ELEC4848: Senior Design Project 2018-19

Brain-Computer Interfacing System

Final Report

Supervisor

Dr Anthony Tam

Department of Computer Science

University of Hong Kong

Team Members

Rohak Singhal *(Author)*	3035242475
Sagar Gupta	3035238785

8th April 2019

Abstract

Recent developments in technology has enabled us to interact with our machines in novel ways - touch interaction in smartphones, voice recognition in home assistants like Alexa and biometrics using fingerprint scanners. These developments seemed no more than science fiction a few decades ago, and so did an interface linking human thoughts to commands on a computer. Brain-Computer Interface (BCI) is a technology with a variety of applications from medicine to gaming. This final year project aims to make use of a relatively low cost and commercially available BCI devices to create a product demonstrating the applications of a BCI system. EEG Hardware is used to extract electroencephalography (EEG) signals non-invasively from a user. The signals are then processed and Machine Learning techniques are used to map the signals to left and right commands. The best performing model uses 6 temporal and spatial filters with LDA as the classification algorithm. The commands are then used to control a simple game to demonstrate the power of the BCI system.

Acknowledgements

I would like to thank my project supervisor, Dr Anthony Tam for his continued support and guidance throughout this project. In addition, I would like to thank the Faculty of Engineering and the EEE Department for giving me the opportunity to work on such a project and providing me with the resources necessary for the same.

Contents

List of Figures

List of Tables

Abbreviations

ALS	Amyotrophic lateral sclerosis
BCI	Brain Computer Interface
EEG	Electroencephalogram/Electroencephalography
EMG	Electromyography
fMRI	Functional Magnetic Resonance Imaging
GUI	Graphical User Interface
OpenBCI	Open-source Brain Computer Interface
ERP	Event Related Potential
SVM	Support Vector Machines
LDA	Linear Discriminant Analysis
DTFT	Discrete-Time Fourier Transform
FFT	Fast Fourier Transform

1 Introduction

Recent developments in technology have enabled us to interact with our machines in novel ways - touch interaction in smartphones, voice recognition in home assistants like Alexa and biometrics using fingerprint scanners. These ideas seemed no more than science fiction a few decades ago, and so did an interface linking human thoughts to commands on a computer.

1.1 Brain Computer Interfaces (BCI)

BCIs are a mode of interacting with computers using signals generated from the human brain, in response to an internal or external stimulus. More specifically, a BCI system functions such that the signals produced by an individual do not travel through “normal output pathways” in neuromuscular systems and encodes them in measurable activity (like EEG signals) [1]. In practice this has the potential to translate human thought or muscle movement into a domain-specific command such as moving a player in a video game or controlling a prosthetic limb. Conventionally, BCI systems have been used in medicine, in attempts to treat patients with disorders involving neuromuscular impairments such as cerebral palsy or amyotrophic lateral sclerosis (ALS) [1]. The issue with sophisticated BCI systems is that they are usually invasive and may cost between $5,000-$10,000 [2]. As a consequence, the end product may not be commercially viable at present. The project aims at exploring the potential of BCI systems and exploring the feasibility of utilizing them to provide a better user experience. The motivation behind this project is to also develop a BCI system that is low in cost and non-invasive, thereby making it both commercially viable and safe for use by healthy individuals.

1.2 Project Objectives and Significance

For this final year project, the team has proposed to make use of low-cost, open-source and commercially available devices capable of recording brain signals (EEG) to build a Brain Computer Interface (BCI) application. This project aims to explore the accuracy of such a system built using a relatively low resource and low cost hardware. Low cost devices in this domain tend to be less accurate, which creates an additional challenge; Current systems making use of such technology do exist, however they face issues in terms of classification accuracy. To tackle this issue, we plan to use tools such as Machine Learning to aid the mechanism of classification and employ statistical methods to extract the most useful information from the produced signals. Lastly, we aim to establish a well-documented process of developing a BCI system from scratch so that other university students / researchers and hobbyists are able to continue research in this field.

1.3 Expected Deliverables

The primary deliverable for the project is a fully functional BCI system capable of extracting relevant signals from a user and translating it into corresponding commands, visible in real time. The system is comprised of a hardware device for the extraction, a software module that processes and classifies the recorded signals, and an application that aids in visualizing the results of the thought to command mapping (such as a video game). Experiments are carried out using the OpenBCI 4-channel board (OpenBCI Ganglion), 8-channel board (OpenBCI Cyton) and 16-channel board (OpenBCI Cyton and Daisy).

1.4 Outline of the Report

The major components of the report along with a brief summary of each chapter is listed below. Chapter 1 introduces the topic and objectives of the project and describes its scope and expected deliverables. Chapter 2 goes over the theoretical base in key areas on which the project is built. Chapter 3 describes the technologies used in this project and elaborates on our methodology. Chapter 4 discusses all the results obtained from various experiments conducted. Chapter 5 summarizes the key ideas of the report and concludes by explaining how the current findings influence the work to be carried out in the future.

2 Theoretical Background

In this section, we go through the major advancements in three fields which are important to this project: Brain Computer Interface research, Signal Processing and Machine Learning.

2.1 Brain Computer Interface (BCI) research

This section talks about the different functions of the Human Brain and the signals produced by it, the techniques to capture different signals and the latest developments in BCI.

2.1.1 The Human Brain

The brain is arguably the most important organ of the human body, but it is yet to be completely understood. For the purpose of building a BCI application, certain areas of the brain are of key importance. Figure 2-1 shows the anatomy and functional areas of the brain.

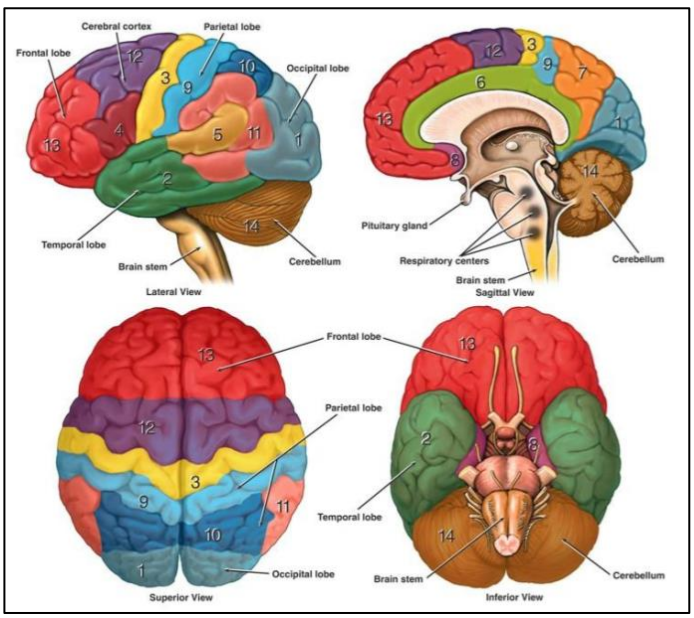

Figure 2-1 Functional Areas of the Brain [3]

The 4 main lobes which are of primary importance in order to acquire signals are Frontal, Parietal, Occipital and Temporal lobes. The frontal lobe is situated at the front and contains the primary motor cortex, which is the region associated with movement and control of the body [4]. The Parietal lobe is responsible for sensory perception, integration and processing of the sensory information [4]. The occipital lobe takes charge of tasks related to sight and the temporal lobe performs auditory processing [4].

2.1.2 Brain Waves – EEG

Neurons form the basic building blocks of the brain and communicate with each other via electrical signals, in the form of ionic currents [5]. EEG is a non-invasive and safe method to record neural activity by measuring electrical potentials from the scalp. The difference in electrical potential between pairs of electrodes on the scalp is measured and tracking these measurements over time generates an estimate of brain activity [6]. The signal obtained from EEG recordings is generally weak due to the non-invasive nature of the technique as well as noise from external sources. However, there are several ways to amplify the signal to acquire more accurate data. One such method is to appropriately position the electrodes on the scalp to correspond to areas where the signals recorded are the strongest and most distinct. There is an international standard, called the 10-20 system, for the placement of these electrodes which results in a better accuracy of recording. The 10-20 system will be discussed in section 2.1.3.

EEG signals comprise of five main bands which correspond to different electrical frequencies. Each band is linked to a specific set of neural activities and may represent different actions or stages of consciousness [7]. Table 2.1 presents the five major bands (waves) classified based on their frequencies and the corresponding set of activities or mental states that they represent.

Frequency Band	Frequency (in Hz)	Mental State Represented
Alpha	7.5 – 12	Alpha waves are associated with a relaxed state of mental activity. Feelings of peace and disengagement lead to an increase of alpha level activity. These waves typically arise from the frontal cortex of the brain. Visualizing calming or peaceful scenarios with the eyes closed typically leads to increased alpha activity [6].
Beta	12 – 30	Beta waves are produced due to a state of focus and concentration. They are best recorded at the central area of the brain. Focused thinking such as that involved in solving a mathematical problem or a challenging puzzle causes an increase in beta activity.

Gamma	31 and above	Gamma waves do not represent a clear state of mental activity like alpha or beta waves but represent the mechanism of consciousness itself. Tasks involving attention, cognition and perception correspond to an increase in both gamma and beta waves [2].
Delta	0.5 – 3.5	Delta waves correspond to a state of complete rest and unconsciousness. They are quite low in frequency and are generally recorded when sleeping. However, if these waves are recorded in a conscious state, it is most often an indication of brain damage [2].
Theta	3.5 – 7.5	Theta waves are found in small amounts in adults and are generally related to a state of daydreaming. They could also manifest as a result of heightened emotion such as anger or disappointment. It is generally difficult to generate theta waves consciously and can be considered as a neutral state.

Table 1 EEG frequency bands

The level of activity in various frequency bands presented in Table 1 can be increased by performing certain activities such as solving a puzzle, focusing on an image or moving a certain body part. This principle forms the basis for mapping various mental states onto commands in the BCI system.

2.1.3 10 – 20 system

The 10-20 system is an international standardization for the placements of electrodes on the scalp [8]. It ensures that the parts of the brain associated with EEG activity are captured and maintains an equal inter-electrode spacing. It utilizes proportional and relative measurements with respect to the size of the skull of the user. Figure 2-2 shows the 10-20 placement, with relative positioning of the electrodes on the skull. It uses the Nasion, Inion and the two Pre-auricular points as reference points to measure the skull. Sub-divisions are then made in 10%, 20%, 20%, 20%, 20% and 10% of the nasion-inion and pre-auricular distances.

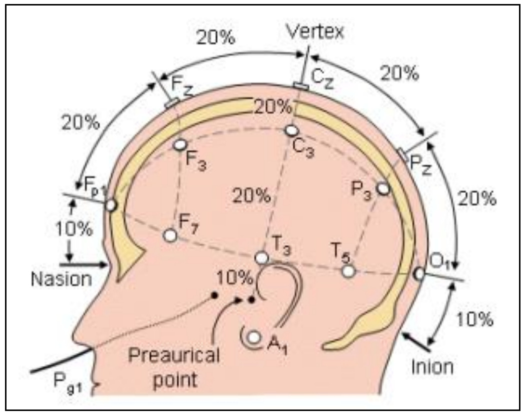

Figure 2-2 10-20 EEG Placement [8]

2.1.4 Motor Imagery

The different functions of the lobes of the human brain are discussed in section 2.1.1. There is substantial activity generated in the areas which correspond to motor control when an individual moves their forelimbs. Motor imagery relies on the fact that when the individual rehearses or simulates the movement of their forelimbs in their minds, without actually performing the action, the same activity is generated in the areas of the brain associated with movement [9]. Figure 2-3 shows the difference in location and power spectrum of the activity recorded in the brain. According to the graphs, the electrode placed at C3 contributes more to the power of the signal recorded while mentally simulating the movement of the left hand.

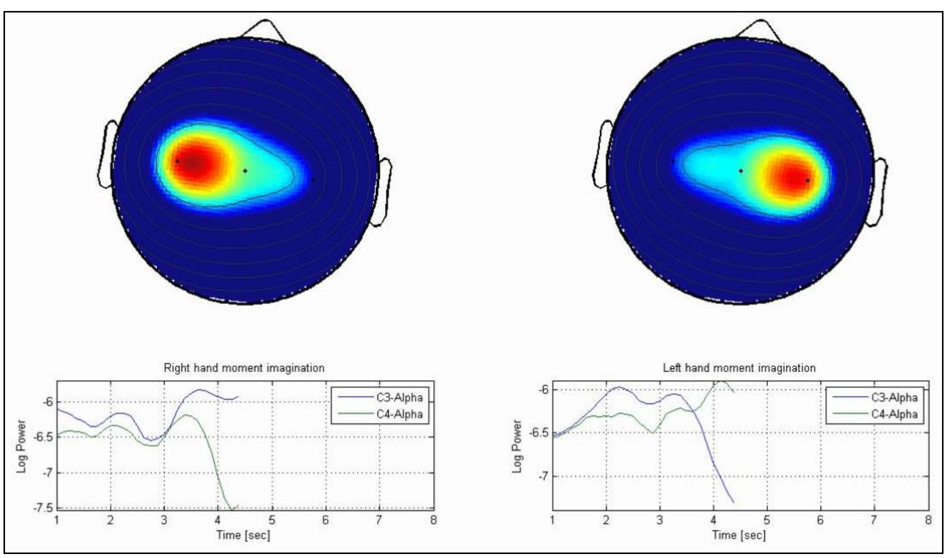

Figure 2-3 Motor Imagery by moving left and right hand respectively [10]

2.1.5 Advancements in BCI research

BCIs have had applications in bioengineering in the past where these devices were used to assist disabled users in performing routine tasks. However, recent years have seen a shift in the application of these systems from being not just a medical aid, but also devices aimed at improving the interactive experience between humans and machines. This subsection explores two major domains where research has led to the development of various kinds of BCI systems.

In the field of medicine, research in BCIs has mainly been in trying to improve the ability of disabled users to communicate with their environment. In 2010 an experiment was carried out on patients with severe epilepsy who were going to undergo surgery to remove the parts of the brain causing repeated seizures. During the surgery, microelectrodes were placed on the brain surface in order to find the seizure causing area. The scientists involved in the procedure took this opportunity to place the electrodes directly over the posterior temporal gyrus (pSTG) which is responsible for speech production (Figure 2-4). Using the EEG signals obtained from this procedure, the scientists were able to correctly match a set of words to the corresponding signal almost 48% of the time [6]. This result provided evidence that BCIs could someday be used to allow disabled people to communicate without having to speak.

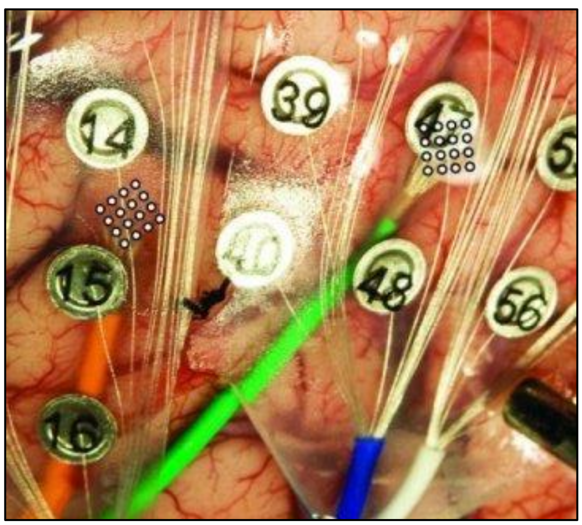

Figure 2-4 Mindspeech invasive BCI

BCIs are also becoming a part of the entertainment industry by providing new gaming experiences. NeuroSky is one of the companies dedicated to providing low cost EEG headsets (biosensors) for the purposes of BCI research. There have been attempts in using their product, the “NeuroSky Mindset” to develop BCI systems. One such example is a functional prototype of the game called "Snake" controlled using EEG signal inputs. It was created by Erik Andreas Larsen, and makes use of signals generated while blinking to control the direction taken by the snake object in navigating the game space [11].

2.2 Signal Processing

This section will elaborate the theoretical basis of some general signal processing techniques applicable to EEG signals. Since our focus is on time based signals, signal processing techniques in the time, frequency and complex frequency domain enable us to derive a greater insight into raw EEG signals.

2.2.1 Epoch

In the context of signal processing, Epochs can be defined as chunks of data formed after slicing the original data into segments. Different epochs can refer to stimulus presentation of different tasks / actions within a single experiment. Figure 2-5 showcases epoching, which is done on the original signal. Four out of the six epochs are marked as 'X', to symbolize the first class of stimulus and the middle two are marked as 'O', symbolizing a different class of stimuli.

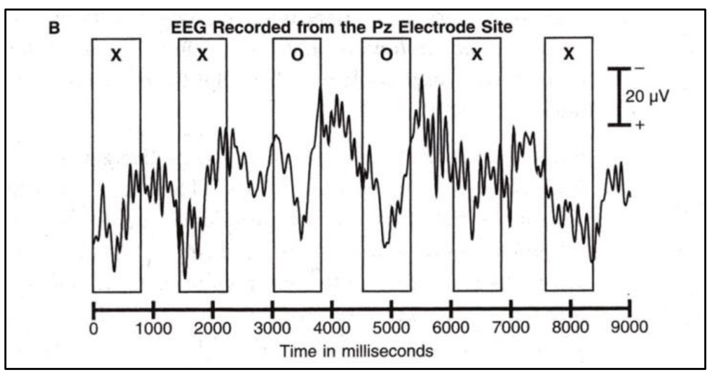

Figure 2-5 Epoching [5]

2.2.2 Signal Average

In signal processing, calculating the average or the mean value for each time-point across all epochs can substantially help out with improving the quality of the raw signal. It helps in increasing the signal-to-noise ratio since the noise, which is assumed to be uncorrelated by definition, averages out with time [12].

2.2.3 Event Related Potential (ERP)

ERP are minute voltages detected by the EEG hardware as a result of specific events or stimuli [13]. ERPs are of great importance since we want to correlate this specific brain activity / voltages with specific tasks in the brain. Thus, ERPs are the meaningful signal extracted from the overall noisy signal. The above mentioned techniques, Epoching and Signal Averaging are useful to calculate the ERPs. Figure 2-6 shows how averaging the epochs in the case of 'X's and 'O's compute two sets of ERPs which may be of use.

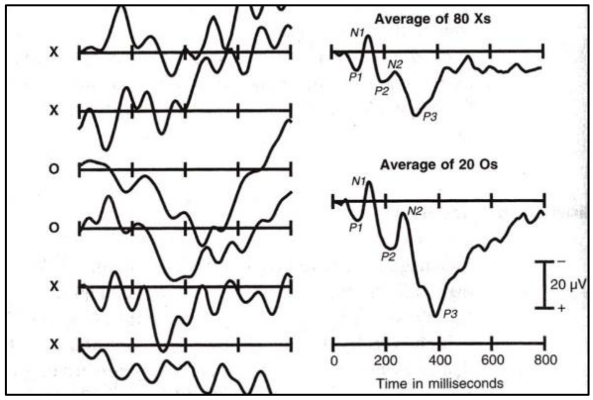

Figure 2-6 ERPs emerging from EEG as trials are averaged [5]

2.2.4 Artifacts in EEG signal

Figure 2-7 shows the most common EEG artifacts.

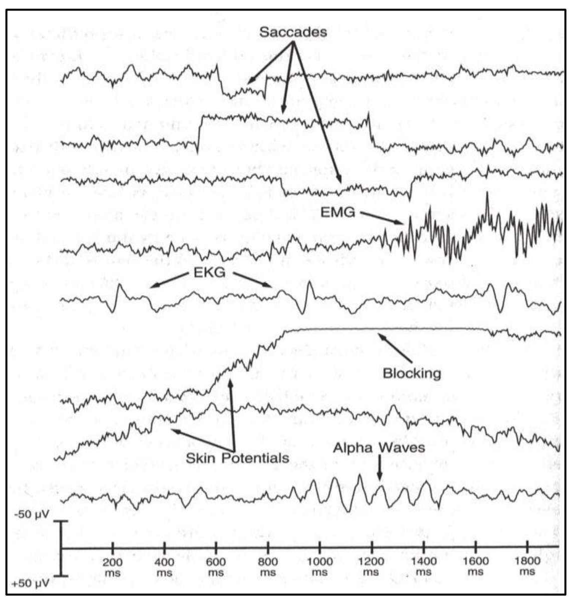

Figure 2-7 EEG Artifacts [5]

Saccades are Eye Blinks, which is a natural phenomenon likely to occur during any BCI experiment procedure. The electrodes can also pick up skin potentials or Electromyography (EMG) signals, which is the activity related to skeletal muscles. The goal of signal processing techniques is to get rid of these artifacts to extract more meaning out of the raw signal.

2.2.5 Spatial Filters

Spatial Filtering is a technique to combine multichannel EEG data recorded from the user linearly, in order to obtain processed information from the raw data. The purpose of such a linear combination of the signal is to obtain new “channels” of data, in which the signal-to-noise ratio is increased as compared to the raw data [14]. Another purpose of Spatial Filters is to remap channel signals in such a way that they closely resemble the source signals coming from the regions of the brain [15]. Figure 2-8 shows how the signals matrix Y is linearly combined using a weight matrix w. Different sets of weights can be applied to the same input signal matrix, to result in a multi-dimensional output matrix [16].

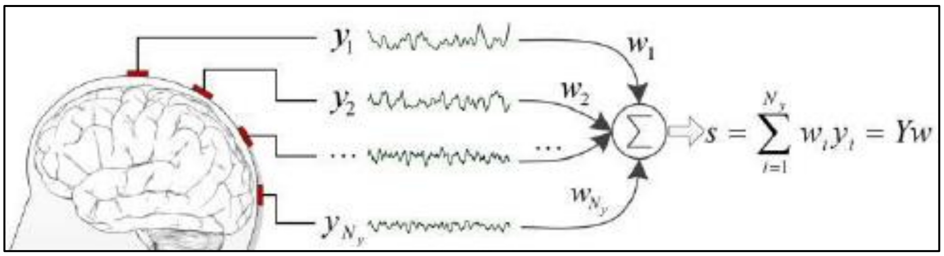

Figure 2-8 The Concept of Spatial Filtering [14]

A wide variety of methods can be applied to learn the set of weights to optimally combine the original signals. Common Spatial Pattern (CSP) is an extensively used algorithm for learning Spatial Filters in EEG applications [15]. Once the signal has been divided into certain classes (for the purpose of classification), CSP computes the weights to maximize the class variance of one and minimize it for the other.

2.2.6 The Fourier Transform

The Fourier Transform converts a time-signal to its frequency domain representation. It divides the frequencies present in the signal into sinusoidal components, by mapping the magnitude of the frequency components onto the unit-circle. This is the Continuous Time Fourier Transform (CTFT):

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega\mathrm{t}}\, dt$$

Where $x(t)$ is the time-signal domain and $X(\omega)$ is its frequency domain representation. Since digital signals are discrete-time in nature, the Discrete Time Fourier Transform (DTFT) is utilized to perform the frequency domain conversion:

$$X\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} x\,[n]e^{-j\omega \mathrm{n}}$$

In practice, we sample the signal at $\omega_{\mathrm{k}} = \frac{2\pi k}{N}$ to perform the Fourier Transform, and we hence obtain the Discrete Fourier Transform (DFT):

$$X[k] = X\left(e^{j\omega_{\mathrm{k}}}\right) = \sum_{n=0}^{N-1} x\,[n]e^{\frac{-j2\pi nk}{N}}$$

Most software incorporate the Fast Fourier Transform (FFT) to calculate the DFT efficiently [17]. The transformation of a time based signal (such as EEG) to the frequency domain allows an ease of analysis, since filters can be efficiently applied such that certain frequencies are allowed / blocked out (further discussed in section 2.2.7). As seen in figure 2-9, conversion to the frequency domain provides an ease in visualization and recognition of spectral patterns [18].

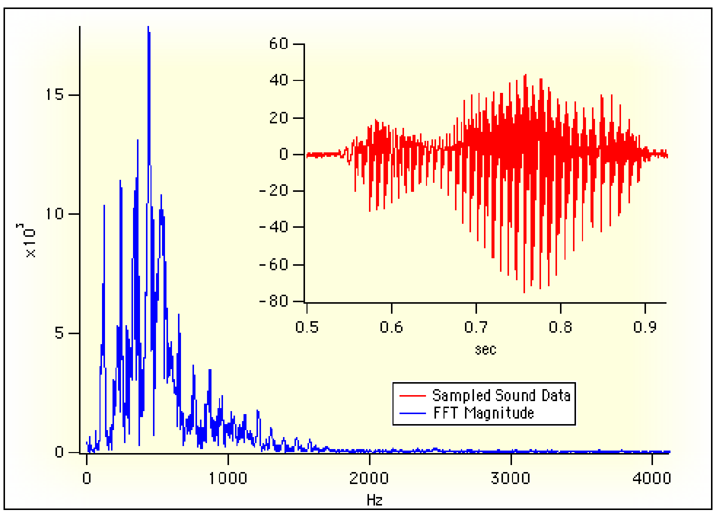

Figure 2-9 Application of FFT [19]

2.2.7 Temporal or Spectral Filters

These are filters designed to act on the Spectrum of the signal. The main purpose of such filters Wis to isolate the waves or ERPs in a particular frequency band [15]. Some examples of such filters would be High-pass, Low-pass, Bandpass, and Notch filters. Figure 2-10 shows the effect of some commonly used filters.

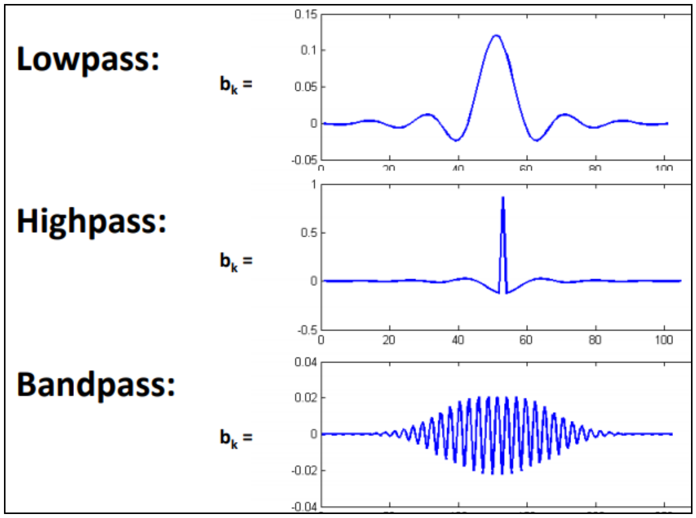

Figure 2-10 Spectral Filters [15]

The Butterworth Filter is a commonly used frequency domain filter used in signal processing. The equation below gives the relation of frequency response G(ω) with ω the angular frequency, n the number of poles, ε the maximum passband gain and ω_c the cutoff frequency [20].

$$G(\omega) = \frac{1}{\sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

2.2.8 Short-Time Fourier Transform

Time based data can also be visualized as a spectrogram, which displays the relative strength of all the frequencies present in a signal at any given time, for all time points. This can be calculated using the Short-Time Fourier Transform given by:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]w[m-n]e^{-j\omega m}$$

Where $w[n]$ is the analysis window, ω the angular frequency and n is the sample number.

Figure 2-11 shows the spectrogram obtained by plotting the time-based EEG data. The red artifacts on the spectrogram are due to the EEG waves generated when the eyes of the individual are closed for 10 seconds. It can be easily seen through the spectrogram that the red artifacts lie in the alpha region (8-12 Hz), which means that the activity produced in that frequency band is relatively stronger.

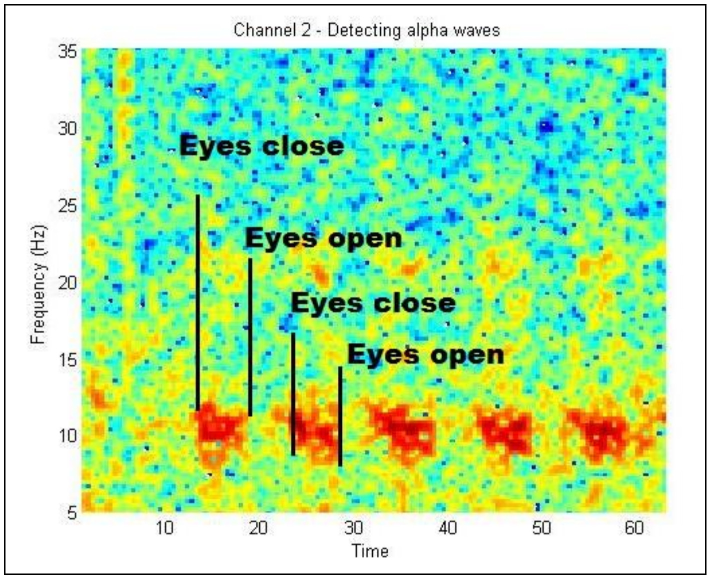

Figure 2-11 Spectrogram using STFT [21]

2.2.9 Power Spectral Density (PSD)

PSD of a signal is the distribution of its power along the spectrum of frequencies [22]. The average power P of a signal x:

$$P = \lim_{T \to \infty} \frac{1}{T} \int_0^T |x(t)|^2 dt$$

To estimate the power spectral density, Welch's Power Method is widely used. It windows overlapping segments to compute the FFT such that the noise is reduced [23]. The power axis is sometimes Log-Scaled for ease in visualization, since the power usually decreases at higher frequencies. In EEG signal processing, power spectral analysis is widely used for assessing which frequency bands are more predominant in the signal. As an example, PSD is plotted by researchers to quantify the effect of anesthetics on the brain [24].

2.3 Machine Learning

Machine Learning has been an upcoming topic in all fields of research related to computer science and bioinformatics. It has lately been incorporated by some researchers interested in BCI as well, who want to explore its usage in BCI based prosthetics and exoskeletons [25]. In this section, commonly used machine learning techniques in BCI research are elucidated.

2.3.1 Perceptron

The Rosenblatt's perceptron was one of the first machine learning techniques formulated, and forms a basis for more advanced techniques (such as Deep Learning) widely in use [26]. It is a kind of linear classifier used in supervised binary classification. The functional equation of the perceptron is:

$$Y = W^T \cdot X + b$$

Where X is the input, W is the learnt weight matrix, b the learnt bias and Y the output. The training dataset goes through a training procedure for training the weights and bias of the perceptron, which utilizes optimization techniques such as Gradient Descent. Various EEG research teams have utilized perceptron based machine learning architectures to classify signals. Multilayer perceptron classifier were utilized to learn and classify EEG signals of individuals while listening to emotional music [27]. The classification accuracies of experiments utilizing this machine learning technique has been relatively low.

2.3.2 Support Vector Machines (SVM)

Support Vectors refer to the data points which lie closest to the separating hyperplane of the classes. SVMs are called large margin classifiers since they classify the data by maximizing the margin around the separating hyperplane using the support vectors [28]. Various EEG research studies have incorporated this machine learning technique with relatively better accuracies. The same research group as in Section 2.3.1 utilized SVMs for classification of EEG signal while listening to emotional music, and reported a maximum classification accuracy of 92.7% [29]. SVMs deliver a greater classification accuracy mostly attributed to its non-linearity and kernel tricks.

2.3.3 Linear Discriminant Analysis (LDA)

Also called Fisher's LDA, LDA is a machine learning algorithm which aims to linearly combine features in a way which efficiently separates the classes. LDA computes a vector which distinguishes the classes in the best way [30]. LDA defines two feature vectors by linearly combining the given features for all the samples of all classes, the first one is Within Class Scatter Matrix:

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_j^i - \mu_j)\,(x_j^i - \mu_j)^T$$

The second is Between Class Scatter Matrix:

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)\,(\mu_j - \mu)^T$$

x_j^i is the ith sample of the jth class and μ_j is the jth class mean. c is the total number of classes, N_j the total sample number and μ is the total class mean. LDA aims to maximizes the ratio $\frac{det|S_b|}{det|S_w|}$. LDA has been used in many EEG related studies since it is simple and fast to apply and performs optimally under certain assumptions [15]. In a study comparing LDA to other machine learning methods for preictal stage detection using EEG, LDA outperformed SVMs by having a classification accuracy of 88.06% [31].

2.4 Summary

We conclude this section by reiterating the importance of the developments in BCI, Signal Processing and Machine Learning for this project. Understanding the functions of the different brain lobes and the types of EEG signals and adopting the 10-20 system for recording EEG has been a crucial step for the team. The team was also able to extract more meaning out of raw signals by appreciating signal processing theory, especially by converting the data to the frequency domain and by applying filters. Lastly, machine learning theory formed an important part of the project as we explored different techniques to classify the data. Section 3 will elaborate on the specific techniques used by the team and section 4 will describe the results we obtained.

3 Methodology

This section details the Hardware and Software apparatus required and then elaborates the specific details of the various experiments conducted by the team.

3.1 Hardware

For this project, the team selected hardware based on the number of channels offered and the cost effectiveness of the device. Thus, OpenBCI hardware was selected for this project since they are much cheaper than their functional alternatives and offer abundant support from the open source community with tutorials and bug fixes. OpenBCI is an open-source platform that provides biosensing hardware devices at relatively affordable prices [32]. They also provide a graphical user interface (GUI) based software that can be used to visualize the data being recorded by the hardware.

3.1.1 OpenBCI Ganglion

The OpenBCI Ganglion is a hardware device used to measure EEG/EMG/ECG signals. It consists of 4 input channels with high impedance differential inputs for measuring EEG and EMG signals. It can even connect to a computer wirelessly or via Bluetooth and comes pre-programmed with the firmware necessary for signal extraction. A radio module called Simblee is used as the on-board wireless connection and microcontroller [33]. The advantage of using Ganglion is that it comes at a lower cost than any other EEG hardware on the market. Figure 3-1 shows the OpenBCI Ganglion board.

Figure 3-1 OpenBCI Ganglion [33]

3.1.2 OpenBCI Cyton

OpenBCI Cyton is an 8-channel hardware device used to measure EEG signals, with a 32-bit processor [34]. The board communicates with the computer wirelessly, using the RFDuino radio module, and comes with a custom programmable dongle. The Cyton biosensing board has a tradeoff in the form of cost-channels. Although there are double the channels as compared to the Ganglion, it also comes at more than double its price. There are many well-documented experiments available on the internet which utilize the Cyton board because of its ease in setup and running. Figure 3-2 shows the 8-channel OpenBCI Cyton board.

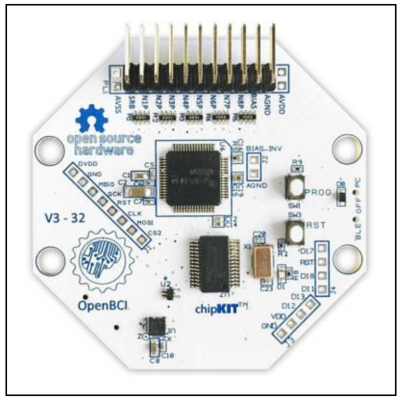

Figure 3-2 OpenBCI Cyton [34]

3.1.3 OpenBCI Daisy

The OpenBCI Daisy board is an 8-channel, Arduino-compatible biosensing module from OpenBCI, made to run in conjunction with the OpenBCI Cyton Board [35]. When using the Daisy module with Cyton, the boards together can capture 16 channels of EEG data from different places on the scalp. Both the boards come pre-loaded with OpenBCI software and can very easily be combined with the most commonly used BCI software on the internet. For this project, we decided to utilize the Cyton board with the Daisy module since there are numerous well-documented experiments and tutorial resources available on the internet, which gave us a baseline to build our work on. The Daisy module costs about the same as the Cyton Board, doubling the total cost of the hardware used. Figure 3-3 shows the Cyton Board and Daisy module.

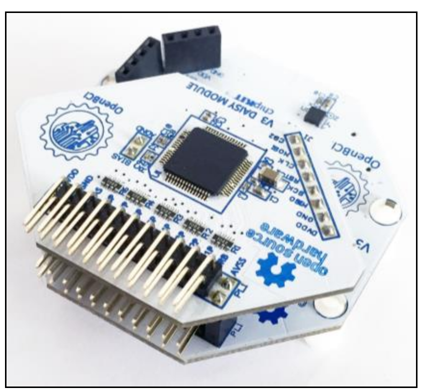

Figure 3-3 OpenBCI Daisy and Cyton [35]

3.2 Software

This section describes the softwares used for the different experiments conducted by the team. We enlist the advantages and disadvantages below, before choosing the main software on which we built our application.

3.2.1 OpenBCI GUI

The OpenBCI Graphical User Interface (GUI) is a software that enables visualization of the raw signal data being recorded by OpenBCI hardware devices such as the Ganglion [32]. The raw signals recorded by the hardware can be visualized in real time in the form of graphs and numerical data and simple operations to process the data can also be performed (Figure 3-4). It can be run on any operating system (Windows, macOS and Linux) and the data recorded by the device can also be converted to specialized file formats using this software and saved for further processing. Figure 3-4 depicts the time series visualization of the raw signal data. The top-right depicts the amplitude vs frequency representation of the data and the bottom-right displays the points on the scalp where the electrodes are connected [6].

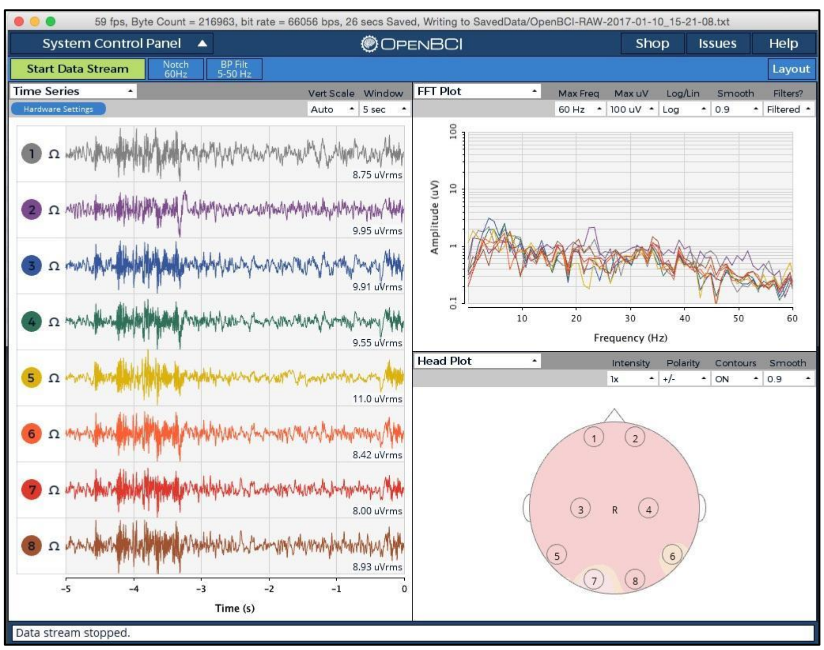

Figure 3-4 OpenBCI GUI

The team utilized OpenBCI GUI as a troubleshooting software to make sure that the OpenBCI hardware can effectively collect and stream the data to the computer.

3.2.2 OpenBCI Python

The OpenBCI Python repository contains a module to help receive and manipulate signals from the OpenBCI hardware. The module is extremely useful in creating a pipeline between the signals obtained from the board and any kind of analytical code written in python. There are several advantages with this approach. Firstly, the OpenBCI hardware utilizes its own protocol for data transfer, which is challenging to stream without using a library. By using OpenBCI's official Python library, we can make use of APIs to ensure reliable data transfer. We use OpenBCI Python in experiments to generate a dataset from the recorded signals. Figure shows the raw EEG data obtained by the OpenBCI Ganglion board through OpenBCI python and stored in a csv file.

```
%OpenBCI Raw EEG Data
%Number of channels = 4
%Sample Rate = 200.0 Hz
%First Column = SampleIndex
%Last Column = Timestamp
%Other Columns = EEG data in microvolts followed by Accel Data (in G) interleaved with Aux Data
0, -177.15, -176.09, -136.90, -231.26, 0.000, 0.000, 0.000, 23:48:10.771, 1547020090771
1, -167.03, -165.72, -124.81, -220.07, 0.000, 0.000, 0.000, 23:48:10.785, 1547020090785
2, -171.47, -172.58, -127.16, -221.88, 0.000, 0.000, 0.000, 23:48:10.785, 1547020090785
```

Figure 3-5 OpenBCI Python Data

3.2.3 Python Libraries

After collecting data through OpenBCI Python, Python libraries were used to perform preliminary analysis on the EEG data to extraction features. The data was prepared for analysis by loading it in a Pandas dataframe and NumPy was used to reshape and slice the dataframe. Matplotlib was used to plot the raw time-data to get an idea of the relative strengths of the EEG signal. Figure 3-6 shows 4 channels of EEG data recorded with the OpenBCI Ganglion board and plotted using Matplotlib. The yellow and blue streams are highly correlated whereas the streams in red and green differ from the first two.

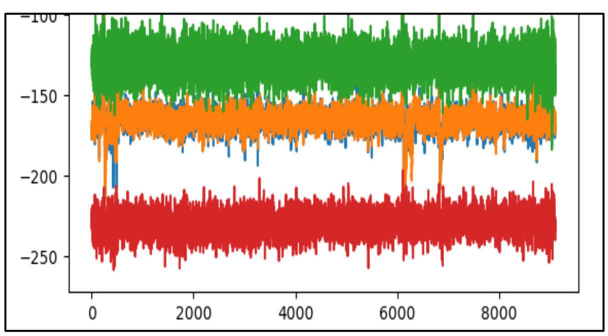

Figure 3-6 EEG with Matplotlib.

SciPy is an open-source python library for scientific computing [36]. In figure 3-7, we calculate the power spectral density using Welch's method (as discussed in section 2.2.9) by calling a function in SciPy on the signal displayed in Figure 3-6.

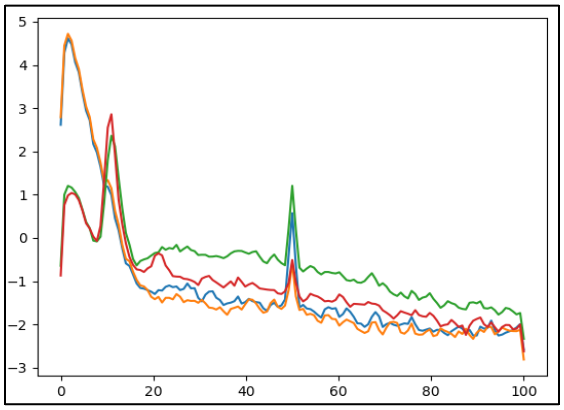

Figure 3-7 Power Spectral Density in SciPy

Plotting a spectrogram in python by computing the Short-Time Fourier transform (as discussed in section 2.2.8) is convenient as well, with inbuilt functions in Matplotlib and SciPy. Figure 3-8 shows the spectrogram obtained by using Matplotlib's spectrogram function on the data in Figure 3-6.

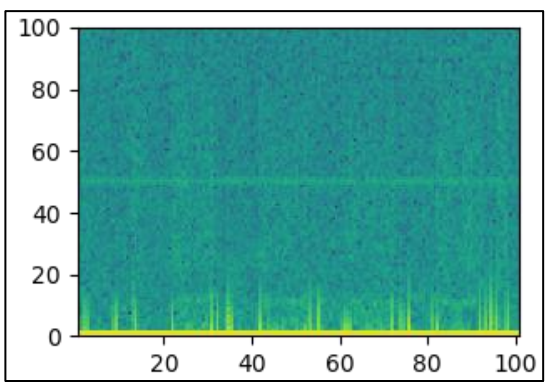

Figure 3-8 Matplotlib Spectrogram

Although Python and its libraries NumPy, Pandas, SciPy and Matplotlib proved to be resourceful for offline analysis, the team required software that could be immediately modified to a greater extent with minimal effort, so that maximum insight could be derived from each experiment at a faster pace. This led us towards real-time EEG signal processing softwares such as OpenViBe, discussed in the next section.

3.2.4 OpenViBe

OpenViBe is an open source software platform that can be used for development of real-time neurofeedback technology. It has tools for acquiring, processing and visualizing real-time EEG and EMG data. OpenViBe allows the user to create a module or "box" that takes an input, performs the set operation and returns an output. These boxes can be chained, and the data can be passed from one box to another. A collection of these boxes is called a scenario which can be saved as an xml or config file. Since these boxes can be created, destroyed and moved around very conveniently, the team was able to derive more meaning out of the EEG experiments at a faster pace.

The boxes allow the application of digital signal processing (DSP) techniques on real time signal data- Operations such as adding, subtracting or squaring signals can be performed by entering the equation into the Simple DSP box. OpenViBe provides tools for data visualization, which comes mainly in the form of simple time data visualization, streamed matrix and 3D plots. The time data visualization is useful in confirming if all the electrodes have been successfully attached to the scalp and getting a basic idea of whether the hardware is functioning as expected. The matrix visualization is particularly effective in judging data from the feature vectors or probability values returned on classification. Figure 3-9 shows 3D visualization in OpenViBe.

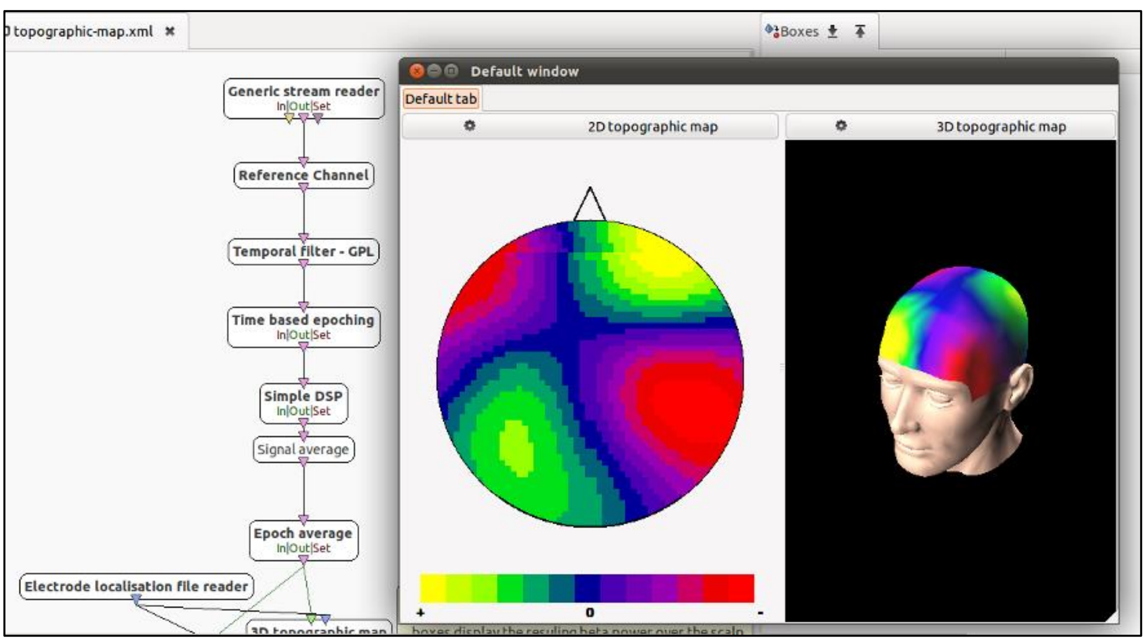

Figure 3-9 Visualization in OpenViBe

OpenVibE provide signal filtering modules with parameters that can be adjusted and configured. Among them, the Temporal filter and CSP Spatial filter modules proved to be particularly useful for manipulating EEG data, as explained in Section 2.2. OpenVibE also provides modules for classification and Machine Learning. These include algorithms such as Perceptron and SVMs and LDA, explained in Section 2.3. Moreover, not only do they have adjustable parameters, but also provide the accuracy values on the training data using cross validation.

Additionally, in order to set up experiments that require visualization to either induce a stimulus or facilitate training, OpenVibe comes with Lua programming language support. Lua scripts can simply be written for visualization and triggered when a scenario is played. Lastly, OpenViBe supports OSC, which is a communication protocol for computers, sound synthesizing and media processing hardware in real time [37]. The OSC controller in OpenViBe can be used to connect it to other 3D visualization softwares such as processing, which is explained in Section 3.2.5.

3.2.5 Processing

Processing is a framework for programming 3D and interactive visualization / games [38]. The code in a Processing 'sketch' is written in either Java or Python, and is compiled within the framework itself. Since Processing supports OSC as well, it was chosen by the team to build our end-user visualization after the signals are used as a control for the BCI app. Open-source support and a huge collection of libraries made Processing an ideal platform.

3.3 Experimental Setup

This section details the experiments conducted by the team in order to develop the BCI system. We experimented with the different hardware and software enlisted in sections 3.1 and 3.2 by building several architectures of the BCI application. Listed below is the setup of the important experiments conducted by the team.

3.3.1 Offline Preliminary Analysis

The aim of this experiment was to derive the most basic insight related to EEG data and gain confidence in the OpenBCI hardware and signal processing techniques. Another goal of this experiment was to empirically test out the theoretical research conducted by the team and identify the different types of EEG signals and scalp areas important for the BCI application.

The hardware used in this experiment is the OpenBCI Ganglion board, the details for which are in section 3.1.1. The electrodes are applied onto the scalp of the user according to Figure 3-10, given by the official setup guide for Ganglion [33]. Since the Ganglion board is the most hardware restricted (4 EEG channels), the electrodes are attached such that the activities from only the Frontal and Occipital lobes are captured.

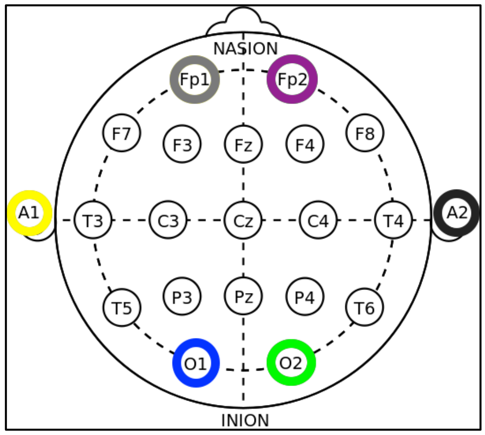

Figure 3-10 10-20 system for Preliminary Analysis [33]

For the software, OpenBCI Python is used to stream the data from the board onto the computer via Bluetooth. The user was given two sets of tasks - in the first task, the user opened their eyes for 10 seconds and then closed them for 10 seconds repeatedly for about 2 minutes. In the second task, the users had to move their right leg for a few seconds, 30 seconds after the start of the experiment.

Once the experiment was performed, the data was saved into a file for offline analysis. Python libraries like NumPy, Pandas, SciPy and Matplotlib as discussed in section 3.2.3. Simple plots in the time and frequency domain were plotted and signal processing algorithms such as PSD using Welch's method were applied in Python to the EEG data.

3.3.2 Online BCI system

After performing offline preliminary analysis on EEG signals in experiment 3.3.1 and 3.3.2, the team proceeded to explore signal processing in real time (online). On successfully processing and classifying the signal, the team aimed to use these signals as a control for the BCI system.

In a related experiment which explores Motor Imagery for BCI, the author developed software to distinguish the EEG activity obtained when visualizing the movement of the right vs the left arm [39]. Our team utilized the methodology given by the author as the basis of our work, and aimed at increasing the accuracy of the system by utilizing various hand movements / motor imagery, signal processing and machine learning techniques. In the end, we connect our signal processing and classification pipeline to a 3D application, so that the signals can act as the controls.

In this experiment, the user was shown a set of left and right arrows repeatedly with a gap of a few seconds, in a random order. Each experiment comprised of 20 left and 20 right trials where the user had to imagine the full range of motion of their left or right arm, based on the direction of the arrow. This formed the collection stage of the EEG test data. The EEG data was then separated for the left and right arm trials and filtered in the temporal and spatial domain for feature extraction. A machine learning classifier was then trained on the features to separate the two classes of data. The classifier was then tested in real time by showing the user another set of left and right arrows randomly and calculating the test data accuracy and the confusion matrix. After obtaining a satisfactory accuracy by testing and training, the system was connected to a 3D visualization / game to test out the control of the signals.

The hardware used in this experiment is the OpenBCI Cyton board with Daisy module, the details of which are given in section 3.1.2 and 3.1.3. Figure 3-11 (left) shows the placement of the electrodes in line with the 10-20 standardization. 16 electrodes from Cyton and Daisy are attached to the scalp of the user by meticulously taking the measurements of the user's forehead. The regions on which the electrodes are attached are selected by the author such that maximum EEG signals associated with motor activity are captured [39].

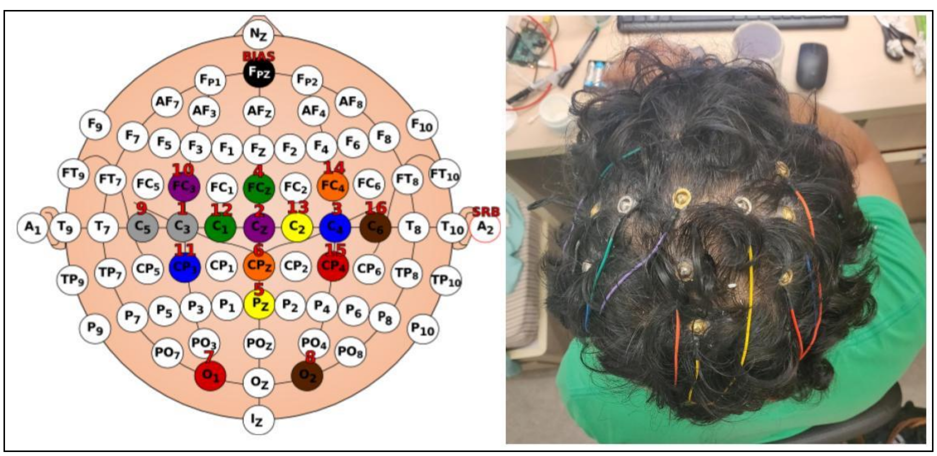

Figure 3-11 10-20 placement for Motor Imagery [39]

This experiment used OpenViBE software, as described in section 3.2.4. Figure 3-12 shows the software architecture of acquiring the EEG test data. OpenBCI Cyton board connects to OpenViBe Acquisiton server wirelessly and start streaming the data of the user. This data is further passed on to OpenVibe for writing the data onto a file (Stream Writer) and for the control and visualization of the experiment.

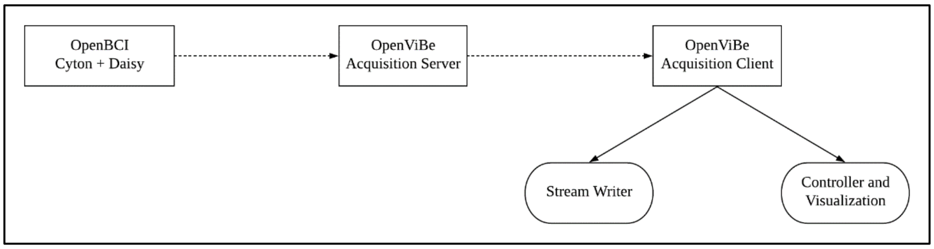

Figure 3-12 1-OpenViBe Acquisition

The controller and visualization unit is built in Lua and its purpose is to mark the special events during the experiment, for example 'right arrow on' event, 'right arrow off' event, start and end of experiment etc. The controller marks these events on the data stream for the stream writer and the visualization displays these events on the screen. Figure 3-13 shows the Visualization which takes place at a 'left arrow on' event. The user performs an action according to the direction of the arrow present on the screen. In our experiment, as further explained in section 3.3.2.2, we collect the activity from 3 different types of user actions in different experiments. In Figure 3-13, the user is moving their entire arm towards their body when an arrow comes on screen.

Figure 3-13 Creating test data

The next step was to process the EEG data collected from the users to extract meaningful information which can act as features. In the most basic architecture, as shown in Figure 3-14, a single temporal filter is applied to select the frequencies lying within 8-30Hz, which comprises of the alpha and beta region. The signals collected from the right trials vs the left trials are then separated and given to a CSP spatial filter trainer. As discussed in section 2.2.5, the goal of the CSP trainer is to linearly combine the data to maximize the class variance of one and minimize it for the other class. The output dimension of the CSP spatial filter was chosen to be 6, by empirical analysis. This means that the 16 channel data from the Cyton and Daisy boards gets combined to form 6 features. Multiple temporal filters are also used for comparison, as explained in section 3.3.2.1.

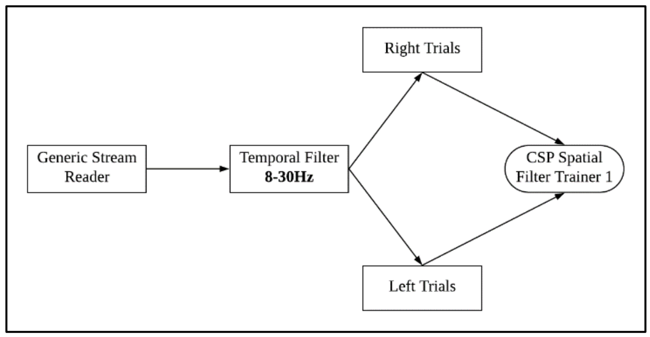

Figure 3-14 CSP Training

After the CSP filter weights are trained, the classifier was trained next. Figure 3-15 shows the pipeline for the training of the classifier. Similar to Figure 3-14, the right and left trials are separated and a classifier is trained on the PSD of the signals. We use the LDA algorithm (as explained in section 2.3.3) as our classifier in this experiment, but other algorithms are used too for a comparison, the details of which can be found in section 3.3.2.3. The classifier trainer also computes the training and cross validation accuracy, which we use to compare the different algorithms.

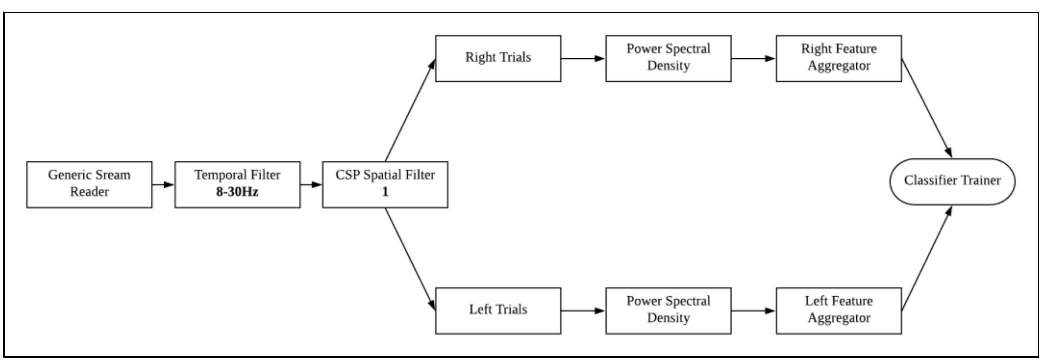

Figure 3-15 Classifier Training

Lastly, we test the training accuracy of our system by using the pipeline shown in Figure 3-16. The EEG signal acquired from the user goes through a temporal filter and CSP spatial filter before its PSD is computed. The classifier then proceeds to classify if the signal is either of a right or left command. The accuracy of the classification is calculated as well, with the help of a scatter matrix. For this testing, a similar visualization is incorporated as in shown in Figure 3-13, for the creation of the test dataset.

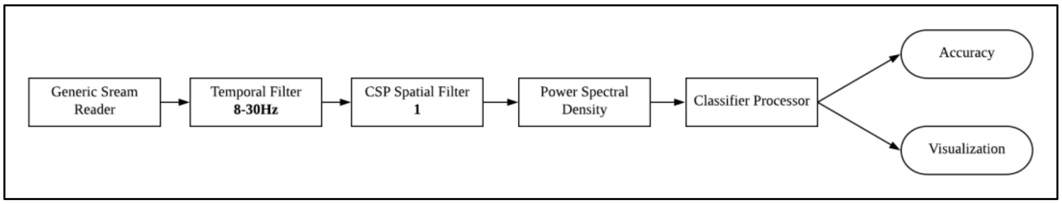

Figure 3-16 Classifier Testing

3.3.2.1 Varied Signal Processing Architectures

In an aim to increase the training and testing accuracies of the system, the team developed 2 more Signal Processing Architectures / Pipelines, in addition to the one displayed in Figures 3-14 to 3-16. For the training of the CSP filter and classifier weights as shown in Figure 3-14 and Figure 3-15, we utilized more than a single temporal filter with an aim to increase the classification accuracy of the signal. This technique called FBCSP aims to extract CSP features which are specific to a given frequency band [40].

Figure 3-17 shows the Alpha and Beta band separated version of the architecture shown in Figure 3-14. Two temporal filters are used, one for choosing the EEG activities in the Alpha range (8-12Hz) and the other in the Beta range (12-38Hz). Since a single CSP filter is trained for each Temporal Filter, two sets of CSP filters weights are trained, with an output of 6 features each.

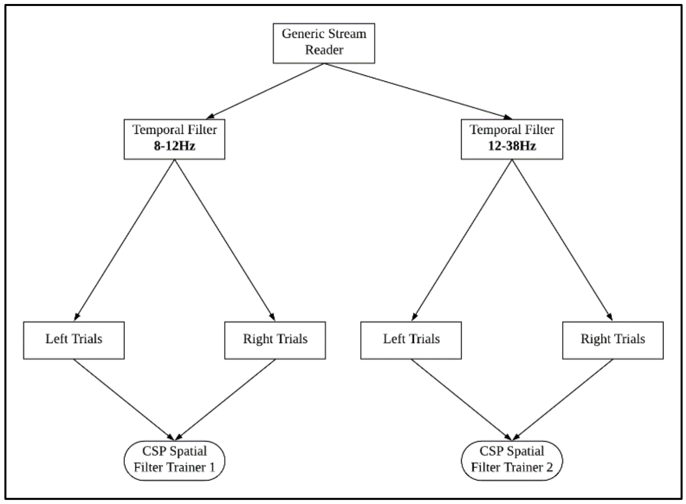

Figure 3-17 CSP Training Alpha Beta

Figure 3-18 shows the the Alpha and Beta band separated version of the architecture shown in Figure 3-15. The two CSP filters are used to obtain features which are then separated on the basis of the left and right trails. A single classifier is then trained after the left and the right features are aggregated from both the CSP spatial filters.

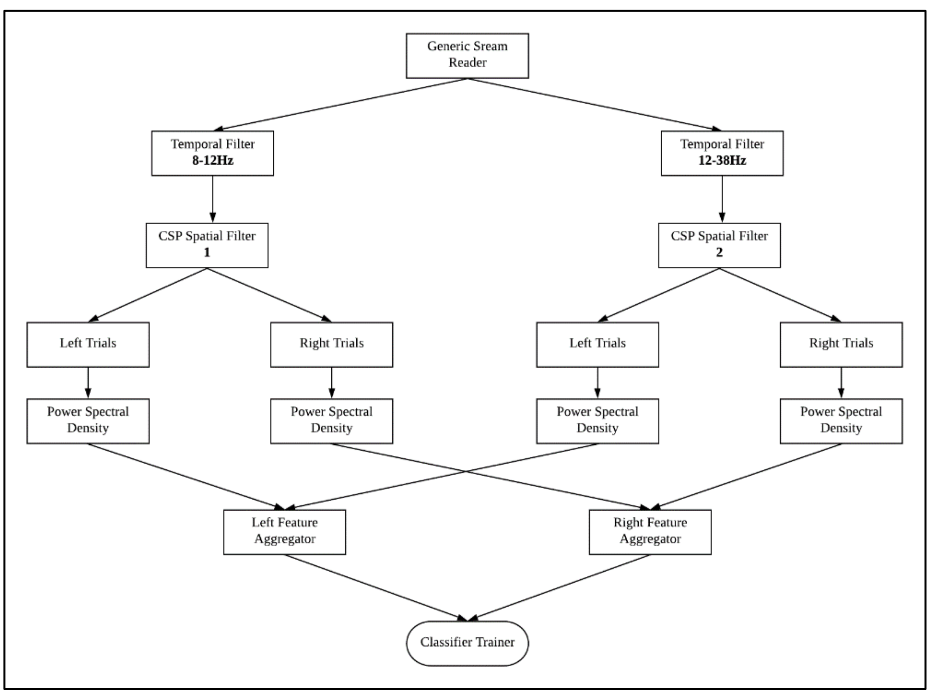

Figure 3-18 Classifier Training Alpha Beta

Similarly, the temporal filters are applied while testing and computing the accuracy of the system. The pipeline shown in Figure 3-16 gets extended to incorporate 2 temporal and spatial filters.

In addition to this training and testing architecture with 2 temporal filters, the team developed another architecture for further increasing the accuracy of the system with 6 temporal filters in the following frequency bands: Theta (4-8Hz), Alpha (8-12Hz), Beta1 (12-16Hz), Beta2 (16-20Hz), Beta3 (20-25Hz), Beta4 (25-38Hz). The results from these 3 different architectures (Single band, Alpha-Beta and 6 Band temporal filters) are discussed in Section 4.2.

3.3.2.2 Varied User Actions

The EEG signal processing and classification architectures were trained on experiments which utilized 3 different types of movements / motor imagery. In the most basic experiment, the user was asked to practice the full range of motion of their arm, as shown in Figure 3-13. Because of the explicit nature of this action, it served as a good comparison for other more subtle actions. The second action, as shown in Figure 3-19, was one where the user slightly moves the fingers of their left or right hand, based on the direction of the arrow displayed.

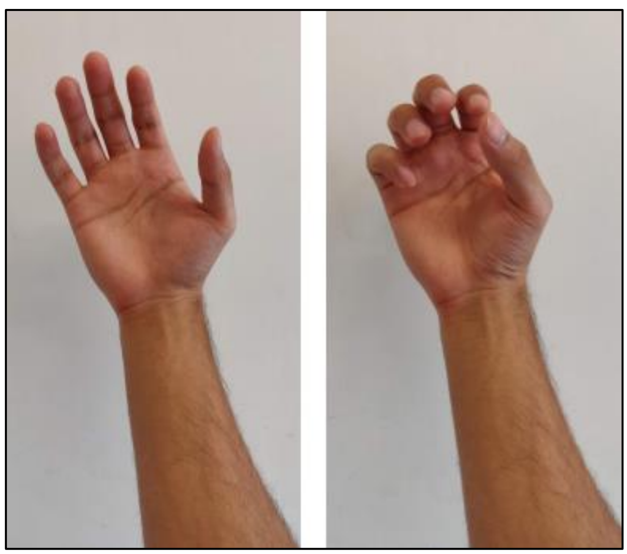

Figure 3-19 Moving Fingers

The third action, which utilized Motor Imagery, was to just imagine the full range of movement of the user's arm, without actually performing the movement. In a single experiment, a single action type was chosen and was utilized for both the training and testing procedure of the system. The results of the different actions over the different signal processing architectures are discussed in Section 4.2.

3.3.2.3 Varied Machine Learning Models

In addition to the varied signal processing architectures and action, the team compared various machine learning models on the EEG data collected during the experiments. We chose the Perceptron, SVM and LDA algorithms as classifiers which are available as modules in OpenViBe as the classification techniques. The results obtained from these algorithms are discussed in Section 4.2.

3.3.2.4 Processing 3D Game

The final aspect of the BCI system is the visualizartion software, which the team chose to build in the Processing framework, as explained in Section 3.2.5. We chose a 3d version of the popular arcade game Pong. The game is an abstract version of table tennis, where the user controls a 'paddle' and hits the ball until the opponent's paddle misses it [41]. An open source version of the processing code was used as a basis of the development of our visualization software [42]. Since our signal processing and classification system can classify two commands, we restrict the movement of the paddle in only the X direction instead of both X and Y. Figure 3-20 shows a screenshot of the game built in Processing.

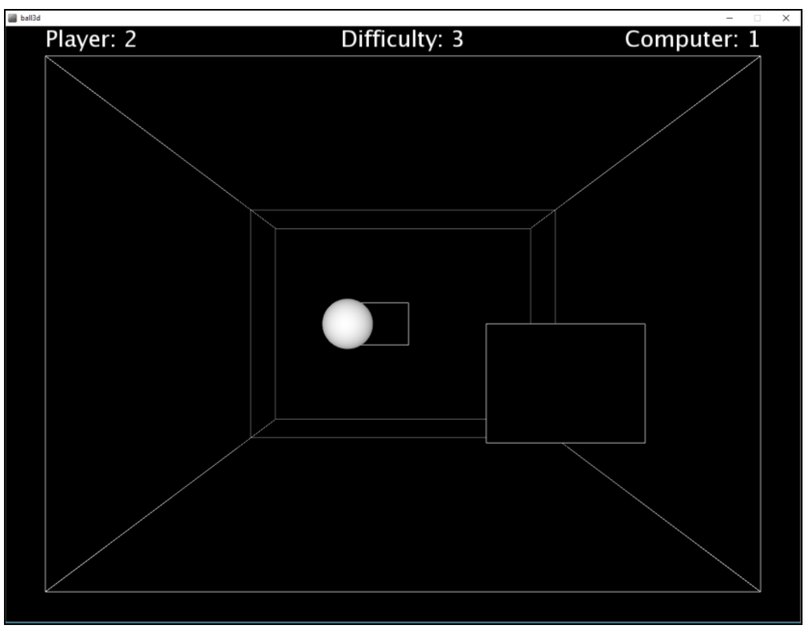

Figure 3-20 Processing 3D Pong

Since processing supports OSC, the classified signals from OpenViBe go through its OSC controller box for being sent to Processing as a control.

Each OSC message contains an OpenViBe stimulus which corresponds to the classification by the Classifier Processor box. ‘OVTK_GDF_LEFT’ stimulus received via OSC from OpenViBe to processing is mapped such that the paddle travels left in the X direction and ‘OVTK_GDF_Right’ is mapped to the right direction. The users used various actions, as detailed in 3.3.2.2 to try out the game by using the EEG signals produced by either Motor movement or Motor imagery. The results are explained in Section 4.2.

3.4 Summary

In section 3, we went over the hardware and software used in the various experiments and the justification of using the appropriate technology for each type of experiment. The team performed a preliminary offline analysis using OpenBCI Ganglion, OpenBCI Python and other Python libraries which was crucial in understanding the EEG signals. We then connected the Cyton board and Daisy Module to OpenViBe and attempted to capture, process and classify the EEG signals. We also computed the accuracies of the classification to compare the various actions, Signal Processing techniques and Machine Learning algorithms used during the experiment. Finally, the signal was used to control a 3D game built in Processing, with the help of OSC.

4 Results

This section analyses and compares the results obtained from the experiments performed in section 3.3 to quantify the performance of the BCI system built by the team.

4.1 Offline Preliminary Analysis

This experiment was done by the team with an aim to empirically test out the theoretical studies on EEG signals using the 4-channel OpenBCI Ganglion Board, OpenBCI Python and Python libraries. The results from the experiments performed with the user are given below.

Figure 4-1 shows the spectrogram plotted with the data collected from the user who had to repeatedly open and close his eyes for 10 seconds.

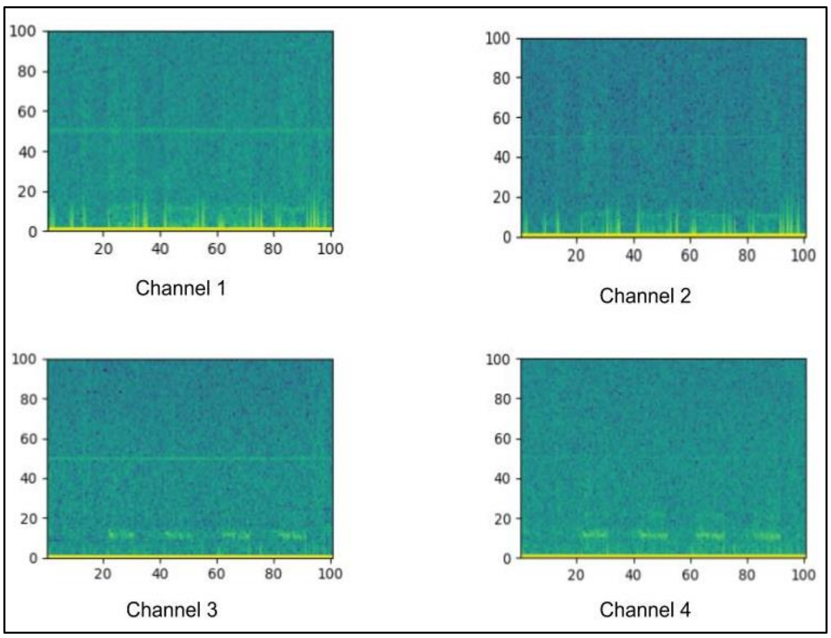

Figure 4-1 Eyes Closed Results

It can be clearly seen from the spectrogram that there are alternating 10 seconds of artifacts in the spectrograms of the data obtained from all 4 channels. These artifacts correspond to alpha waves (7.5-12 Hz) which are generated when the user closes their eyes.

Figure 4-2 shows the spectrogram plotted with the data collected from the user who had to move their right leg, 30 seconds into the experiment.

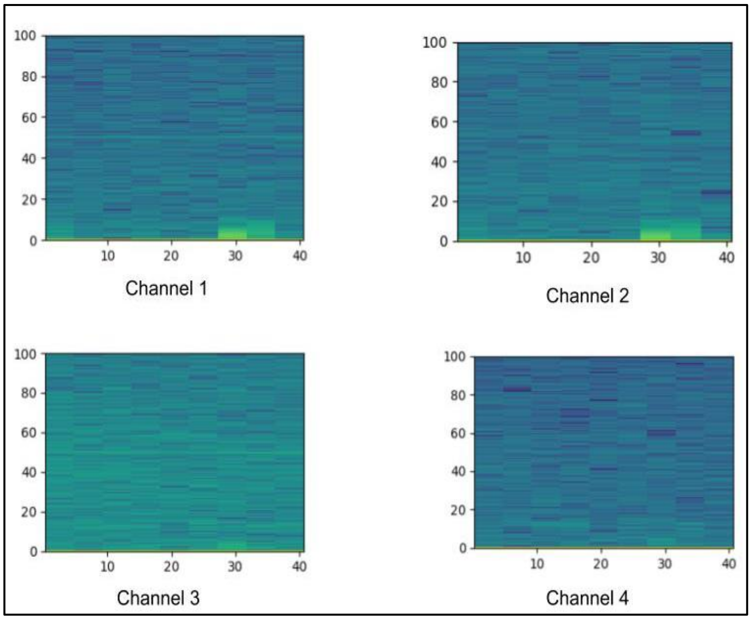

Figure 4-2 Move Leg Results

The experimental results showed artifacts in channel 1 and 2 around 30 seconds, while the spectrograms obtained from channel 3 and 4 did not contain any interesting information. These results can be explained by the fact that activity in the primary motor cortex is associated with the user moving their leg [13], as discussed in Section 2.1.1 as well. The primary motor cortex is located in the frontal lobe, which is at the front-end of the brain and since the electrodes for channel 1 and 2 are placed at the front of the head, they are able to pick up the EEG signals generated in the primary motor cortex.

These experiments provide a valuable insight into the power of the 4-channel Ganglion board. The results obtained demonstrate that we can differentiate between thoughts like closing eyes and moving legs purely based on the channels and frequency of the waves generated, as theoretically stated.

4.2 Online BCI system

This section details the results from the experiments performed using the real-time BCI application developed using the OpenBCI Cyton board and Daisy Module as Hardware to extract EEG data from the user and OpenViBe and Processing software for processing and classifying the signals.

The team conducted experiments, as enlisted in Section 3.3.2, by collecting data from users who incorporated different actions / motor imagery, by utilizing different signal processing architectures and by classifying the signals using different machine learning methods.

The figures in this section discuss the accuracy measured during Training, Cross-Validation (CV) and Testing phases. We define our accuracy by the Mis-Classification rate and convert it into a percentage. The mis-classification rate is given by:

$$L\,(p,t) = \frac{1}{N}\sum_{k}\begin{cases}1, p_k = t_k \\ 0, p_k \neq t_k\end{cases}$$

Where p is the prediction of the classifier and t is the target.

Figure 4-3 shows the Training Accuracy calculated for the different types of actions / motor imagery (MI) incorporated by the user over different signal processing architectures. The 'Move Arm' curve refers to the user moving their arm as the action, 'Move Fingers' refers to subtle movement of the fingers and 'Arm MI' refers to using only Motor Imagery of the movement of the arm for training. '1 Filter', '2 Filter' and '6 Filter' are the three different signal processing pipelines used for the experiment, as described in Section 3.3.2.2.1.It is evident that the 6-Filter signal processing architecture designed by our team gives a superior training accuracy as compared to the 1-Filter and 2-Filter pipelines. Intuitively, the actual movement of the left or right arm is more easily distinguishable as compared to more subtle actions like the movement of fingers or motor imagery. For MI, our model substantially improves the accuracy from 67% to 89% training accuracy.

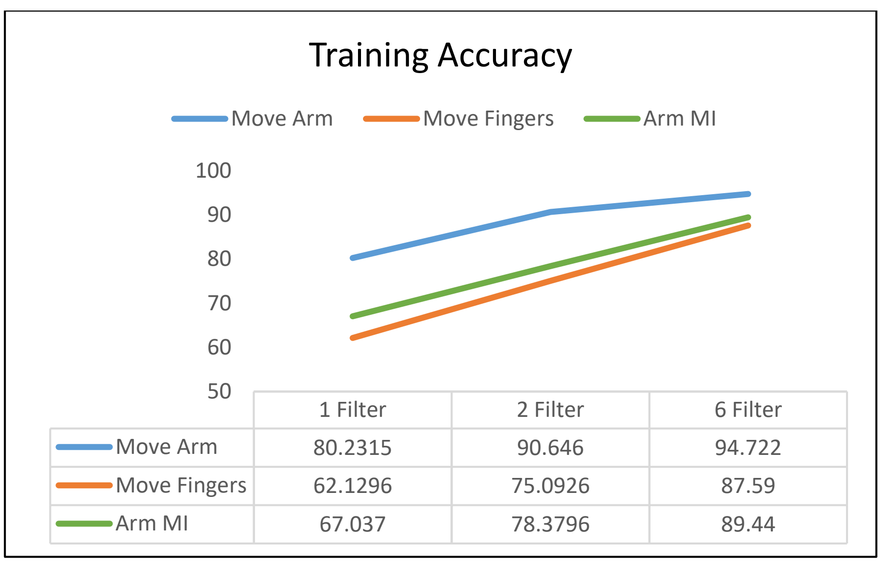

Figure 4-3 Training Accuracy

Similar to Figure 4-3. Figure 4-4 shows the Cross-Validation Accuracy for the different actions and different architectures. Motor Imagery gives a better CV accuracy as compared to Moving Fingers, but a worse CV accuracy as compared to actual movement of the arm. 6-Filter architectures outperforms 1 and 2 Filters in CV accuracy as well.

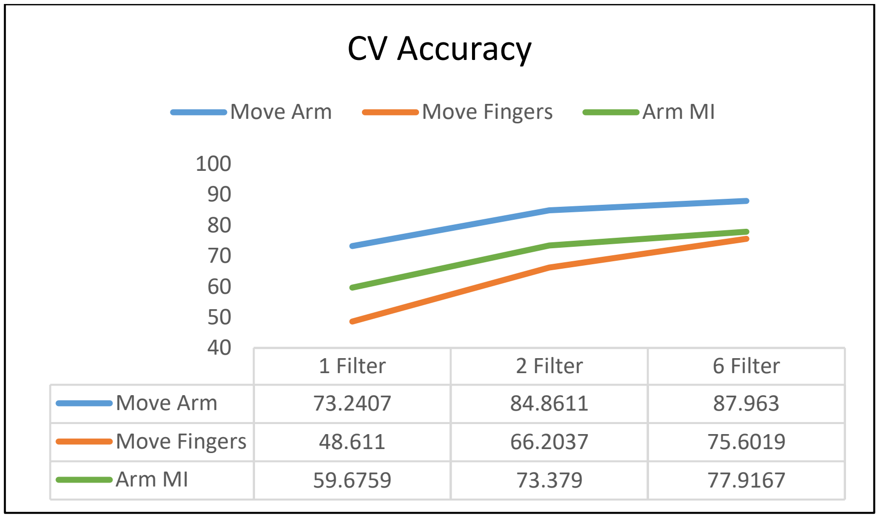

Figure 4-4 Cross-Validation Accuracy

Figure 4-5, similar to Figure 4-3, shows the Testing accuracy of the BCI system. The 6-Filter pipeline delivers a higher testing accuracy as compared to the 1 and 2 Filter architecture. The actual movement of the arm is more easily classified by our system as compared to Moving fingers and Motor Imagery. For Motor Imagery, the 6-Filter pipeline significantly improves the accuracy which has the potential to give a greater degree of control in the BCI application.

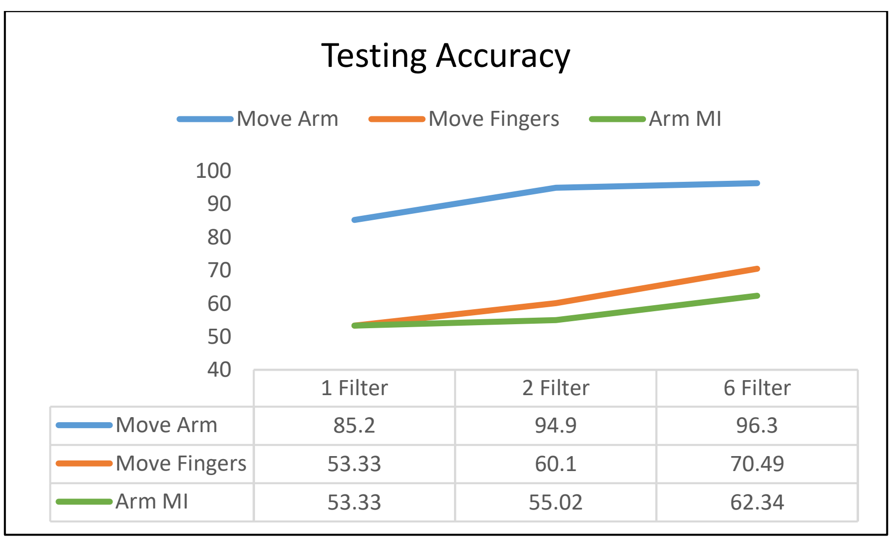

Figure 4-5 Testing Accuracy

The team also performed an analysis of different ML techniques which can be used to classify the signal. Figure 4-6 shows the CV accuracy computed on the dataset with Motor Imagery for Perceptron, SVM and LDA classification algorithms. The signal processing architecture used is the 2-Filter one. LDA and SVM give a very similar performance on most experiments and this accuracy is substantially higher to the Perceptron. This may be attributed to the linear nature of perceptron, which has difficulty in separating non-linear data.

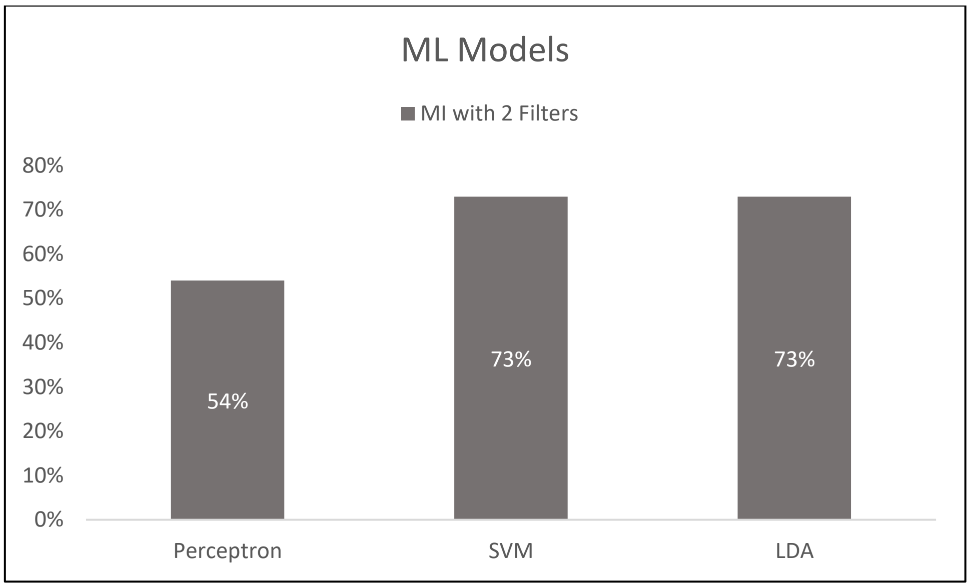

Figure 4-6 ML Models Results

In addition to the experiments with the models given above, the team also tested out the quality of control of the BCI system by connecting it to the Processing 3D Pong game. As explained through Figures 4-3 to 4-5, the user experienced greater control when utilizing the 6-Filter architecture with LDA as the classification algorithm. Using merely motor imagery gives an unpredictable control, whereas the slight movement in fingers acts as better control with lag of about half a second. The system developed by the team works better as a controller when the user incorporates the full movement of their arm rather than just motor imagery, as shown in Figure 4-5, but these are limitations due to the noisy nature of EEG signals and restrictions in the number of channels and amount of data generated.

5 Conclusion

This project was initiated to explore the potential of Brain Computer Interface systems for creating a device that could provide a convenient method for users to interact with their devices. The main motivation behind this project was to not only create a low cost and low resource based BCI system, but also to promote interest and research into the field of brain-computer interface technology. The most salient feature of this project so far is that it can recognize mental simulations of a given action / motor imagery which is used to link thoughts to specific virtual actions. The team has classified complex motor imagery like imagining the movement of the left vs right hand to a great extent by training a machine learning model on the acquired data to map thoughts onto commands.

5.1 Project Limitations

There are a few constraints which lead to certain limitations in functionality of the BCI system. Monetary cost has been a major factor in several decisions made during the progress of the project. Since the funds allocated for the Senior Design Project are limited, development was carried out with devices with at most 16 EEG channels. This may lead to limitations in accuracy of the trained models since more having data from more channels would result in a more reliable command classification.

5.2 Future Research

One potential direction could be to develop multiplayer or competitive video games using the BCI system created. Such an application would only require multiple hardware devices which can all run the same software. Research can also be carried out to create BCI applications in domains other than entertainment such as business or finance. Moreover, although this project is restricted to using 16 EEG channels at maximum, experiments can be carried out with hardware devices which support more channels using the software developed by the team in order to provide quantitative evidence on the number of channels versus the reliability of the command classification. There is immense potential for research and development in the field of brain computer interfacing, and all that is required is the initiative to tap into it.

6 References

[1] J. J. Shih, D. J. Krusienski and J. R. Wolpaw, "Brain-Computer Interfaces in Medicine," *Mayo Foundation,* March 2012.

[2] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and T. M. Vaughan, "“Brain–computer interfaces for communication and control,”," [Online]. Available: http://www.cs.cmu.edu/~tanja/BCI/BCIreview.pdf. .

[3] K. Sukel, "Neuroanatomy-A Primer(1)," The Dana Foundation, [Online]. Available: http://www.dana.org/News/Details.aspx?id=43515.

[4] "Lobes of the Brain," SpinalCord.com, [Online]. Available: https://www.spinalcord.com/lobes-of-the-brain.

[5] N. E. and d. S. F.L., "Electroencephalography: Basic Principles, Clinical Applications, and Related Fields," *Lippincott Williams & Wilkins,* 2004.

[6] Egyptian Informatics Journal, "Brain computer interfacing: Applications and challenges," 06-Jul-2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866515000237.

[7] " “EEG,”," Kinesia, [Online]. Available: https://glneurotech.com/FAQ/eeg.html.

[8] A. Morley, L. Hill and A. G. Kaditis, "10-20 system EEG Placement," [Online]. Available: https://www.ers-education.org/lrmedia/2016/pdf/298830.pdf.

[9] D. R and D. JE, "Motor imagery in physical therapist practice," *PubMed,* 2007.

[10] J. F. D. S. [jfdelgad], "Brain Computer Interface Basics, Topographic distribution of scalp potentials during motor imagery," [Online]. Available: https://www.youtube.com/watch?v=QUa-7ipeLWM.

[11] E. A. Larsen, ""Classification of EEG Signals in a BrainComputer"," *Norwegian University of Science and Technology,* 2011.

[12] P. Tagare, "Signal Averaging," [Online]. Available: https://pdfs.semanticscholar.org/5941/95a91dea1a76a6865df707d4e5688f3ed9bc.pdf.

[13] B. D. H. R and M. W. J., "Cognitive brain potentials and their application," *The British Journal of Psychiatry,* 1990.

[14] A. Materka and P. A. Poryzala, "A Robust Asynchronous SSVEP Brain-Computer Interface Based On Cluster Analysis of Canonical Correlation Coefficients," *Springer International Publishing,* vol. 2014.

[15] C. A. Kothe, "Lecture 3: Signal Processing In BCIs," [Online]. Available: ftp://sccn.ucsd.edu/pub/bcilab/lectures/03_Signal_Processing_in_BCIs.pdf.

[16] "Documentation," OpenViBe, [Online]. Available: http://openvibe.inria.fr/documentation/1.2.0/Doc_BoxAlgorithm_SpatialFilter.html.

[17] C. Furse and O. Gandhi, "Why the DFT is faster than the FFT for FDTD time-to-frequency domain conversions," *IEEE.*

[18] I. J. Rampil, "Fast Fourier Transformation of EEG Data," *JAMA Network,* 1984.

[19] WaveMetrics, "signalprocessing," WaveMetrics, [Online]. Available: https://www.wavemetrics.com/products/igorpro/dataanalysis/signalprocessing.

[20] S. Butterworth, "On the Theory of Filter Amplifiers," *Wireless Engineer,* vol. 1930.

[21] T. Vo, "Exploring EEG with STM32F4 and ADS1299," [Online]. Available: http://eegexplore.blogspot.com/2016/05/capturing-alpha-waves-with-weegee.html.

[22] P. Stoica and R. Moses, "Spectral Analysis of Signals," in *Spectral Analysis of Signals*, Prentice Hall, 2005.

[23] P. Donhauser, "Spectral Analysis: How to process neural oscillatory signals," [Online]. Available: https://www.mcgill.ca/bic/files/bic/spec_decomp_peterdonhauser-expanded.pdf.

[24] D. O, S. G, S. G and K. EF, "Awareness and the EEG power spectrum: analysis of frequencies.," *PubMed,* 2004.

[25] A. Chan, C. E. Early, S. Subedi, Y. Li and H. Lin, "Systematic analysis of machine learning algorithms on EEG data for brain state intelligence," *IEEE Explore.*

[26] S. Haykin, Neural Networks and Learning Machines, Pearson Education, Inc.

[27] Y.-P. Lin, C.-H. Wang, T.-L. Wu and S.-K. Jeng, "Multilayer perceptron for EEG signal classification during listening to emotional music," *IEEE Explore.*

[28] R. Berwick, "An Idiot's guide to Support vector machines," [Online]. Available: http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf.

[29] Y.-P. Lin, C.-H. Wang, T.-L. Wu, S.-K. Jeng and J.-H. Chen, "Support vector machine for EEG signal classification during listening to emotional music," *IEEE Explore.*

[30] A. M. MartõÂnez and A. C. Kak, "PCA versus LDA," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 23.

[31] M. Yildiz, E. Bergil and C. Oral, "Comparison of different classification methods for the preictal stage detection," *Biomedical Research,* vol. 28, pp. 858-865, 2017.

[32] OpenBCI, "OpenBCI/OpenBCI_GUI," OpenBCI, 19-Aug-2018. [Online]. Available: https://github.com/OpenBCI/OpenBCI_GUI.

[33] OpenBCI, ""Ganglion Board (4-channels)"," OpenBCI, [Online]. Available: https://shop.openbci.com/products/pre-order-ganglion-board?variant=13461804483.

[34] OpenBCI, "Cyton Biosensing Board (8-channels)," OpenBCI, [Online]. Available: https://shop.openbci.com/collections/openbci-products/products/cyton-biosensing-board-8-channel?variant=38958638542.

[35] OpenBCI, "Cyton + Daisy Biosensing Boards (16-Channels)," OpenBCI, [Online]. Available: https://shop.openbci.com/products/cyton-daisy-biosensing-boards-16-channel?variant=38959256526.

[36] SciPy developers, "Scientific Computing Tools for Python," SciPy developers, [Online]. Available: https://www.scipy.org/about.html.

[37] opensoundcontrol.org, "Introduction to OSC," opensoundcontrol.org, [Online]. Available: http://opensoundcontrol.org/introduction-osc.

[38] Processing Foundation, "Processing.org," Processing Foundation, [Online]. Available: https://processing.org/.

[39] J. Frey, "OpenBCI crossing swords with motor imagery," 3 March 2015. [Online]. Available: http://blog.jfrey.info/2015/03/03/openbci-motor-imagery/.

[40] K. K. Ang and C. Guan, "EEG-Based Strategies to Detect Motor Imagery for Control and Rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* 2017.

[41] The International Arcade Museum, "Pong," [Online]. Available: https://www.arcade-museum.com/game_detail.php?game_id=9074.

[42] J. Counts, "Jared Counts," [Online]. Available: http://www.jaredcounts.com/.

[43] V. Morash, O. Bai, S. Furlani, P. Lin and a. M. Hallett, "Classifying EEG Signals Preceding Right Hand, Left Hand, Tongue, and Right Foot Movements and Motor Imageries," *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology,* vol. 2570–2578.

[44] "What Is FMRI? - Center for Functional MRI - UC San Diego," [Online]. Available: http://fmri.ucsd.edu/Research/whatisfmri.html.

[45] "“EMOTIV EPOC 14 Channel Mobile EEG,”," Emotiv, [Online]. Available: https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/.

[46] "“intendiX: Specs & Features,”," g.tec medical engineering. , [Online]. Available: http://www.gtec.at/Products/Complete-Solutions/intendiX-Specs-Features.

[47] "“The brain speaks: Scientists decode words from brain signals,”," ScienceDaily, [Online]. Available: https://www.sciencedaily.com/releases/2010/09/100907071249.htm.

[48] P. Welch, "“The use of the fast Fourier transform for the estimation of power spectra:," *IEEE Trans. Audio Electroacoust,* vol. vol. 15, no. 1967, pp. pp. 70-73.

[49] "“The Anatomy of Movement”," Brain Connection, [Online]. Available: https://brainconnection.brainhq.com/2013/03/05/the-anatomy-of-movement.

[50] P. Tomalski and K. C. Kadosh, "Pre-processing for MEG and EEG," [Online]. Available: https://www.fil.ion.ucl.ac.uk/mfd_archive/2007/powerpoint%20presentations/Preprocessing%20EEG%20&%20MEG_final.ppt.