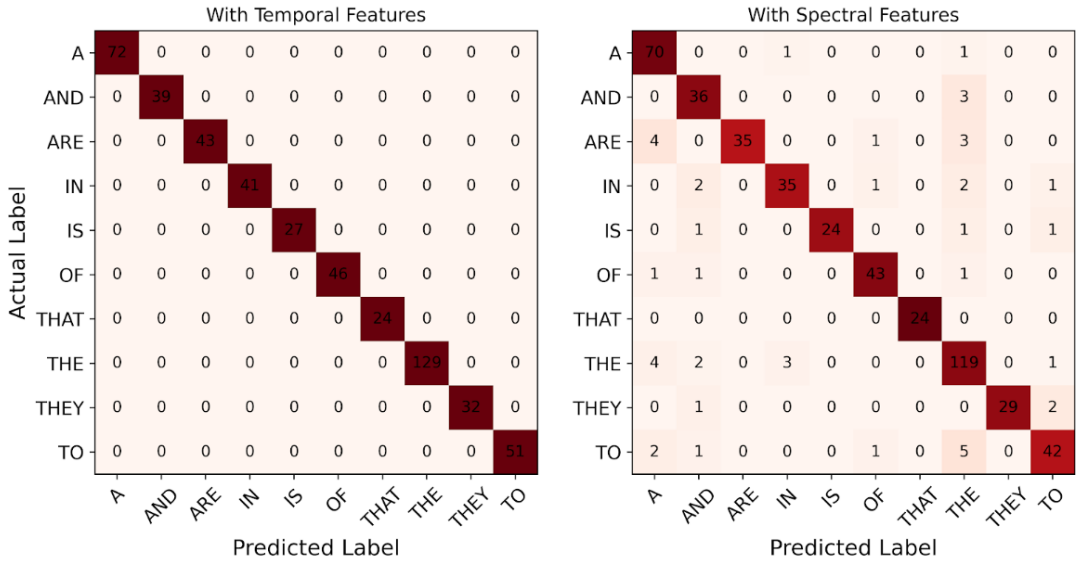



Figure 9-20: CNN Confusion Matrix for Silent Mode

### 9.2.1.4 Model Summary

The summary of all three models for both temporal and spectral features are shown in the tables 9-1, tables 9-2 and table 9-3. The tables contain the train accuracy, test accuracy, precision, recall and F1 score for all variation of the models. From table 9-1, it can be deduced that the best performing model for audible mode is MLP with spectral features as it has the highest precision and recall scores with relatively high test accuracy. However, the MLP model's difference in train and test accuracy is large which makes it

essential to analyze the model carefully before considering it as the optimum model. Despite the KNN model with spectral features having the highest test accuracy, the model was not the optimum choice as the model's precision and recall scores are remarkably low.

Table 9-1: Classifier Model Summary Table for Audible Mode

| **Models** | **Feature Type** | **Train Accuracy** | **Test Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|---|---|
| KNN | Temporal | 68.76 | 33.72 | 0.53 | 0.48 | 0.49 |
| MLP | Temporal | 99.01 | 35.81 | 0.98 | 0.97 | 0.97 |
| CNN | Temporal | 98.39 | 39.62 | 0.99 | 0.98 | 0.98 |
| KNN | Spectral | 58.01 | 47.90 | 0.55 | 0.52 | 0.52 |
| MLP | Spectral | 100 | 40.90 | 1 | 1 | 1 |
| CNN | Spectral | 99.74 | 36.28 | 1 | 1 | 1 |

The table 9-2 shows that the classifying models perform better when using spectral features for the whisper mode. The models with spectral features have marginally high scores in all respects and the optimum model among them is MLP which has the test accuracy of 42.25% and both precision and recall scores as 100%. This model is most likely to be overfitting the data as it has a large difference between the train and the test accuracy.

Table 9-2: Classifier Model Summary Table for Whisper Mode

| **Models** | **Feature Type** | **Train Accuracy** | **Test Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|---|---|
| KNN | Temporal | 44.14 | 26.79 | 0.44 | 0.39 | 0.39 |
| MLP | Temporal | 97.88 | 28.07 | 0.96 | 0.99 | 0.99 |
| CNN | Temporal | 96.36 | 19.28 | 0.91 | 0.85 | 0.87 |
| KNN | Spectral | 45.25 | 31.57 | 0.40 | 0.37 | 0.38 |
| MLP | Spectral | 100 | 42.10 | 1 | 1 | 1 |
| CNN | Spectral | 99.01 | 28.07 | 0.99 | 0.99 | 0.99 |

From the table 9-3, it can be seen that the MLP and CNN models have a similar performance in terms of accuracies. The KNN model has good test accuracies but the precision and recall scores are much lower. The optimum model for the silent mode seems to be MLP model with spectral features as it has good precision and recall scores in comparison to CNN. Further analysis of both CNN and MLP models is required before selecting between the two.

Table 9-3: Classifier Model Summary Table for Silent Mode

| **Models** | **Feature Type** | **Train Accuracy** | **Test Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|---|---|
| KNN | Temporal | 41.52 | 28.11 | 0.38 | 0.32 | 0.31 |
| MLP | Temporal | 89.57 | 21.05 | 0.94 | 0.93 | 0.94 |
| CNN | Temporal | 91.07 | 19.39 | 0.90 | 0.89 | 0.89 |
| KNN | Spectral | 43.45 | 29.82 | 0.41 | 0.37 | 0.37 |
| MLP | Spectral | 99.05 | 24.56 | 0.99 | 0.99 | 0.99 |
| CNN | Spectral | 92.46 | 24.61 | 0.94 | 0.92 | 0.93 |

### 9.2.2 Unsupervised Model Response

The EMG spectrogram feature space was fed to the unsupervised learning model imposed with K-means algorithm. The cluster size was equal to number of labels which was 10. The clustered output of the K-means algorithm in audible, whispered and silent modes are as shown in figure 9-24, figure 9-25, and figure 9-26 respectively. The circles colored red, blue, black, yellow, magenta, cyan, green, pink, orange and purple represent the labels sorted in alphabetical order. The scattered points of various colors represent the data points. In all three modes the pink colored circle can be seen separate while the remaining circles are overlapped. This shows that the data points with label represented by the pink colored circle i.e. 'THE' is clustered well because of its dominance in the data set while the remaining labels due to relatively less quantity are superimposed.

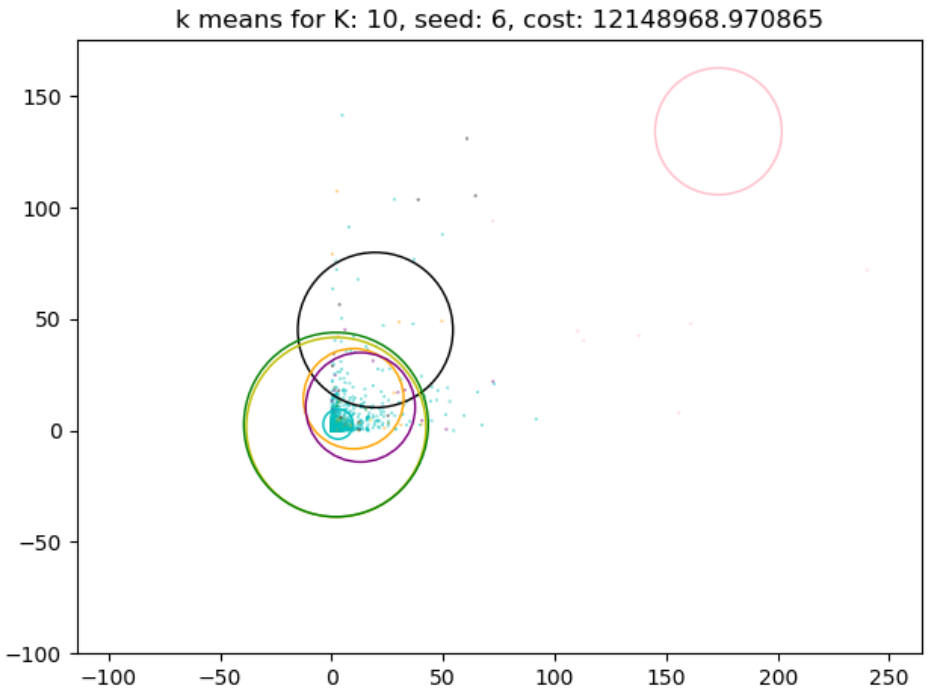



Figure 9-21: K-means Output Plot for K=10 in Audio Mode

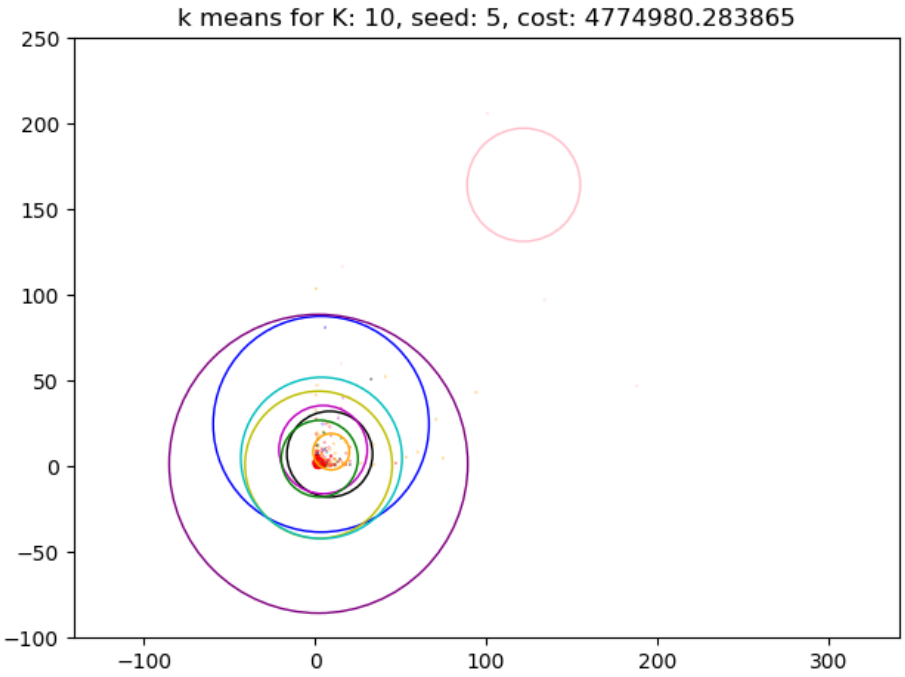



Figure 9-22: K-means Output Plot for K=10 in Whispered Mode

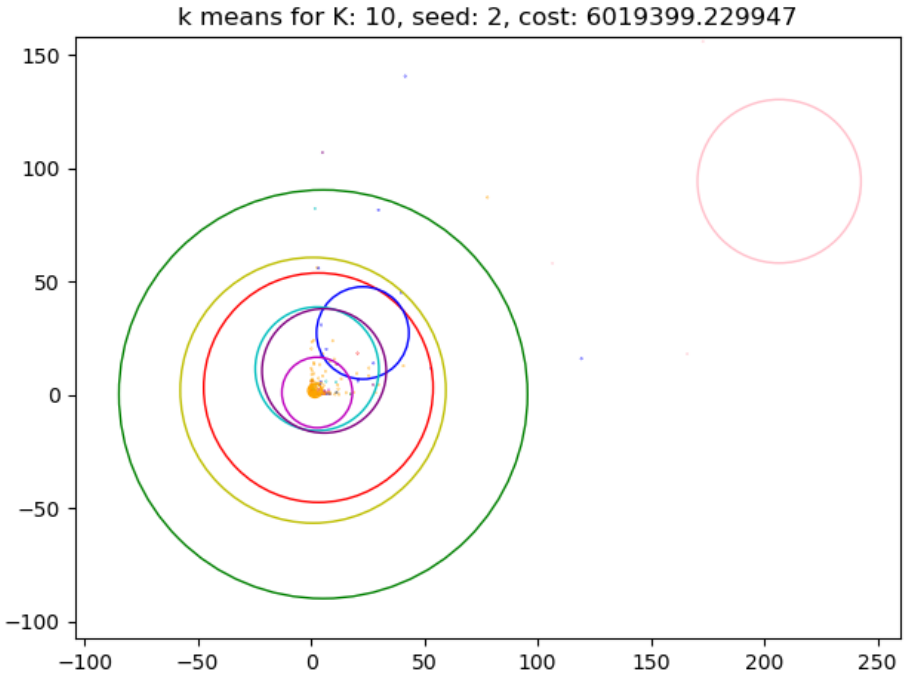



Figure 9-23: K-means Output Plot for K=10 in Silent Mode

The table above shows the cost values obtained in audible, whispered and silent modes for EMG spectrogram feature space with cluster size 10. The number of audible data is relatively higher than that of whispered and silent data so the cost values of audible mode are higher than that of others. Moreover, due to biased, improperly tuned and inadequate data set and some software constraints the cost values of all the modes are very high than expected from the theoretical analysis of the model.

Table 9-4: Cost Values of Feature Space in All Three Modes

| **No. of Clusters (K)** | **Cost values in various modes** | | |
|---|---|---|---|
| | **Audible** | **Whispered** | **Silent** |
| 1 | 48403892.367 | 9941683.388 | 12414276.628 |
| 2 | 24620486.454 | 8754925.771 | 11003163.448 |
| 3 | 20483158.822 | 7375674.222 | 10194351.654 |
| 4 | 18985628.517 | 6906657.715 | 8761844.147 |
| 5 | 15206828.721 | 6428972.626 | 8288547.830 |
| 6 | 14087568.160 | 6028904.735 | 7474337.261 |
| 7 | 13447004.529 | 5535020.849 | 7026519.250 |
| 8 | 12976077.841 | 5418646.888 | 6864662.314 |
| 9 | 12509659.173 | 4545957.983 | 6593630.427 |
| 10 | 12148968.970 | 4774980.284 | 6509246.797 |

## 10. ANALYSIS AND DISCUSSION

Due to the inherent noises, imprecise values of resistors and capacitors, stray inductance and capacitance, wire resistance and many other parameters, the output of the hardware component was not obtained as ideal expectation rather some magnitude shift, phase shift along with noise were found in the output signal. During the design of amplifier and filter circuits the non-significant values beyond the decimal place were neglected and the resistor value as calculated were unavailable in the market. Moreover, the tolerance of the passive elements like resistor and capacitor were not taken into account during the calculation and design of the circuit. The effect of such neglected parameters was later found while analyzing the circuit in practice. The amplification factor of the amplifiers was attenuated due to which the output signal was attenuated by some factor. Similarly, the roll-off factor of the filters was not as calculated as a result of which transition band was extended and the noises from the frequency bands which were expected to be eliminated by the filter were introduced in the output signal. Also, magnitude as well as phase shifts were found due to imperfections in the filter and amplifier circuit.

The data between the EMG acquisition hardware and a laptop was shared using a wired connection. This setup introduced noise in the signals, especially when the charger of the laptop was connected while receiving the data. The crosstalk between the electrode wires also introduced further noise in the system. Furthermore, the Arduino's ADC is unipolar which means it cannot sample the negative voltages from the signal which causes the loss of all the data in the negative domain.

The placement of the signal and ground electrodes has a direct impact on the magnitude and frequency of signal generated. Only a slight displacement in the electrode position affected the output signal by a large factor. The signal output was also affected by the attenuation introduced in the skin layer that lies between the electrode and the muscle. Also, after prolonged use, the Ag in the Ag-AgCl electrode starts diminishing resulting in weaker ionic potential for the same muscle activity.

Due to the contemporary global issue of the COVID-19 pandemic, the hardware parts could not be imported and thus the proposed 6 channel hardware could not be fabricated.

As a consequence, only dual channel EMG acquisition hardware could be fabricated. This reduced the information contained in the signals for each utterance by a significant amount.

The dataset for this project has been optimized from the EMG-UKA Trial Corpus. The original dataset was initially collected to perform Automatic Speech Recognition tasks. The corpus has audio signals, corresponding EMG signals and their transcripts in three different modes: Spoken, Whispered and Silent. The EMG signals for each of the modes were trimmed using their transcripts which was not totally accurate [29]. Due to this, the optimized dataset was imperfectly transcribed which introduced major errors in the machine learning models. Another major problem faced was inadequate data in each mode and uneven distribution of data for all the words which ultimately created a bias. This bias made the machine learning models to generalize better for classes with more number of samples and failed to generalize overall. The distribution of data after word-by-word segmentation is shown in the figure 10-1. It can be clearly observed that the number of samples for the words “THE”, “TO” and “A” in all the modes is overwhelmingly high in comparison to other classes.

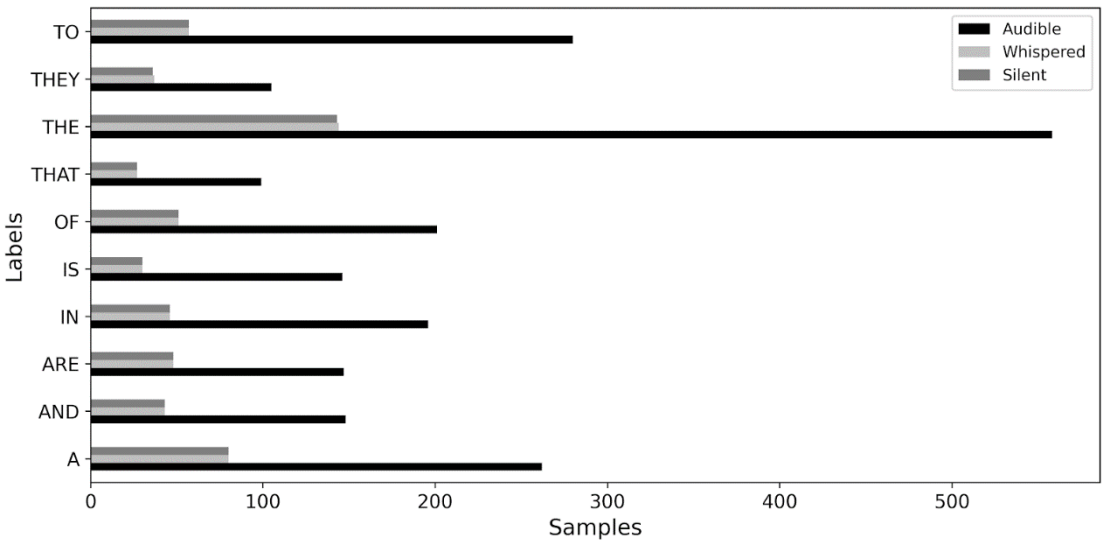



Figure 10-1: Distribution of Data after Segmentation

The segmented dataset used in the training of the machine learning models is not the only reason for the generalization error of models. From the preliminary analysis of the test

and train accuracies of the models (as seen in the tables 9-1, 9-2 and 9-3), it can be suspected that the models are highly overfitting the training data. Moreover, the accuracy from the models tested before was only trained on only one variation of train-test set. This does not give the correct approximation of how well the model will generalize for unseen data. Thus, a 10-fold cross validation was performed on each of the models in all the speaking modes. The output of the 10-fold cross validation was then plotted as box plots as shown in the figure 10-2. The x-axis of the plot represents the different speaking modes and the y-axis represents the plot. Each plot is divided into two sections by a dashed line in between that separates the models using the different feature type. The minimum, maximum and median of the box plot corresponds to the minimum, maximum and median accuracies of the models.

The 10-fold cross validation was first performed on KNN and the output is shown in the figure 10-2 below. From the figure, it can be deduced that the KNN model has a high generalization error with accuracies below 30% in whisper and silent mode. The model does generalize better for audible mode with spectral features but the accuracy is still negatively skewed with the majority of accuracies lying below the median accuracy.

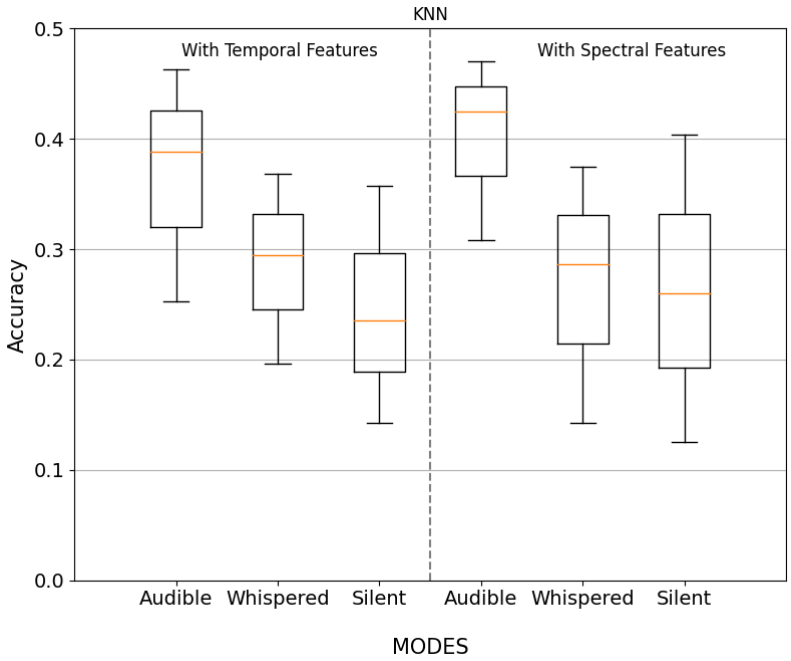



Figure 10-2: KNN Box Plot (10 Fold CV)

The box plot of the MLP model derived after performing 10-fold cross validation is shown in the figure 10-3 below. It can be seen from the figure that the MLP model too fails to generalize for unseen data as the median accuracies for all the modes is low with highest being only 36.50% for audible mode using spectral features. It can also be concluded that the model is overfitting the training data as the difference between the median accuracy and the train accuracy is quite large. The overfitting problem can be addressed by using regularization.

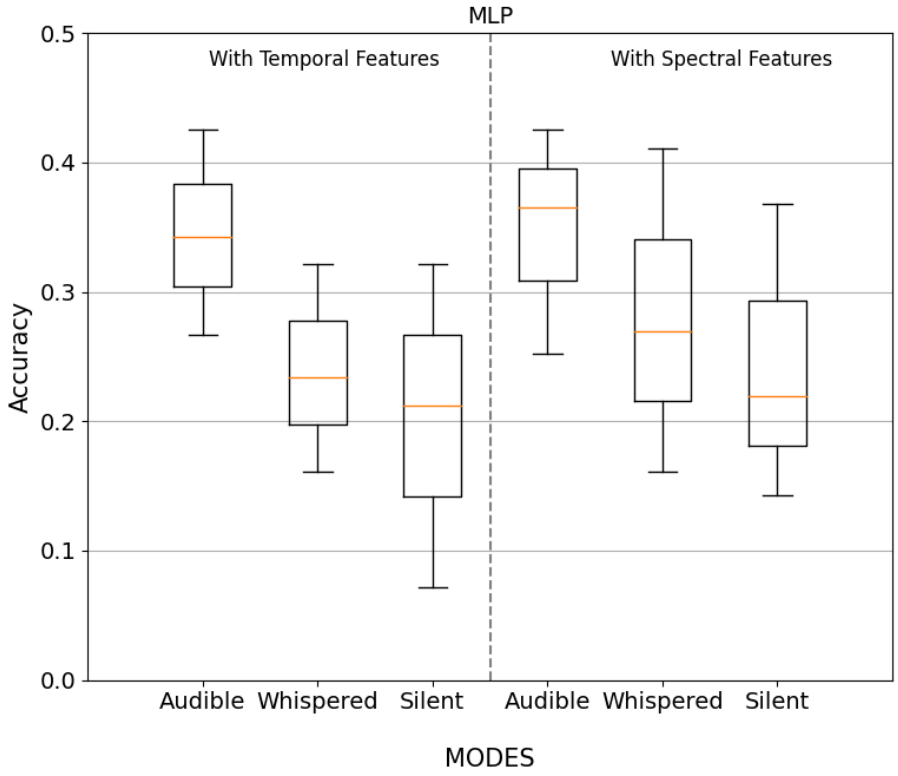



Figure 10-3: MLP Model Box Plot (10 Fold CV)

Finally, the 10-fold cross validation was performed on CNN and the box plot of the output was plotted as shown in the figure 10-4. It can be interpreted from the figure that the CNN too has failed to generalize for the unseen data. This may be due to overfitting of the model or less amount of training data. The maximum accuracy observed for CNN is 45.32% for audible mode using spectral features and the median values for all the modes lies below 37.50%.

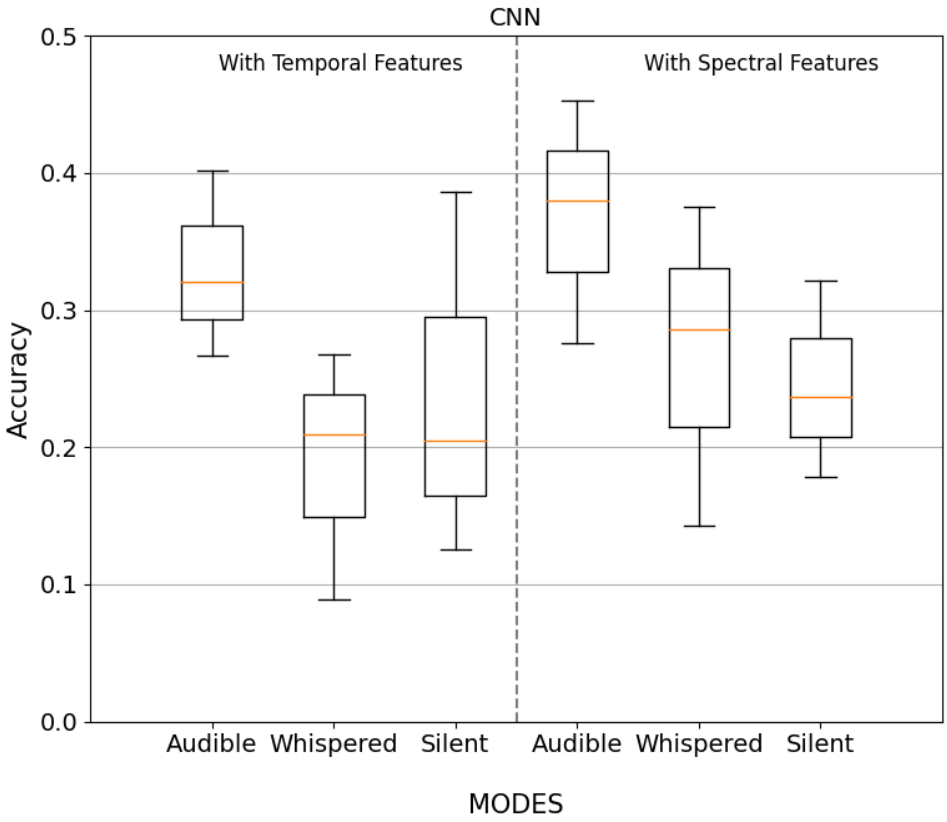



Figure 10-4: CNN Model Box Plot (10 Fold CV)

A summary table for the minimum, maximum and the median values for all the models in different speaking modes while using temporal features and spectral features is shown in the tables 10-1 and 10-2.

Table 10-1: 10 Fold Cross Validation for Temporal Features

| **MODES** | **KNN** | | | **MLP** | | | **1D CNN** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Min** | **Max** | **Avg** | **Min** | **Max** | **Avg** | **Min** | **Max** | **Avg** |
| Audible EMG | 25.23 | 46.26 | 38.84 | 26.55 | 42.52 | 34.2 | 26.63 | 40.18 | 32.07 |
| Whisper EMG | 19.64 | 36.84 | 29.45 | 16.07 | 32.17 | 23.43 | 8.92 | 26.78 | 20.95 |
| Silent EMG | 14.28 | 35.74 | 23.51 | 7.14 | 32.14 | 21.20 | 12.50 | 38.59 | 20.46 |

Table 10-2: 10 Fold Cross Validation for Spectral Features

| MODES | KNN | | | MLP | | | 1D CNN | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Min** | **Max** | **Avg** | **Min** | **Max** | **Avg** | **Min** | **Max** | **Avg** |
| Audible EMG | 30.84 | 46.98 | 42.93 | 25.23 | 42.85 | 36.50 | 27.57 | 45.32 | 37.95 |
| Whisper EMG | 14.28 | 37.50 | 28.60 | 16.07 | 41.74 | 26.99 | 14.28 | 37.50 | 28.62 |
| Silent EMG | 12.50 | 40.35 | 25.99 | 14.28 | 36.84 | 21.89 | 17.85 | 32.14 | 23.69 |

For the selection of the optimum model for the classification task, all three models were analysed together using box plots. Since the spectral features were seen optimal in the previous plots, only the models using the spectral features have been analysed. The box plot for model selection can be seen in the figure 10-5. At a glance, KNN seems to be the optimal model as it has high maximum accuracies and high median values. But after careful analysis, it can be observed that the accuracy for KNN has a wider spread and ranges as low as 12.50%. Meanwhile, the spread of MLP is low compared to KNN but the median value ranges from 21.89% to 36.50%. Lastly, the CNN model has a narrower spread and the median value ranges from 23.69% to 37.95%. Both the CNN and MLP model while using spectral features have comparatively similar performance. The selection between the two can be done after tuning the hyperparameters for maximum accuracy for both the models.

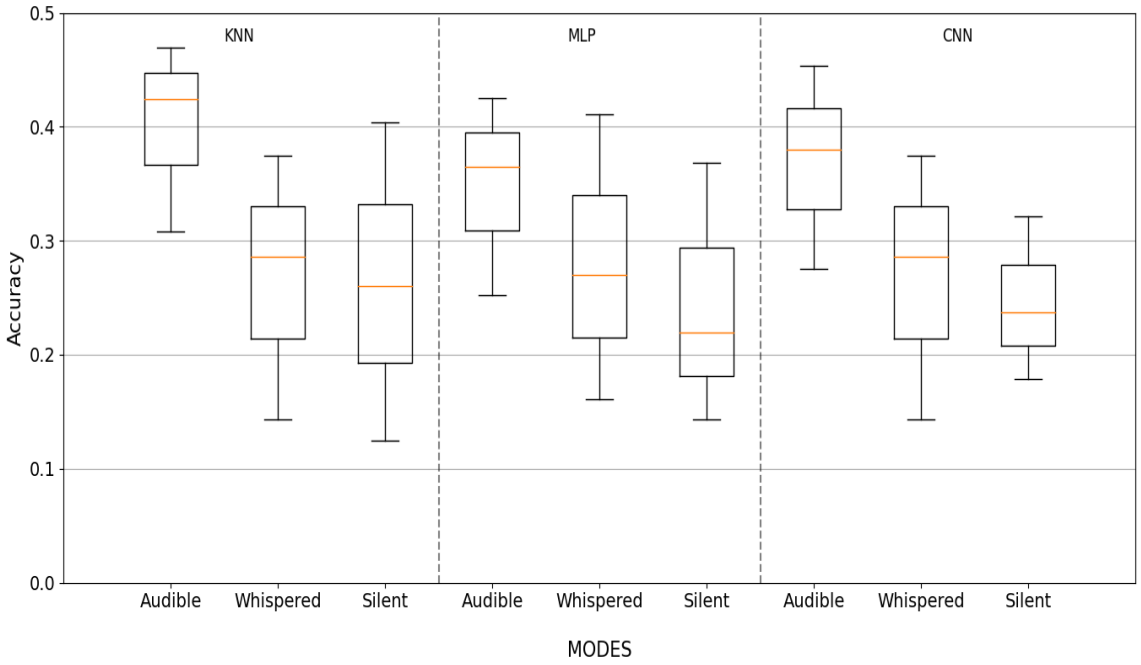



Figure 10-5: Model Comparison with Spectral Features

The hyperparameters in a machine learning model plays a key role in its performance. The table 10-3 below shows the testing done using different hyperparameters of the MLP model. The table shows the test parameters are the optimum parameters found during the test and their corresponding accuracy.

The MLP model was tested using varying numbers of epochs, batch sizes, hidden nodes and dropout rates. The activation functions and optimizing functions were also varied during the testing. The testing of the parameters was done using a grid search algorithm with ten cross validation sets. The best parameters for the MLP model were found to be 100 epochs with a batch size of 80 when using ReLU activation function and Adam as an optimizer. The learning rate of 0.003 gave the highest performance and the best number of hidden nodes was found to be 64 for each hidden layer. Also, the 70% dropout rate for the regularization gave the highest accuracy. All the hyperparameters were tested individually (except for learning rate and dropout rate which were tested in conjunction) because of the limitation in computing power. Due to this, the optimum parameters found are not yet optimal.

Table 10-3: Hyperparameter Tuning for MLP Model

| **Hyperparameters** | **Tested Parameters** | **Optimum Parameter** | **Accuracy(%) for Optimum Parameter** |
|---|---|---|---|
| Epochs | 20, 50, 100, 200 | 100 | 18.89 |
| Batch Size | 20, 50, 70 , 80, 100, 150 | 80 | 18.89 |
| Hidden Nodes | 64, 100, 150, 200, 300 | 64 | 20.14 |
| Activation Function | ReLU, Sigmoid, Tanh, Linear | ReLU | 16.76 |
| Optimizer | SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam | Adam | 17.82 |
| Learning Rate | 0.000001, 0.00001, 0.00003, 0.0001, 0.0003, 0.003,0.03,0.3 | 0.003 | 24.06 |
| Dropout Rate | 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 | 0.7 | 24.06 |

The table below shows the mean and standard deviation of the clustered data by K-means algorithm in all three modes. Mean is Kxd dimension array with each row corresponding to Gaussian component mean while S.D is the standard deviation of the component. The audible and whispered EMG data have the highest number of data with label ‘THE’ while the silent mode seems to have highest data of word ‘THAT’. Some labels have zero standard deviation due to very less number of data in that feature space. From the analysis of the statistical data tabulated above it can be concluded that the dataset of all three modes are highly biased and even some of the data points are so much less that any feature sample extracted may not be the best estimation of the population.

Table 10-4: Mean and SD for K-means

| Labels | Audible Mode(K =10) | | Whispered Mode(K=10) | | Silent Mode(K =10) | |
|---|---|---|---|---|---|---|
| | **Mean(μ1, μ2)** | **S.D(σ)** | **Mean (μ1, μ2)** | **S.D(σ)** | **Mean(μ1, μ2)** | **S.D(σ)** |
| A | (0.13, 3.18) | 0.00 | (2.08, 2.04) | 3.11 | (0.81, 2.97) | 63.96 |
| AND | (2.08, 0.18) | 0.00 | (3.63, 24.70) | 62.99 | (22.95, 28.54) | 19.10 |
| ARE | (19.16, 45.11) | 34.78 | (8.67, 7.22) | 25.01 | (0.88, 1.97) | 50.95 |
| IN | (2.06, 1.51) | 40.28 | (2.18, 0.99) | 42.96 | (7.02, 10.22) | 33.99 |
| IS | (0.78, 0.10) | 0.00 | (4.82, 9.84) | 25.75 | (1.92, 12.70) | 26.56 |
| OF | (2.91, 2.85) | 6.68 | (3.82, 4.99) | 47.06 | (1.08, 0.11) | 0.00 |
| THAT | (1.97, 2.66) | 41.32 | (2.78, 4.59) | 22.38 | (206.59,94.14) | 36.07 |
| THE | (173.43,134.21) | 28.42 | (121.93,164.42) | 33.03 | (6.85, 0.22) | 76.38 |
| THEY | (9.82,14.29) | 22.48 | (9.53, 8.64) | 10.62 | (4.81, 2.75) | 28.73 |
| TO | (12.92, 10.44) | 24.53 | (2.09, 1.57) | 87.16 | (1.77, 1.75) | 4.60 |

## 11. ACCOMPLISHED AND REMAINING TASKS

Table 11-1: Accomplished Task and Task Remaining

| S.N | Tasks Accomplished | Tasks Remaining |
|---|---|---|
| 1. | Testing and recording of facial EMG signals from designed circuit | Fine tuning the circuit parameters for optimal data acquisition |
| 2. | Extraction of features from the EMG signals | Fine tuning neural network model parameters and improving the performance |
| 3. | Refining EMG-UKA Trial corpus on the basis of different speech modes | Training and testing of models with dataset obtained from designed hardware |
| 4. | Trial dataset preparation of discretely uttered words | |
| 5. | Testing of both supervised and unsupervised machine learning algorithms | |

# 12. APPENDICES

## 12.1 Project Budget

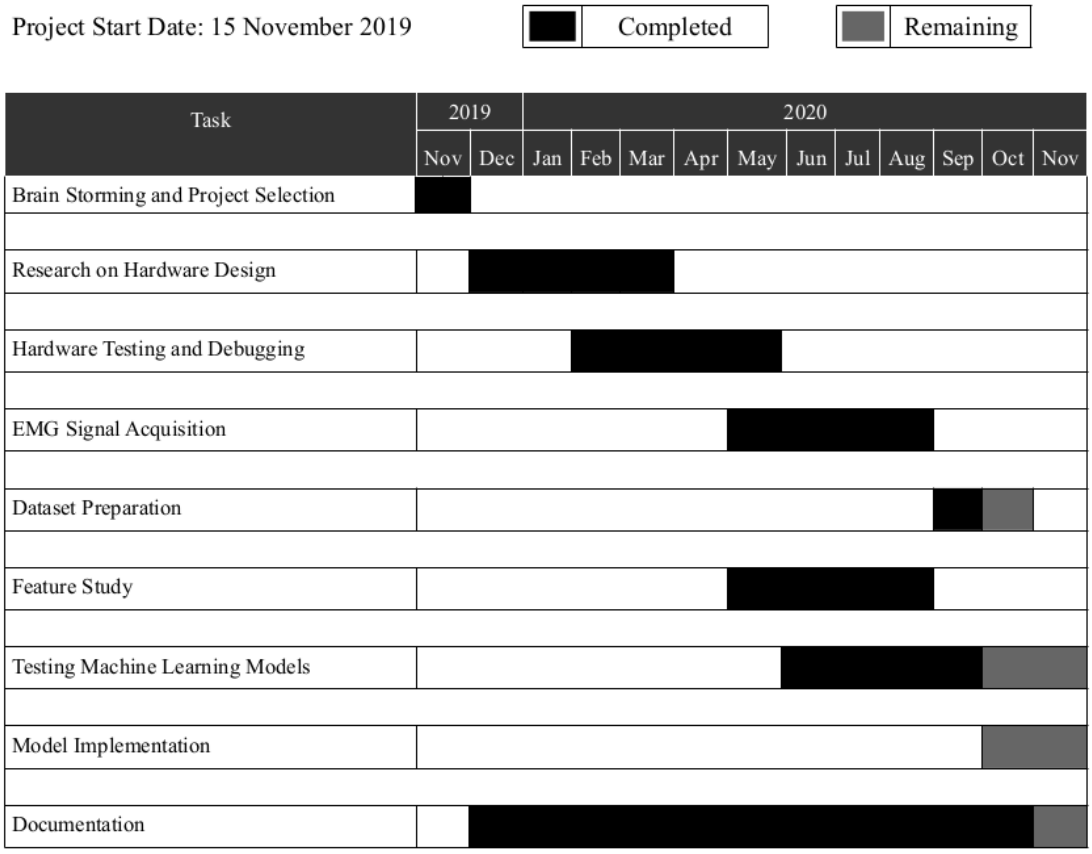
Table 12-1: Budget of Purchased Items

| S.N. | Title | | Model | Quantity (pcs) | Rate (NRs.) | Price (NRs.) |
|---|---|---|---|---|---|---|
| **1** | Ag-AgCl Electrode | | ECG EMG | 150 | 10/- | 1,500/- |
| **2** | Instrumentation amplifier | | AD620 | 5 | 114/- | 570/- |
| **3** | Op-amp | | OP37AJ | 10 | 114/- | 1,140/- |
| **4** | Passive electronic components | • Resistors<br>• Capacitors<br>• Header pins<br>• Diodes | - | - | - | 2,000/- |
| **5** | Electrolyte | | AgCl | 250ml | 50/- | 50/- |
| **6** | Arduino | | Uno | 2 | 1,000/- | 2,000/- |
| **7** | PCB board | | Single sided | 2 | 250/- | 500/- |
| **8** | Shielded Cable | | RCA | 8(1m) | 215/- | 1720/- |
| **9** | Miscellaneous | | - | - | - | 3,000/- |
| | **Total** | | | | | 12,480/- |

## 12.2 Project Timeline

Table 12-2: Gantt Chart

Project Start Date: 15 November 2019




Task Completed: 86.67 %

Task Remaining: 13.33 %

## 12.3 Module Specifications

Table 12-3: Specifications of Instrumentation Amplifier AD620

| S.N. | Parameters | Specifications |
|---|---|---|
| 1. | Gain Range | 1-10,000 |
| 2. | Power Supply Range | ±2.3 V to ±18 V |
| 3. | Max. Supply current | 1.3 mA |
| 4. | Input Voltage Noise | 0.28 μV p-p (0.1 Hz to 10 Hz) |
| 5. | Bandwidth | 120 KHz (G=100) |
| 6. | CMRR | 100 dB min (G=10) |

Table 12-4: Specifications of Amplifier OP37G

| S.N | Parameters | Specifications |
|---|---|---|
| 1. | Open-Loop Gain | 1.8 Million |
| 2. | Max. Supply Voltage | 22 V |
| 3. | Max. Supply Current | 25 mA |
| 4. | Bandwidth | 63 MHz (Common Voltage @ 11V) |
| 5. | Input Voltage Noise | 80 nV p-p (0.1 Hz to10 Hz) |

Table 12-5: Specifications of ADC of Arduino Uno

| S.N | Parameters | Specifications |
|---|---|---|
| 1. | Type | Successive Approximation Register |
| 2. | Resolution | 10 Bit |
| 3. | Absolute Accuracy | ±2 LSB |
| 4. | Conversion Time | 13 - 260μs |

### 12.4 Raw Speech EMG Plot

A sample data from the dataset is shown in the following figures.

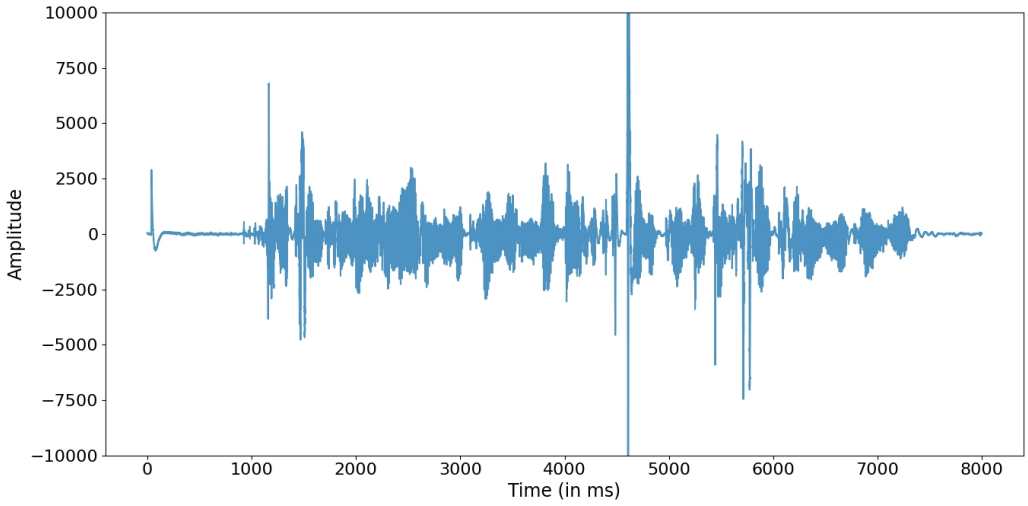



Figure 12-1: Audio Signal

EMG signal data from different channels are as below:

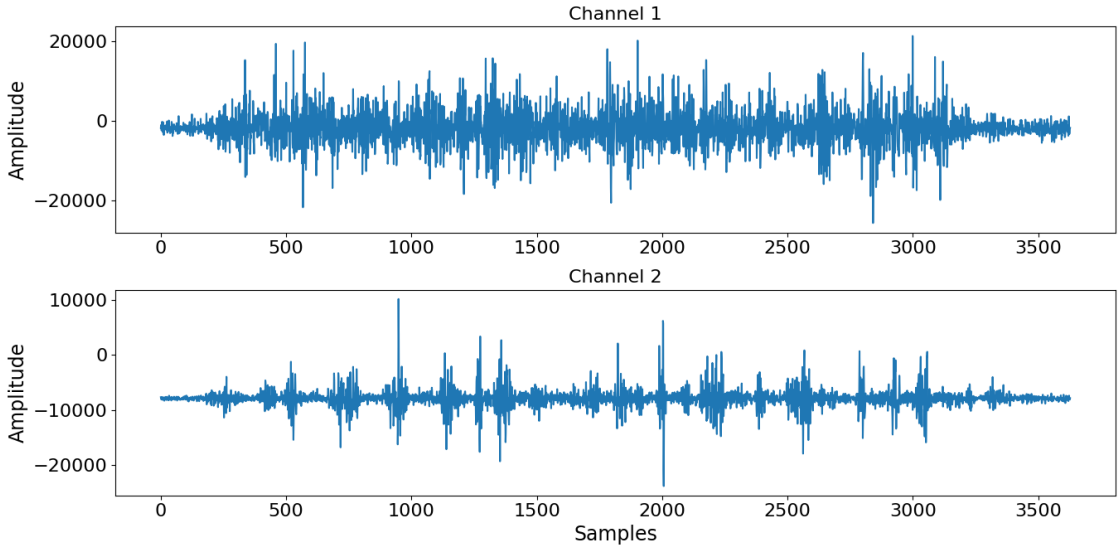



Figure 12-2: EMG Channel 1 and 2 from the Dataset

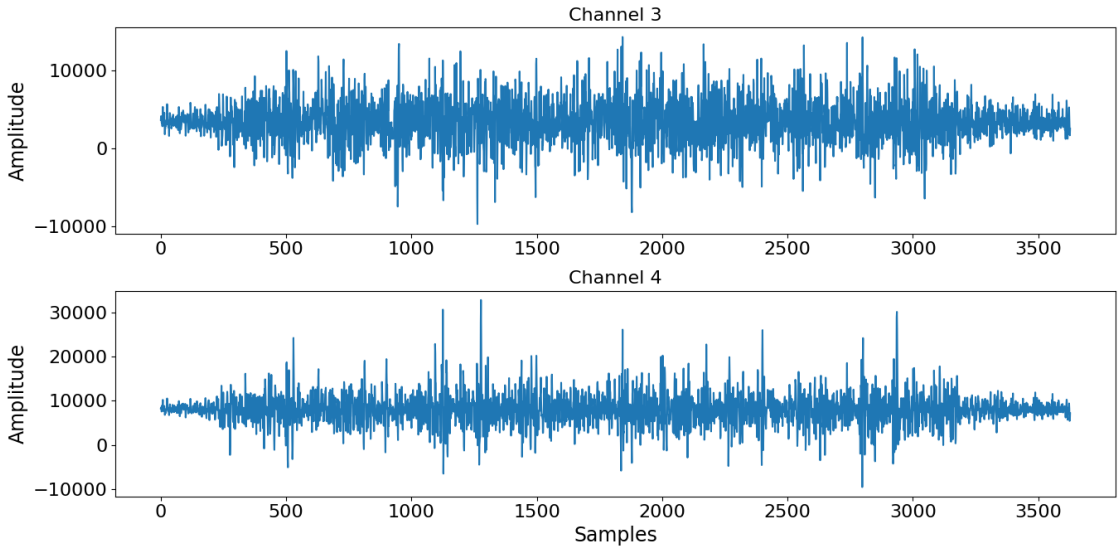



Figure 12-3: EMG Channel 3 and 4 from the Dataset

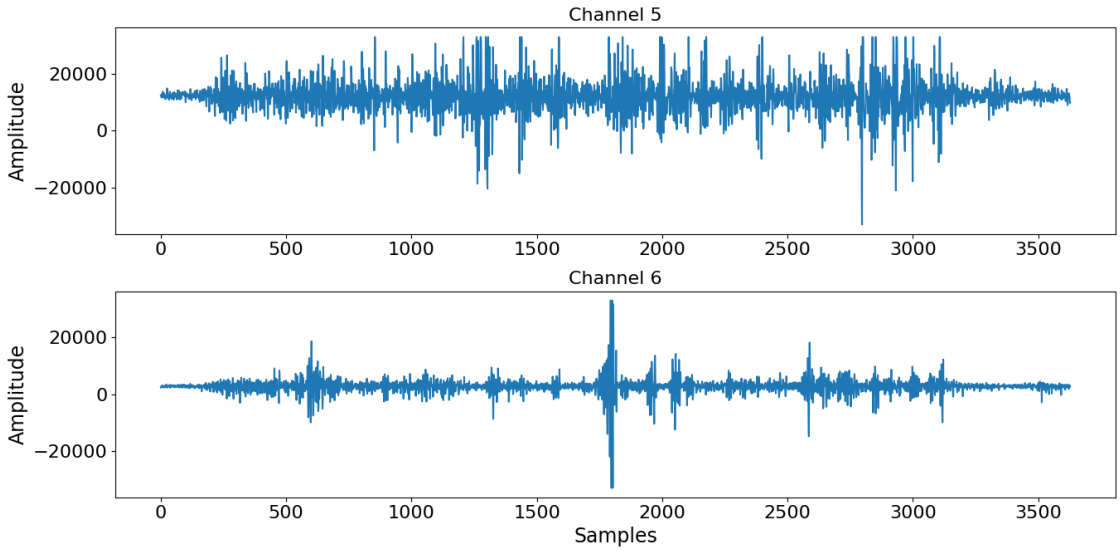



Figure 12-4: EMG Channel 5 and 6 from the Dataset

Above figures show EMG signals obtained from the dataset during citation of the sentence “THIS COUNTRY HAS RELIED ON IMMIGRANTS AND IS FOUNDED UPON A PRINCIPLE OF WELCOMING IMMIGRANTS” by the first speaker (profiled as “002”) during the first session. The figure 12-1 is the audio signal of the utterance and figures 12-2, 12-3 and 12-4 are the EMG signals of the utterance of the sentence from 6 different articulatory muscles as 6 different channels.

## References


[1] Greg Gage,Timothy Marzullo, 2009. [Online]. Available: https://backyardbrains.com/about/. [Accessed december 2019].

[2] A. Kapur, "Human-Machine Cognitive Coalescence through," MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2018.

[3] Arnav Kapur,Shreys Kapur, Pattie Maes, "AlterEgo," *Multimodel Interface,* 2018.

[4] G. Gage, "Control Machines with your Brain," *Backyardbrains,* 2009-2017.

[5] Michael Wand and Tanja Schultz, "SESSION-INDEPENDENT EMG-BASED SPEECH RECOGNITION," *Cognitive Systems Lab, Karlsruhe Institute of Technology, Adenauerring 4, 76131 Karlsruhe, Germany,* pp. 1-3.

[6] Vanderthommen Marc, Duchateau Jacques, "Electrical Stimulation as a Modality to Improve Performance of the Neuromuscular System," *Exercise and Sport Sciences Reviews,* vol. 35, pp. 180-185, 2007.

[7] J. Herbert, "The principles of neuromuscular electrical stimulation," *Nursing Times,* vol. 99, p. 54, May 2003.

[8] M.Khan, M. Jahan, "The Application of AR Coefficients and Burg Method in Sub-vocal EMG Pattern Recognition," *Journal of Basic and Applied Engineering Research,* vol. 2, pp. 813-815, April-June 2015.

[9] Geoffrey S. Meltzner, James T. Heaton, "Development of sEMG sensors and algorithms for silent speech recognition," *Journal of Neural Engineering*, vol. 15, no. 4, 25 June 2018.

[10] Chuck Jorgensen and Kim Binsted, "Web Browser Control Using EMG Based Sub Vocal Speech Recognition," in *38th Hawaii International Conference on System Sciences, IEEE*, Hawaii, 2005.

[11] Jennifer C. Shieh, Matt Carter, Guide to Research Techniques in Neuroscience, Second Edition ed., 2015.

[12] Arthur C. Guyton, John E. Hall, Textbook of Medical Physiology, Eleventh Edition ed., Elsevier Inc., 2006.

[13] G. Kamen, David A. Gabriel, "Essentials of Electromyography," in *Human Kinetics*, 2010, p. 57.

[14] K Sembulingam, Prema Sembulingam, Essentials of Medical Physiology, Sixth ed., Jaypee Brothers Medical Publishers (P) Ltd, 2012, pp. 197-199.

[15] "Phonation," [Online]. Available: https://www2.ims.uni-stuttgart.de/EGG/page6.htm.

[16] Jacob MIllman and Christos C. Halkias, Integrated Electronics: Analog and Digital Circuits and systems, McGraw-Hill Kogakusha. Ltd., 1972, pp. 501-534.

[17] F. Najmabadi, "ECE65 Lecture Notes,"," Spring 2007.

[18] M. Valkenburg, Analog Filter Design,, CBS College Publishing, 1982, pp. 157-167.

[19] "Understanding FFTs and Windowing," National Instruments, 05 03 2019. [Online]. Available: :https://www.ni.com/en-us/innovations/white-papers/06/understanding-ffts-and-windowing.html.

[20] Ulysse Cote-Allard, Evan Campbell, Angkoon Phinyomark, Francois Laviolette, Beniot Gosselin, Erik Scheme, "Interpreting Deep Learning Features for Myoelectric Control: A Comparison With Handcrafted Features," *Frontiers in Bioengineering and Biotechnology,* 2020.

[21] A. Mertins, Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications, John Wiley & Sons Ltd, 1999.

[22] Mariusz Kubanek , Janusz Bobulski and Joanna Kulawik, "A Method of Speech Coding for Speech Recognition Using a Convolutional Neural Network," *Symmetry*, vol. 11, p. 9, 19 09 2019.

[23] "Wikipedia," Wikipedia Inc., [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

[24] "Automatic Speech Recognition," [Online]. Available: https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC.

[25] McAdams, Eric, in *Bioelectrodes*, 2006.

[26] M. Jamal, "Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis," *Computational Intelligence in*

*Electromyography Analysis - A Perspective on Current Applications and Future Challenges,* 2012.

[27] I. Analog Devices, *AD620 Datasheet.*

[28] I. Analog Devices, *OP37 Datasheet.*

[29] Michael Wand, Matthias Janke, Tanja Schultz, "The EMG-UKA Corpus for Electromyographic Speech Processing".

[30] Choi, Heung & Kim, Jung-Ho & Kwon, Jang-Woo, "Performance Improvement of EMG-Pattern Recognition Using MFCC-HMM-GMM," *Journal of Biomedical Engineering Research,* 2006.

[31] "CS231n Convolutional Neural Networks for Visual Recognition," Stanford University, 2019.

[32] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING,* vol. 22, no. 10, 10 2014.

[33] J. Herbert, "The principles of neuromuscular electrical stimulation," *Nursing Times,* vol. 99, p. 54, May 2003.

[34] B. Farnsworth, "https://imotions.com/blog/electromyography-101/," 24 july 2018. [Online]. Available: https://imotions.com/blog/electromyography-101/. [Accessed December 2019].

[35] S.C. Prasanna Kumar, Arathi Chandrasekar, Arushi Nagaraj, Parul Gupta, Sheetal Sekhar, "Design of an ElectroEncephaloGram (EEG) Amplification Circuit for Neonates," *International Conference on Communication and Signal Processing, IEEE,* pp. 1-3, 2016.

[36] M. K. Das, Frequency response of high pass Butterworth RC filters using operational amplifiers, vol. 2, International Journal of Advanced Science and Research, Sepetember 2017, pp. 135-138.

[37] "ADC Tutorial : Analog to Digital Conversion," [Online]. Available: http://www.robotplatform.com/knowledge/ADC/adc_tutorial.html. [Accessed Feb 2020].

[38] "Arduino ADC," [Online]. Available: https://www.best-microcontroller-projects.com/arduino-adc.html. [Accessed Feb 2020].

[39] Mc Adams, Eric, in *Bioelectrodes*, vol. 148, Ann. New York Acad. Sci, 2006.

[40] J. Herbert, "The principles of neuromuscular electrical stimulation," *Nursing Times,* vol. 99, p. 54, 2003.

[41] Eduardo Lopez-Larraz, Oscar M. Mozos, Javier M. Antelis, Javier Minguez, "Syllable-Based Speech Recognition Using EMG," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* 2010.

[42] Anthony J. Seikel, Douglas W. King, David G. Drumright, Anatomy and Physiology for Speech, Language, and Hearing Textbook, Fourth Edition ed., 2009.

[43] Kusuma Mohanchandra, Snehanshu Saha, "A Communication Paradigm Using Subvocalized Speech: Translating Brain Signals into Speech," *Augmented Human Research*, vol. 1, no. 3, 2016.

[44] Shibli Nisar, Omar Usman Khan and Muhammad Tariq, "An Efficient Adaptive Window Size Selection Method for Improving Spectrogram Visualization," *Computational Intelligence and Neuroscience*, 2016.

[45] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, 2017.