# Self Paced Brain Computer Interface On Sensorimotor Rhythms For Virtual Objects Controlling

**- Semester 8 Mid Report -**

**Avishka Athapaththu**
**Prageeth Dassanayake**
**Sewwandie Nanayakkara**

Department of Computer Engineering
University of Peradeniya

Final Year Project (courses CO421 & CO425) report submitted as a requirement of the degree of
*B.Sc.Eng. in Computer Engineering*

January 2021

Supervisors: Dr. Isuru Nawinne (University of Peradeniya), Prof. Roshan Ragel (University of Peradeniya), and Mr. Theekshana Dissanayake (Queensland University of Technology )

We would like to dedicate this report to our loving parents and teachers.

# Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is our own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgments.

Avishka Athapaththu
Prageeth Dassanayake
Sewwandie Nanayakkara
January 2021

# Acknowledgements

We would like to acknowledge the guidance given by our supervisors Dr. Isuru Nawinne, Prof. Roshan Ragel, Mr. Theekshana Dissanayake to achieve the milestones of our research project. We would also like extend our gratitude for everyone who supported us.

# Abstract

Non-invasive EEG based BCI systems have been an interesting research area for many fields. However most of the research done on this subject is synchronous, therefore the state of mind of the user is not similar to its natural behaviour. Considering to provide possible experience in practical applications, self-paced BCI systems started gaining popularity in recent years. However, there are certain challenges yet to be addressed when following this method. Out of the research done on self-paced BCI systems, most of them are focused on motor-imagery control whereas research on non-motor imagery mental tasks is limited. In this research, we analyse the possibility of using the techniques used in the motor-imagery method for non-motor imagery mental tasks to be fed into virtual object controlling applications. In this research, different artifact removal methods such as Independent Component Analysis (ICA), filtration methods, and wavelet transformation with threshold have been tested out. Feature extraction was done using signals representations such as fast Fourier transformation, wavelet transformation and statistical representation of EEG data. Their ability to perform in real-time manner and their information resolution were obtained using different methods. For this research, an OpenBCI Cyton board was used to capture EEG data in experimental setup and signals were processed using python. Research was done with 5 different classification models with the use of features from Fast Fourier Transform (FFT) and Wavelet Transform (WT). K-nearest neighbor model with features obtained with FFT showed the highest accuracy.

# Table of contents

# List of figures

# List of tables

# Nomenclature

**Acronyms / Abbreviations**

| | |
|---|---|
| BCI | Brain Computer Interface |
| CNN | Convolutional Neural Network |
| CSV | Comma-separated Values |
| DFT | Discrete Fourier Transform |
| DWT | Discrete Wavelet Transform |
| EEG | Electroencephalography |
| EOG | Electrooculogram |
| ERD | Event-related Desynchronization |
| FFT | Fast Fourier Transform |
| FPR | False Positive Rate |
| IC | Intentional Control |
| ICA | Independent Component Analysis |
| LDA | Linear Discriminant Analysis |
| LSL | Lab Streaming Layer |
| MI | Motor Imagery |
| NC | Non-Control |
| RNN | Recurrent Neural Network |

| | |
|---|---|
| SMR | Sensorimotor Rhythm |
| SSVEP | Steady-State Visual Evoked Potential |
| SVM | Support Vector Machine |
| TPR | True Positive Rate |
| WT | Wavelet Transform |

# Chapter 1

# Introduction

## 1.1 Background

Brain-Computer Interfaces have been thrust into the limelight in recent years as a source of innovative applications for many fields like medicine and gaming industry. The research on this area belongs to a wide range from improving the conditions of patients with damages in the central nervous system to developing entertaining gaming accessories for healthy individuals. New directions are introduced to the BCI applications often and, using BCI to control virtual environments is one of them. Unlike the early days, hardware components had to be built to test and develop applications, with virtual environments researchers get a good testing ground to develop their applications.

Obtaining the thought commands of a person is a major task of a BCI system. The laboratory-based BCI systems often use a synchronous method where the subjects are guided towards certain intentions by visual or verbal cue. Not only this method is far from the natural behaviour of the mind but also can have certain limitations when it comes to practical applications. Alternatively, researchers have introduced self-paced systems where the subjects are free to decide when to have controlling commands. Here the BCI recognizes them based on the state of their thought commands which may belong to either Intentional-Control state or Non-Control state. However self-paced BCI systems are more challenging than the former as there are major concerns with the false positive rate that comes with signal detection. Brain-sensing is done in three types of devices; invasive, partially invasive and non-invasive. The most common approach that has been used in research is non-invasive EEG based devices. The key advantages of EEG for BCI are the maturity of the technology, relative ease of use and low in cost, as well as the robustness, portability and versatility of recent EEG devices [1]. However, there are

major concerns regarding receiving distorted signals as the devices are located far from the neurons. So far the motor imagery method has been the most used asynchronous method in BCI research where the subjects imagine the movement of their arm or leg without moving them physically. However, there are cases where some subjects like patients who have been paralyzed for long periods were not able to obtain the expected results. Alternatively, researchers have focused on non-motor imagery mental tasks. However, this area is still novel to the field of BCI research. In this paper, we focus on the possibility of using motor imagery signal processing and classification techniques in non-motor imagery mental tasks for feed into virtual object controlling applications.

### 1.1.1 Problem and proposed solution

The current methods used in non-MI based BCI applications such as virtual object controlling has low accuracy. More novel solutions have been introduced in MI based researches and our goal is to apply some of those latest approaches and determine whether they increase the system accuracy as well as whether those approaches applicable for real time systems. Different signal processing and artifact removal methods such as ICA, filtration methods, wavelets methods can be used for better artifact removal. BCI systems have shifted towards deep learning algorithms rather than traditional machine learning algorithms. Convolution Neural Networks and Recurrent Neural Networks have been some of the latest algorithms that were used to classification of data. Classification of data will be our next milestone, here we will be focused on removing signal artifacts and processing the signal. .

The first chapter of this reports provides an overview of the research. The next chapter discusses the related work that have been done. The methodology and implementation details are discussed in third and fourth chapters respectively. The next chapter includes the results and the last chapter provides the conclusions and future work.

# Chapter 2

# Related Work

In our literature review, we observed that there are certain challenges yet to be addressed when using non-invasive BCI for a real-time application. Out of the research done on self-paced BCI systems, most of them are focused on motor-imagery control whereas research on non-motor imagery mental tasks was limited.

## 2.1 EEG based different approaches that have been used for controlling virtual objects

### 2.1.1 P300

P300 is the positive component of the evoked potential that may develop after about 300 ms an item is flashed. The user focuses on one flashing item while ignoring other stimuli. Whenever the target stimulus flashes, it yields a larger P300 than the other possible choices. P300 BCIs are typically used to spell but have been validated with other tasks such as control of a mobile robot, smart home control and even hybrid versions of virtual object controlling. Using external stimuli is one of the major limitations of this system when it comes to controlling virtual objects.

### 2.1.2 SSVEP

SSVEP (Steady-State Visual Evoked Potential) occurs when sensory stimuli are repetitively delivered rapidly enough that the relevant neuronal structures do not return to their resting states. In this approach, the user focuses on some external stimuli usually flickering lights according to a given frequency. Use of external stimuli is also considered as one of the limitations of this design. When this approach comes to virtual object

control it is not a very attractive idea for better users' experience. Because instead of looking at the virtual object the user has to focus on another external stimulus. In 2013 Lei Cao and Jie Li used P300 and SSVEP approaches to control a wheelchair that had an average TPR of 14.18% and FPR of 0.49% [2]. In 2014 the same group used motor imagery with SSVEP to control the wheelchair. It improved in accuracy up to a value of 85.7% [3].

### 2.1.3 ERD-based BCI (ERD BCI, SMR BCI)

ERD-based BCI describes any BCI system that relies on the detection of amplitude changes in sensorimotor rhythms (mu and central beta rhythms) or other brain oscillations, including short-lasting post imagery beta bursts (beta ERS, beta rebound). According to neuroscience, when a person performs motor execution (ME) or motor imagery (MI), the amplitude or energy of mu and beta rhythms in a specific area of the primary sensorimotor will decrease, resulting in event-related desynchronization (ERD) [4], [5]. Motor imagery results in a somatotopically organized activation pattern, similar to that observed when the same movement is really executed. This is the widely used approach for controlling virtual object research because it is much more convenient and can have a better user experience. Our research is also based on ERD, where the ability to controlling virtual objects using mind intent is used, rather than mapping it to a MI task.

There are widely used two methods that can properly evaluate the performance of an asynchronous BCI. They are the true-positive rate (TPR) and the false-positive rate (FPR). A TPR outcome results from correctly classifying a command as an IC state, and FPR outcome results from the BCI misclassifying a no control as an IC state.

In 2007, Graz university of technology [6] conducted a research where MI was used to operate a person in a virtual environment which resulted in 50% in TPR and average FPR to be less 10%. The classification method used here was LDA. In 2010 Graz university of technology was able to increase TPR to 79% and reduce the FPR down to 0.67% per minute [7] with the usage of SVMs. This is a major improvement. In research by Faradj et.al [8], they explored the idea of rotation of a virtual object in 3D space in a more natural way. They used auto scalar auto-regressive methods for feature extraction and the classification was done with quadratic discriminant analysis. They obtained a TPR value of 54.6% TPR and 0.01% FPR. Both non-motor imagery EEG signals related to virtual object manipulation and motor imagery EEG signals are sensorimotor rhythms (SMR). These are specific brain waves over the sensorimotor cortex that are generated after MI or ME. Although there are numerous researches on using motor imagery to control virtual objects that give us higher accuracy [9], research done by Faradji explores

the possibility of controlling objects in a more natural way. It was stated that although the TPR is relatively low compared to MI related research, this method is more preferable in real-time applications since this method requires less computational power.

## 2.2 Challenges of EEG Based Self Paced Brain-Computer Interface

One of the main difficulties in self-paced BCI's is to properly determine the intended control state and non-intended state. Although there have been some advanced methods that have been introduced there has not been enough research to study whether those methods are reliable when it comes to controlling virtual objects using non-motor imagery mental tasks. Research done by Graz University of technology states that they faced difficulties in properly determining the threshold that IC state is detected because the threshold value varied from subject to subject [6]. They have also detected a delay in EEG components related to IC or NC state to appear, which results in a longer period of activity time. Operating BCI for a longer period of time affects its performance. Proper feature extraction methods, and short trials with a limited time range (0 s–15 s) for subjects, would resolve these issues. Another major issue is that not everyone is able to perform as expected with BCI systems. This is sort of a barrier in adapting BCI systems to the needs of people. A research done by Blankertz et al. in 2020 states that only 80% of subjects were able to control the BCI system successfully [3]. This issue is addressed in the research done by Krusienski et al. They suggest solutions like covariate shift, feature adaptation and classifier adaptation [10].

Large amounts of artifacts within the EEG signal make it hard to identify the proper brain waves related to non-MI or MI tasks. In the research done by Zhang et al. in 2017 this issue is addressed [11].

To overcome these issues different mechanisms and algorithms have been introduced in MI-based research. It is yet to be determined whether they are applicable in non-MI intent. Our goal is to introduce these solutions into non-MI based BCI systems and improve the TPR and FPR values. We explore different artifact removal methods such as ICA, wavelet transformation, and filtering methods. In order to identify the IC and NC states, used machine learning algorithms that are more advance than the previously used ones.

# Chapter 3

# Methodology

The main focus of this research is to apply advancements in MI-based BCI to non-MI-based BCI systems to improve its TPR and FPR values than prior researches. Since this system should possess the capability to function in real-time, the computational cost is a major concern in selecting algorithms and methods to follow. In terms of determining the IC and NC states, classical machine learning algorithms were used.

## 3.1 Conceptual Design

The procedure of self-paced BCI module for virtual object controlling consists of 8 steps.

1. The subject should know what are the expected mind intents need to be performed since it is important to induce brain waves related to those activities.For the case of controlling a virtual object, the activities would be moving the object up and down, left and right.

2. Subjects should be trained for specific amount of mind intents without receiving visual feedback. Here the subject would be trained on moving a stationary virtual object.

3. Pre-processing the data by artifact reduction(Electrooculogram (EOG), Electromyogram (EMG)) and signal filtering methods such as low-pass/high pass or band-pass filter.

4. Feature extraction is done to find a suitable representation of the electrophysiology data that simplify the subsequent classification or detection of specific brain patterns.

5. With the extracted features, classifiers will be trained until an acceptable accuracy level is reached.

6. The predictions done by the trained model will be fed to the virtual environment in real time.

7. Subject is trained with visual feedback.

8. Update the classifier if the frequency band or EEG pattern changes(Post-processing).

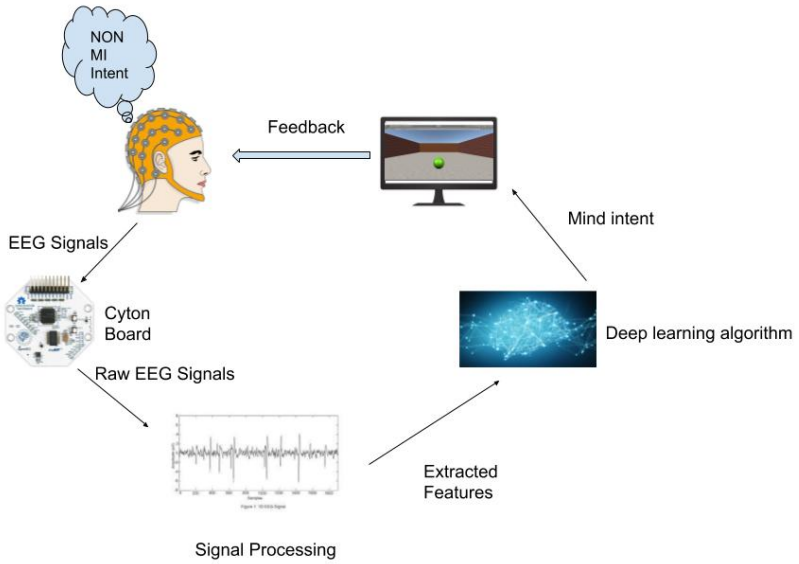


Fig. 3.1 Full experimental setup

Figure 3.1 visualizes the milestones through our research. At this point, we have completed up to signal processing in order to extract required features from EEG signals. Initially, the mind intents that we planned to identify should be decided and the subjects should be trained to focus on those intents. In 2010 Graz university of technology stated that not all subjects are able to control virtual objects with prior training. So the subject was trained without visual feedback on three intents (left, right, and none).

Brain waves are captured as EEG signals using the Cyton Board by OpenBCI. The raw EEG signals are sent to the computer to be processed from the Cyton board using the dongle that is connected to the computer. This procedure uses Bluetooth communication. Communication from dongle to the computer is done through serial communication.

After getting the raw EEG signals different signal processing methods are used to denoise the signal. Then different signal representations such as fast Fourier transform, wavelet transform are used to represent the signal and remove the signal artifacts. These signal representation are used to train the classical machine learning algorithms.

## 3.2 Methodological Approach

When a person engages in activities electro-chemical signals are passed through neurons. An electric potential is generated as a result. We measure this electric potential using a non-invasive EEG based approach.

### 3.2.1 Signal Representation

Brains waves are measured with analog EEG signals which include a representation of magnitude and time stamp. The magnitude is in the micro voltage range. Brain waves consist of alpha, beta, gamma, theta and delta signals as shown in Table 3.1.

Table 3.1 Frequency bands of brainwaves

| Brainwave | State |
|---|---|
| Gamma (32-100 Hz) | Heightened perception, learning, problem-solving tasks, cognitive processing |
| Beta (13-32 Hz) | Awake, alert conscious, thinking, excitement |
| Alpha (8-13 Hz) | Physically and mentally relaxed |
| Theta (4-8 Hz) | Creativity,insights, deep states,dreams,deep medication |
| Delta (0.5-4 Hz) | Deep sleep, loss of bodily awareness, repair |

Signals related to controlling virtual object exists within the beta and gamma ranges. Therefore we have focused more on Beta and Gamma ranged signals throughout our research. Signals in the range of 13Hz-60 Hz are considered in here.

As this system functions in real-time it is really important to reduce computation cost. So we downsample EEG signals that we receive, in a way that the information in those is preserved. Downsampling is done with Nyquist Sampling Theorem. It states that a bandlimited continuous-time signal can be sampled and perfectly reconstructed from its samples if the waveform is sampled over twice as fast as it's highest frequency component.

Nyquist limit: the highest frequency component that can be accurately represented.

$$f_{max} = f_{s/2}$$

Nyquist frequency: sampling rate required to accurately represent up to.

$$f_s > 2 * f_{max}$$

In this research, raw EEG signals are mainly transformed into two representations. They are Fast Fourier Transformation and Wavelet transformation.

**Fast Fourier Transformation**

Fourier analysis converts the signal from its original domain (time or space) to its frequency domain components and vice versa. FFT is an algorithm that computes Discrete Fourier Transform (DFT) or its inverse. DFT is defined as

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$$
$$k = 0, ..., N-1$$

Where:

$N$ : number of time samples we have

$n$ : current sample we're considering (0,.., $N-1$)

$x_n$ : value of the signal at time $n$

$k$ : current frequency we're considering (0 Hz up to $N-1$ Hz)

$X_k$ : amount of frequency $k$ in the signal (amplitude and phase, a complex number)

Direct computation of DFT has a higher computational cost of $O(N^2)$. Since we are working with a real-time system it is important to optimize the computational cost. FFT has a lesser computational cost of $O(NlogN)$ in comparison to the direct method. When it comes to FFT, it has high resolution in frequency domain but zero resolution in the time domain, so we can determine what frequencies are the frequencies present in the signal but not the time that they occurred.

Given below is an overview on the advantages and limitations of FFT representation.

Table 3.2 FFT advantages and limitations

| Advantages | Limitations |
|---|---|
| Low computational cost | FFT representation has no temporal information |
| We can determine the magnitude of each frequency component | EEG signals are non-stationary signals FFT is not the best representation when it comes to non-stationary signals.<br>FFT has large noise sensitivity, and it does not have a shorter duration data record. |

EEG signals are non-stationary since it does not keep the time period and spectral content values inconsistent.

**Wavelet Transformation**

Wavelet transformation is a better approach to represent signals with a dynamical frequency spectrum. As EEG signals are non-stationary signals, the most applicable way for feature extraction from raw data(time series data) is the use of time frequency-domain methods like wavelet transform, where any general function can be expressed as an infinite series of wavelets. Wavelet transform has high resolution in both frequency domain and time domain. To get more flexible representation, variable size time windows are used in wavelet transformation. To capture low frequencies long time windows are used and for high frequencies, short time windows are used. This is a multi-scale structure method so it resolves the issues of EEG representations which is a non-stationary signal. The original EEG signal is represented by known blocks of wavelet signals. For EEG signal representation we have been using discrete wavelet transform. Every scale represents a unique thickness of EEG signal. EEG signal undergoes multi-scale decomposition of raw EEG data where each step consists of two digital filters with high pass nature and low pass nature.

Where:

$g[n]$ : high pass nature wavelet

$h[n]$ : low pass nature wavelet

DWT (Discrete Wavelet Transform) algorithm is as follows.

$$\Psi_{j,k}(t) = 2^{j/2}\Psi(2^j t - k)$$

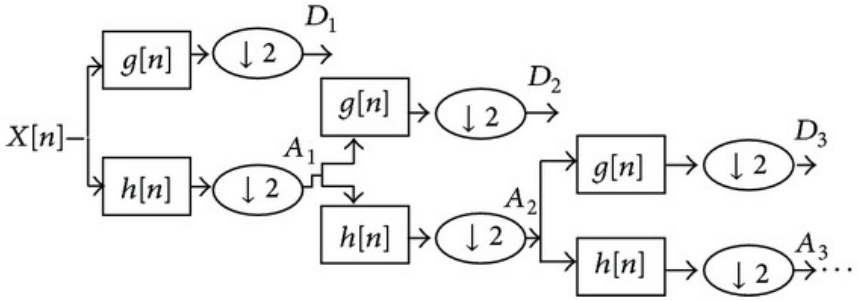


Fig. 3.2 Multi structure signal decomposition

where $j$ and $k$ are integers.Then the wavelet transform can be performed by:

$$W_\Psi =< f, \Psi_{j,k}$$

which means the inner product of time-domain signal and wavelet function.
The DWT derived fro the continuous wavelet can be applied when the input signal, and the decomposition can be expresses as:

$$X_{a,L}[n] = \sum_{k=1}^{N} X_{a-1,L}[2n-k]g[k]$$

$$X_{a,H}[n] = \sum_{k=1}^{N} X_{a-1,L}[2n-k]h[k]$$

In FFT the frequency coefficients will be its features which will be fed into our next stage.

Table 3.3 WT advantages and limitation

| Advantages | Limitations |
|---|---|
| Varying window size allows having a flexible representation of the signal. | Need to select a proper mother wavelet. |
| More suitable for non-stationary signal analysis. | More computational cost compared to FFT. |

In Wavelet transformation the approximation coefficients and the detailed coefficients will be its features. These features will be fed into deep learning algorithms to determine the mind intent.

### 3.2.2 Artifact Removal

EEG signals have a high temporal resolution, so it is more vulnerable for undesired noise, which will result in various artifacts. These artifacts may occur from either equipment or test subjects. Equipment artifacts can be removed by following strict experimental methods but the removal of psychological artifacts is harder. Examples for psychological artifacts are eye blinking, eye movement and cardiac activity or muscle activity. These psychological artifacts contaminate the neural information and also can mislead the BCI to identify it as a common scenario. A number of efficient methods have been introduced to remove artifacts. In our research, we have followed the following two methods.

1. **Artifact removal prior signal transformation**

   **Independent Component Analysis (ICA)**
   Considering that signal sources are instantaneous linear mixtures of cerebral and artifactual sources, we can decompose the signal into independent components. Independent components which are contaminated with artifacts can be removed and clean signals will be reconstructed. This signal has a higher information percentage since only artifacts are removed and information within those frequency ranges are preserved yet it has a higher computational cost. Ocular artifacts and muscle artifacts that contaminate the signal can be removed using this method.

   **Filtering Method**
   Oculus artifact occurs in the frequency range of 0 Hz–5 Hz and muscle artifact occurs and in the range of 40 Hz–100 Hz. We use a bandpass filter to filter out these ranges and get an EEG signal in the range of 5 Hz–60 Hz. Although this removes the non-contaminated information in these ranges. As we mentioned above non-MI intent occurs in the frequency range of 13 Hz–60 Hz so it would still give out information in a higher resolution. This method has a lesser computational cost.

2. **Artifact removal after signal transformation**

   **Fast Fourier Transformed Data**
   After Fourier transformation, we can remove the frequency coefficients related to ocular artifacts and muscle artifacts easily.

**Wavelet Transformed Data**
**Threshold Method**
In wavelet transformation, we can introduce a threshold to remove the artifact and remaining details can be added up to create clean signals. Following are the steps for applying threshold approach.

(a) To apply stationary wavelet transform to the contaminated EEG signals and decompose it up to eight levels with Symlet (Sym3) as a basis function.

(b) To identify the spikes in the contaminated EEG at each level.

(c) To identify the ocular artifact spike zones using coefficient of variation, a statistical approach.

(d) To apply denoising technique – To fix the suitable threshold value and threshold function for the artifact zones.

(e) To apply wavelet reconstruction procedure to reconstruct the EEG signal.

**Non-Threshold Method**
By decomposing the signal into a few levels by using a suitable mother wavelet the bandwidth can be divided and the signal can be reconstructed by rejecting unwanted detailed coefficients.

### 3.2.3 Classification Methods

After features were extracted classification methods were used to determine commands the EEG signals are related to. There has been a shift towards deep learning algorithms when it comes to MI based BCI research. Some of the classification methods that we used are random forest, linear discriminant analysis (LDA) ,quadratic discriminant analysis (QDA),KNearest neighbour algorithm and Catboost Algorithm.

1. **Random Forest**

Random forest consists of multiple decision trees that operate as ensembles. Each tree provides a class prediction where the class with the most number of votes will be the model's prediction. EEG data has more probability towards making the model overfit towards the training data set. Random forest models have the ability not to overfit this increases the accuracy of the prediction. By changing the models hyper parameters we can further increase the accuracy of the model.

2. **Quadratic Discriminant Analysis (QDA)**

QDA is a variant of LDA in which an individual covariance matrix is estimated for every class of observations. QDA is particularly useful if there is prior knowledge that individual classes exhibit distinct covariances.Observation of each class is drawn from a normal distribution.QDA assumes that each class has its own covariance matrix .When these assumptions hold, QDA approximates the Bayes classifier very closely and the discriminant function produces a quadratic decision boundary.A disadvantage of QDA is that it cannot be used as a dimensionality reduction technique.

3. **K-Nearest Neighbour Algorithm (KNN)**

K-nearest neighbors algorithm is a non-parametric machine learning which is used for classification and regression. In both cases, the input consists of the k closest training examples in feature space.K-nearest neighbors algorithm uses 'feature similarity' to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

4. **Catboost algorithm**

CatBoost is based on gradient boosted decision trees.It is developed by Yandex researchers and engineers, and is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks at Yandex and in other companies, including CERN, Cloudflare, Careem taxi. During training of this algorithm a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees.The number of trees is controlled by the starting parameters. To prevent overfitting, use the overfitting detector. When it is triggered, trees stop being built.

5. **Support Vector Machine (SVM)**

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points to separate the K(the number of classes) classes of data points, there

are many possible hyperplanes that could be chosen. Objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of K classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

# Chapter 4

# Experimental Setup and Implementation

Our subject was a male volunteer, of age 24. Initially the subject had to train for three mind intents which are left, right, and none with a virtual stationary object (sphere). This training was done in a limited time trial like 0-10 seconds, because the performance of the mental task degrades over time. Subjects' brain waves will be captured with the Cyton board which is shown in Figure 4.2. EEG time-series signals will be fed for processing and denoising from OpenBCI GUI to a Python application through Lab Stream Layer (LSL) for the feature extraction. This setup is visualized in Figure 4.1

## 4.1 Research Instruments

### 4.1.1 Cyton Board

Cyton board is an Arduino compatible wireless device which is able to capture EEG signals. It consists of 8 biopotential input channels. It must be powered up with 3 V–6 V DC battery only. It has the ability to send samples at a frequency rate of 250 Hz. Each packet contains a header followed by a sample counter, 8 Analog to Digital Converter (ADC) channel data, the three axes values of the accelerometer, and a footer. The USB dongle is connected to the laptop where the Cyton board communicates with it using Bluetooth to transfer data.

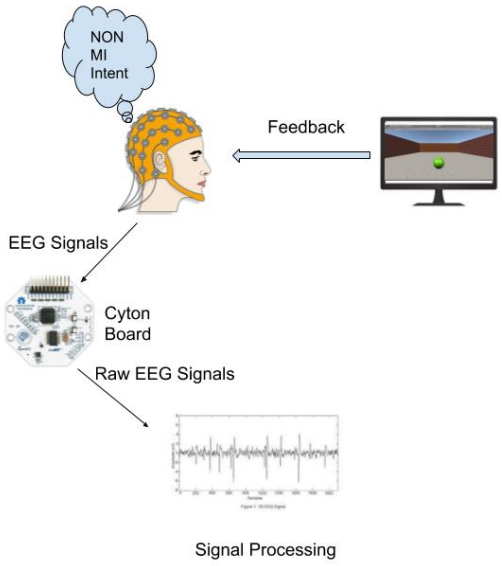


Fig. 4.1 Experimental setup to feature extraction

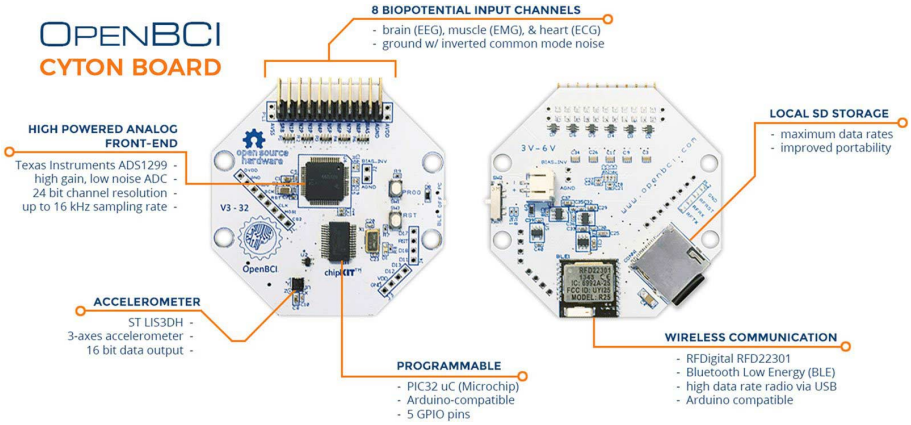


Fig. 4.2 Cyton board

### 4.1.2 Electrodes and electrode placement

Golden cup electrodes are used to sample EEG data. These are placed on the subject according to the 10-20 method. The 10–20 system or International 10–20 system is an internationally recognized method to describe and apply the location of scalp electrodes in the context of an EEG exam. EEGs are placed in 10% and 20% spaces on the scalp as shown in Figure 4.3.

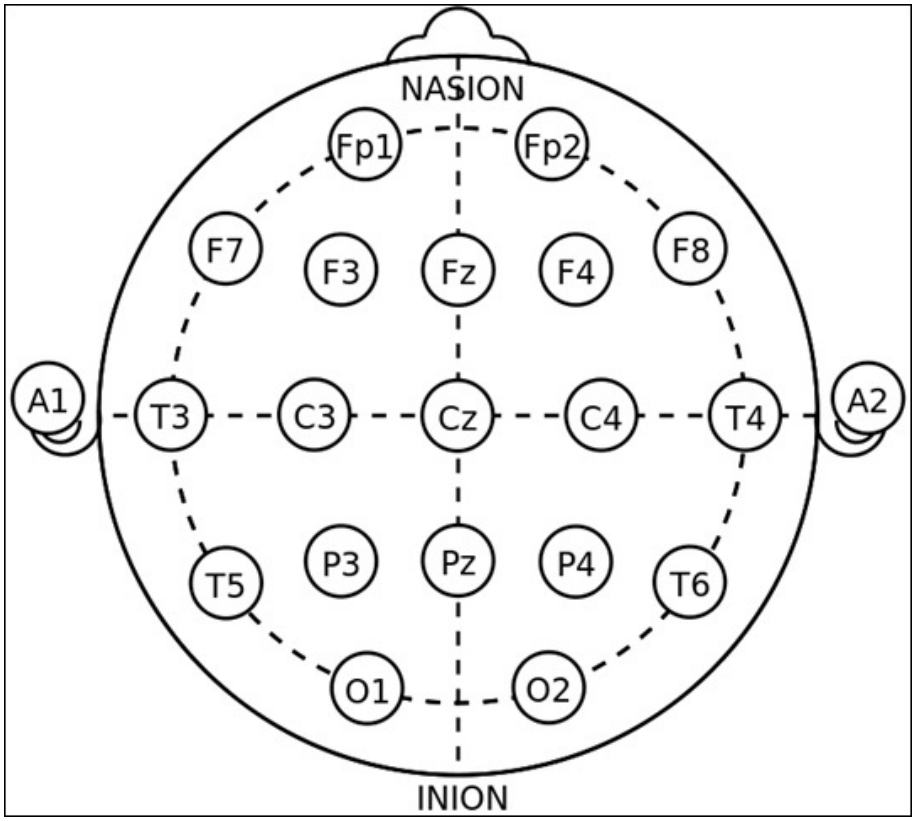


Fig. 4.3 10-20 method electrode placement

We placed eight electrodes in eight of these positions. The brain waves related to controlling virtual object are induced in the motor cortex so electrode placement positions are chosen so as we can extract the maximum amount of information. In our experiment, we placed electrodes as shown in Figure 4.4.

FP1, FP2, C3, C4, T3, T4, P3, P4 positions have been chosen in order to extract motor cortex information related to virtual object controlling. A1 and A2 act as the relative points. These electrodes act as references for all of the EEG electrodes placed on the head. Pinna or the outer part of the ear does not contain any neurons therefore electro chemical reactions don't occur. Due to this pinna can be considered as a neutral source compared to other parts of the body.

### 4.1.3 Graphical User Interface

Virtual objects that were meant for controlling are created with Unity. The subject is trained on a virtual environment where the display is 15.6 inch, monitor resolution of

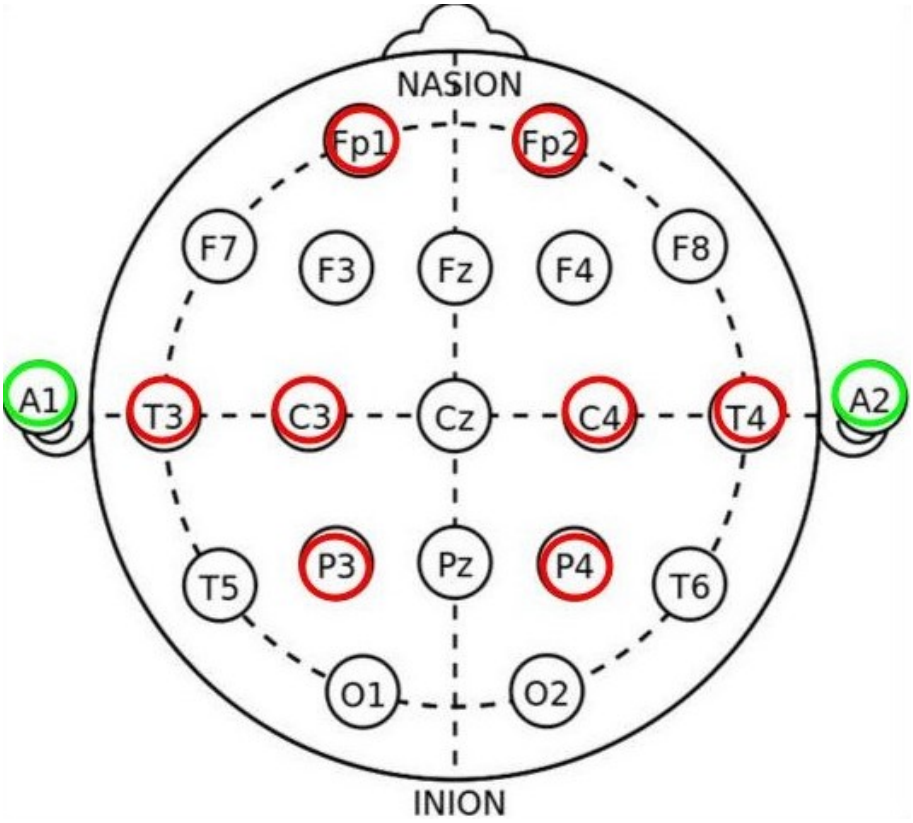


Fig. 4.4 10-20 experimental electrode placement

1920 x 1080 p and 60Hz refresh rate. Data of mind intent will be recorded where the subject will focus on moving the objects along axes. The GUI is shown in Figure 4.5.

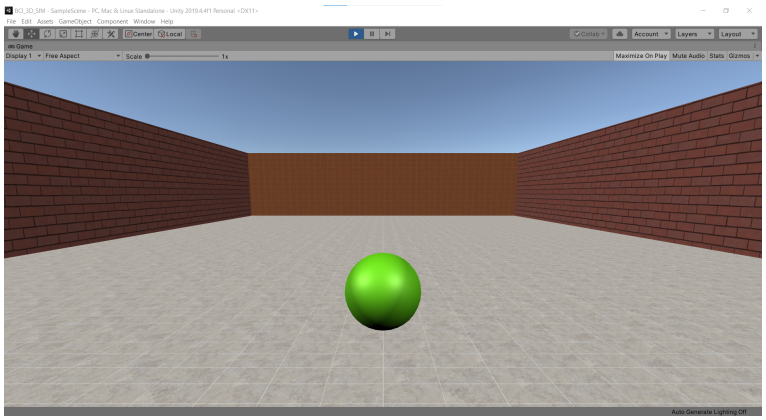

Fig. 4.5 GUI interface

### 4.1.4 OpenBCI GUI and LSL

OpenBCI GUI is a powerful software that is used to visualize, record and stream data from OpenBCI boards. This GUI helps to visualize data coming from eight channels of Cyton board to understand if there are any faults in connections. If there are external disturbances that interfere with the visualization of EEG signals it can be recognized as well. It also visualizes the real-time representations of FFT, power spectral distribution and time series Figure 4.6.

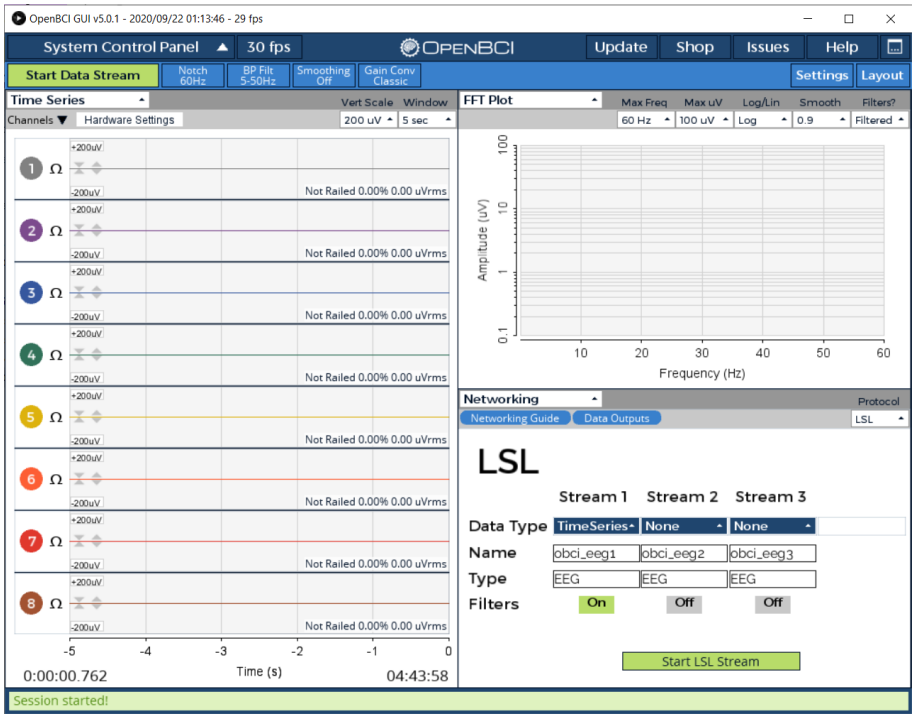


Fig. 4.6 OpenBCI GUI

Lab Streaming Layer Figure 4.7 is a system developed for synchronising streaming data for real-time analysis and recording. This is used to send the raw EEG data as time series, into a Python application for signal processing. PyLSL library is used to feed the data to the Python application. Data is transferred at 250 Hz. Each sample contains data of the 8 channels as floats.

## 4.2 Data Manipulation

Initially, EEG data is notched on 50Hz from the OpenBCI GUI and transferred to python applications. This data was saved in CSV format. Data manipulation was done according

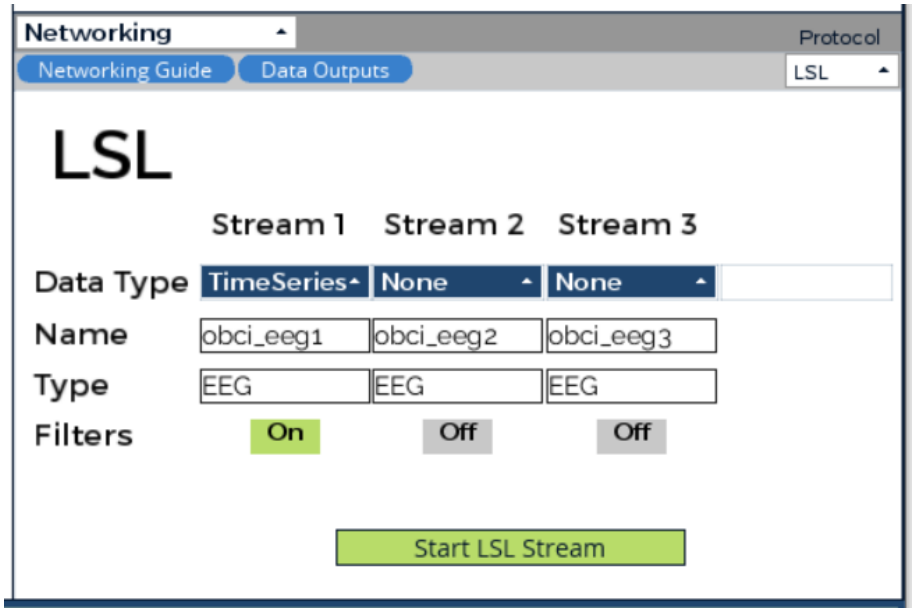


Fig. 4.7 LSL Settings

to the different artifact removal methods that were followed.
For Independent Component Analysis (ICA) and artifact removal after signal representation methods, the raw data was used as it is. For artifact removal method prior signal representation EEG data is band passed in the range of 10 Hz–60 Hz. This range is chosen as the beta and gamma signals are within this range. The components above 60 Hz are weak so information that can be extracted from them is low. Band passing in the range of (10 Hz–60 Hz) won't affect a significant loss of data.

## 4.3 Implementation

The feature extraction of this research is based on various kinds of signal representation methods. Since EEG signals are non-stationary signals, the signal representation should be powerful enough to represent the information gain of the signal. Base libraries that have been used in this research are given below.

- **Scipy** - SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering.
- **Numpy** - NumPy is a python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

- **Pylsl** - Python interface to the Lab Streaming Layer (LSL). LSL is an overlay network for real-time exchange of time series between applications, most often used in research environments.
- **Matplotlib** - Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- **MNE** - Open-source Python package for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, NIRS, and more.
- **PyWavelets** - PyWavelets is open source wavelet transform software for Python. It combines a simple high level interface with low level C and Cython performance.
- **Scikit-Learn** - Scikit learn consists with simple and efficient tools for predictive data analysis, accessible to everybody, and reusable in various contexts. This library is built on NumPy, SciPy, and matplotlib libraries.

### 4.3.1 Signal Representations

There are three main signal representation methods that were used in this research. They are Fast Fourier Transform, Wavelet Transform and Statistical Representation of the signal. Following implementation details are based on channel 1[ FP1 electrode in 10-20 method] data of EEG. Only critical code snippets are mentioned below.

#### Fast Fourier Transform

The predefined method *scipy.fft* is used for implementing fast Fourier transform.
Method signature

```
fft(x[, n, axis, norm, overwrite_x, ...])
```

returns 1 dimensional numpy array
This method Computes the 1-D discrete Fourier Transform. example implementation for channel1_data(1 dimensional array) of the EEG signals

```
from scipy.fft import fft, ifft
fft_signal = fft(channel1_data1)
```

Consider the signal $y[N]$ where $N$ stands for length. For $N$ even, the elements $y[1]...y[N/2-1]$ contain the positive-frequency terms, and the elements $y[N/2]...y[N-1]$ contain the

negative-frequency terms, in order of decreasingly negative frequency. For $N$ odd, the elements $y[1]...y[(N-1)/2]$ contain the positive-frequency terms, and the elements $y[(N+1)/2]...y[N-1]$contain the negative-frequency terms, in order of decreasingly negative frequency. In case the sequence $x$ is real-valued, the values of $y[n]$ for positive frequencies is the conjugate of the values $y[n]$ for negative frequencies (Because the spectrum is symmetric). Typically, only the FFT corresponding to positive frequencies is plotted as shown in Figure 4.8.
Therefore the FFT signal is transformed as follows.

```
fft_pdata  = 2.0/N * np.abs(fft_signal[0:N//2])
```

**np.abs(array)**:give absolute values of the array elements 10 - 60 Hz is the frequency

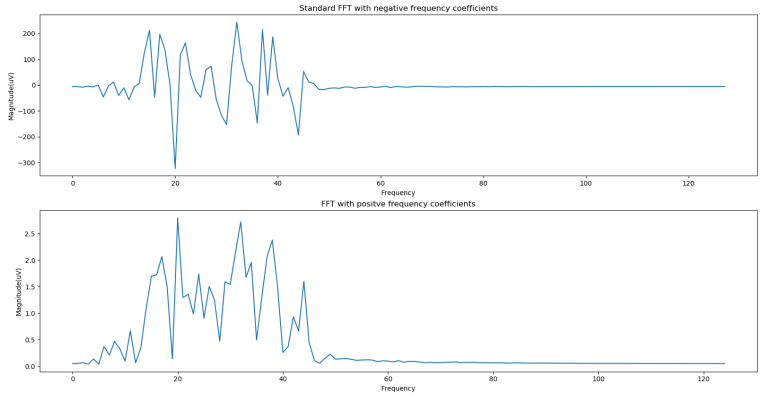


Fig. 4.8 FFT of the signal with positive frequencies

band that is taken into consideration in this research. After that, the FFT is extracted for this bandwidth.

### Wavelet Transform

Daubechies 4 (db4) wavelet Figure 4.10 is used as the mother wavelet in this research since it is most suitable to process biomedical signals. The input signal has a frequency band of 0 Hz–250 Hz. In this research, our interest area is 0 Hz–60 Hz for EEG signals. Since the relevant frequency band lies in the Gamma and Beta range (16 Hz–63 Hz), the filtered signal will be decomposed only up to level 5 to obtain the beta and gamma bands in CD3 and CD4 as shown in Table 4.1.

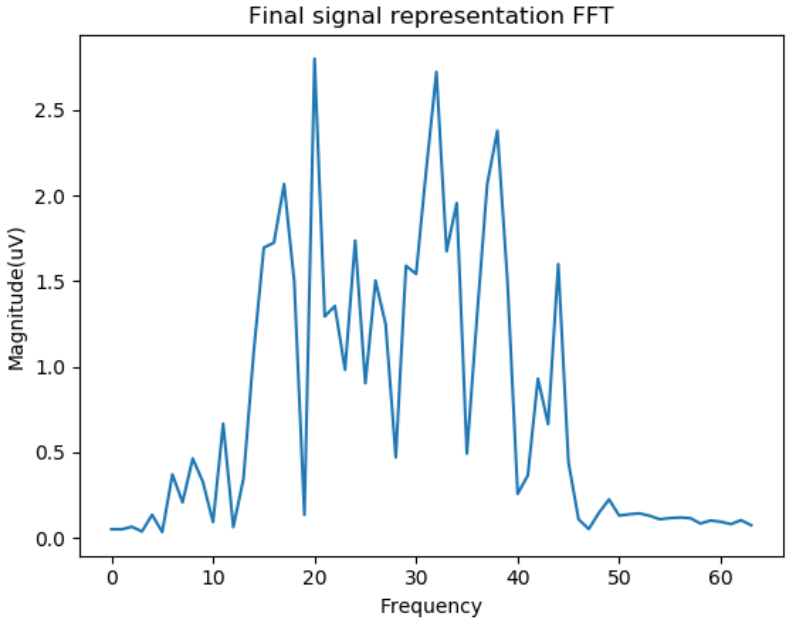


Fig. 4.9 Extracted FFT representation (0 Hz–60 Hz)

Table 4.1 Wavelet transform decompose details

| Wavelet coefficient | Frequency range(Hz) | Signal information |
|---|---|---|
| D1 | 125 - 250 | Noise |
| D2 | 63 -125 | Noise |
| D3 | 32 - 63 | Gamma |
| D4 | 16 - 32 | Beta |
| D5 | 8 - 16 | Alpha |

Predefined method pywt.wavedec is used for Multilevel 1D Discrete Wavelet Transform of data for the implementation.
Method signature

```
pywt.wavedec(data, wavelet, mode='symmetric', level=None, axis=-1)
```

Returns: $[cA\_n, cD\_n, cD\_n-1, \ldots, cD2, cD1]$ : list
Implementation of Daubechies wavelet with 5 levels.

```
#Daubechies wavelet i.e. db5 (level=5).
w_coefficients = wavedec(  data = channel1_data , #data 1-dimensional
                           wavelet = 'db4',    #wavelettype
```

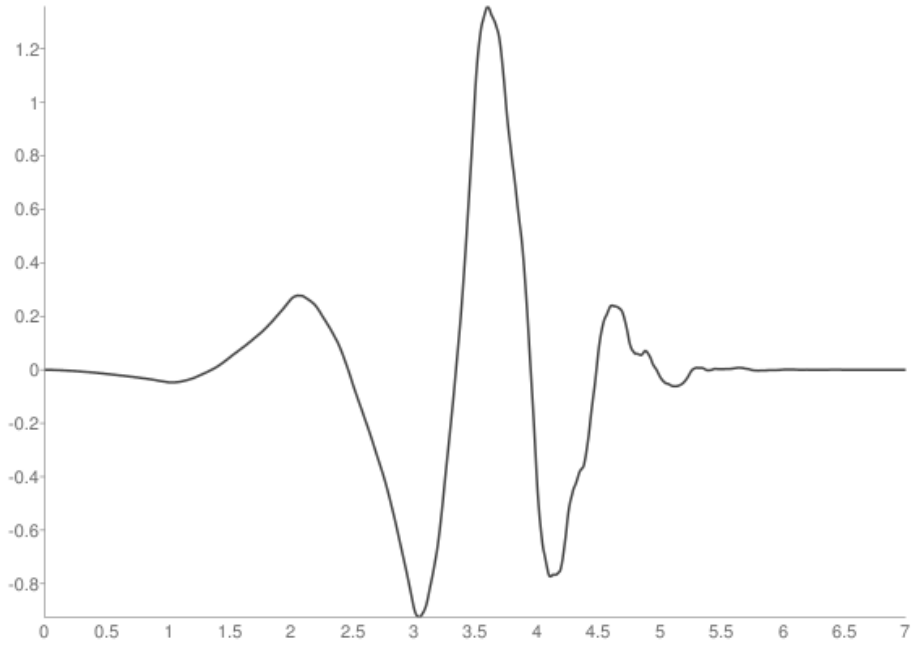


Fig. 4.10 Mother Wavelet

```
                        level = 5)     #number of decomposition level
cA5 , cD5, cD4, cD3, cD2, cD1 = w_coefficients
```

$cAn$ : Approximation coefficient of the decomposition level n

$cDn$ : Detailed coefficient of the decomposition level n

After decomposing data the following graph Figure 4.12 represents the detailed and approximation coefficients. cD4 and cD3 are the detailed coefficients that represent Gamma and Beta bandwidth that are relevant to non motor imagery intent. These coefficients can be used as features for the classification method. Signal also can be reconstructed as time series signal by removing unwanted detailed coefficients which are not related to motor imagery intent. The following redefined method is used to reconstruct the signal.

```
pywt.wavedec
```

Method signature

```
pywt.waverec(coeffs, wavelet, mode='symmetric', axis=-1)
```

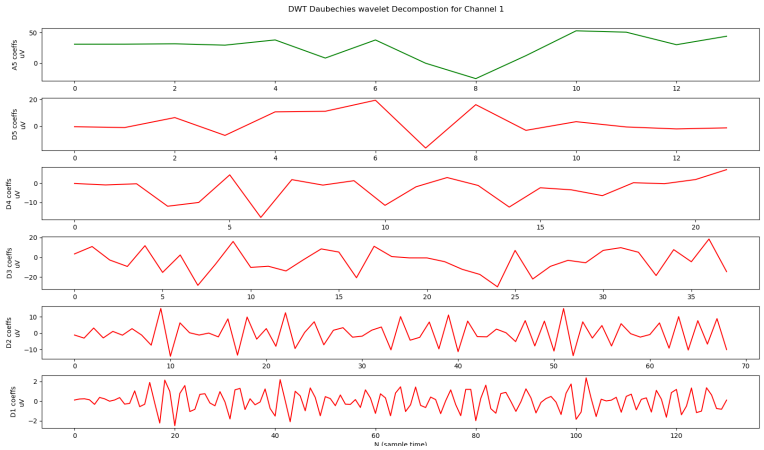


Fig. 4.11 Level 5 decomposition of the signal

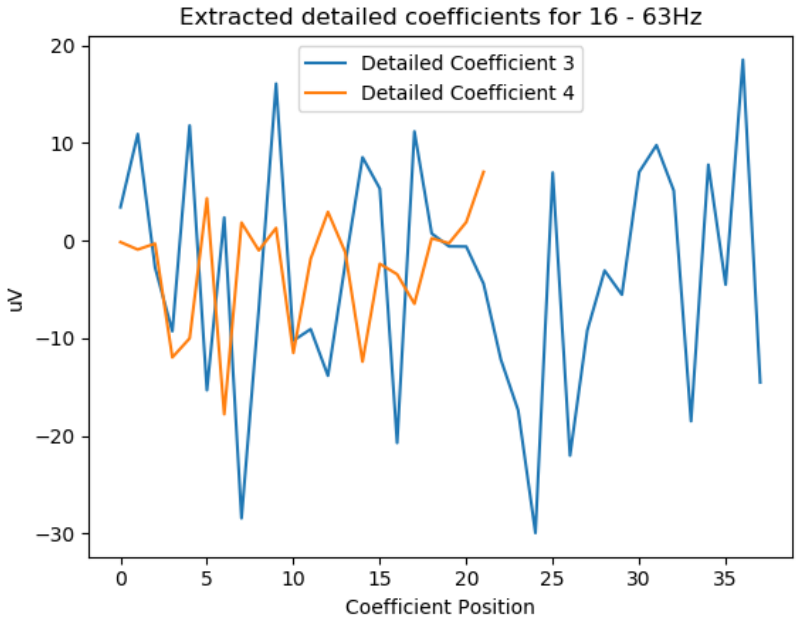


Fig. 4.12 Extracted Features CD4 and CD3

Before reconstructing the signal the unwanted detailed coefficients should be removed. The best way to do this is by setting the corresponding arrays to zero arrays of matching shape and dtype.
Implementation of reconstruction.

```
# to remove 5 th decomposition level
w_coefficients[-5] = np.zeros_like(w_coefficients[-5])
```

```
# to remove 2nd decomposition level
w_coefficients[-2] = np.zeros_like(w_coefficients[-2])
# to remove 1st decomposition level
w_coefficients[-1] = np.zeros_like(w_coefficients[-1])

#reconstruct signal
reconstructed_signal = waverec(w_coefficients,wavelet = 'db4')
```

This method also can be used for extracting wanted frequency bands. This way the ocular artifacts which are appearing in the 0-5 Hz are also rejected.

#### Statistical Representation

Statistical representation of band passed time series is one of the feature extraction methods used in EEG signals. Statistical features such as mean, median, mode, standard deviation, maximum and minimum can be used.
Implementation

```
#calculate time series signal statistics , y: is band passed signal.
ts_mean = statistics.mean(y)  #calculate mean
ts_std = statistics.stdev(y) #calculate standard deviation
ts_max = max(y) #calculate max
ts_min = min(y) #calculate min
ts_median = statistics.median(y) #median
```

### 4.3.2 Artifact Removal

EEG signals are contaminated with artifacts that can defect the real information carried by the signal. Ocular artifact is the main artifact that is focused for removal in this research. For removing those artifacts following methods have been implemented.

#### Independent component analysis (ICA)

This algorithm is implemented using the MNE library . Data of all channels are put into a 2D array and channel data is set up according to the EEG 10-20 methods. Then Apply ICA algorithm and separate components are plotted.

```
raw = mne.io.RawArray(eeg.T, info)
raw.set_montage("standard_1020")  #signal setup to standard 10-20 method
```

```
raw_tmp = raw.copy()  # copy
raw_tmp.filter(1,None) # High pass the signal  over 1 Hz
ica = mne.preprocessing.ICA(method= "infomax",     #apply ICA
                            fit_params = {"extended":True},
                            random_state=1)
ica.fit(raw_tmp)
#plot the independent components
ica.plot_components(inst=raw_tmp,picks= range(22))
```

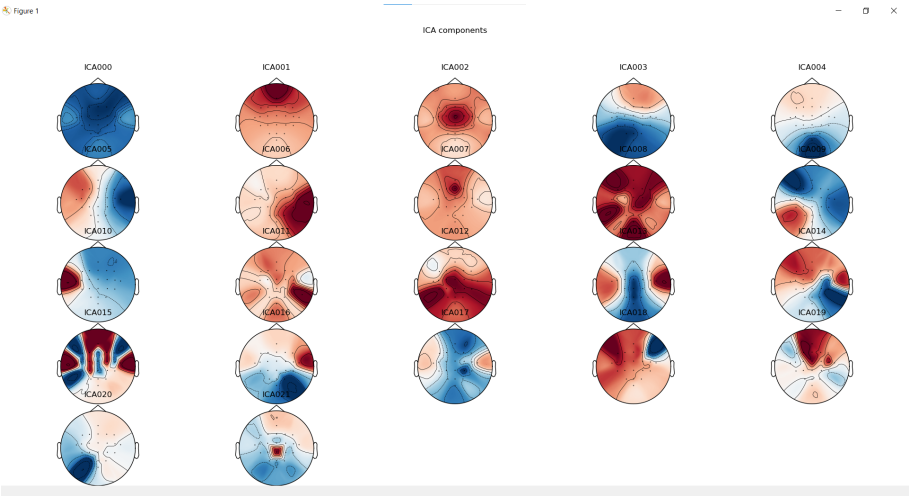


Fig. 4.13 Independent components Mapped to brain area

According to Figure 4.13, ICA001 is the component that contaminates the EOG artifact. So this should be removed to acquire the real signal representation.

```
ica.exclude = [1] #component for removing
raw_corrected = raw.copy() # copy raw data
ica.apply(raw_corrected) # apply removal
```

According to Figure 4.14 it is clear that the ICA algorithm has done great work on removing EOG artifact. But this algorithm takes a long time and cannot be used in real time application in this research.

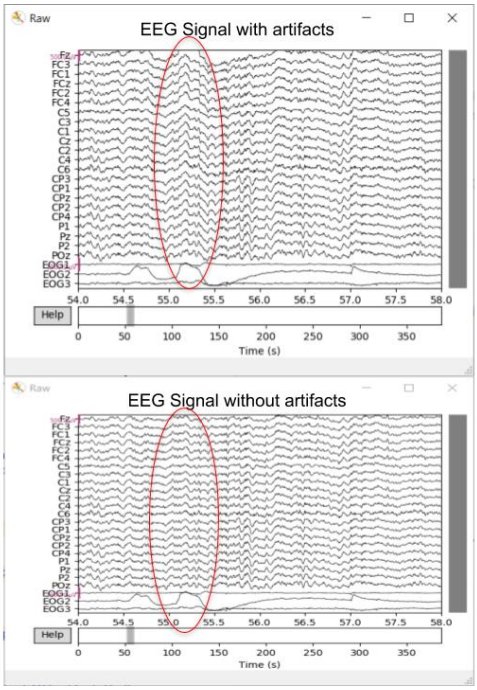


Fig. 4.14 EEG signals after removing EOG artifact

#### Wavelet Transform Threshold Method

This method is mostly used for removing high frequency noises. Artifacts of EEG signals cannot be removed using the universal threshold method directly.

### 4.3.3 Classification

Frequency bins from 15Hz to 60Hz i.e- [15Hz , 16Hz , 17Hz , .... ,60Hz ] is used as features extracted from FFT for training classification models. Detailed coefficients on 4th and 3rd levels are used as features extracted from the wavelet transform method. Following are the classification model implementations.

#### Random Forest

Predefined class RandomForestClassifier from sklearn.ensemble package is used for classification.

```
rfc_channeln = RandomForestClassifier(n_estimators=120,random_state=1,
                    n_jobs=-1,
```

```
                    max_features='log2',
                    min_samples_leaf=2,
                    min_samples_split=5,
                    max_depth=10)
```

Hyper parameters are initially set for a certain value and they are changed to increase the accuracy.

**Parameters**

n_estimators : The number of trees in the forest.

random_state : Controls both the randomness of the bootstrapping of the samples used when building trees.

N_jobs : The number of jobs to run in parallel. fit, predict, decision_path and apply are all parallelized over the trees.-1 means use all processors.

max_features : The number of features to consider when looking for the best split.

min_sample_leaf : The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches.

max_depth : The maximum depth of the tree.

**Quadratic Discriminant Analysis**

QDA is implemented using the QuadraticDiscriminantAnalysis class in the sklearn.discrimina_analysis package.

```
qda_channel1 = QuadraticDiscriminantAnalysis()
```

Default constructor parameters are used when training this algorithm.

**Parameters**

*priors* : default=None. Class priors. By default, the class proportions are inferred from the training data.

*tol* : default=1.0e-4. Absolute threshold for a singular value to be considered significant, used to estimate the rank of Xk where Xk is the centered matrix of samples in class k.

### KNearest Neighbor(KNN) algorithm

KNN is implemented using predefined Class KNeighborsClassifier of sklearn.neighbour package.

```
knn_channel1 = KNeighborsClassifier(n_neighbors=3,n_jobs = -1)
```

**Parameters**

n_neighbors : Number of neighbors to use. This example has used 5 because the number of target classes is 3.

n_jobs : The number of parallel jobs to run for neighbors search.-1 means using all processors.

### Support Vector Machine Algorithm (SVM)

SVM is implemented using predefined class SVC(Support Vector Classification) of sklearn.svm package.

```
svm_channel1 = SVC(kernel='poly', degree=3)
```

**Parameters**

kernel : Specifies the kernel type to be used in the algorithm. The parameter 'poly' stands for polynomial kernel function.

degree : Degree of the polynomial kernel function ('poly'). 3 is used as a degree because there are 3 classes.

### Catboost Algorithm

CatBoost is an open-source software library developed by Yandex. The predefined class CatBoostClassifier is used to implement classification objects.

```
Cat_channeln = CatBoostClassifier(iterations=200,depth=5,
                    learning_rate=1,loss_function='MultiClass',
                    logging_level='Silent',random_state=0)
```

**Parameters**

iterations : The maximum number of trees that can be built when solving machine learning problems.

depth : Depth of the tree.

learning rate : The learning rate.Used for reducing the gradient step.

loss function : The metric to use in training. The specified value also determines the machine learning problem to solve. 'MultiClass' loss function is used because we have 3 classes.

logging_level : The logging level to output to stdout.

### 4.3.4 Python and Unity Communication unit

Unity does not give accessibility for direct process communication. Therefore the communication is done using the socket programming which is done in multiplayer game programming. The Python program which has the data acquisition and identifying thought(AI) acts as the server and the Unity program which has the virtual object act as the client program. But these programs do not work synchronously like typical server-client programs because the life cycle of a virtual object in unity has to synchronize with python response. The output of the python program is a categorical value.

'left':0,'right':1,'none':2

Following Figure 4.15, Figure 4.16 diagrams show how the data flow happens between programs.

Awake() function starts the client program in unity to connect with the python server.The FIFO queue in the background thread is accessed in callable function update() .update function is called once per frame. In this method the locality of the virtual object in quaternion space is updated according to the direction.

Consider moving object in z direction.

```
Vector3 vect = new Vector3(0f, 0f, speed * Time.deltaTime);
transform.localPosition += transform.localRotation * vect;
```

Vector3(x,y,z) : Vector3 Class has a constructor with x,y,z components.

Speed : Speed for moving a certain object.

Time.deltaTime : The time difference between 2 frames.

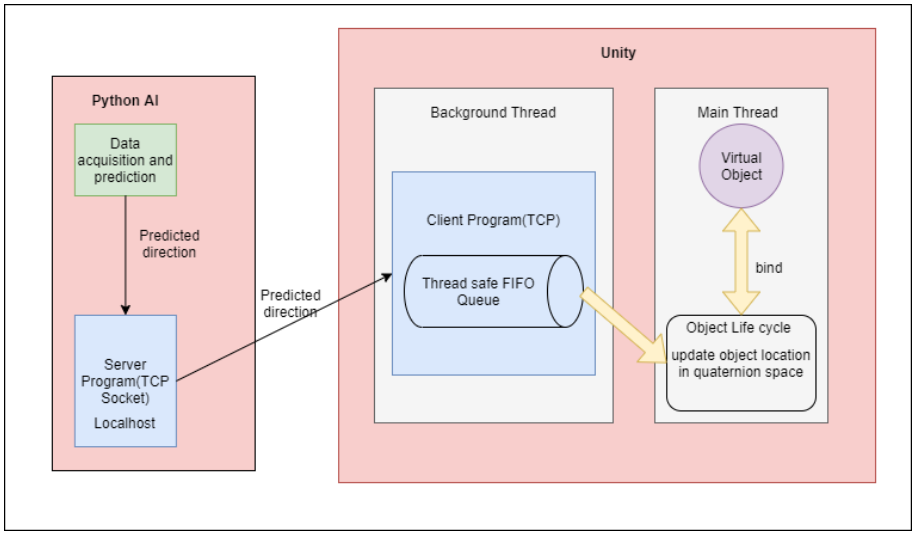


Fig. 4.15 Flow of data server to client program

Vector3 represents the 3D vectors and points. This structure is used throughout Unity to pass 3D positions and directions around. It also contains functions for doing common vector operations.Every object in a Scene has a Transform. It's used to store and manipulate the position, rotation and scale of the object. So update the position of the object the vector is added.

## 4.4 Pitfalls and workarounds

There were certain obstacles that we had faced as we implemented the above step up.

1. The raw EEG data that we got into Python through LSL did not have the EEG signal representation. The representation of raw EEG signals that can be seen in the OpenBCI GUI and in Python plot is quite different. As presented in Figure 4.17 raw EEG data was in a form of pure sinusoidal signal. With the help from OpenBCI community we were able to determine that due to the significant advanced feature of DC coupled EEG amplifiers as ADS1299, DC offset reflects ($\approx$ 27.4 millivolts) that is present on top of the microvolt variations in EEG. The offset was removed using filters.

2. EEG based experiments require an optimal environment to work on since it can easily get contaminated with noise from electrical devices.

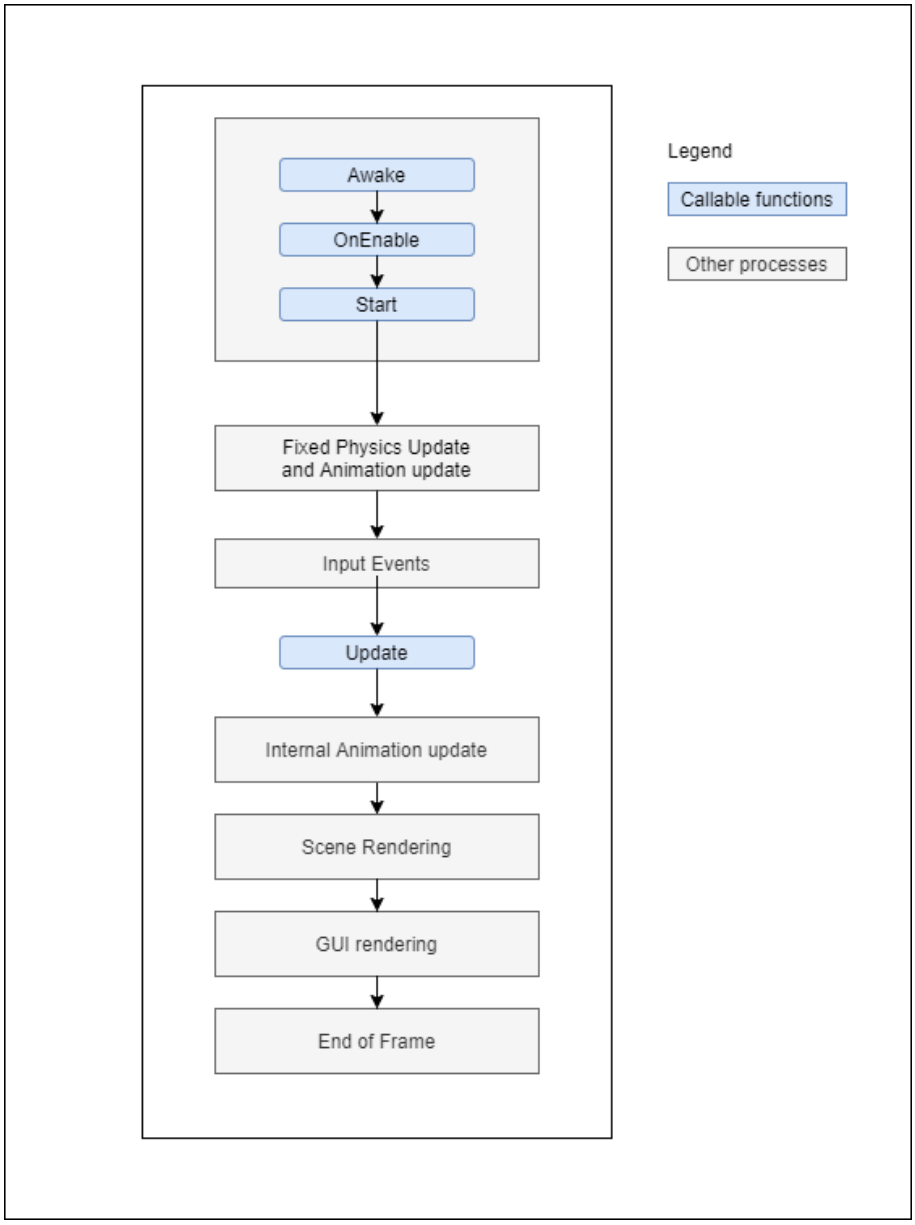


Fig. 4.16 Brief diagram of virtual object life cycle

Yet these experiments were not done in an optimal environment we have considered that the noise affecting EEG signals from electrical equipment are negligible.

3. Since the brainflow library which is the recommended library by OpenBCI and OpenBCI GUI can not function concurrently, we used pylsl

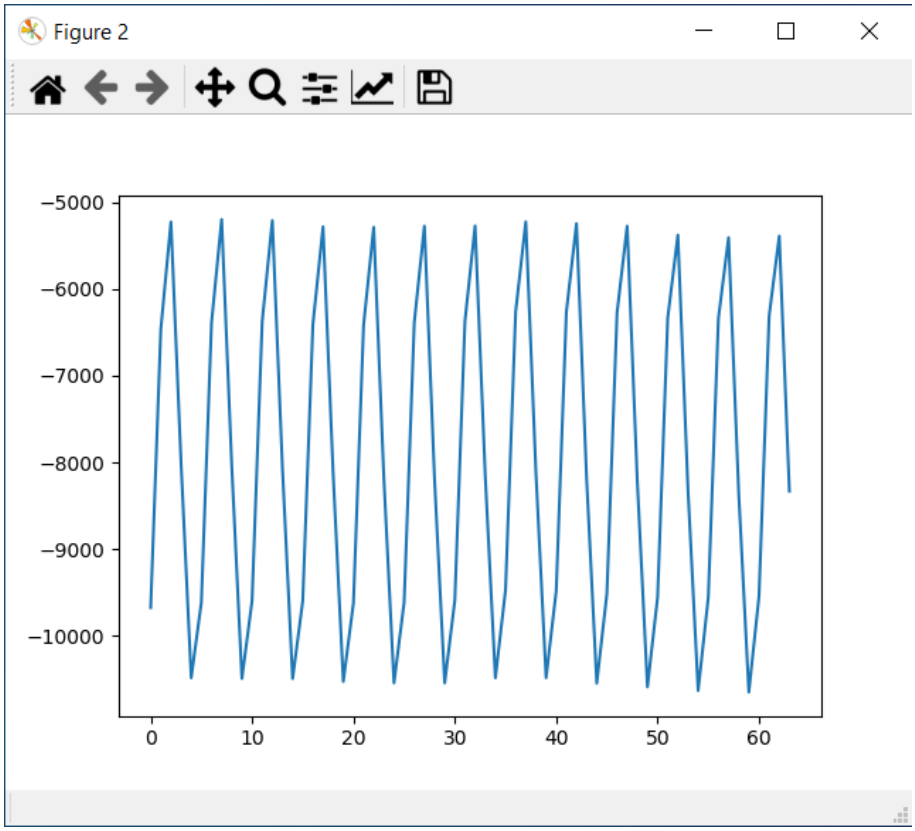


Fig. 4.17 EEG data contaminated with DC offset

(Python Lab Stream Layer) to extract raw EEG data from the OpenBCI.

4. EEG data that were collected for the classification purposes seem to have a different visualization than in the open BCI GUI(version 5.0.1) when we plotted it in python and the data was in mV region where it should have been in microVolt region. We implemented data acquisition through a brainflow library yet the same problem can be seen. After working along with the brainflow community we realized that there was a bug in the software itself, which they updated(version 5.0.2) after a week.

5. Initially the impedance of each electrode showed high value due to default mode set in the open BCI GUI. Default mode was set to classical gain convention after feedback from the open BCI community we set the mode for BodyuV gain convention where the impedance


Fig. 4.18 OpenBCI GUI with high impedance signal

values were in a more acceptable region. Classical gain convention was removed in OpenBCI GUI v5.0.2.

6. The gain convention used in OpenBCI GUI v5.0.1 is faulty. The scale factor of the signal is calculated as follows.

```
Scale Factor (Volts/count) = 4.5 Volts / gain / (2^23 - 1);
```

Note that 2^23 might be an unexpected term in this equation considering that the ADS1299 is a 24-bit device. In the old GUI the gain is used as 1 even though the gain is 24. This fault also corrected in OpenBCI GUI v5.0.2

# Chapter 5

# Results and Analysis

### Independent Component Analysis (ICA)

As shown in Figure 4.14 ICA based artifact removal methods have higher resolution of information preserved within the EEG data yet it took around 37 seconds of processing time to process samples of EEG data, which suggests that this method is not suitable for real time applications.

### Fast Fourier Transformation

Figure 4.9 is the extracted feature for the channel1 which include 0 Hz–60 Hz FFT coefficients.

### Wavelet Transformation

Figure 4.12 is the wavelet coefficient which corresponds to the range 16 Hz–63 Hz. these coefficients can be used as extracted features for training a classification method

### Statistical Representation

Given in Table 5.1 are the results of statistical representation of the bandpass(0 Hz–60 Hz) signal.

In the Table 5.2 below we have compared the different artifact removal and feature extraction methods on information resolution, computational cost and whether the method is applicable for a real-time system. Information resolution is considered as high when both spatial and temporal details are in it and medium when it has one of either temporal or spatial details.

Table 5.1 Statistical representation of bandpass signal

| Statistical Feature | Value(uV) |
|---|---|
| mean | -0.02506917767562681 |
| std | 6.080293459696251 |
| max | 15.555577894359976 |
| min | -14.437421006235299 |
| median | -0.20587660677953962 |

Table 5.2 Feature extraction and artifact removal methods comparison

| Method | Computational Cost<br>K-number of sources<br>N-number of samples | Computational Time |
|---|---|---|
| ICA | $\mathcal{O}(kn \log n)$ | 86 ms |
| FFT frequency coefficient removal method | $\mathcal{O}(n^2)$ | 23 ms |
| WT Threshold Method (Universal/ Bayesian threshold) | $\mathcal{O}(n^2 \log_2 n)$ | 53 ms |
| WT Detail Coefficient Removal Method | $\mathcal{O}(n^2 \log_2 n)$ | 42 ms |

### 5.0.1 Classification Results

The extracted features were applied to the models after performing model based feature selection methods. We used Random Forest, QDA, KNN, SVM and Catboost models for classification. The classification is based on three classes Left, Right and None. The evaluation of classification models is done by train split set. For training the model we used 4800 data points which were collected over 5 days and for the test set we used 480 data points randomly picked out of those.
Following were used as evaluation metrics.

$$\text{Accuracy} = \text{Number of correct predictions / Total number of predictions} \quad (5.1)$$

$$\text{True Positive Rate (TPR)} = \text{True Positives / (True Positives + False Negatives)} \quad (5.2)$$

Frequency bin components extracted by FFT and Detailed coefficients extracted by wavelet transform were used as features for the classification purpose. All the classifications have the ability to perform in real time. We used Random Forest, QDA, KNN, Catboost and SVM for classifying. In Table 5.3 we have compared the accuracies between different classification models. KNN model with features obtained with FFT showed the highest accuracy. Table 5.6 gives the TPR of each class with respect to the model. The confusion matrix of the KNN model is shown in Figure 5.1.
Since we have data collected over 5 days we used a 5-fold cross validation to get an estimation of the consistency of accuracies. This is shown in Table 5.4 and Table 5.5

Table 5.3 Accuracy comparison of each classification model

| **Random Forest** | | **Catboost** | | **QDA** | | **KNN** | | **SVM** | |
|---|---|---|---|---|---|---|---|---|---|
| FFT | WT | FFT | WT | FFT | WT | FFT | WT | FFT | WT |
| 0.473958 | 0.385417 | 0.5 | 0.380208 | 0.46875 | 0.40625 | 0.552083 | 0.40625 | 0.451875 | 0.432292 |

Table 5.4 Five-fold cross validation with fast Fourier Transform

| | **KNN-FFT** | **CatBoost-FFT** | **Random Forest-FFT** | **SVM-FFT** | **QDA-FFT** |
|---|---|---|---|---|---|
| CV_score | 0.56 | 0.54 | 0.55 | 0.54 | 0.55 |
| std_dev(+/-) | 0.02 | 0.01 | 0.07 | 0.07 | 0.09 |

Table 5.5 Five-fold cross validation with Wavelet Transform

| | **KNN-WT** | **CatBoost-WT** | **Random Forest-WT** | **SVM-WT** | **QDA-WT** |
|---|---|---|---|---|---|
| CV_score | 0.47 | 0.50 | 0.47 | 0.49 | 0.47 |
| std_dev(+/-) | 0.03 | 0.05 | 0.09 | 0.09 | 0.06 |

## 5.1 Discussion

Since EEG signals are in the micro voltage range it is more vulnerable towards artifacts. As this is a real-time application, in this research we were more concerned about the time complexities and processing costs of each signal analysis and processing methods. ICA is one of the methods that was implemented which showed higher resolution in information yet it had considerable computational time with respect to other methods. In order to identify the signals that are related to the artifacts, it is a must that training

Table 5.6 True positive rates of each class with respect to model

| Class | Random forest FFT | Catboost FFT | KNN FFT | SVM FFT | QDA FFT |
|---|---|---|---|---|---|
| LEFT | 0.79 | 0.57 | 0.67 | 0.68 | 0.79 |
| RIGHT | 0.36 | 0.47 | 0.51 | 0.44 | 0.49 |
| NONE | 0.062 | 0.44 | 0.44 | 0.00 | 0.06 |

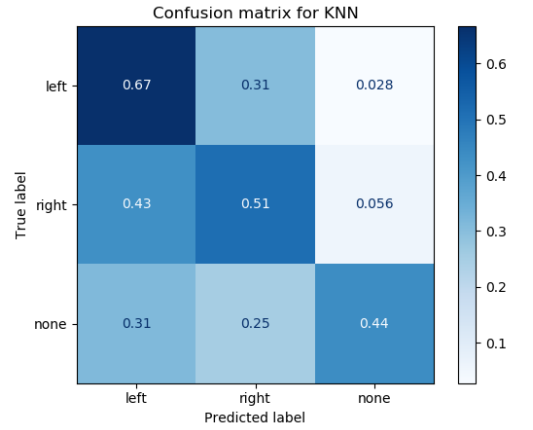


Fig. 5.1 Confusion matrix of KNN model

data is contaminated with those artifacts. Still, ICA can be considered one of the best methods for offline applications since it is more sensitive to discovering artifacts and has the capability to preserve more information while removing them.

WT threshold method has a lower artifact removal ability compared to other methods that were implemented. In our research, we tested out the universal threshold method. Since the universal threshold is sensitive for higher frequencies, it is not applicable for lower frequency reducting. Overall, FFT frequency components as features gave more accuracy than WT detailed coefficients as features when it comes to signal classification. The reason behind that is non-motor imagery signals are induced brain signals which are also known as time phase unlocked signals [5]. Therefore signal characteristics in the time domain variate. Due to this, features extracted from WT can misguide the classification trends. Different filtrations that were done on EEG time series data would cause a phase shift which results in a change in the signal characteristics in the time domain. Hence classification becomes even less accurate when using wavelet features. Although both Catboost and KNN with FFT show a significant accuracy compared to other models, all 5 classification models show high TPR on identifying categorical values

left and right. But TPR of none class (non-intended state) identification is low on QDA, SVM and Random Forest algorithms. KNN and Catboost models show a significant TPR on identifying non-intended states. When comparing both KNN and Catboost models, KNN has overall high accuracy and TPR than other models.

In [8] Faradji et al. were able to obtain a 54.6% accuracy using auto scalar auto-regressive methods for feature extraction and quadratic discriminant analysis as the classification method with 29 EEG channels. However with the KNN algorithm with FFT feature extraction method, we were able to obtain 55% accuracy with 8 EEG channels.

# Chapter 6

# Conclusions and Future Work

In our research, our main goal was to collect advancements done within MI-based research and see how much is applicable when it comes to self-paced non-MI-based applications. It is evident that there are novel signal analysis techniques that show an improved ability to identify and classify the thought commands of the subject but some of these methods are not applicable for a real-time self-paced BCI application. Compared to FFT features, WT features have far less reliability in non-motor imagery signals. The phase unlocked nature of non-motor imagery signals causes the signals within the same class to be distinguished from each other. Filtration techniques that were implemented in the hardware itself cause phase changes for signals which reduces the reliability of WT features. This proved that WT signals do not provide the best features when it comes to identifying thought commands in self-paced BCI. The subject needs to be properly trained prior to using the application. The subject should be able to properly concentrate on the thought commands related to controlling the virtual object. Change of thought patterns and lack of concentration will decrease the accuracy of the application immensely. Changes in the virtual environment also can cause reliability issues in virtual object controlling. Uncertainties of weights for inducing signal patterns for thoughts, from each cerebral area are hard to identify. Changes in the placement of electrodes can cause changes in signals related to thought commands in respective channels. Even the slightest change can cause a considerable impact. This is known as the error of uncertainties in the anatomical localization of electrodes [10]. In a self-paced BCI application, it is important to have more channels or have a proper headset for electrode placement. When the electrode density (number of electrodes) increases we can rectify the anatomical localization errors considerably using statistical methods.

With all the classification models that were trained, KNN algorithm with FFT algorithm would be the ideal choice for features and classification combination. We were able to obtain around 55% TPR value with this combination

Deep learning methods are proved to have a lot of potential in MI-based research in recent history. The possibility of using deep learning approaches in non-motor imagery intent with self-paced brain-computer interfaces is something that can be explored as well.

# References


[1] M. Kaya, M. K. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, "A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces," *Scientific data*, vol. 5, p. 180211, 2018.

[2] Y. Li, J. Pan, F. Wang, and Z. Yu, "A hybrid bci system combining p300 and ssvep and its application to wheelchair control," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 11, pp. 3156–3166, 2013.

[3] L. Cao, J. Li, H. Ji, and C. Jiang, "A hybrid brain computer interface system based on the neurophysiological protocol and brain-actuated switch for wheelchair control," *Journal of neuroscience methods*, vol. 229, pp. 33–43, 2014.

[4] Z. Fang, W. Wang, S. Ren, J. Wang, W. Shi, X. Liang, C.-C. Fan, and Z.-G. Hou, "Learning regional attention convolutional neural network for motion intention recognition based on eeg data," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 1570–1576, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[5] G. Pfurtscheller, R. Leeb, J. Faller, and C. Neuper, "Braincomputer interface systems used for virtual reality control," *Virtual Reality*, vol. 1, pp. 3–20, 2011.

[6] R. Leeb, V. Settgast, D. Fellner, and G. Pfurtscheller, "Self-paced exploration of the austrian national library through thought," *International Journal of Bioelectromagnetism*, vol. 9, no. 4, pp. 237–244, 2007.

[7] A. A. Nooh, J. Yunus, and S. M. Daud, "A review of asynchronous electroencephalogram-based brain computer interface systems," in *International Conference on Biomedical Engineering and Technology IPCBEE*, vol. 11, pp. 55–59, 2011.

[8] F. Faradji, R. K. Ward, and G. E. Birch, "A self-paced two-state mental task-based brain-computer interface with few eeg channels," in *New Frontiers in Brain-Computer Interfaces*, IntechOpen, 2019.

[9] C.-H. Han, K.-R. Müller, and H.-J. Hwang, "Brain-switches for asynchronous brain–computer interfaces: A systematic review," *Electronics*, vol. 9, no. 3, p. 422, 2020.

[10] D. J. Krusienski, M. Grosse-Wentrup, F. Galán, D. Coyle, K. J. Miller, E. Forney, and C. W. Anderson, "Critical issues in state-of-the-art brain–computer interface signal processing," *Journal of neural engineering*, vol. 8, no. 2, p. 025002, 2011.

[11] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Eeg-based intention recognition from spatio-temporal representations via cascade and parallel convolutional recurrent neural networks," *arXiv preprint arXiv:1708.06578*, 2017.